TOSHIBA

TOSHIBA Original CMOS 16-Bit Microcontroller

TLCS-900/H Series

TMP95CS64/TMP95C265

TOSHIBA CORPORATION

Preface

Thank you very much for making use of Toshiba microcomputer LSIs.
Before use this LSI, refer the section, "Points of Note and Restrictions".
Especially, take care below cautions.

CAUTION

How to release the HALT mode

Usually, interrupts can release all halts status. However, the interrupts = ($\overline{\text{NMI}}$, INT0), which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 3 clocks of X1) with IDLE1 or STOP mode (IDLE2 is not applicable to this case). (In this case, an interrupt request is kept on hold internally.)

If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficultly. The priority of this interrupt is compare with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.

CMOS 16-Bit Microcontrollers

TMP95CS64F / TMP95C265F

1. Outline and Features

TMP95CS64/265 is a high-speed 16-bit microcontroller designed for the control of various mid- to large-scale equipment. TMP95CS64 incorporates masked ROM, while TMP95C265 has no ROM. Otherwise, all the functions of the products are the same.
TMP95CS64/265 comes in a 100-pin flat package.
Listed below are the features.

(1) High-speed 16-bit CPU (900/H CPU)

- Instruction mnemonics are upward-compatible with TLCS-90/900
- 16 Mbytes of linear address space
- General-purpose registers and register banks
- 16-bit multiplication and division instructions; bit transfer and arithmetic instructions
- Micro DMA: Four-channels (640 ns / 2 bytes at 25 MHz)

(2) Minimum instruction execution time: 160 ns (at 25 MHz)

(3) Built-in RAM: 2 Kbytes
Built-in ROM:

TMP95CS64	64 Kbyte ROM
TMP95C265	No ROM

(4) External memory expansion

- Expandable up to 16 Mbytes (shared program/data area)
- External data bus width select pin (AM8/$\overline{16}$)
- Can simultaneously support 8/16-bit width external data bus
  ··· Dynamic data bus sizing

(5) 8-bit timers: 8 channels

- With event counter function: 2 channels

(6) 16-bit timer/event counter: 2 channels

000707EBP1

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

(7) General-purpose serial interface: 3 channels

(8) 10-bit A/D converter: 8 channels

(9) 8-bit D/A converter: 2 channels

(10) Watchdog timer

(11) Chip select/wait controller: 4 blocks

(12) Interrupts: 45 interrupts

- 9 CPU interrupts: Software interrupt instruction and illegal instruction
- 26 internal interrupts: ⎤ Seven selectable priority levels
- 10 external interrupts: ⎦

(13) Input/output ports

TMP95CS64	81 pins
TMP95C265	55 pins

(14) Standby mode

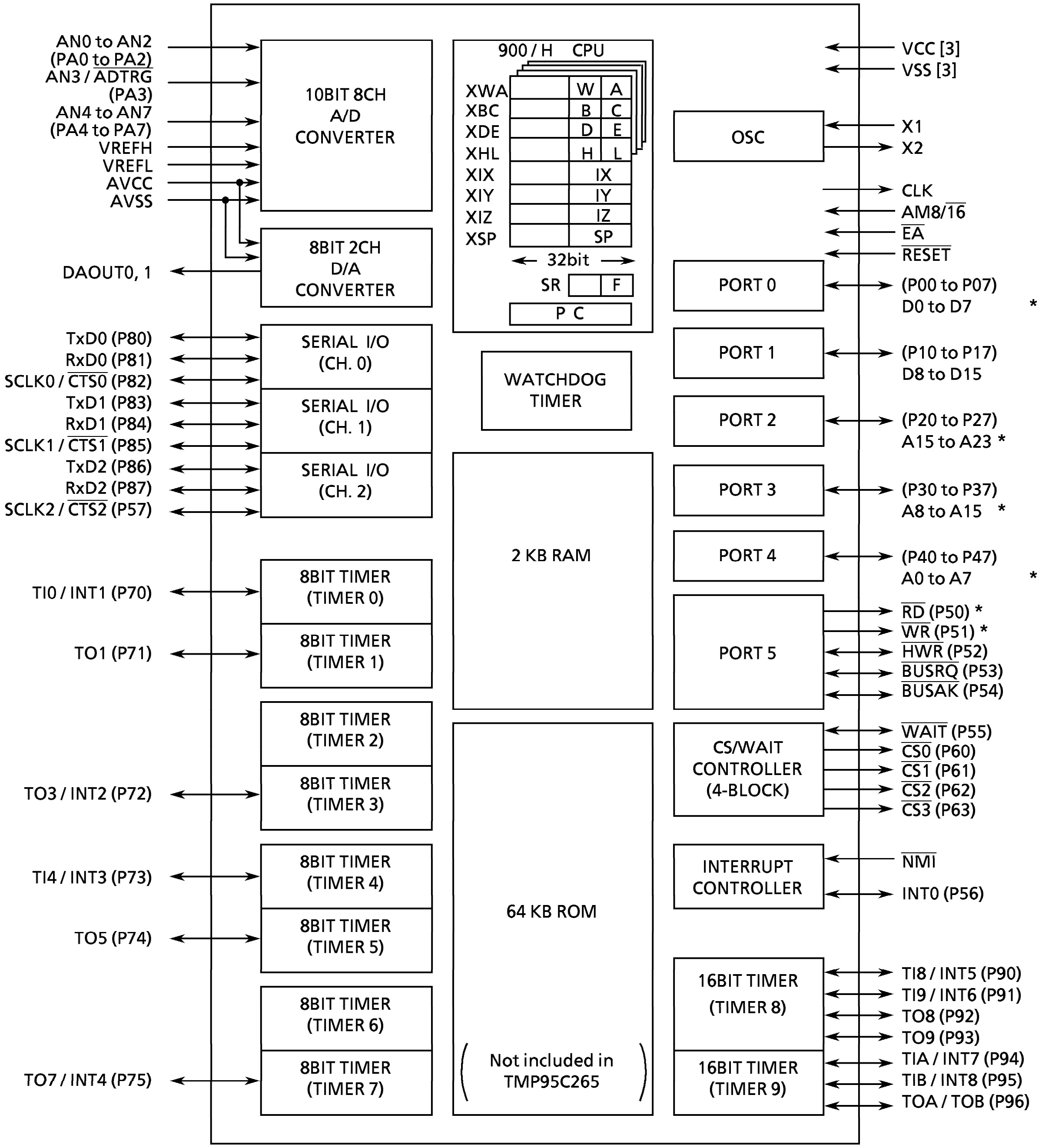
- Four HALT modes: RUN, IDLE2, IDLE1, STOP

(15) Operating voltage

- $V_{CC} = 2.7 - 3.3$ V
- $V_{CC} = 4.5 - 5.5$ V

(16) Package

- P-LQFP100-1414-0.50F



Note: Pin states after reset

Product	AM8/$\overline{16}$	Pin function after reset
TMP95CS64	Fixed to high level	Multi-use pins can select function in parentheses ( ).
TMP95C265	High level	Multi-use pins other than those marked by an asterisk can select functions in parentheses ( ).
	Low level	Multi-use pins other than those marked by asterisk can select function in parentheses ( ). However, port 1 can select functions outside parentheses ( ).

Figure 1 TMP95CS64/TMP95C265 Block Diagram

2. Pin Assignment and Pin Functions

This section shows the TMP95CS64F/265F pin assignment, and the names and an outline of the functions of the input/output pins.

2.1 Pin Assignment Diagram

Figure 2.1 is a pin assignment diagram for TMP95CS64F/265F.

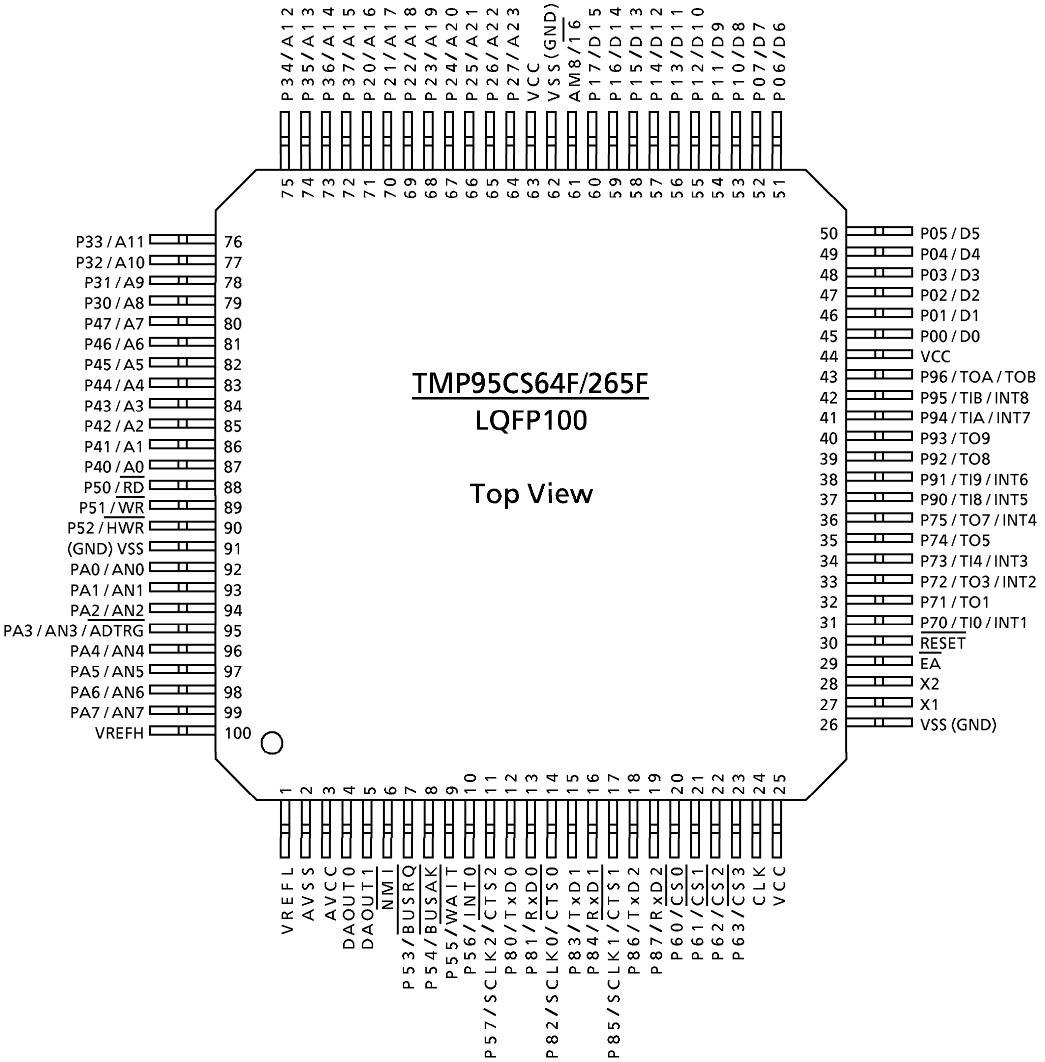


Figure 2.1 Pin Assignment Diagram (100-Pin LQFP)

2.2 Pin Names and Functions

Table 2.2 shows the names and functions of the input/output pins.

Table 2.2 Pin Names and Functions (1/4)

Pin Name	Number of Pins	Input/Output	Function
P00 to P07	8	Input/output	Port 0: I/O port. Input or output specifiable in units of bits
/ D0 to D7		Input/output	Data: Data bus 0 to 7
P10 to P17	8	Input/output	Port 1: I/O port. Input or output specifiable in units of bits
/ D8 to D15		Input/output	Data: Data bus 8 to 15
P20 to P27	8	Input/output	Port 2: I/O port. Input or output specifiable in units of bits
/ A16 to A23		Output	Address: Address bus 16 to 23
P30 to P37	8	Input/output	Port 3: I/O port. Input or output specifiable in units of bits
/ A8 to A15		Output	Address: Address bus 8 to 15
P40 to P47	8	Input/output	Port 4: I/O port. Input or output specifiable in units of bits
/ A0 to A7		Output	Address: Address bus 0 to 7
P50	1	Output	Port 50: Output-only port
/ $\overline{RD}$		Output	Read: Outputs strobe signal to read external memory (setting P5 <P50> = 0 and P5FC <P50F> = 1 outputs strobe signal at all read timings)
P51	1	Output	Port 51: Output-only port.
/ $\overline{WR}$		Output	Write: Outputs strobe signal to write data on pins D0 to D7
P52	1	Input/output	Port 52: I/O port (with built-in pull-up resistor)
/ $\overline{HWR}$		Output	Upper write: Outputs strobe signal to write data on pins D8 to D15
P53	1	Input/output	Port 53: I/O port (with built-in pull-up resistor)
/ $\overline{BUSRQ}$		Input	Bus request: Input pin to request external bus release
P54	1	Input/output	Port 54: I/O port (with built-in pull-up resistor)
/ $\overline{BUSAK}$		Output	Bus acknowledge: Output pin to acknowledge that CPU received $\overline{BUSRQ}$ and released external bus.
P55	1	Input/output	Port 55: I/O port (with built-in pull up resistor)
/ $\overline{WAIT}$		Input	Wait: Bus wait request pin for CPU (Effective when 1 + N WAIT mode, or 0 + N WAIT mode. Set using chip select/wait control register.)
P56	1	Input/output	Port 56: I/O port (with built-in pull-up resistor)
/ INT0		Input	Interrupt request pin 0: Interrupt request pin with programmable level/rising edge.

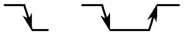
Table 2.2 Pin Names and Functions (2/4)

Pin Name	Number of Pins	Input/Output	Function
P57	1	Input/output	Port 57: I/O port (with built-in pull-up resistor)
/ SCLK2		Input/output	Serial clock input/output 2
/ $\overline{\text{CTS2}}$		Input	Serial data ready to send 2 (Clear-to-send)
P60	1	Output	Port 60: Output-only port
/ $\overline{\text{CS0}}$		Output	Chip select 0: Outputs 0 if address is within specified address range
P61	1	Output	Port 61: Output-only port
/ $\overline{\text{CS1}}$		Output	Chip select 1: Outputs 0 if address is within specified address range
P62	1	Output	Port 62: Output-only port
/ $\overline{\text{CS2}}$		Output	Chip select 2: Outputs 0 if address is within specified address range
P63	1	Output	Port 63: Output-only port
/ $\overline{\text{CS3}}$		Output	Chip select 3: Outputs 0 if address is within specified address range
P70	1	Input/output	Port 70: I/O port
/ TI0		Input	Timer input 0: Input pin for timer 0
/ INT1		Input	Interrupt request pin 1: Rising-edge interrupt request pin ↑
P71	1	Input/output	Port 71: I/O port.
/ TO1		Output	Timer output 1: Output pin for timer 0 or 1
P72	1	Input/output	Port 72: I/O port
/ TO3		Output	Timer output 3: Output pin for timer 2 or 3
/ INT2		Input	Interrupt request pin 2: Rising-edge interrupt request pin ↑
P73	1	Input/output	Port 73: I/O port
/ TI4		Input	Timer input 4: Input pin for timer 4
/ INT3		Input	Interrupt request pin 3: Rising-edge interrupt request pin ↑
P74	1	Input/output	Port 74: I/O port
/ TO5		Output	Timer output 5: Output pin for timer 4 or 5
P75	1	Input/output	Port 75: I/O port
/ TO7		Output	Timer output 7: Output pin for timer 6 or 7
/ INT4		Input	Interrupt request pin 4: Rising-edge interrupt request pin ↑
P80	1	Input/output	Port 80: I/O port (with built-in pull-up resistor)
/ TxD0		Output	Serial transmission data 0
P81	1	Input/output	Port 81: I/O port (with built-in pull-up resistor)
/ RxD0		Input	Serial receive data 0
P82	1	Input/output	Port 82: I/O port (with built-in pull-up resistor)
/ SCLK0		Input/output	Serial clock input/output 0
/ $\overline{\text{CTS0}}$		Input	Serial data ready to send 0 (Clear-to-send)

Table 2.2 Pin Names and Functions (3/4)

Pin Name	Number of Pins	Input/Output	Function
P83	1	Input/output	Port 83: I/O port (with built-in pull-up resistor)
/ TxD1		Output	Serial transmission data 1
P84	1	Input/output	Port 84: I/O port (with built-in pull-up resistor)
/ RxD1		Input	Serial receive data 1
P85	1	Input/output	Port 85: I/O port (with built-in pull-up resistor)
/ SCLK1		Input/output	Serial clock input/output 1
/ $\overline{CTS1}$		Input	Serial data ready to send 1 (Clear-to-send)
P86	1	Input/output	Port 86: I/O port (with built-in pull-up resistor)
/ TxD2		Output	Serial transmission data 2
P87	1	Input/output	Port 87: I/O port (with built-in pull-up resistor)
/ RxD2		Input	Serial receive data 2
P90	1	Input/output	Port 90: I/O port
/ TI8		Input	Timer input 8: Input pin for timer 8
/ INT5		Input	Interrupt request pin 5: Interrupt request pin with programmable rising/falling edge ↑ ↓
P91	1	Input/output	Port 91: I/O port
/ TI9		Input	Timer input 9: Input pin for timer 8
/ INT6		Input	Interrupt request pin 6: Rising edge interrupt request pin ↑
P92	1	Input/output	Port 92: I/O port
/ TO8		Output	Timer output 8: Output pin for timer 8
P93	1	Input/output	Port 93: I/O port
/ TO9		Output	Timer output 9: Output pin for timer 8
P94	1	Input/output	Port 94: I/O port
/ TIA		Input	Timer input A: Input pin for timer 9
/ INT7		Input	Interrupt request pin 7: Interrupt request pin with programmable rising/falling edge ↑ ↓
P95	1	Input/output	Port 95: I/O port
/ TIB		Input	Timer input B: Input pin for timer 9
/ INT8		Input	Interrupt request pin 8: Rising edge interrupt request pin ↑
P96	1	Input/output	Port 96: I/O port
/ TOA		Output	Timer output A: Output pin for timer 9
/ TOB		Output	Timer output B: Output pin for timer 9
PA0 to PA2	3	Input	Port A0 to A2: Input-only port
/ AN0 to AN2		Input	Analog input 0 to 2: A/D converter input pins
PA3	1	Input	Port A3: Input-only port
/ AN3		Input	Analog input 3: A/D converter input pin
/ $\overline{ADTRG}$		Input	External start trigger

Table 2.2 Pin Names and Functions (4/4)

Pin Name	Number of Pins	Input/Output	Function
PA4 to PA7	4	Input	Port A4 to A7: Input-only port
/ AN4 to AN7		Input	Analog input 4 to 7: A/D converter input pins
DAOUT0	1	Output	D/A output 0: D/A converter 0 output pin
DAOUT1	1	Output	D/A output 1: D/A converter 1 output pin
$\overline{\text{NMI}}$	1	Input	Non-maskable interrupt request pin: Interrupt request pin with programmable falling edge or both falling and rising edge 
CLK	1	Output	Clock output: Outputs external clock divided by 4. Pulled up during reset.
$\overline{\text{EA}}$	1	Input	External access: With TMP95CS64, connect to VCC. With TMP95C265, connect to GND.
AM8 / $\overline{16}$	1	Input	Address mode: External data bus width select pin With TMP95CS64: Connect this pin to VCC. Data bus width at external access can be set by chip select/wait control register. With TMP95C265: Connect to GND when external 16-bit bus is fixed or external 8/16-bit buses are mixed. When external 8-bit bus is fixed, connect to VCC.
$\overline{\text{RESET}}$	1	Input	Reset: Initializes TMP95CS64/265 (with built-in pull-up resistor)
VREFH	1	Input	Reference voltage input pin for A/D converter (high)
VREFL	1	Input	Reference voltage input pin for A/D converter (low)
AVCC	1		Power supply pin for A/D converter and reference voltage input pin for D/A converter: Connect to power supply
AVSS	1		GND pin for A/D converter and reference voltage input pin for D/A converter: Connect to GND
X1 / X2	2	Input/output	Oscillator connecting pin
VCC	3		Collector supply pin: Connect all VCC pins to power supply
VSS	3		GND pin: Connect all VSS pins to GND (0 V)

Note: Disconnect the pull-up resistors from pins other than $\overline{\text{RESET}}$ pin by software.

3. Operation

The following describes block by block the functions and basic operation of TMP95CS64/265.
Notes and restrictions for each block are outlined in "7, Use Precautions and Restrictions" at the end of this manual.

3.1 CPU

TMP95CS64/265 incorporates a high-performance 16-bit CPU (900/H-CPU). For CPU operation, see the "TLCS-900/H CPU".
The following describes the unique functions of the CPU used in TMP95CS64/265; these functions are not covered in the TLCS-900/H CPU section.

3.1.1 Reset

When resetting the TMP95CS64/265 microcontroller, ensure that the power supply voltage is within the operating voltage range, and that the internal high-frequency oscillator has stabilized. Then hold the $\overline{\text{RESET}}$ input to low level for at least 10 system clocks (ten states: 0.8 μs at 25 MHz).
When the reset is accepted, the CPU:

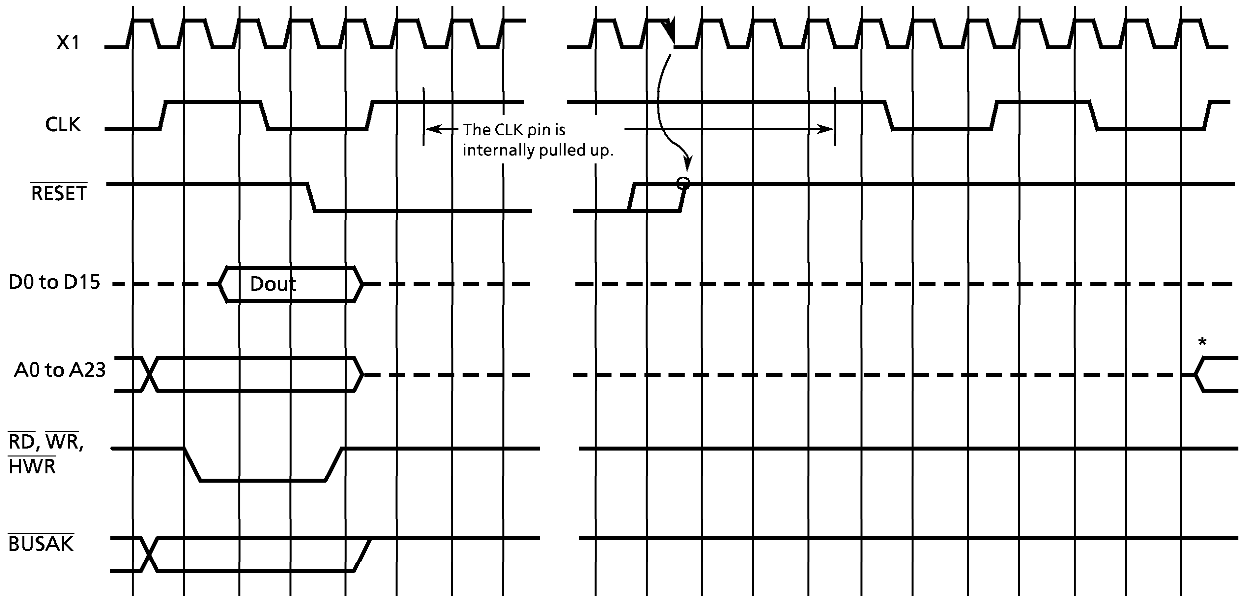
- Sets as follows the program counter (PC) in accordance with the reset vector stored at address FFFF00H to FFFF02H:
  PC (7: 0) ← value at FFFF00H address
  PC (15: 8)← value at FFFF01H address
  PC (23: 16) ← value at FFFF02H address
- Sets the stack pointer (XSP) to 100H.
- Sets bits <IFF2:0> of the status register (SR) to 111 (sets the interrupt level mask register to level 7).
- Sets the <MAX> bit of the status register to 1 (MAX mode).
  (Note: As this product does not support a MIN mode, don't write 0 to <MAX>.)
- Clears bits <RFP2:0> of the status register to 000 (sets the register bank to 0).

When reset is released, the CPU starts executing instructions in accordance with the program counter settings. CPU internal registers not mentioned above do not change when the reset is released.
When the reset is accepted, the CPU sets internal I/O, ports, and other pins as follows.

- Initializes the internal I/O registers.
- Sets the port pins, including the pins that also act as internal I/O, to general-purpose input or output port mode.
- Pulls up the CLK pin to high level.

  (Note: During reset, do not reduce the external voltage level as this can cause malfunction.)

Figure 3.1 shows an example of the basic timing of the reset operation.



* After confirmation that RESET = high, A0 to A23 are output at the X1 rising edge of the 10th or 12th clock.

Figure 3.1 TMP95CS64/265 Reset Timing Example

3.1.2 External Data Bus Width Selection (AM8/16 Pin)

(1) With TMP95CS64 (EA high level)

Connect the input pin to VCC. After a reset, this pin accesses ROM by the internal 16-bit bus.
The data bus width for an external access depends on the setting in the <B0BUS>, <B1BUS>, <B2BUS>, <B3BUS>, or <BEXBUS> bit of the chip select/wait control registers. To access the 16-bit bus, set port 1 to D8 to D15.

(2) With TMP95C265 (EA low level)

Selects the width of the external data bus by sampling the AM8/16 input pin at the rising edge of the reset signal.

- When AM8/16 = low level
  P00 to P17 function as a 16-bit data bus (D0 to D15) (8- and 16-bit data bus width mixed, or 16-bit data bus width fixed).
  The data bus width for an external access depends on the setting in the <B0BUS>, <B1BUS>, <B2BUS>, or <BEXBUS> bit of the chip select/wait control registers.

- When AM8/16 = high level
  P00 to P07 function as an 8-bit data bus (D0 to D7) (external 8-bit data bus fixed).
  The <B0BUS>, <B1BUS>, <B2BUS>, or <BEXBUS> setting is ignored.

3.2 Memory Map

TMP95CS64/265 uses 160 bytes of address space as an internal I/O area.
This is allocated to address area 000000H to 00009FH. The CPU can access this internal I/O by direct addressing mode using short command code.

Figure 3.2 shows the memory map and the access widths for the CPU addressing modes.

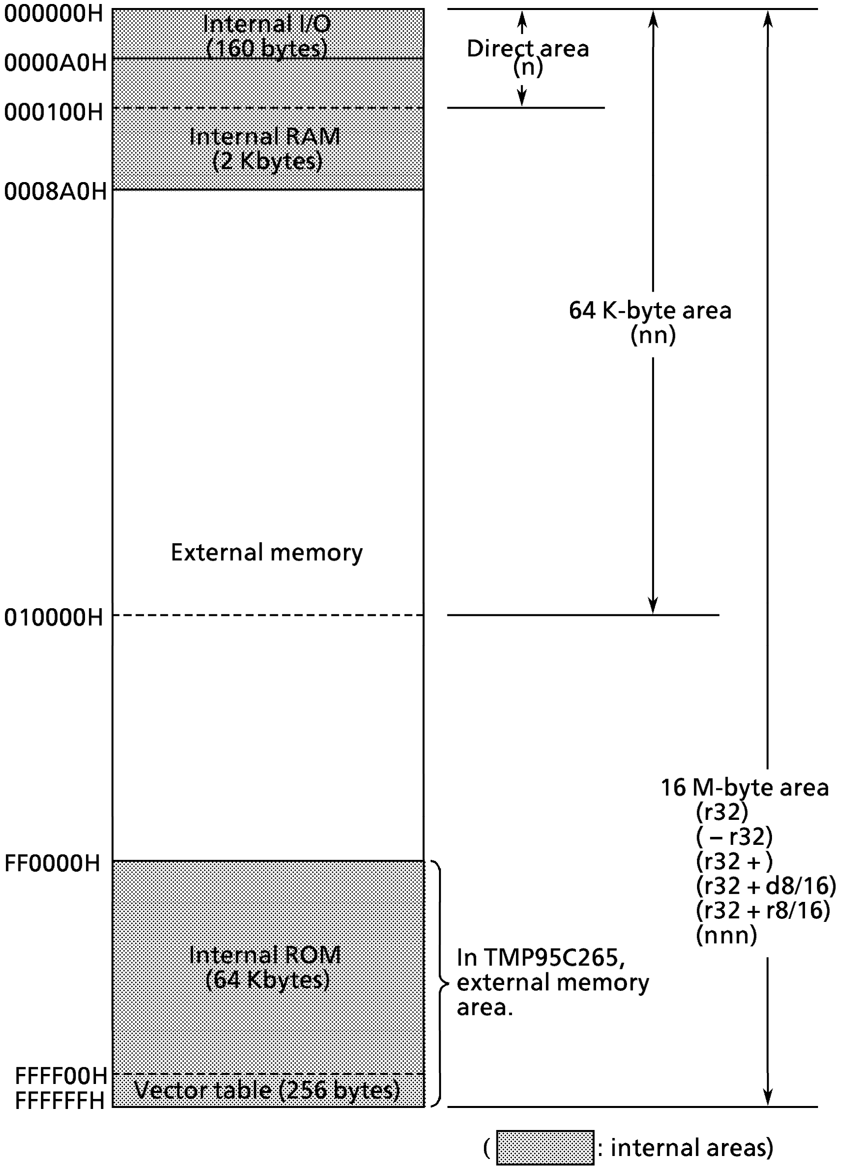


Figure 3.2 TMP95CS64/265 Memory Map

3.3 Interrupts

Interrupts are controlled by the CPU interrupt mask register <IFF2:0> (bits 14 to 12 of the status register) and by the built-in interrupt controller.

TMP95CS64/265 has a total of 45 interrupts divided into the following five types:

Interrupts generated by CPU: 9

- Software interrupts: 8
- Illegal instruction: 1

Internal interrupts: 26

- Internal I/O interrupts: 22
- Micro DMA transfer end interrupts: 4

External interrupts: 10

- Interrupts from external pins ($\overline{\text{NMI}}$, INT0 to INT8)

A (fixed) individual interrupt vector number is assigned to each interrupt.

One of seven (variable) priority levels can be assigned to each maskable interrupt. The priority level of non-maskable interrupts is fixed at 7, the highest level.

When an interrupt is generated, the interrupt controller sends the priority of that interrupt to the CPU. If multiple interrupts are generated simultaneously, the interrupt controller sends the interrupt with the highest priority to the CPU. (The highest priority possible is level 7, used for non-maskable interrupts.)

The CPU compares the priority level of the interrupt with the value of the CPU interrupt mask register <IFF2:0>. If the priority level of the interrupt is higher than the value of the interrupt mask register, the CPU accepts the interrupt. However, software interrupts and illegal instruction interrupts generated by the CPU are processed without comparison with the <IFF2:0> value.

The interrupt mask register <IFF2:0> value can be updated using the value of the EI instruction (executing EI num sets the content of <IFF2:0> to num). For example, specifying EI 3 enables the acceptance of maskable interrupts whose priority level set in the interrupt controller is 3 or higher, and enables the acceptance of non-maskable interrupts. However, if EI or EI 0 is specified, maskable interrupts with a priority level of 1 or higher and non-maskable interrupts are accepted (operationally identical to "EI1").

Operationally, the DI instruction (<IFF2:0> is 7) is identical to the EI 7 instruction, but as the priority level of maskable interrupts is 0 to 6, the DI instruction is used to disable maskable interrupts. The EI instruction is valid immediately after execution begins. (With TLCS-90, the EI instruction is valid after execution of the instruction following the EI instruction.)

In addition to the general-purpose interrupt processing mode described above, TLCS-900/H interrupts have a micro DMA processing mode as well.
Because the CPU transfers data (byte transfer, word transfer, or 4-byte transfer) automatically in micro DMA mode, this mode can be used for speeding up interrupt processing, such as transferring data to I/O. TMP95CS64/265 also has a micro DMA soft start function for requesting micro DMA processing by software not by interrupt.
Figure 3.3 (1) shows the overall interrupt processing flow.

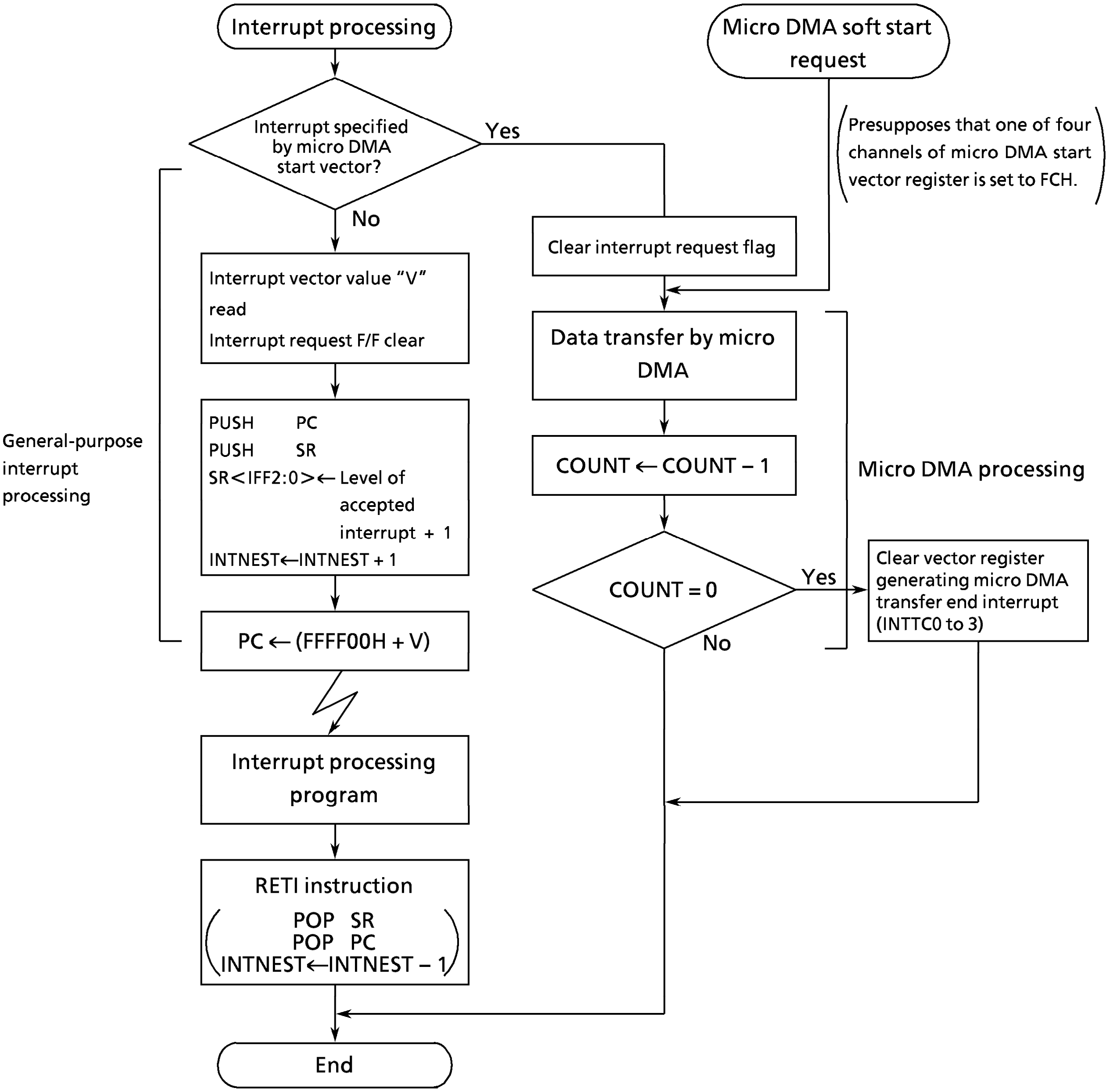

Interrupt processing
Interrupt specified by micro DMA start vector?
Yes
No
Interrupt vector value "V" read
Interrupt request F/F clear
PUSH PC
PUSH SR
SR<IFF2:0>← Level of accepted interrupt + 1
INTNEST←INTNEST + 1
PC ← (FFFF00H + V)
Interrupt processing program
RETI instruction
(POP SR
POP PC
INTNEST←INTNEST – 1)
End
General-purpose interrupt processing
Micro DMA soft start request
(Presupposes that one of four channels of micro DMA start vector register is set to FCH.)
Clear interrupt request flag
Data transfer by micro DMA
COUNT ← COUNT – 1
COUNT = 0
Yes
No
Clear vector register generating micro DMA transfer end interrupt (INTTC0 to 3)
Micro DMA processing

Figure 3.3 (1) Interrupt and Micro DMA Processing Flow

3.3.1 General-Purpose Interrupt Processing

When the CPU accepts an interrupt, the CPU performs the following processing. However, in the case of software interrupts and illegal instruction interrupts generated by the CPU, the CPU skips ① and ③ and executes steps ②, ④, and ⑤.

① The CPU reads the interrupt vector from the interrupt controller. If there are simultaneous interrupts set to the same level, the interrupt controller generates an interrupt vector in accordance with the default priority and clears the interrupt request.
(The default priority is already fixed for each interrupt: the smaller the vector value, the higher the priority level.)

② The CPU saves the contents of the program counter (PC) and status register (SR) to the stack area (indicated by XSP).

③ The CPU sets the value of the CPU's interrupt mask register <IFF2:0> to the received interrupt level incremented by 1. However, if the incremented value level is 7 or higher, the CPU just sets the register to 7.

④ The CPU increments interrupt nesting counter INTNEST by 1.

⑤ The CPU jumps to the address indicated by the data at address FFFF00H + interrupt vector, and starts the interrupt processing routine.

Table 3.3 (1) shows the times for the above processing.

Table 3.3 (1) Interrupt Processing Times for Bus Widths

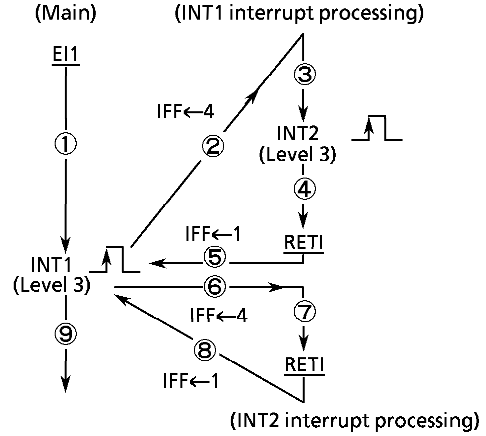
Stack Area Bus Width (Bits)	Interrupt Vector Area Bus Width	Number of Interrupt Processing Execution States	Interrupt Processing Time (μs) @ fc = 25 MHz
8	8	28	2.24
	16	24	1.92
16	8	22	1.76
	16	18	1.44

When the CPU completed the interrupt processing, use the RETI instruction to return to the main routine. This instruction restores the contents of the program counter and status register from the stack, and decrements interrupt nesting counter INTNEST by 1.
Non-maskable interrupts cannot be disabled by program. Maskable interrupts can be enabled or disabled by program. The program can set a priority level for every interrupt source. (Setting the priority level to 0 (or 7) disables the interrupt request.)

If a request is received for an interrupt with a higher priority level than that set in the CPU interrupt mask register <IFF2:0>, the CPU accepts the interrupt. Set the CPU interrupt mask register <IFF2:0> to the received interrupt priority level incremented by 1.
If, during interrupt processing, an interrupt is generated with a higher level than the interrupt being currently processed, or if, during non-maskable interrupt processing, a non-maskable interrupt request is generated from another source, the CPU suspends the currently processing routine and accepts the later interrupt. Then, after the CPU finished processing the later interrupt, the CPU returns to the interrupt it previously suspended and resumes processing.
If the CPU receives a request for another interrupt while performing processing in steps ① to ⑤, the second interrupt is sampled immediately after execution of the start instruction for its interrupt processing routine. Specifying DI as the start instruction disables maskable interrupt nesting. (Note: In the 900 and 900/L, sampling is performed before execution of the start instruction.)

After a reset, the interrupt mask register <IFF2:0> is initialized to 111, thus disabling maskable interrupts.
The following steps (1) through (5) show the interrupt processing flow.

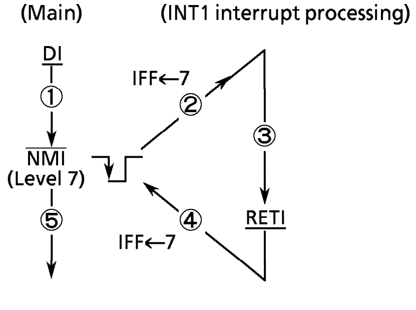
(1) Maskable interrupts



When the CPU accepts an interrupt, it sets IFF to the priority level of the interrupt incremented by 1.
Accordingly, if during interrupt processing an interrupt request is received with the same or a lower priority than that of the interrupt being processed, because this priority level is lower than the IFF value, the second interrupt cannot be accepted until the processing of the prior interrupt is complete.

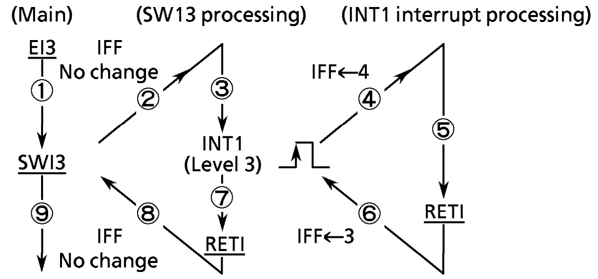
Note:	__ (underline)	:	Instruction
	①, ②,	:	Execution flow
	IFF	:	Interrupt mask register

(2) Non-maskable interrupts (NMI, INTWD)



When the DI instruction is executed (IFF is 7), only non-maskable interrupts can be received (because the priority level of non-maskable interrupts is fixed to 7.)
When the EI instruction is executed, the CPU sets IFF to 7 upon acceptance of an NMI or INTWD interrupt.

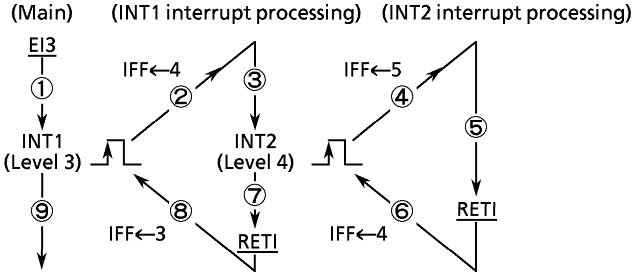
(3) Non-maskable interrupts (Software interrupts, illegal instruction interrupts)



When the DI instruction is executed (IFF is 7), the CPU can accept interrupts. However, unlike with NMI or INTWD interrupts, IFF does not change upon acceptance of an interrupt.

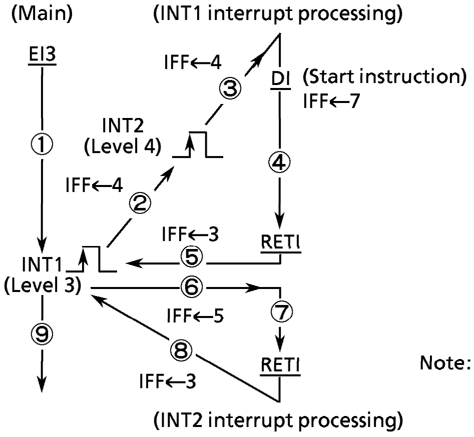
Therefore, during processing of a software interrupt, if a request is received for an interrupt with a priority the same or higher than the IFF value, the interrupt is nested.

(4) Interrupt nesting



During interrupt processing, if a request is received for an interrupt with a priority the same or higher than the interrupt being processed (the interrupt priority level is the same as or higher than the IFF value), the CPU receives the second interrupt and nests it.

(5) Interrupt sampling (Maskable interrupt nesting disabled)



If, after the time the CPU accepted an interrupt but before the CPU begins processing it, the CPU receives a request for another interrupt with a higher priority, the second interrupt is nested after execution of the start instruction for processing the interrupt accepted first.

Accordingly, issuing the DI instruction as the start instruction disables nesting of maskable interrupts.

Note: __ (underline) : Instruction.
①, ②, : Execution flow
IFF : Interrupt mask register

Table 3.3 (2) shows the TMP95CS64/265 interrupt vectors and micro DMA start vectors. With the TMP95CS64/265, FFFF00H to FFFFFFH (256 bytes) is allocated to the interrupt vector area.

Table 3.3 (2) TMP95CS64/265 Interrupt Vectors and Micro DMA Start Vectors

Default priority	Type	Interrupt source and source of micro DMA request	Vector value V	Vector reference address	Micro DMA start vector
1	Non-maskable	Reset or [SWI0] instruction	0000H	FFFF00H	–
2		[SWI1] instruction	0004H	FFFF04H	–
3		Illegal instruction or [SWI2] instruction	0008H	FFFF08H	–
4		[SWI3] instruction	000CH	FFFF0CH	–
5		[SWI4] instruction	0010H	FFFF10H	–
6		[SWI5] instruction	0014H	FFFF14H	–
7		[SWI6] instruction	0018H	FFFF18H	–
8		[SWI7] instruction	001CH	FFFF1CH	–
9		NMI : $\overline{\text{NMI}}$ pin input	0020H	FFFF20H	–
10		INTWD : Watchdog timer	0024H	FFFF24H	–
–	–	Micro DMA (Note)	–	–	–
11	Maskable	INT0 : INT0 pin input	0028H	FFFF28H	28H
12		INT1 : INT1 pin input	002CH	FFFF2CH	2CH
13		INT2 : INT2 pin input	0030H	FFFF30H	30H
14		INT3 : INT3 pin input	0034H	FFFF34H	34H
15		INT4 : INT4 pin input	0038H	FFFF38H	38H
16		INT5 : INT5 pin input	003CH	FFFF3CH	3CH
17		INT6 : INT6 pin input	0040H	FFFF40H	40H
18		INT7 : INT7 pin input	0044H	FFFF44H	44H
19		INT8 : INT8 pin input	0048H	FFFF48H	48H
20		INTT0 : 8-bit timer 0	004CH	FFFF4CH	4CH
21		INTT1 : 8-bit timer 1	0050H	FFFF50H	50H
22		INTT2 : 8-bit timer 2	0054H	FFFF54H	54H
23		INTT3 : 8-bit timer 3	0058H	FFFF58H	58H
24		INTT4 : 8-bit timer 4	005CH	FFFF5CH	5CH
25		INTT5 : 8-bit timer 5	0060H	FFFF60H	60H
26		INTT6 : 8-bit timer 6	0064H	FFFF64H	64H
27		INTT7 : 8-bit timer 7	0068H	FFFF68H	68H
28		INTTR8 : 16-bit timer 8 (TREG8)	006CH	FFFF6CH	6CH
29		INTTR9 : 16-bit timer 8 (TREG9)	0070H	FFFF70H	70H
30		INTTRA : 16-bit timer 9 (TREGA)	0074H	FFFF74H	74H
31		INTTRB : 16-bit timer 9 (TREGB)	0078H	FFFF78H	78H
32		INTTO8 : 16-bit timer 8 (Overflow)	007CH	FFFF7CH	7CH
33		INTTO9 : 16-bit timer 9 (Overflow)	0080H	FFFF80H	80H
34		INTRX0 : Serial receive (Channel 0)	0084H	FFFF84H	84H
35		INTTX0 : Serial transmission (Channel 0)	0088H	FFFF88H	88H
36		INTRX1 : Serial receive (Channel 1)	008CH	FFFF8CH	8CH
37		INTTX1 : Serial transmission (Channel 1)	0090H	FFFF90H	90H
38		INTRX2 : Serial receive (Channel 2)	0094H	FFFF94H	94H
39		INTTX2 : Serial transmission (Channel 2)	0098H	FFFF98H	98H
40		INTAD : A/D conversion end	009CH	FFFF9CH	9CH
41		INTTC0 : Micro DMA end (Channel 0)	00A0H	FFFFA0H	–
42		INTTC1 : Micro DMA end (Channel 1)	00A4H	FFFFA4H	–
43		INTTC2 : Micro DMA end (Channel 2)	00A8H	FFFFA8H	–
44		INTTC3 : Micro DMA end (Channel 3)	00ACH	FFFFACH	–
–		(Reserved)	00B0H	FFFFB0H	–
to		to	to	to	to
–		(Reserved)	00FCH	FFFFFCH	–
–	–	Micro DMA soft start request	–	–	FCH

Note: Micro DMA default priority

If an interrupt request is generated by a source specified by micro DMA, the interrupt has the highest priority of the maskable interrupts (irrespective of the default priority allocated to all channels).

Setting reset vectors and interrupt vectors

① Reset vector

FFFF00H	PC (7:0)
FFFF01H	PC (15:8)
FFFF02H	PC (23:16)
FFFF03H	XX

XX: Don’t care

② Interrupt vectors (Other than reset vector)

Vector reference address + 0	PC (7:0)
+ 1	PC (15:8)
+ 2	PC (23:16)
+ 3	XX

XX: Don’t care

(Setting example)

Where the reset vector is defined as FF0000H, the NMI vector as FF9ABCH, and the INT1 vector as FF3456H

```
ORG     0FF0000H
LD      A, B
          ⋮
ORG     0FF9ABCH
LD      B, C
          ⋮
ORG     0FF3456H
LD      C, A
          ⋮
ORG     0FFFF00H
DL      0FF0000H      ; reset vector = FF0000H

ORG     0FFFF20H
DL      0FF9ABCH      ; NMI vector = FF9ABCH

ORG     0FFFF2CH
DL      0FF3456H      ; INT1 vector = FF3456H
```

Reference:

ORG and DL are assembler directives

ORG : For location counter control

DL : To define (32-bit) long word data

3.3.2 Micro DMA Processing

In addition to general-purpose interrupt processing, TMP95CS64/265 supports a micro DMA function. Interrupt requests set by the micro DMA perform micro DMA processing at the highest priority level of maskable interrupts (level 6), regardless of the priority level of the particular interrupt source.
Because the function of micro DMA has been implemented with the cooperative operation of CPU, when CPU is a state of stand-by by HALT instruction, the requirement of micro DMA will be ignored (pending).

(1) Micro DMA Operation

When an interrupt request is generated by an interrupt source specified by the micro DMA start vector register, the micro DMA triggers a micro DMA request to the CPU at interrupt priority level 6 and starts processing the request. The four micro DMA channels allow micro DMA processing to be set for up to four types of interrupts at any one time.
When micro DMA is accepted, the interrupt request flip-flop assigned to that channel is cleared. The data are automatically transferred from the transfer source address to the transfer destination address set in the control register, and the transfer counter is decremented by 1. If the decremented counter reads other than 0, DMA processing ends with no change in the value of the micro DMA start vector register. If the decremented reading is 0, the micro DMA transfer end interrupt (INTTC0 to 3) passes from the CPU to the interrupt controller. In addition, the micro DMA start vector register is cleared to 0, the next micro DMA is disabled, and micro DMA processing completes.
If a micro DMA request is set for more than one channel at a time, the priority is not based on the interrupt priority level but on the channel number: the smaller the channel number the higher the priority. (Channel 0 (high) --> channel 3 (low)).
If an interrupt request is triggered for the interrupt source in use during the interval between the clearing of the micro DMA start vector and the next setting, general-purpose interrupt processing executes at the interrupt level set. Therefore, if only using the interrupt for starting the micro DMA (not using the interrupt as a general-purpose interrupt), first set the interrupt level to 0 (interrupt requests disabled).
If using micro DMA and general-purpose interrupts together as described above, first set the level of the interrupt used to start micro DMA processing lower than all the other interrupt levels. In this case, the cause of general interrupt is limited to the edge interrupt.

Example: When using external interrupt INT0 to 3 to start micro DMA0 to 3, set:
External interrupt INT0 to 3 interrupt level ...... "1"
Level of other interrupts .... "2" to "6"

Like other maskable interrupts, the priority of the micro DMA transfer end interrupt is determined by the interrupt level and the default priority.

While the register for setting the transfer source/transfer destination addresses is a 32-bit control register, this register can only effectively output 24-bit addresses. Accordingly, micro DMA can access 16M bytes (the upper eight bits of the 32 bits are not valid).
Three micro DMA transfer modes are supported: 1-byte transfer, 2-byte (one word) transfer, and 4-byte transfer. After a transfer in any mode, the transfer source / destination addresses are incremented, decremented, or remain unchanged. This simplifies the transfer of data from I/O to memory, from memory to I/O, and from I/O to I/O. For details of the transfer modes, see 3.3.2 (4), Transfer Mode Register.
As the transfer counter is a 16-bit counter, micro DMA processing can be set for up to 65536 times per interrupt source. (The micro DMA processing count is maximized when the transfer counter initial value is set to 0000H.)
Micro DMA processing can be started by the 30 interrupts (INT0 to INTAD) shown in the micro DMA start vectors of Table 3.3 (2) and by the micro DMA soft start, making a total of 31 interrupts.
Figure 3.3 (2) shows the micro DMA cycle in transfer destination address INC mode (except for COUNTER mode, the same as for other modes).

① Word transfer (the conditions for this cycle are based on an external 16-bit bus, 0 waits, transfer source/transfer destination addresses both even-numbered values)

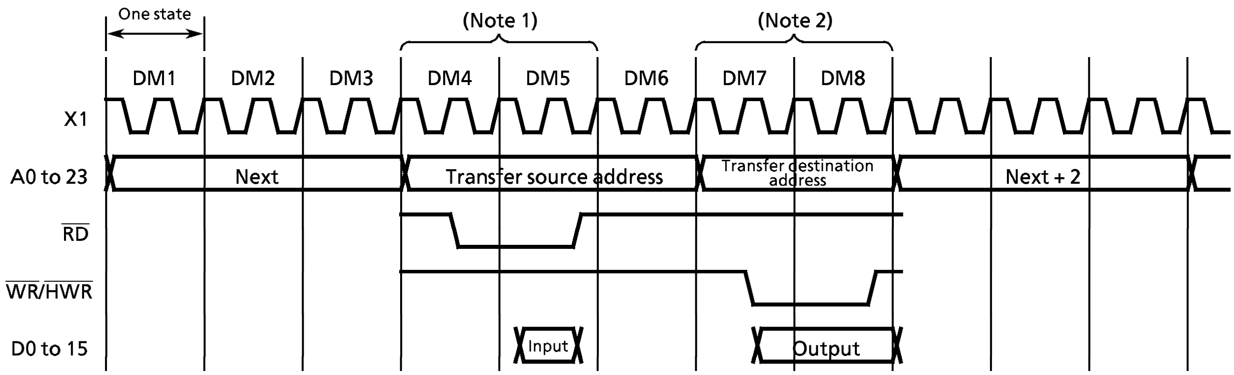


Figure 3.3.(2)-1 Timing of Micro DMA Cycle

States 1 to 3 : Instruction fetch cycle (gets next address code).
If three or more instruction codes are inserted in the instruction queue buffer, this cycle becomes a dummy cycle.

States 4 to 5 : Micro DMA read cycle
State 6 : Dummy cycle (the address bus remains as in state 5)
States 7 to 8 : Micro DMA write cycle

Note 1: If the source address area is an 8-bit bus, it is incremented by two states.

Note 2: If the destination address area is an 8-bit bus, it is incremented by two states.

② Word transfer (the conditions for this cycle are based on a 16-bit external bus, 0 waits, transfer source/transfer destination addresses both odd-numbered values)

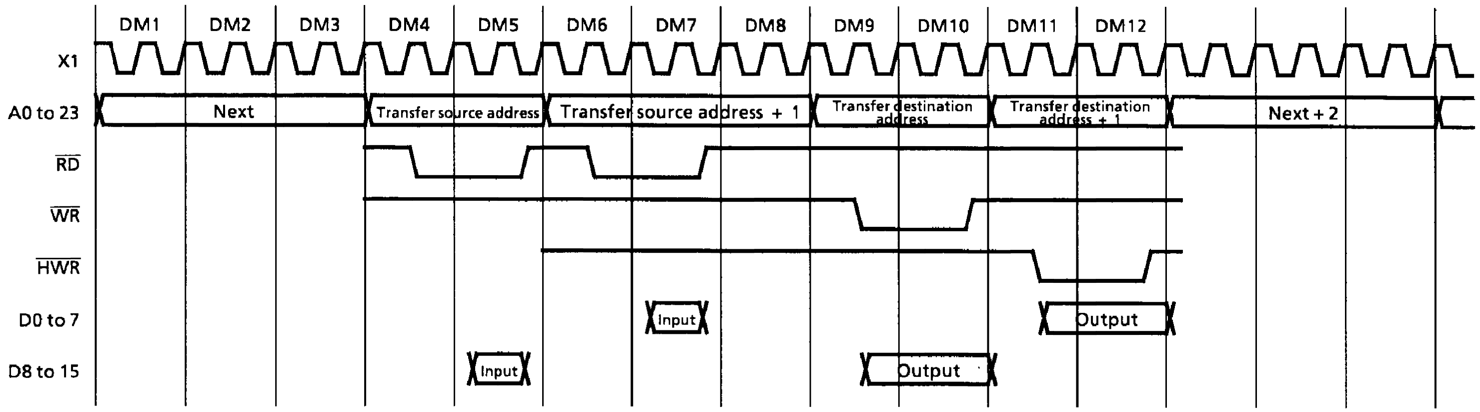


Figure 3.3 (2)-2 Timing of Micro DMA Cycle

③ 4-byte transfer (the conditions for this cycle are based on a 16-bit external bus, 0 waits, transfer source/transfer destination addresses both even-numbered values

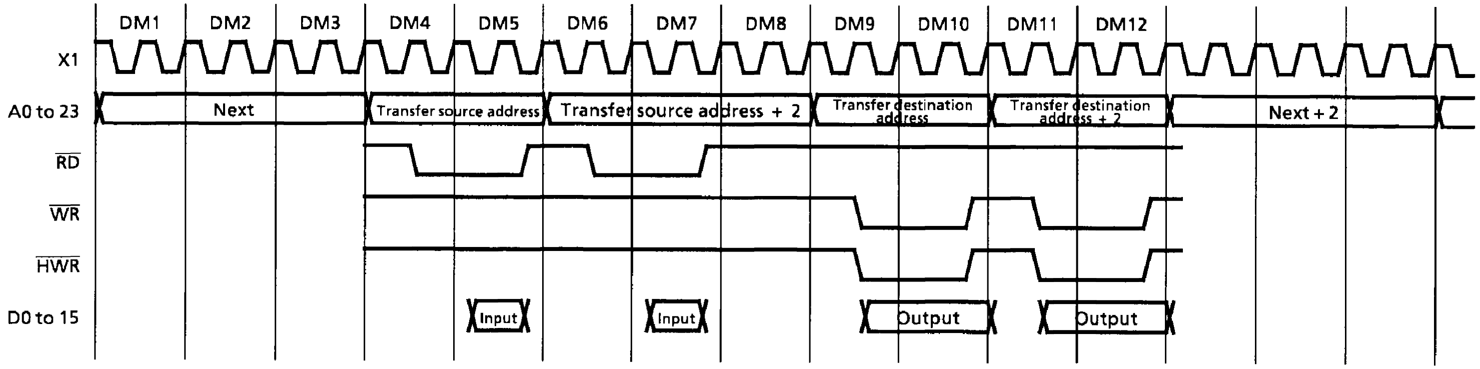


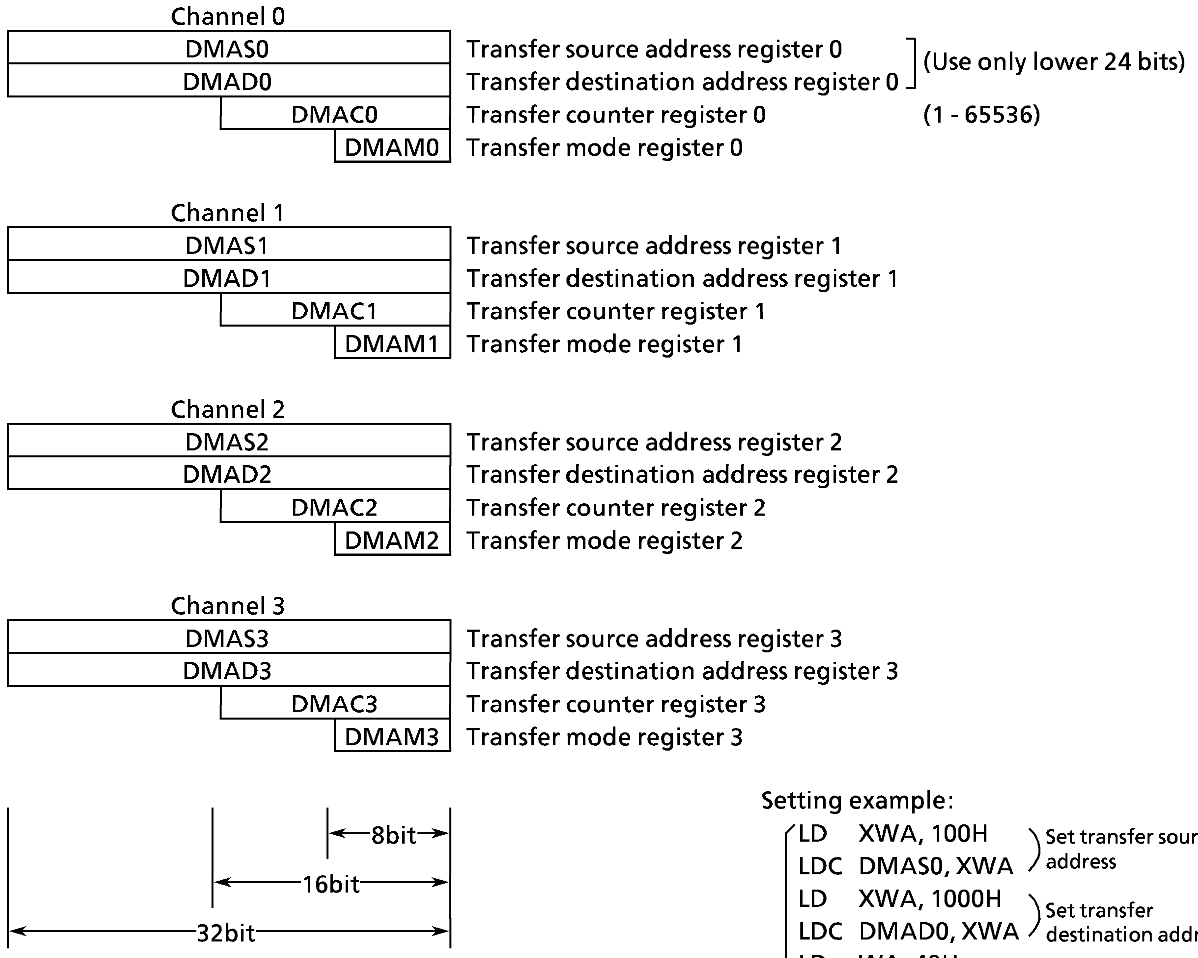
Figure 3.3 (2)-3 Timing of Micro DMA Cycle

(2) Micro DMA Soft Start Function

In addition to starting micro DMA by interrupt, TMP95CS64/265 supports a micro DMA soft start function. This starts micro DMA by generating a cycle to write to the soft DMA control register.
To code a soft start, write micro DMA start vector FCH to micro DMA start vector register DMA0V:3V (at memory addresses 5AH, 5BH, 5CH, and 5DH).
Then, write any data to soft DMA control register SDMACR0:3 (at memory addresses 6AH, 6BH, 6CH, and 6DH). (The value of the data has no effect on the operation of the soft start.) This starts micro DMA of the applicable channel once. Then, whenever data are written again to the soft DMA control register, as long as the micro DMA transfer counter register values are other than 0, a soft start can be continuously triggered (without rewriting the micro DMA start vector).
Setting the micro DMA start vector is a prerequisite for generating a micro DMA software start. (The software start request is a one-shot request and not saved. Therefore, even if a cycle which writes to the soft DMA control register is generated, unless the micro DMA start vector is already set, a soft start cannot be generated.)

(3) Structure of Micro DMA-Only Register

Figure 3.3 (3) shows the micro DMA-only registers. These registers are incorporated in the CPU. (See 3.2.5, Control Registers in Chapter 3, TLCS-900/H CPU.) To set the registers use the LDC instruction.
Set the transfer source address in the transfer source address register; the transfer destination address, in the transfer destination address register. These address registers use only the lower 24 bits. They support a 16M-byte address space.
Use the transfer counter register to set the number of times micro DMA is performed between 1 and 65536.
For details on setting the transfer mode register, see 3.3.2 (4), Transfer Mode Register.
Only the LDC cr, r instruction can load data into the micro DMA-only registers.



Transfer source address register 0 (Use only lower 24 bits)
Transfer destination address register 0
Transfer counter register 0 (1 - 65536)
Transfer mode register 0

Transfer source address register 1
Transfer destination address register 1
Transfer counter register 1
Transfer mode register 1

Transfer source address register 2
Transfer destination address register 2
Transfer counter register 2
Transfer mode register 2

Transfer source address register 3
Transfer destination address register 3
Transfer counter register 3
Transfer mode register 3

Setting example:

```
LD   XWA, 100H      ) Set transfer source
LDC  DMAS0, XWA     / address
LD   XWA, 1000H     ) Set transfer
LDC  DMAD0, XWA     / destination address
LD   WA, 40H        )
LDC  DMAC0, WA     /  Set transfer count
LD   A, 05H         ) Select transfer
LDC  DMAM0, A      /  mode
```

Figure 3.3 (3) Micro DMA-Only Registers

(4) Transfer Mode Register

To set micro DMA transfer mode, use transfer mode register DMAM0:3. Table 3.3 (3) shows the settings for each mode and the numbers of execution states.

Table 3.3 (3) Micro DMA Transfer Mode

	← 8-bit →			
DMAM0 to 3	0	0	0	Mode

Note: When setting a value in this register, write 0 to the upper three bit = s.

			Number of Transfer Bytes	Mode Description	Number of Execution States (※)	Minimum Execution Time @ fc = 25 MHz
000 (Fixed)	000	00	Byte transfer	Transfer destination address INC mode .... For I/O to memory (DMADn +) ← (DMASn) DMACn ← DMACn – 1 If DMACn = 0, then INTTCn generated	8 states	640 ns
		01	Word transfer			
		10	4-byte transfer		12 states	960 ns
	001	00	Byte transfer	Transfer destination address DEC mode .... For I/O to memory (DMADn –) ← (DMASn) DMACn←DMACn – 1 If DMACn = 0, then INTTCn generated	8 states	640 ns
		01	Word transfer			
		10	4-byte transfer		12 states	960 ns
	010	00	Byte transfer	Transfer source address INC mode .... For memory to I/O (DMADn) ← (DMASn +) DMACn←DMACn – 1 If DMACn = 0, then INTTCn generated	8 states	640 ns
		01	Word transfer			
		10	4-byte transfer		12 states	960 ns
	011	00	Byte transfer	Transfer source address DEC mode .... For memory to I/O (DMADn) ← (DMASn –) DMACn←DMACn – 1 If DMACn = 0, then INTTCn generated	8 states	640 ns
		01	Word transfer			
		10	4-byte transfer		12 states	960 ns
	100	00	Byte transfer	Address fixed mode .... For I/O to I/O (DMADn) ← (DMASn) DMACn←DMACn – 1 If DMACn = 0, then INTTCn generated	8 states	640 ns
		01	Word transfer			
		10	4-byte transfer		12 states	960 ns
	101	00	Counter mode ...... For counting number of times interrupts generated DMASn←DMASn + 1 DMACn←DMACn – 1 If DMACn = 0, then INTTCn generated		5 states	400 ns

(※) For external 16-bit bus, 0 waits, word/4-byte transfer mode, transfer source/transfer destination addresses both have even-numbered values.

Note: n: Corresponding micro DMA channels 0 to 3
DMADn + / DMASn + : Post increment (increments register value after transfer)
DMADn -/ DMASn- : Post decrement (decrements register value after transfer)
The I/Os in the table mean fixed addresses; memory means incremented and decremented addresses.
Do not use undefined code, that is, codes other than those listed above for the transfer mode register.

3.3.3 Interrupt Controller Control

Figure 3.3 (4) is a block diagram of the interrupt controller circuit. The left-hand side of this diagram shows the interrupt controller. The right-hand side shows the CPU interrupt request signal circuit and CPU halt release circuit. (For details on halt modes, see 3.4, Standby Function.)
The interrupt controller has a total of 36 interrupt channels, consisting of NMI, INTWD, INT0 to 8, INTT0 to 7, INTTR8 to O9, INTRX0 to TX2, INTAD, and INTTC0 to 3.
Each interrupt channel supports:

- Interrupt request flag (36 channels)
- Interrupt priority setting register (34 channels (NMI and INTWD excluded)).

In addition, there are also four channels of start vector registers for performing micro DMA processing.

(1) Interrupt request flags

The function of the interrupt request flag is to indicate the generation of an interrupt request. Apart from NMI and INTWD, each channel has a clear bit <IxxC> for clearing the interrupt requests (see Figure 3.3 (5), Interrupt Priority Setting Registers). Reading clear bit <IxxC> reads the state of the interrupt request flag and indicates whether an interrupt request is generated or not.
The interrupt request flags are zero-cleared by the following operations:

① A reset (clears all interrupt request flags)

② When the CPU accepts an interrupt and reads the vector of the accepted interrupt channel

③ When the CPU accepts the micro DMA request of the specified channel

④ When 0 is written to clear bit <IxxC> of the interrupt priority setting register

Note: ②, ③, and ④ operations do not include INT0 level mode or INTRX0, 1, or 2.

In addition, flags are also cleared by the following operations.

Table 3.3 (4) Other Flag Clearing Operations

Flag clearing source Interrupt source		Other operations that clear interrupt flags
INT0	Edge mode	Switching to level mode
	Level mode	Change in pin input after interrupt is generated (high level → low level)
INTRX0, 1, 2		Reading serial channel receive buffer

Before clearing an interrupt request by writing 0 to the clear bit or by performing a Table 3.3 (4) operation to clear the interrupt request flag, first execute the DI instruction.

(INT0 interrupt cautions)

Note the following cautions when using the INT0 interrupt in level mode.
In level mode, the INT0 pin input must be held continuously at high level until the interrupt response sequence completes. Likewise, when releasing the halt in this mode, the INT0 pin must be held continuously at high level until the halt is released.
When using INT0 level mode, be sure that an low level is not input as a result of noise (as this can cause malfunction).

When switching the INT0 pin operation mode from level to edge mode, first disable the INT0 interrupt as follows. (In level mode, an accepted interrupt request must be cleared.)

Setting example:

```
        DI                          ; disable interrupt
        LD (IIMC), XX0XXX0XB        ; switch from level to edge
        LD (INTE0AD), XXXX0nnnB     ; clear interrupt request flag and set INT0 interrupt
                                      level to n

        EI                          ; enable interrupt
```

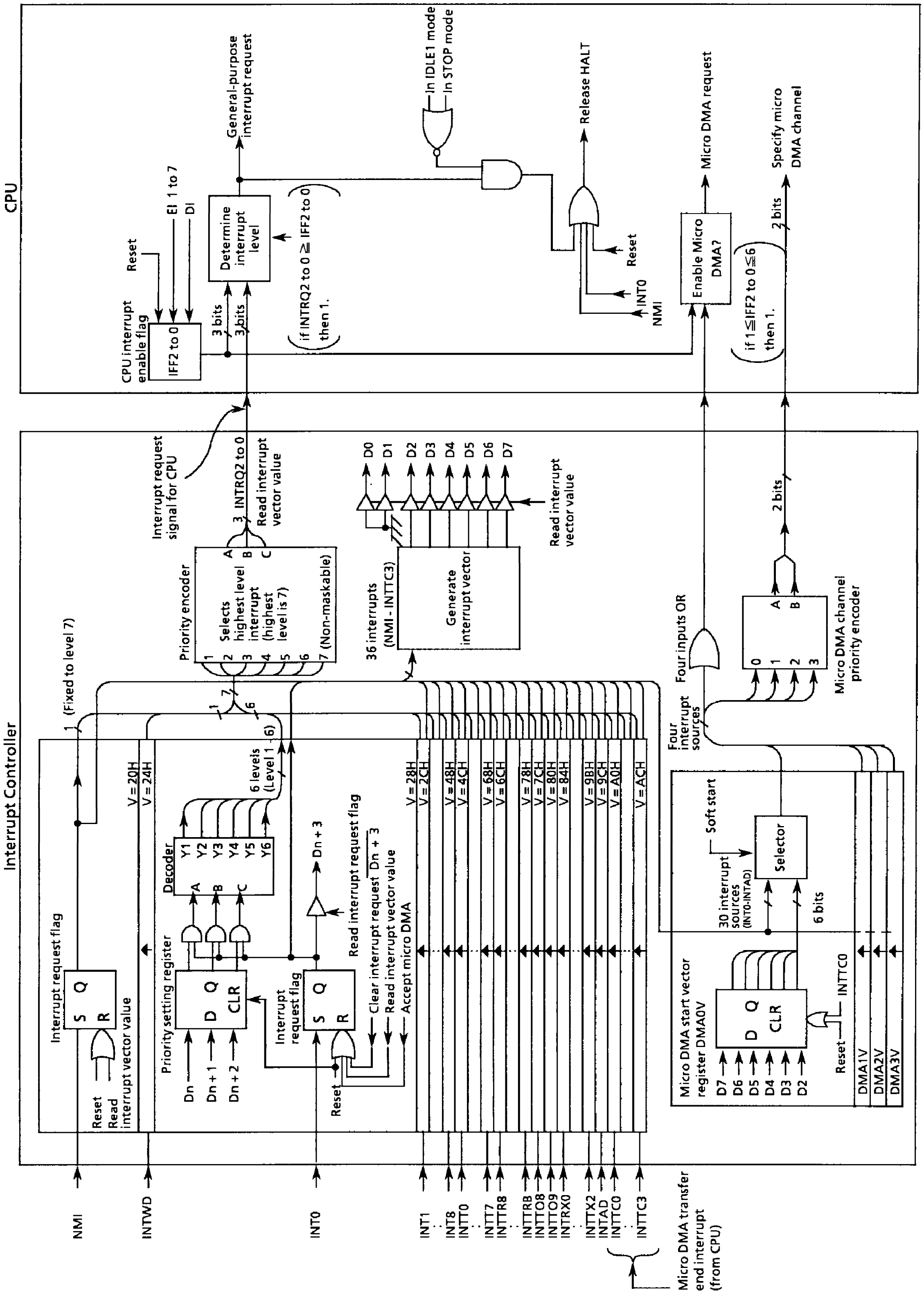


Figure 3.3 (4) Block Diagram of Interrupt Controller

(2) Interrupt Priority Setting Register

Figure 3.3 (5) shows the interrupt priority setting registers. Each of the 34 interrupt channels (INT0 to AD, INTTC0 to 3) has an interrupt request level setting bit <IxxM2:0>. An interrupt request is generated at six interrupt levels (levels 1 through 6). Setting the priority level to 0 (or 7) disables the corresponding interrupt request. The priority level for non-maskable interrupts ($\overline{\text{NMI}}$ pin input) is fixed to 7. If two or more interrupts with the same level occur simultaneously, the interrupts are accepted in accordance with the default priority.

Symbol	Address	7	6	5	4	3	2	1	0	
INTE0AD	70H	INTAD				INT0				←Interrupt source
		IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0	←bit Symbol
		R/W	W			R/W (Note)	W			←Read/Write
		0	0	0	0	0	0	0	0	←After reset
INTE12	71H	INT2				INT1				
		I2C	I2M2	I2M1	I2M0	I1C	I1M2	I1M1	I1M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTE34	72H	INT4				INT3				
		I4C	I4M2	I4M1	I4M0	I3C	I3M2	I3M1	I3M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTE56	73H	INT6				INT5				
		I6C	I6M2	I6M1	I6M0	I5C	I5M2	I5M1	I5M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTE78	74H	INT8				INT7				
		I8C	I8M2	I8M1	I8M0	I7C	I7M2	I7M1	I7M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTET01	75H	INTT1 (Timer 1)				INTT0 (Timer 0)				
		IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTET23	76H	INTT3 (Timer 3)				INTT2 (Timer 2)				
		IT3C	IT3M2	IT3M1	IT3M0	IT2C	IT2M2	IT2M1	IT2M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	

(Do not use read-modify-write instructions.)

Note: In INT0 level mode, writing 0 to <I0C> does not clear the interrupt request flag.

IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Disables interrupt request
0	0	1	Sets interrupt priority level to 1
0	1	0	Sets interrupt priority level to 2
0	1	1	Sets interrupt priority level to 3
1	0	0	Sets interrupt priority level to 4
1	0	1	Sets interrupt priority level to 5
1	1	0	Sets interrupt priority level to 6
1	1	1	Disables interrupt request

IxxC	Function (Read)	Function (Write)
0	No interrupt request	Clears interrupt request flag
1	Interrupt request	----- Don't care -----

Figure 3.3 (5) Interrupt Priority Setting Registers (1/2)

Symbol	Address	7	6	5	4	3	2	1	0	
INTET45	77H	INTT5 (Timer 5)				INTT4 (Timer 4)				←Interrupt source
		IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0	←bit Symbol
		R/W	W			R/W	W			←Read/Write
		0	0	0	0	0	0	0	0	←After reset
INTET67	78H	INTT7 (Timer 7)				INTT6 (Timer 6)				
		IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTET89	79H	INTTR9 (TREG9)				INTTR8 (TREG8)				
		IT9C	IT9M2	IT9M1	IT9M0	IT8C	IT8M2	IT8M1	IT8M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTETAB	7AH	INTTRB (TREGB)				INTTRA (TREGA)				
		ITBC	ITBM2	ITBM1	ITBM0	ITAC	ITAM2	ITAM1	ITAM0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTEOV	7BH	INTTO9				INTTO8				
		ITO9C	ITO9M2	ITO9M1	ITO9M0	ITO8C	ITO8M2	ITO8M1	ITO8M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTES0	7CH	INTTX0				INTRX0				
		ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0	
		R/W	W			R (Note)	W			
		0	0	0	0	0	0	0	0	
INTES1	7DH	INTTX1				INTRX1				
		ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0	
		R/W	W			R (Note)	W			
		0	0	0	0	0	0	0	0	
INTES2	7EH	INTTX2				INTRX2				
		ITX2C	ITX2M2	ITX2M1	ITX2M0	IRX2C	IRX2M2	IRX2M1	IRX2M0	
		R/W	W			R (Note)	W			
		0	0	0	0	0	0	0	0	
INTETC01	7FH	INTTC1				INTTC0				
		ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTETC23	80H	INTTC3				INTTC2				
		ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	

(Do not use read-modify-write instructions.)

IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Disables interrupt request.
0	0	1	Sets interrupt priority level to 1
0	1	0	Sets interrupt priority level to 2
0	1	1	Sets interrupt priority level to 3
1	0	0	Sets interrupt priority level to 4
1	0	1	Sets interrupt priority level to 5
1	1	0	Sets interrupt priority level to 6
1	1	1	Disables interrupt request

Note: As <IRX0C>, <IRX1C>, and <IRX2C> are read-only registers, writing 0 to them does not clear the interrupt request flag.

IxxC	Function (Read)	Function (Write)
0	No interrupt request	Clears interrupt request flag
1	Interrupt request	- - - - - Don't care - - - - -

Figure 3.3 (5) Interrupt Priority Setting Registers (2/2)

From among simultaneous interrupts, the interrupt controller selects the interrupt request with the highest level and sends its vector address to the CPU.

Then, the CPU compares the priority level of the interrupt request with the value of the interrupt mask register <IFF2:0> in the status register. If the priority level of the interrupt request is higher than the value of the interrupt mask register, the CPU accepts the interrupt. When the CPU side interrupt mask register <IFF2:0> is set to the priority level of the received interrupt incremented by 1, subsequent interrupt requests are only accepted if their level is equal to or greater than the incremented value.

(3) Micro DMA Start Vector

The interrupt controller has four channels of micro DMA start vector registers. Writing the micro DMA start vector value (Table 3.3 (2)) for each interrupt source to these registers makes the applicable interrupt request into a micro DMA request. But first set values in the registers for micro DMA parameters (DMAS, DMAD, DMAC, DMAM). Figure 3.3 (6) shows the micro DMA start vector registers.

The function of the micro DMA start vector registers is to select the interrupt to use with micro DMA processing. The micro DMA start source is assigned to the interrupt source whose micro DMA start vector matches the vector value set in the micro DMA start vector register.

When the value of the micro DMA transfer counter is set to 0 after micro DMA processing, the CPU generates a micro DMA transfer end interrupt (INTTC0 to 3) corresponding to the micro DMA start vector register. When the micro DMA start vector register is cleared, the micro DMA startup source is released. Therefore, when continuously performing micro DMA processing, set the start vector value in the micro DMA start vector register again during processing of the micro DMA transfer end interrupt.

When the same vector is set in the micro DMA start vector registers of multiple channels, the lower the channel number the higher the priority.

The channel with the lowest number is executed until the micro DMA transfer end interrupt. Unless the micro DMA start vector is set again during the processing of the micro DMA transfer end interrupt, the subsequent micro DMA startup moves to the next smallest channel number. (This operation is called a micro DMA chain.)

Micro DMA0 start vector register

DMA0V
(005AH)
(Do not use read-modify-write instructions.)

	7	6	5	4	3	2	1	0
bit Symbol	DMA0V7	DMA0V6	DMA0V5	DMA0V4	DMA0V3	DMA0V2		
Read/Write	W							
After reset	0	0	0	0	0	0		
Function	Set startup interrupt source for micro DMA channel 0							

Micro DMA1 start vector register

DMA1V
(005BH)
(Do not use read-modify-write instructions.)

	7	6	5	4	3	2	1	0
bit Symbol	DMA1V7	DMA1V6	DMA1V5	DMA1V4	DMA1V3	DMA1V2		
Read/Write	W							
After reset	0	0	0	0	0	0		
Function	Set startup interrupt source for micro DMA channel 1							

Micro DMA2 start vector register

DMA2V
(005CH)
(Do not use read-modify-write instructions.)

	7	6	5	4	3	2	1	0
bit Symbol	DMA2V7	DMA2V6	DMA2V5	DMA2V4	DMA2V3	DMA2V2		
Read/Write	W							
After reset	0	0	0	0	0	0		
Function	Set startup interrupt source for micro DMA channel 2							

Micro DMA3 start vector register

DMA3V
(005DH)
(Do not use read-modify-write instructions.)

	7	6	5	4	3	2	1	0
bit Symbol	DMA3V7	DMA3V6	DMA3V5	DMA3V4	DMA3V3	DMA3V2		
Read/Write	W							
After reset	0	0	0	0	0	0		
Function	Set startup interrupt source for micro DMA channel 3							

Setting micro DMA startup source

Micro DMA startup source	Value set in micro DMA start vector register	Micro DMA startup source	Value set in micro DMA start vector register
INT 0 interrupt	28H	INTT 7 interrupt	68H
INT 1 interrupt	2CH	INTTR 8 interrupt	6CH
INT 2 interrupt	30H	INTTR 9 interrupt	70H
INT 3 interrupt	34H	INTTR A interrupt	74H
INT 4 interrupt	38H	INTTR B interrupt	78H
INT 5 interrupt	3CH	INTTO 8 interrupt	7CH
INT 6 interrupt	40H	INTTO 9 interrupt	80H
INT 7 interrupt	44H	INTRX 0 interrupt	84H
INT 8 interrupt	48H	INTTX 0 interrupt	88H
INTT 0 interrupt	4CH	INTRX 1 interrupt	8CH
INTT 1 interrupt	50H	INTTX 1 interrupt	90H
INTT 2 interrupt	54H	INTRX 2 interrupt	94H
INTT 3 interrupt	58H	INTTX 2 interrupt	98H
INTT 4 interrupt	5CH	INTAD interrupt	9CH
INTT 5 interrupt	60H	Micro DMA soft start	FCH
INTT 6 interrupt	64H		

Figure 3.3 (6) Setting Micro DMA Start Vector Register and Startup Source

(4) External Interrupt Control

Table 3.3 (5) shows the function settings for the external interrupt pins.
TMP95CS64/265 can select the operating mode for the $\overline{\text{NMI}}$, INT0, INT5, or INT7 pins from among external interrupt functions. (For details on the external interrupt function pulse width, see "4.8 Interrupt Operations".)

Table 3.3 (5) Setting Functions on External Interrupt Pins

Interrupt pin	Shared pin	Mode	Setting method
$\overline{\text{NMI}}$	—	Falling edge	IIMC<NMIREE> = 0
		Both falling and rising edges	IIMC<NMIREE> = 1
INT0	P56	Rising edge	IIMC<I0LE> = 0, <I0IE> = 1
		Level	IIMC<I0LE> = 1, <I0IE> = 1
INT1	P70	Rising edge	———
INT2	P72	Rising edge	———
INT3	P73	Rising edge	———
INT4	P70	Rising edge	———
INT5	P90	Rising edge	T8MOD<CAP12M1:0> = 0, 0 or 0, 1, or 1, 1
		Falling edge	T8MOD<CAP12M1, 0> = 1, 0
INT6	P91	Rising edge	———
INT7	P94	Rising edge	T9MOD<CAP34M1:0> = 0, 0 or 0, 1, or 1, 1
		Falling edge	T9MOD<CAP34M1, 0> = 1, 0
INT8	P95	Rising edge	———

The input mode of the NMI and INT0 interrupts can be controlled by interrupt input mode control register IIMC.
Figure 3.3 (7) shows the interrupt input mode control register.

Interrupt Input Mode Control Register

IIMC
(0059H)

(Do not use read-modify-write instructions.)

	7	6	5	4	3	2	1	0
bit Symbol			–			I0IE	I0LE	NMIREE
Read/Write			W			W		
After reset			0			0	0	0
Function			Note: Be sure to write 0.			INT0 input control 0: Disabled 1: Enabled	INT0 input control 0: Rising edge 1: High level	NMI interrupt control 0: Falling edge 1: Both edges

Note: The INT0 pin can also be used for the halt release described later.
When using as port 56 in HALT mode, be sure to set <I0IE> to 0 before entering HALT mode.

NMI interrupt control

0	Generates interrupt request on falling edge
1	Generates interrupt request on both edges

INT0 interrupt control

0	Generates interrupt request on rising edge
1	Generates interrupt request on high level

INT0 input control

0	INT0 disabled
1	INT0 enabled

Figure 3.3 (7) Interrupt Input Mode Control Register

(5) Caution

When the CPU fetches an instruction to clear the interrupt request flag for the interrupt controller immediately before an interrupt is generated, the CPU may execute the instruction between receiving the interrupt and reading the interrupt vector.
To avoid the above occurring, clear the interrupt request flag by entering the instruction to clear the flag after the DI instruction. In the case of setting an interrupt enable again by EI instruction after the execution of clearing instruction, execute EI instruction after clearing instruction and following more than one instruction are executed. When EI instruction is placed immediately after clearing instruction, an interrupt becomes enable before interrupt request flags are cleared.
In the case of changing the value of the interrupt mask register<IFF2:0> by execution of POP SR instruction, disable an interrupt by DI instruction before execution of POP SR instruction.

3.4 Standby Function

(1) HALT modes

When TMP95CS64/265 executes the HALT instruction, WDMOD<HALTM1:0> of the watchdog timer mode register can be used to set one of the following HALT modes: RUN, IDLE2, IDLE1, STOP. Figure 3.4 (1) shows the watchdog timer mode control register.

Watchdog Timer Mode Control Register

WDMOD (006EH)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
Read/Write	R/W							
After reset	1	0	0	0	0	0	0	0
Function	Controls watchdog timer 0: Disable 1: Enable	Selects watchdog timer detection time 00: 2^{16}/fc 01: 2^{18}/fc 10: 2^{20}/fc 11: 2^{22}/fc		Warmup time 0: 2^{14}/fc 1: 2^{16}/fc	Selects HALT mode 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		Controls internal reset at runaway detection 1: Performs internal reset by detecting runaway	Pin control at STOP mode 1: Drives pins even during STOP mode

Controls pin state in STOP mode

0	Input/output off
1	Maintains state prevailing before HALT

Controls internal reset at runaway detection

0	Don't care
1	Executes internal reset on runaway detection

Sets HALT mode

00	RUN mode (CPU only halted)
01	STOP mode (all circuits halted)
10	IDLE1 mode (oscillator only operating)
11	IDLE2 mode (some I/O only operating)

Selects warmup time when returning from STOP mode

0	2^{14}/fc (around 655 μs @ 25 MHz)
1	2^{16}/fc (around 2.6 ms @ 25 MHz)

Selects watchdog timer detection time

00	2^{16}/fc (around 2.6 ms @ 25 MHz)
01	2^{18}/fc (around 10.5 ms @ 25 MHz)
10	2^{20}/fc (around 41.9 ms @ 25 MHz)
11	2^{22}/fc (around 167.8 ms @ 25 MHz)

Controls watchdog timer enable/disable

0	Disable
1	Enable

Figure 3.4 (1) Watchdog Timer Mode Control Register

The characteristics of RUN, IDLE2, IDLE1, and STOP modes are as follows:

① RUN : In this mode, the CPU only is halted. Power dissipation is almost the same as when the CPU is operating.

② IDLE2 : The internal oscillator and specific internal I/O only operate. Power dissipation is around one half that when the CPU is operating.

③ IDLE1 : Only the internal oscillator operates; all other circuits are halted. Power dissipation is one tenth of operating mode dissipation.

④ STOP : All internal circuits, including the internal oscillator, are halted. In this mode power dissipation drops considerably.

Table 3.4 (1) shows the operation of all blocks in HALT modes.

Table 3.4 (1) Blocks and I/O Pin Operation in Halt Modes

Halt mode		RUN	IDLE2	IDLE1	STOP
WDMOD <HALTM1, 0>		00	11	10	01
Operating block	CPU	Halted			
	I/O ports	Maintains state prevailing at HALT instruction execution			See Table 3.4 (3)
	8-bit timers	Operating	Operating	Halted	Halted
	16-bit timers	Operating	Operating	Halted	Halted
	Serial channels	Operating	Operating	Halted	Halted
	A/D converter	Operating	Halted	Halted	Halted
	D/A converter	Operating	Halted	Halted	Halted
	Watchdog timer	Operating	Operating	Halted	Halted
	Interrupt controller	Operating	Operating	Halted	Halted

(2) Release from HALT mode

Release from HALT mode can trigger an interrupt request or a reset. A combination of the interrupt mask register <IFF2:0> state and the halt mode determine the useable halt release source. (For details, see Table 3.4 (2).)

- Release by interrupt request

The operation to release HALT mode by using an interrupt request differs according to the interrupt enable state. If the interrupt request level set prior to the execution of the HALT instruction is higher than the interrupt mask register value, after HALT mode is released, interrupt processing is performed by this source, and processing starts from the next instruction following the HALT instruction. If the interrupt request level is lower than the interrupt mask register value, HALT mode is not released. (At a non-maskable interrupt, interrupt processing is performed after HALT mode release irrespective of the mask register value.)
However, in the case of the INT0 interrupt only, HALT mode can be released if the interrupt request level is lower than the interrupt mask register value. In this case the interrupt processing is not performed. Processing always starts from the next instruction following the HALT instruction. (The INT0 interrupt request flag is held at 1.)

Note: Usually, interrupts can release all halts status. However, the interrupts = ($\overline{\text{NMI}}$, INT0), which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 3 clocks of X1) with IDLE1 or STOP mode (IDLE2 is not applicable to this case). (In this case, an interrupt request is kept on hold internally.)
If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficultly. The priority of this interrupt is compare with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.

- Release by reset

  All HALT modes can be released by a reset. However, when releasing STOP mode, allow sufficient reset time (at least 3 ms) for the oscillator to stabilize.
  When releasing HALT mode by a reset, the internal RAM retains the data prevailing immediately prior to entering the HALT mode. However, other settings are initialized.

Table 3.4 (2) Halt Release Sources and Halt Release Operation

Interrupt accept state			Interrupt enabled (interrupt request level) ≧ (interrupt mask)				Interrupt disabled (interrupt request level) < (interrupt mask)			
HALT mode			RUN	IDLE2	IDLE1	STOP	RUN	IDLE2	IDLE1	STOP
HALT release source	Interrupt source	NMI	◎	◎	◎	◎ *1	–	–	–	–
		INTWD	◎	×	×	×	–	–	–	–
		INT0	◎	◎	◎	◎ *1	○	○	○	○ *1
		INT1 to 8	◎	◎	×	×	×	×	×	×
		INTT0 to 7	◎	◎	×	×	×	×	×	×
		INTTR8, 9, A, B	◎	◎	×	×	×	×	×	×
		INTTO8, 9	◎	◎	×	×	×	×	×	×
		INTRX0, TX0	◎	◎	×	×	×	×	×	×
		INTRX1, TX1	◎	◎	×	×	×	×	×	×
		INTRX2, TX2	◎	◎	×	×	×	×	×	×
		INTAD	◎	×	×	×	×	×	×	×
	RESET		◎	◎	◎	◎	◎	◎	◎	◎

◎ : After HALT mode release, the CPU starts interrupt processing (a reset initializes the LSI).

○ : After HALT mode release, processing starts from the next instruction following the HALT instruction. (No interrupt processing)

× : Not used for HALT release.

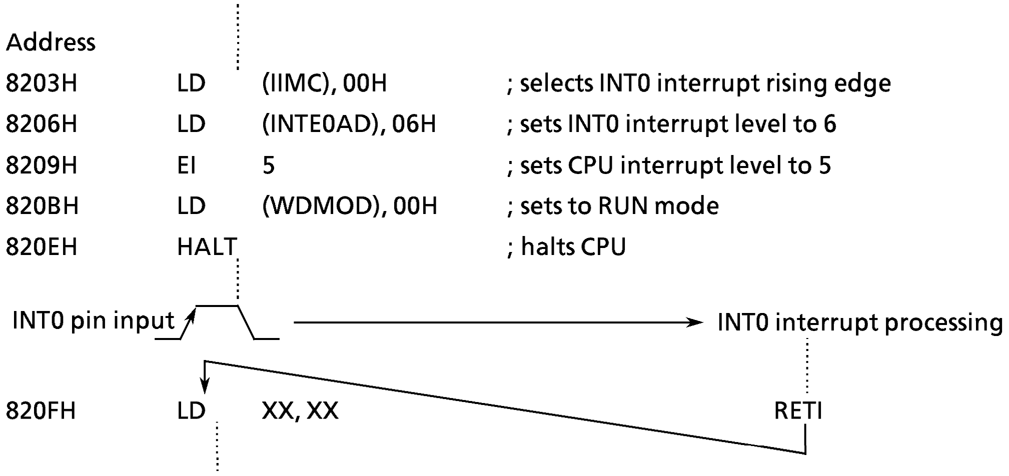
– : As the highest priority level (interrupt request level) for a non-maskable interrupt is fixed to 7, this combination is not available.

*1 : Releases HALT after the warmup time has elapsed.

Note: When releasing HALT in an interrupt enabled state by using a level mode INT0 interrupt, maintain high level on pin INT0 until interrupt processing begins. If pin INT0 changes to low level before interrupt processing begins, interrupt processing cannot start correctly.

(Example of release from HALT mode)

Releasing HALT mode using the edge mode INT0 interrupt when the CPU is in RUN mode:



```
Address
8203H     LD     (IIMC), 00H        ; selects INT0 interrupt rising edge
8206H     LD     (INTE0AD), 06H     ; sets INT0 interrupt level to 6
8209H     EI     5                  ; sets CPU interrupt level to 5
820BH     LD     (WDMOD), 00H       ; sets to RUN mode
820EH     HALT                      ; halts CPU

 INT0 pin input                     INT0 interrupt processing

820FH     LD     XX, XX                                    RETI
```

(3) Operation in each mode

① RUN mode

In RUN mode, the system clock continues operating even after execution of the HALT instruction. Only the CPU instruction execution operations stop.
In HALT mode, interrupt requests are sampled on the falling edge of the CLK signal.
All the external and internal interrupts can be used for releasing RUN mode. (See Table 3.4 (2), Halt Release Sources and Halt Release Operation.)

Figure 3.4 (2) shows the timing example for releasing HALT mode using an interrupt.

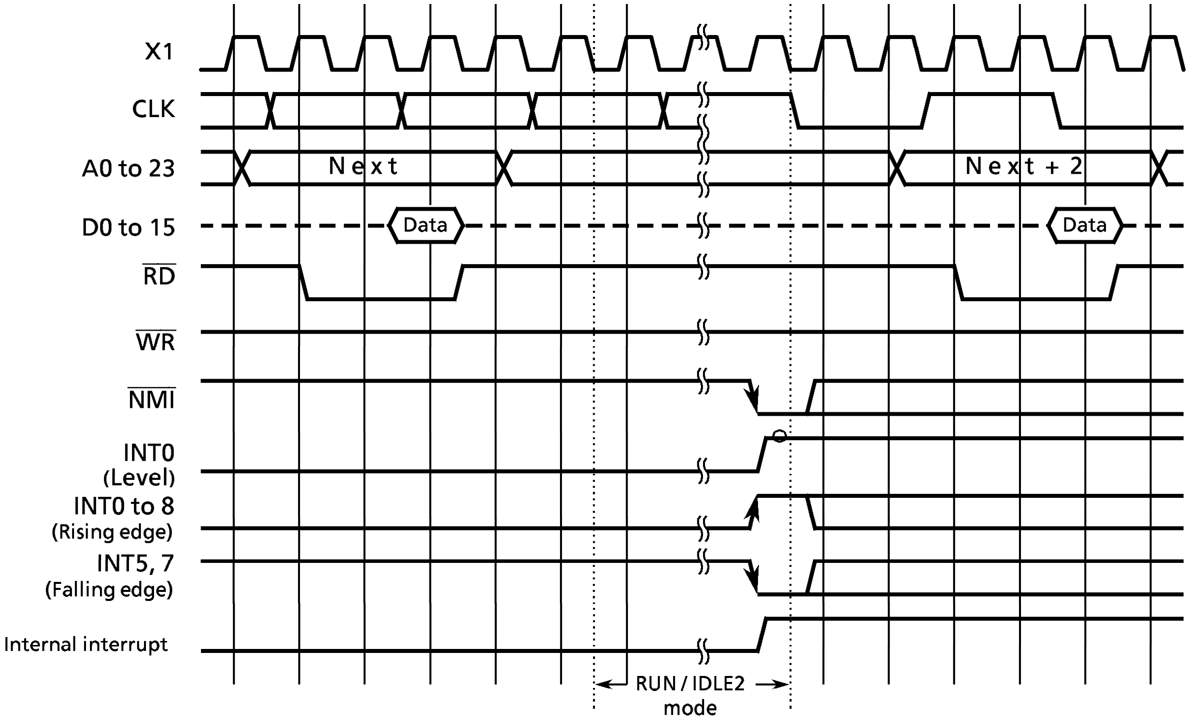


Figure 3.4 (2) Example of Timing for Releasing Halt by Interrupt (RUN or IDLE2 Mode)

② IDLE2 Mode

In IDLE2 mode, the system clock is supplied only to specific internal I/O. CPU instruction execution halts.
In IDLE2 mode, the timing for releasing HALT mode by interrupt is the same as in RUN mode.
External and internal interrupts, apart from INTWD/INTAD, can release IDLE2 mode. (See Table 3.4 (2), Halt Release Sources and Halt Release Operation.)
Before entering HALT mode in IDLE2 mode, disable the watchdog timer (to prevent the generation of a watchdog timer interrupt immediately after halt mode release).

③ IDLE1 Mode

In IDLE1 mode, only the internal oscillator operates. The system clock stops. The CLK pin outputs high level.
The interrupt request sampling in HALT mode is asynchronous to the system clock. However, the release (resumption of operation) is synchronous.
Release IDLE1 mode by an external interrupt (NMI, INT0). (See Table 3.4 (2), Halt Release Sources and Halt Release Operation.)

Figure 3.4 (3) shows the timing example for releasing HALT mode by interrupt.

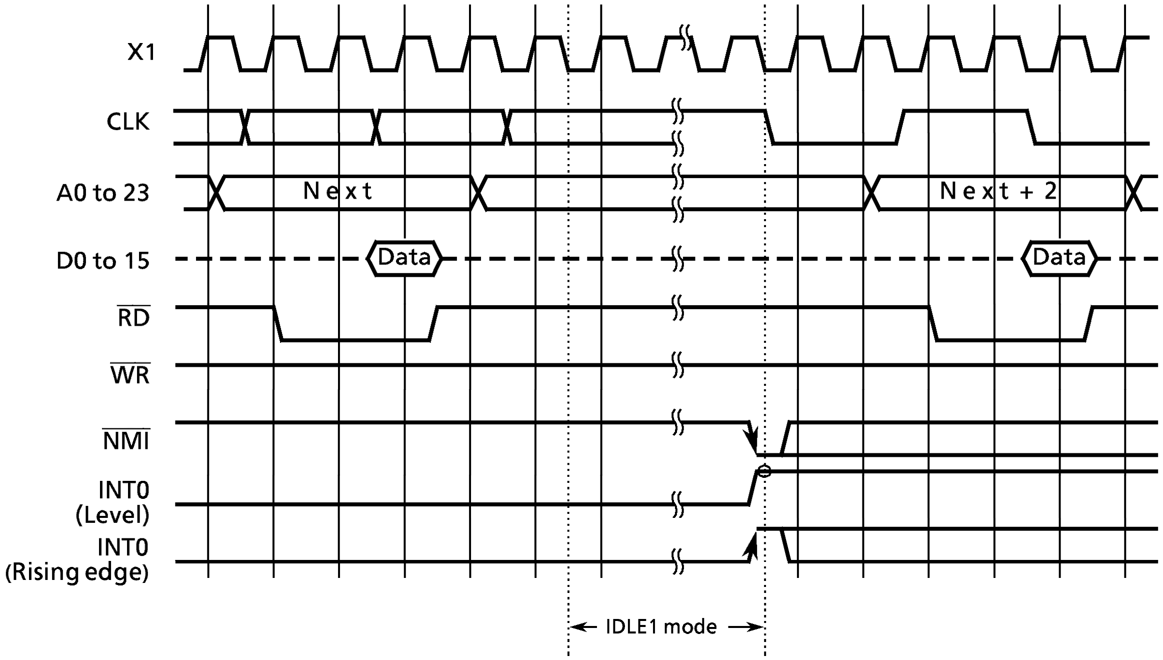


Figure 3.4 (3) Example of Timing for Releasing HALT by Interrupt (IDLE1 Mode)

④ STOP Mode

In STOP mode, all internal circuits, including the internal oscillator, are halted. The pin states in STOP mode differ according to the setting of watchdog timer mode register WDMOD<DRVE>. (For details on the WDMOD<DRVE> settings, see Figure 3.4 (1)). Figure 3.4 (3) shows the pin states in STOP mode.
Release STOP mode by an external interrupt (NMI, INT0). When releasing STOP mode, system clock output starts after the elapse of the warmup time (as set in the warmup counter) to stabilize the internal oscillator. Set the warmup time in the WDMOD<WARM> register.

Figure 3.4 (4) shows an example of the timing for releasing HALT by interrupt.

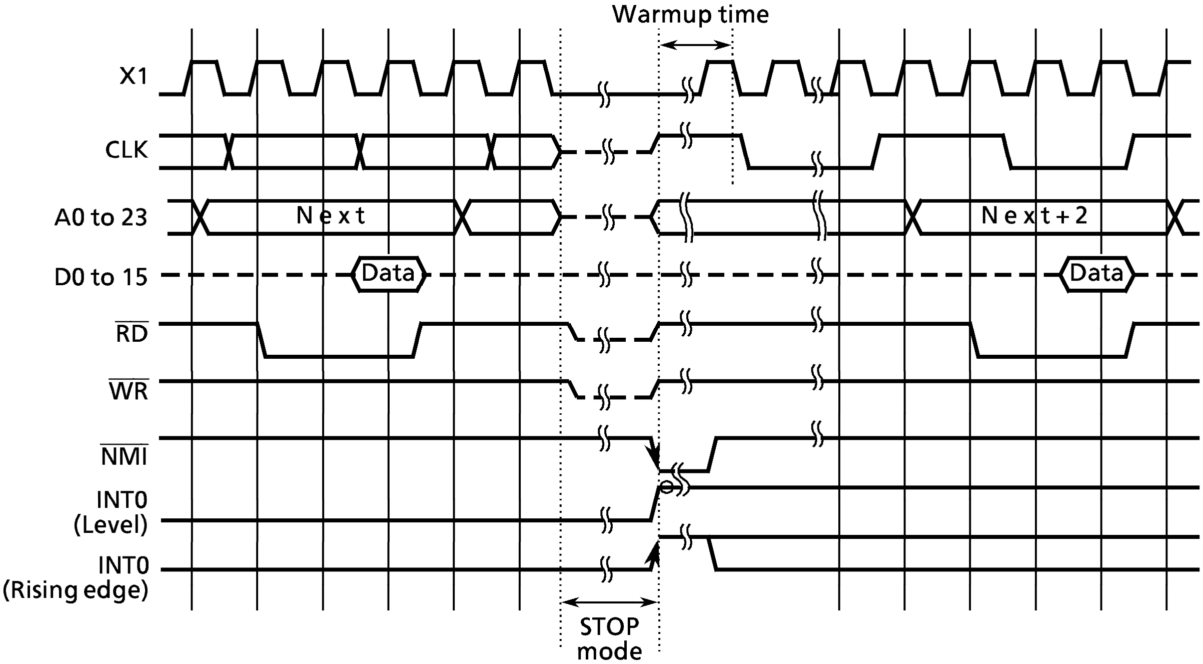


Figure 3.4 (4) Example of Timing for Releasing HALT by Interrupt (STOP Mode)

Table 3.4 (3) Pin States in Stop Mode

Pin Name	Input/Output	TMP95CS64		TMP95C265	
		<DRVE> = 0	<DRVE> = 1	<DRVE> = 0	<DRVE> = 1
P00 to 07	Input mode Output mode Input/output (D0 to D7)	▲ ▲ –	▲ Output –	X X –	X X –
P10 to 17	Input mode Output mode Input/output (D8 to D15)	▲ ▲ –	▲ Output –	▲ ▲ –	▲ Output –
P20 to 27	Input mode Output mode Output (A16 to A23)	▲ ▲ –	▲ Output Output	▲ ▲ –	▲ Output Output
P30 to 37	Input mode Output mode Output (A8 to A15)	▲ ▲ –	▲ Output Output	X X –	X X Output
P40 to 47	Input mode Output mode Output (A0 to A7)	▲ ▲ –	▲ Output Output	X X –	X X Output
P50 ($\overline{RD}$), P51 ($\overline{WR}$)	Output mode Output ($\overline{RD}$, $\overline{WR}$)	▲ –	Output High level output	X –	X High level output
P52 to 55, P57	Input mode Output mode	PU* PU	PU Output	Same as at left	
P56 (INT0)	Input mode Output mode Input mode (INT0)	PU PU Input	PU Output Input		
P60 to 63	Output mode	–	Output		
P70 to 75	Input mode Output mode	– –	Input Output		
P80, 83, 86	Input mode Output mode	PU* PU*	PU Output		
P81, 82, 84, 85, 87	Input mode Output mode	PU* PU	PU Output		
P90 to 97	Input mode Output mode	– –	Input Output		
PA0 to 7 (AN0 to 7)	Input Input ($\overline{ADTRG}$)	▲ –	▲ Input		
DAOUT 0, 1	Output	Output (0 V)	Output (0 V)		
$\overline{NMI}$	Input	Input	Input		
CLK	Output	–	High level output		
$\overline{RESET}$	Input	Input	Input		
$\overline{EA}$	Input	Fixed to High level	Fixed to High level	Fixed to low level	Fixed to low level
AM8/$\overline{16}$	Input	Fixed to High level	Fixed to High level	Input	Input
X1	Input	–	–	Same as at left	
X2	Output	High level	High level		

– : Indicates that input is invalid for an input pin or a pin in input mode. Also, that the pin is set to high impedance for an output pin or a pin in output mode.

Input : The input gate is functioning. To prevent the input pin from floating, fix the input voltage to low or high.

Output : Output state

PU : Programmable pull-up pin. The input gate is functioning. Pins without pull-up set must be fixed to prevent through current.

PU* : Programmable pull-up pin. The input gate is disabled. A through current does not occur even if high impedance is set.

▲ : The input gate continues to operate if the HALT instruction is executed and the CPU is halted at the port register address value. To prevent a through current in this case, either fix the pin or ensure by software that the situation does not occur. In other cases, input is invalid.

X : Cannot be used.

Note : The port register controls the programmable pull-up. However, if the function is set for a pin shared with an output function (eg, TxD0), the pull-up selection for the pin depends on the output function data. For pins that are shared with input functions, the port register setting alone determines whether or not a pull-up resistor is used.

3.5 Port Functions

TMP95CS64 has a total of 81 bits for input/output ports. Assuming that external memory is connected to TMP95C265, the total number of bits available for input/output ports is 47. (Ports 0 and 1 are the data bus, ports 3 and 4 are the address bus, and P50 and P51 are used exclusively as the read and write pins respectively.)

As well as being used as general-purpose I/O ports, port pins are also used for internal CPU and built-in I/O functions. Table 3.5 (1) lists port pin functions; Table 3.5 (2), pin settings.

Table 3.5 (1) Port Pin Functions

Port Name	Pin Name	Number of Pins	Direction	R	Direction Setting Unit	Pin Name for Built-In Function
Port 0	P00 to P07	8	Input/output	–	Bit	D0 to D7
Port 1	P10 to P17	8	Input/output	–	Bit	D8 to D15
Port 2	P20 to P27	8	Input/output	–	Bit	A16 to A23
Port 3	P30 to P37	8	Input/output	–	Bit	A8 to A15
Port 4	P40 to P47	8	Input/output	–	Bit	A0 to A7
Port 5	P50	1	Output	–	(Fixed)	$\overline{\text{RD}}$
	P51	1	Output	–	(Fixed)	$\overline{\text{WR}}$
	P52	1	Input/output	↑	Bit	$\overline{\text{HWR}}$
	P53	1	Input/output	↑	Bit	$\overline{\text{BUSRQ}}$
	P54	1	Input/output	↑	Bit	$\overline{\text{BUSAK}}$
	P55	1	Input/output	↑	Bit	$\overline{\text{WAIT}}$
	P56	1	Input/output	↑	Bit	INT0
	P57	1	Input/output	↑	Bit	SCLK2/$\overline{\text{CTS2}}$
Port 6	P60	1	Output	–	(Fixed)	$\overline{\text{CS0}}$
	P61	1	Output	–	(Fixed)	$\overline{\text{CS1}}$
	P62	1	Output	–	(Fixed)	$\overline{\text{CS2}}$
	P63	1	Output	–	(Fixed)	$\overline{\text{CS3}}$
Port 7	P70	1	Input/output	–	Bit	TI0/INT1
	P71	1	Input/output	–	Bit	TO1
	P72	1	Input/output	–	Bit	TO3/INT2
	P73	1	Input/output	–	Bit	TI3/INT3
	P74	1	Input/output	–	Bit	TO5
	P75	1	Input/output	–	Bit	TO7/INT4
Port 8	P80	1	Input/output	↑	Bit	TxD0
	P81	1	Input/output	↑	Bit	RxD0
	P82	1	Input/output	↑	Bit	SCLK0/$\overline{\text{CTS0}}$
	P83	1	Input/output	↑	Bit	TxD1
	P84	1	Input/output	↑	Bit	RxD1
	P85	1	Input/output	↑	Bit	SCLK1/$\overline{\text{CTS1}}$
	P86	1	Input/output	↑	Bit	TxD2
	P87	1	Input/output	↑	Bit	RxD2
Port 9	P90	1	Input/output	–	Bit	TI8/INT5
	P91	1	Input/output	–	Bit	TI9/INT6
	P92	1	Input/output	–	Bit	TO8
	P93	1	Input/output	–	Bit	TO9
	P94	1	Input/output	–	Bit	TIA/INT7
	P95	1	Input/output	–	Bit	TIB/INT8
	P96	1	Input/output	–	Bit	TOA/TOB
Port A	PA0 to PA2	3	Input	–	(Fixed)	AN0 to AN2
	PA3	1	Input	–	(Fixed)	AN3 / $\overline{\text{ADTRG}}$
	PA4 to PA7	4	Input	–	(Fixed)	AN4 to AN7

R: ↑ = With programmable pull-up resistor

Table 3.5 (2) Port Pin Setting Methods (1/3)

n: Corresponding port no. X: Don't care

Port Name	Pin Name	Function	Port Register Setting		
			Pn	PnCR	PnFC
Port 0	P00 to P07	Input port (Note 1)	X	0	None
		Output port (Note 1)	X	1	
		D0 to D7	X	X	
Port 1	P10 to P17	Input port	X	0	0
		Output port	X	1	0
		D8 to D15	X	0	1
Port 2	P20 to P27	Input port	X	0	X
		Output port	X	1	0
		A16 to A23	X	1	1
Port 3	P30 to P37	Input port (Note 1)	X	0	X
		Output port (Note 1)	X	1	0
		A8 to A15	X	1	1
Port 4	P40 to P47	Input port (Note 1)	X	0	X
		Output port (Note 1)	X	1	0
		A0 to A7	X	1	1
Port 5	P50	Output port (Note 1)	X	None	0
		$\overline{\text{RD}}$ output at external access only	1		1
		Always $\overline{\text{RD}}$ output	0		1
	P51	Output port (Note 1)	X		0
		$\overline{\text{WR}}$ output at external access only	X		1
	P52	Input port (no pull-up)	0	0	0
		Input port (with pull-up)	1	0	0
		Output port	X	1	0
		$\overline{\text{HWR}}$ output	X	1	1
	P53	Input port (no pull-up)	0	0	0
		Input port (with pull-up)	1	0	0
		Output port	X	1	X
		$\overline{\text{BUSRQ}}$ input (no pull-up)	0	0	1
		$\overline{\text{BUSRQ}}$ input (with pull-up)	1	0	1
	P54	Input port (no pull-up)	0	0	0
		Input port (with pull-up)	1	0	0
		Output port	X	1	X
		$\overline{\text{BUSAK}}$ output	X	1	1
	P55	Input port/$\overline{\text{WAIT}}$ input (no pull-up)	0	0	None
		Input port/$\overline{\text{WAIT}}$ input (with pull-up)	1	0	
		Output port	X	1	
	P56	Input port/INT0 input (no pull-up) (Note 2)	0	0	
		Input port/INT0 input (with pull-up) (Note 2)	1	0	
		Output port	X	1	
	P57	Input port/ SCLK2/ $\overline{\text{CTS2}}$ input (no pull-up)	0	0	0
		Input port/ SCLK2/ $\overline{\text{CTS2}}$ input (with pull-up)	1	0	0
		Output port	X	1	0
		SCLK2 output	X	1	1
Port 6	P60 to P63	Output port	X	None	0
		$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ output	X		1

Note 1: TMP95C265 does not use this function.

Note 2: When using pin P56 as an INT0 input, enable interrupt input with interrupt input mode control register IIMC<I0LE>.

Table 3.5 (2) Port Pin Setting Methods (2/3)

n: Corresponding port no. X: Don't care

Port Name	Pin Name	Function	Port Register Setting		
			Pn	PnCR	PnFC
Port 7	P70	Input port/TI0/INT1 input	X	0	None
		Output port	X	1	
	P71	Input port	X	0	X
		Output port	X	1	0
		TO1 output	X	1	1
	P72	Input port/INT2 input	X	0	X
		Output port	X	1	0
		TO3 output	X	1	1
	P73	Input port/TI4/INT3 input	X	0	None
		Output port	X	1	
	P74	Input port	X	0	X
		Output port	X	1	0
		TO5 output	X	1	1
	P75	Input port/INT4 input	X	0	X
		Output port	X	1	0
		TO7 output	X	1	1
Port 8	P80	Input port (no pull-up)	0	0	0
		Input port (with pull-up)	1	0	0
		Output port	X	1	0
		TxD0 output (Note 3)	X	1	1
	P81	Input port/RxD0 input (no pull-up)	0	0	None
		Input port/RxD0 input (with pull-up)	1	0	
		Output port	X	1	
	P82	Input port/SCLK0/$\overline{\text{CTS0}}$ input (no pull-up)	0	0	0
		Input port/SCLK0/$\overline{\text{CTS0}}$ input (with pull-up)	1	0	0
		Output port	X	1	0
		SCLK0 output	X	1	1
	P83	Input port (no pull-up)	0	0	0
		Input port (with pull-up)	1	0	0
		Output port	X	1	0
		TxD1 output (Note 3)	X	1	1
	P84	Input port/ RxD1 input (no pull-up)	0	0	None
		Input port/ RxD1 input (with pull-up)	1	0	
		Output port	X	1	
	P85	Input port/SCLK1/$\overline{\text{CTS1}}$ input (no pull-up)	0	0	0
		Input port/SCLK1/$\overline{\text{CTS1}}$ input (with pull-up)	1	0	0
		Output port	X	1	0
		SCLK1 output	X	1	1
	P86	Input port (no pull-up)	0	0	0
		Input port (with pull-up)	1	0	0
		Output port	X	1	0
		TxD2 output (Note 3)	X	1	1
	P87	Input port/ RxD2 input (no pull-up)	0	0	None
		Input port/ RxD2 input (with pull-up)	1	0	
		Output port	X	1	

Note 3: Open drain enable register ODE<ODE0:2> is used to set the open drain output mode for pins TxD0 to 2.

Table 3.5 (2) Port Pin Setting Methods (3/3)

n: Corresponding port no. X: Don’t care

Port Name	Pin Name	Function	Port Register Setting		
			Pn	PnCR	PnFC
Port 9	P90	Input port/TI8/INT5 input	X	0	None
		Output port	X	1	
	P91	Input port/TI9/INT6 input	X	0	
		Output port	X	1	
	P92	Input port	X	0	X
		Output port	X	1	0
		TO8 output	X	1	1
	P93	Input port	X	0	X
		Output port	X	1	0
		TO9 output	X	1	1
	P94	Input port/TIA/INT7 input	X	0	None
		Output port	X	1	
	P95	Input port/TIB/INT8 input	X	0	
		Output port	X	1	
	P96	TOA/TOB output (Note 4)	X	1	1
Port A	PA0 to PA7	Input port	X	None	
		AN0 to AN7 input (Note 5)	X		

Note 4: P9FC<TOS1> is used to switch between the TOA and TOB timer outputs to pin P96.

Note 5: When PA0 to PA7 are used as A/D converter input channels, A/D mode control register 1ADMOD1<ADCH2:0> is used to select the channel.

3.5.1 Port 0 (P00 to P07)

Port 0 is an 8-bit general-purpose input/output port with each port bit settable as an input or output.
In addition to functioning as a general-purpose input/output port, port 0 also functions as the data bus (D0 to D7). The port 0 control register P0CR sets the pins as inputs or outputs.
A reset sets all the bits of the P0CR register to 0, and sets all pins to input mode.
When external memory is accessed, the port automatically functions as the data bus (D0 to D7) and all bits of P0CR are cleared to 0.
In the external ROM version of TMP95C265, port 0 always functions as the data bus (D0 to D7) irrespective of the settings in the P0CR control register.

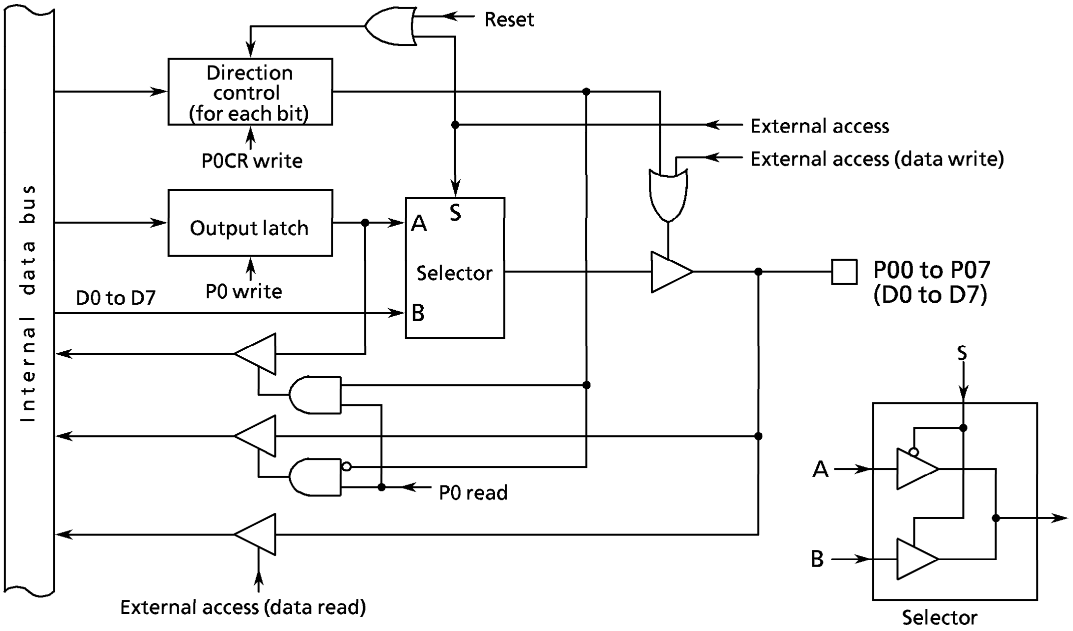


Figure 3.5 (1) Port 0 (P00 to P07)

Port 0 Register

P0
(0000H)

	7	6	5	4	3	2	1	0
bit Symbol	P07	P06	P05	P04	P03	P02	P01	P00
Read/Write	R/W							
After reset	Input mode (output latch register undefined)							
Function	Also functions as D7 to D0							

Port 0 Control Register

P0CR
(0002H)

Read-modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Port 0 input/output settings 0: Input 1: Output							

Note: When functioning as a data bus (D0 to D7), P0CR is cleared to 0.

Figure 3.5 (2) Port 0 Related Registers

3.5.2 Port 1 (P10 to P17)

Port 1 is an 8-bit general-purpose input/output port with each port bit settable as an input or output.
In addition to functioning as a general-purpose input/output port, port 1 also functions as a data bus (D8 to D15). The port 1 control register P1CR and function register P1FC set the port 1 functions.
Reset sets all the bits of the P1 output latch register and all bits of the P1CR and P1FC registers to 0, and sets port 1 to input mode.
In the external ROM version of TMP95C265, port 1 functions as the data bus (D8 to D15) if the AM8/$\overline{16}$ pins are at low level after a reset (fixed 16-bit external data bus or mixed 8/16-bit external data bus). Port 1 functions as a port if the AM8/$\overline{16}$ pins are at high level after a reset (fixed 8-bit external data bus).

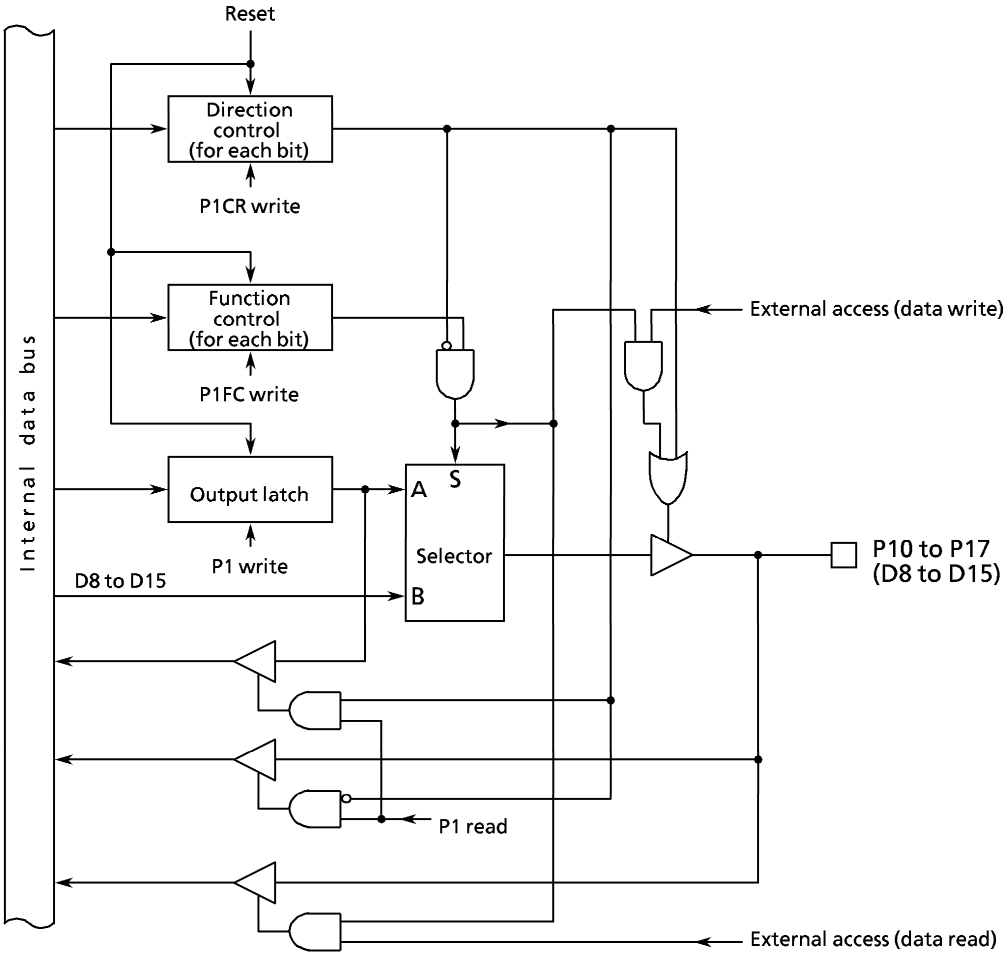

Reset
Internal data bus
Direction control (for each bit)
P1CR write
Function control (for each bit)
P1FC write
Output latch
P1 write
D8 to D15
A
S
Selector
B
External access (data write)
P10 to P17 (D8 to D15)
P1 read
External access (data read)

Figure 3.5 (3) Port 1 (P10 to P17)

Port 1 Register

P1 (0001H)

	7	6	5	4	3	2	1	0
bit Symbol	P17	P16	P15	P14	P13	P12	P11	P10
Read/Write	R/W							
After reset	Input mode (output latch register cleared to 0)							
Function	Also functions as D15 to D8							

Port 1 Control Register

P1CR (0004H)

Read-modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Port 1 function settings							

Port 1 Function Register

P1FC (0005H)

Read-modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Port 1 function settings							

Port 1 function settings

P1FC<P1xF> P1CR<P1xC>	0	1
0	Input port	Data bus (D15 to D8)
1	Output port	Do not set

Note 1: In TMP95C265, when the AM8/$\overline{16}$ pin is set to low, P1FC is fixed to 1. Therefore, do not set P1CR to 1. (After a reset, P1CR is cleared to 0.)

Note 2: In TMP95C265, when the AM8/$\overline{16}$ pin is set to high, setting port 1 as a data bus (D15 to D8) sets pins P17 to P10 to high impedance.

Figure 3.5 (4) Port 1 Related Registers

3.5.3 Port 2 (P20 to P27)

Port 2 is an 8-bit general-purpose input/output port with each port bit settable as an input or output.
In addition to functioning as a general-purpose input/output port, port 2 also functions as an address bus (A16 to A23). The port 2 control register (P2CR) and function register (P2FC) set the port 2 functions.
Reset sets all the bits of the P2 output latch register and all bits of the P2CR and P2FC registers to 0, setting port 2 to input mode.
In the external ROM version of TMP95C265, after a reset, port 2 functions as an address bus (A16 to A23). However, depending on the settings of the P2CR and P2FC registers, port 2 can also function as a general-purpose input/output port.

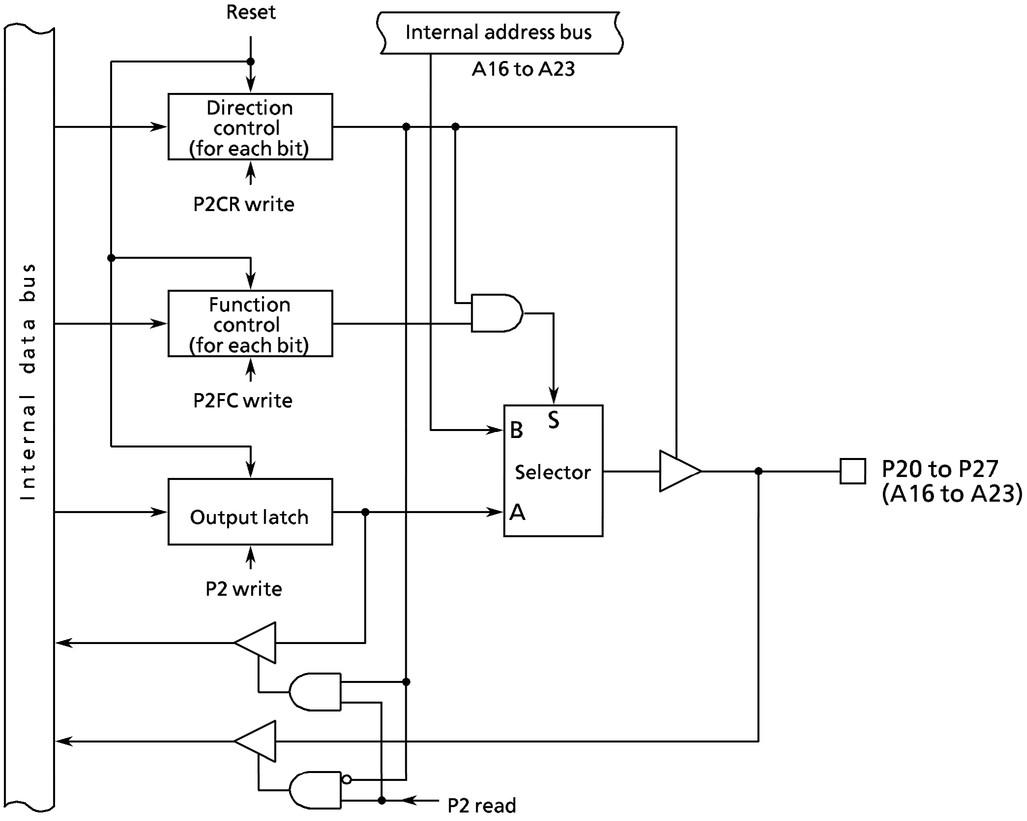


Figure 3.5 (5) Port 2 (P20 to P27)

Port 2 Register

P2
(0006H)

	7	6	5	4	3	2	1	0
bit Symbol	P27	P26	P25	P24	P23	P22	P21	P20
Read/Write	R/W							
After reset	Input mode (output latch register set to 0)							
Function	Also functions as A23 to A16							

Port 2 Control Register

P2CR
(0008H)

Read-modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Port 2 function settings							

Port 2 Function Register

P2FC
(0009H)

Read-modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Port 2 function settings							

Port 2 function settings

P2FC<P2xF> P2CR<P2xC>	0	1
0	Input port	
1	Output port	Address bus (A23 to A16)

Note: When setting the address bus (A23 to A16), first set P2CR, then P2FC.
In TMP95C265, P2CR and P2FC are set to 1 after a reset, thus selecting the address bus (A23 to A16).

Figure 3.5 (6) Port 2 Related Registers

3.5.4 Port 3 (P30 to P37)

Port 3 is an 8-bit general-purpose input/output port with each port bit settable as an input or output.

In addition to functioning as a general-purpose input/output port, port 3 also functions as an address bus (A8 to A15). The port 3 control register (P3CR) and function register (P3FC) set the port 3 functions.

Reset sets all the bits of the P3 output latch register and all bits of the P3CR and P3FC registers to 0, setting port 3 to input mode.

In the external ROM version of TMP95C265, port 3 functions as an address bus (A8 to A15) irrespective of the settings of the P3CR and P3FC registers.

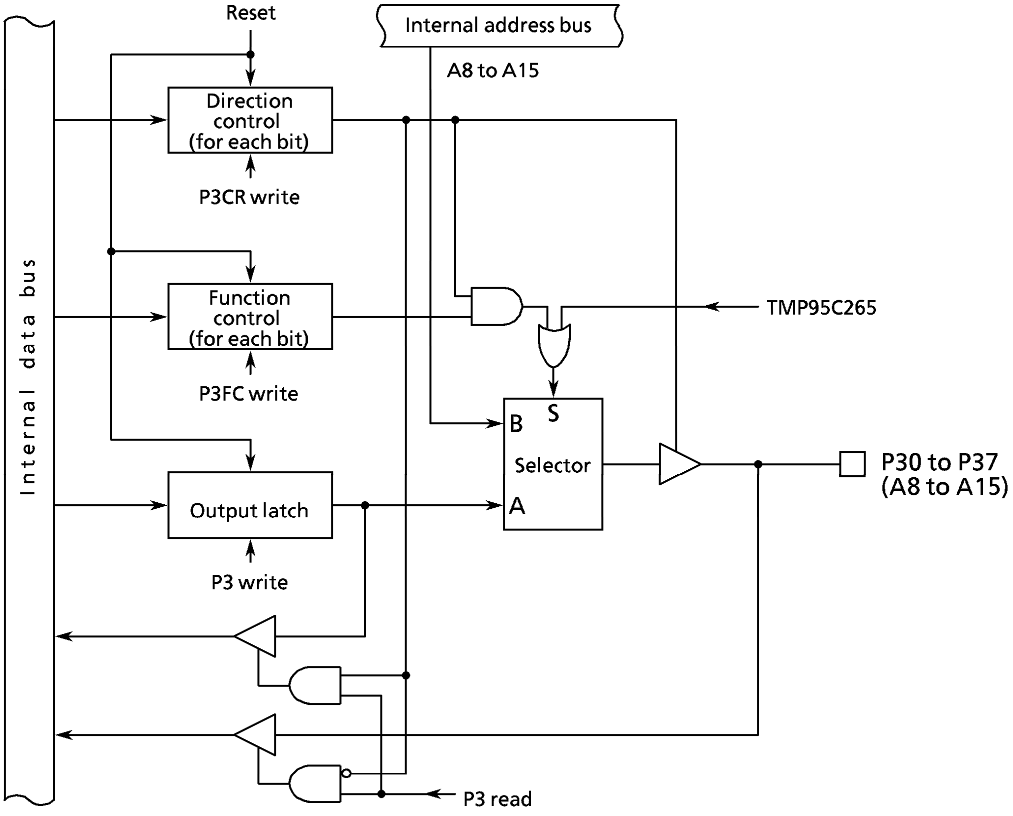


Figure 3.5 (7) Port 3 (P30 to P37)

Port 3 Register

		7	6	5	4	3	2	1	0
P3 (0007H)	bit Symbol	P37	P36	P35	P34	P33	P32	P31	P30
	Read/Write	R/W							
	After reset	Input mode (output latch register cleared to 0)							
	Function	Also functions as A15 to A8							

Port 3 Control Register

		7	6	5	4	3	2	1	0
P3CR (000AH)	bit Symbol	P37C	P36C	P35C	P34C	P33C	P32C	P31C	P30C
Read-modify-write instructions prohibited.	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	Port 3 function settings							

Port 3 Function Register

		7	6	5	4	3	2	1	0
P3FC (000BH)	bit Symbol	P37F	P36F	P35F	P34F	P33F	P32F	P31F	P30F
Read-modify-write instructions prohibited.	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	Port 3 function settings							

Port 3 function settings

P3FC<P3xF> P3CR<P3xC>	0	1
0	Input port	
1	Output port	Address bus (A15 to A8)

Note: When setting the address bus (A15 to A8), first set P3CR, then P3FC.
In TMP95C265, the address bus (A15 to A8) is selected after a reset.

Figure 3.5 (8) Port 3 Related Registers

3.5.5 Port 4 (P40 to P47)

Port 4 is an 8-bit general-purpose input/output port with each port bit settable as an input or output. In addition to functioning as a general-purpose input/output port, port 4 also functions as an address bus (A0 to A7). The port 4 control register (P4CR) and function register (P4FC) set the port 4 functions. Reset sets all the bits of the P4 output latch register and all bits of the P4CR and P4FC registers to 0, setting port 4 to input mode.

In the external ROM version of TMP95C265, port 4 functions as an address bus (A0 to A7) irrespective of the settings of the P4CR and P4FC registers.

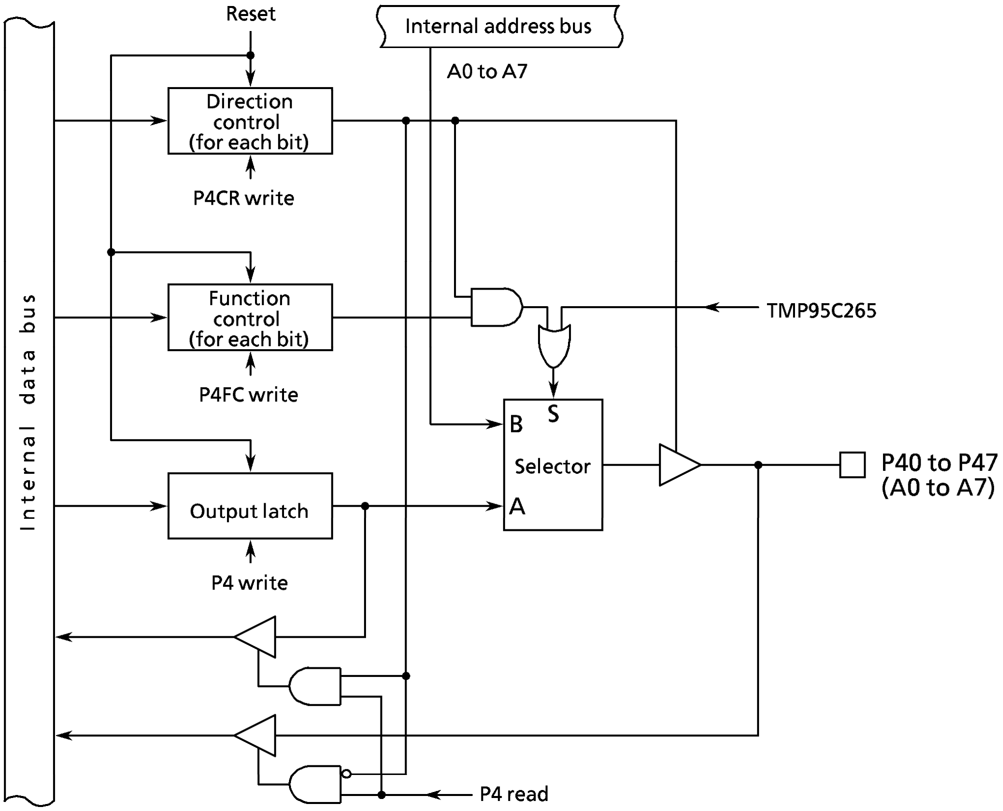


Figure 3.5 (9) Port 4 (P40 to P47)

Port 4 Register

		7	6	5	4	3	2	1	0
P4 (000CH)	bit Symbol4	P47	P46	P45	P44	P43	P42	P41	P40
	Read/Write	R/W							
	After reset	Input mode (output latch register cleared to 0)							
	Function	Also functions as A7 to A0							

Port 4 Control Register

		7	6	5	4	3	2	1	0
P4CR (000EH) Read-modify-write instructions prohibited.	bit Symbol	P47C	P46C	P45C	P44C	P43C	P42C	P41C	P40C
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	Port 4 function settings							

Port 4 Function Register

		7	6	5	4	3	2	1	0
P4FC (000FH) Read-modify-write instructions prohibited.	bit Symbol	P47F	P46F	P45F	P44F	P43F	P42F	P41F	P40F
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	Port 4 function settings							

Port 4 function settings

P4FC<P4xF> P4CR<P4xC>	0	1
0	Input port	
1	Output port	Address bus (A7 to A0)

Note: When setting the address bus (A7 to A0), first set P4CR, then P4FC.
In TMP95C265, the address bus (A7 to A0) is selected after a reset.

Figure 3.5 (10) Port 4 Related Registers

3.5.6 Port 5 (P50 to P57)

Port 5 is an 8-bit general-purpose input/output port with each port bit settable as an input or output. However, P50 and P51 are output-only ports.

In addition to functioning as a general-purpose input/output port, port 5 also has a CPU control/status signal input/output function, a $\overline{\text{WAIT}}$ input function, an INT0 external interrupt input function, and a serial channel SCLK2/$\overline{\text{CTS2}}$ function. The port 5 control register (P5CR) and function register (P5FC) set the port 5 functions.

Reset sets all the bits of the P5 output latch register to 1 and clears all bits of P5CR (bits 0 and 1 are unused) and P5FC (bits 5 and 6 are unused) to 0. Pins P50 and P51 output 1 and P52 to P57 are set to input mode with resistors pulled up.

When P50 is set as the $\overline{\text{RD}}$ pin (when P5FC<P50F> = 1) or, in the case of TMP95C265, when P5<P50> is cleared to 0, the P50 $\overline{\text{RD}}$ signal is output even when an internal address area is accessed, and external PSRAM (pseudo SRAM) can be refreshed. If <P50> is set to 1, the $\overline{\text{RD}}$ signal is output only when an external area is accessed.

In the external ROM version of TMP95C265, P50 functions as the $\overline{\text{RD}}$ pin and P51 as the $\overline{\text{WR}}$ pin irrespective of the <P50F> and <P51F> settings.

(1) Port 50 ($\overline{\text{RD}}$)

In addition to functioning as a general-purpose output-only port, port 50 can also function as the $\overline{\text{RD}}$ pin. In TMP95C265, port 50 always functions as the $\overline{\text{RD}}$ pin.

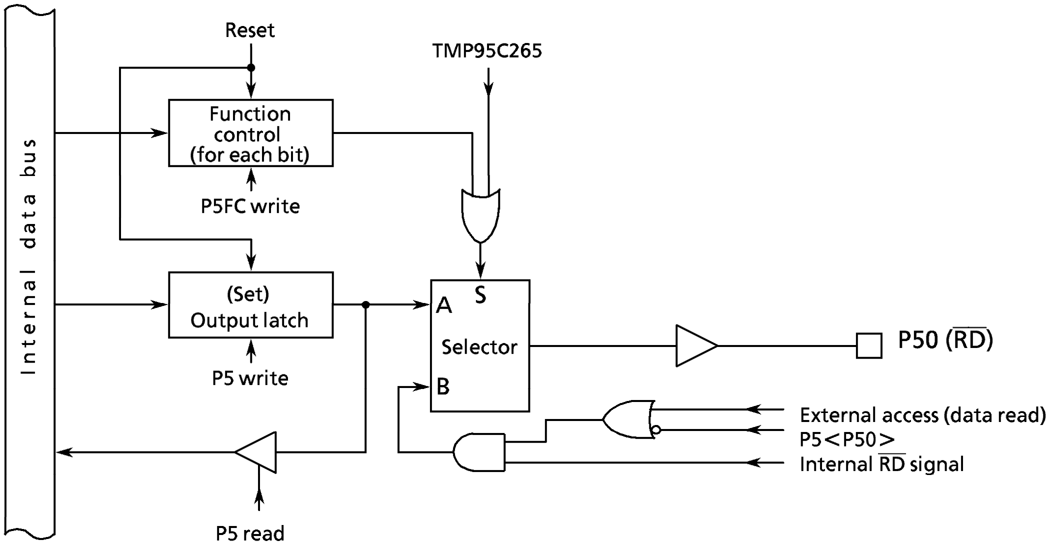

Internal data bus
Reset
TMP95C265
Function control (for each bit)
P5FC write
(Set) Output latch
P5 write
A
S
B
Selector
P50 ($\overline{\text{RD}}$)
External access (data read)
P5<P50>
Internal $\overline{\text{RD}}$ signal
P5 read

Figure 3.5 (11) Port 5 (P50)

(2) Port 51 ($\overline{\text{WR}}$)

In addition to functioning as a general-purpose output-only port, port 51 can also function as the $\overline{\text{WR}}$ pin. In TMP95C265, port 51 always functions as the $\overline{\text{WR}}$ pin.

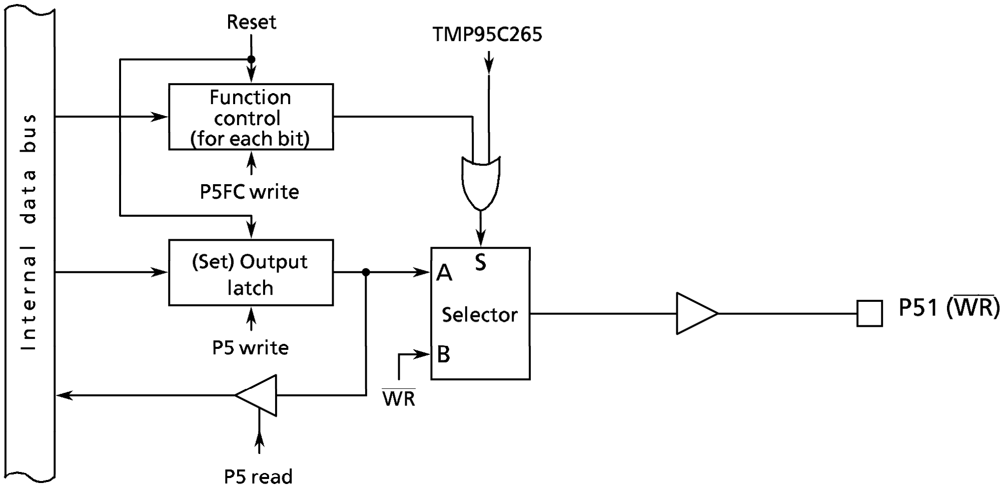


Figure 3.5 (12) Port 5 (P51)

(3) Ports 52, 54 ($\overline{\text{HWR}}$, $\overline{\text{BUSAK}}$)

In addition to being general-purpose input/output ports, port 52 can also function as the $\overline{\text{HWR}}$ pin, and port 54 can also function as the $\overline{\text{BUSAK}}$ pin.

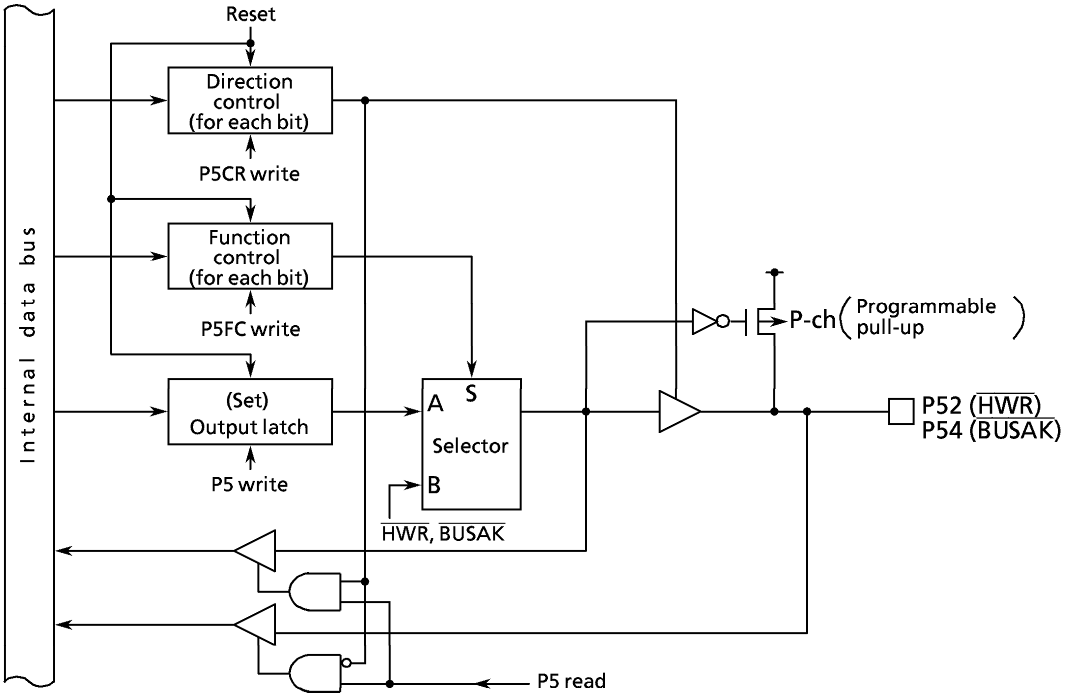


Figure 3.5 (13) Port 5 (P52, P54)

(4) Port 53 ($\overline{\text{BUSRQ}}$)

In addition to being a general-purpose input/output port, port 53 also functions as the $\overline{\text{BUSRQ}}$ pin.

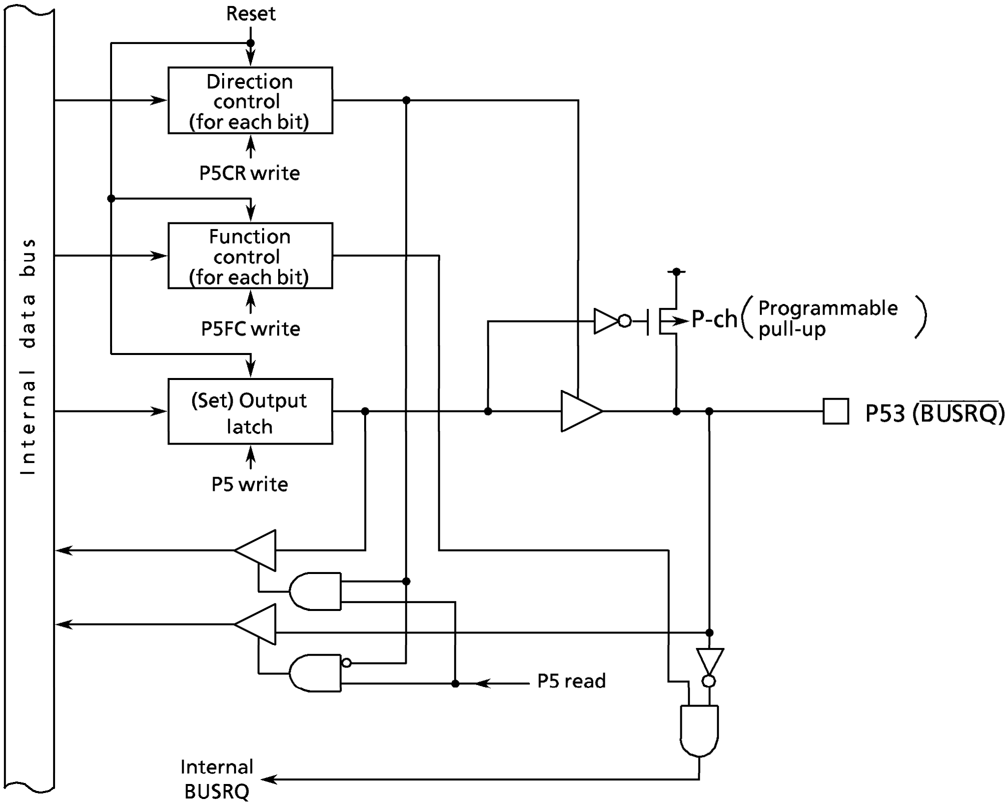

Reset
Direction control (for each bit)
P5CR write
Function control (for each bit)
P5FC write
(Set) Output latch
P5 write
Internal data bus
P-ch (Programmable pull-up)
P53 ($\overline{\text{BUSRQ}}$)
P5 read
Internal BUSRQ

Figure 3.5 (14) Port 5 (P53)

(5) Port 55 ($\overline{\text{WAIT}}$)

In addition to being a general-purpose input/output port, port 55 also functions as the $\overline{\text{WAIT}}$ pin.

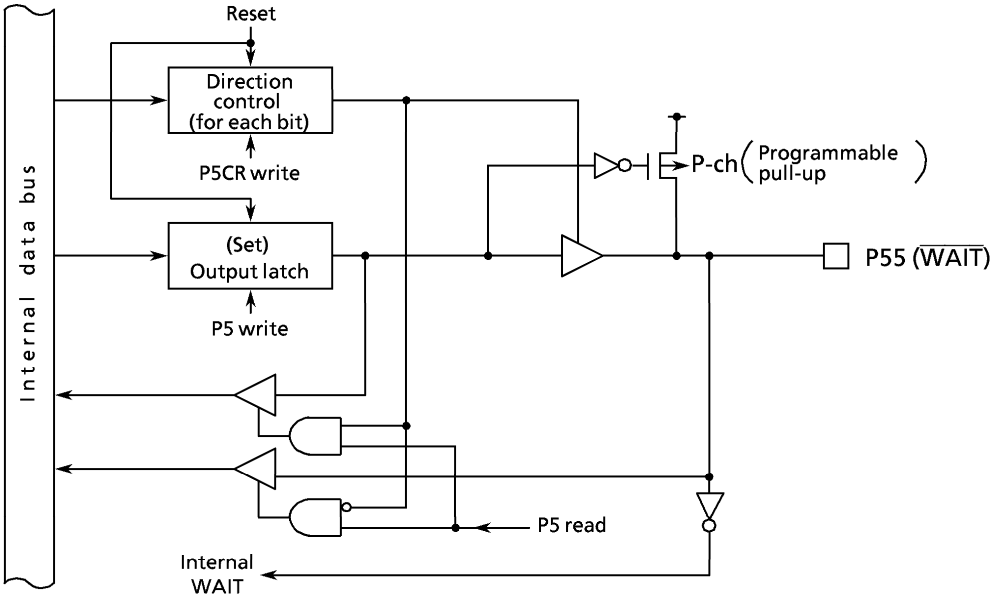


Figure 3.5 (15) Port 5 (P55)

(6) Port 56 (INT0)

In addition to being a general-purpose input/output port, port 56 also functions as the external interrupt request input INT0 pin.

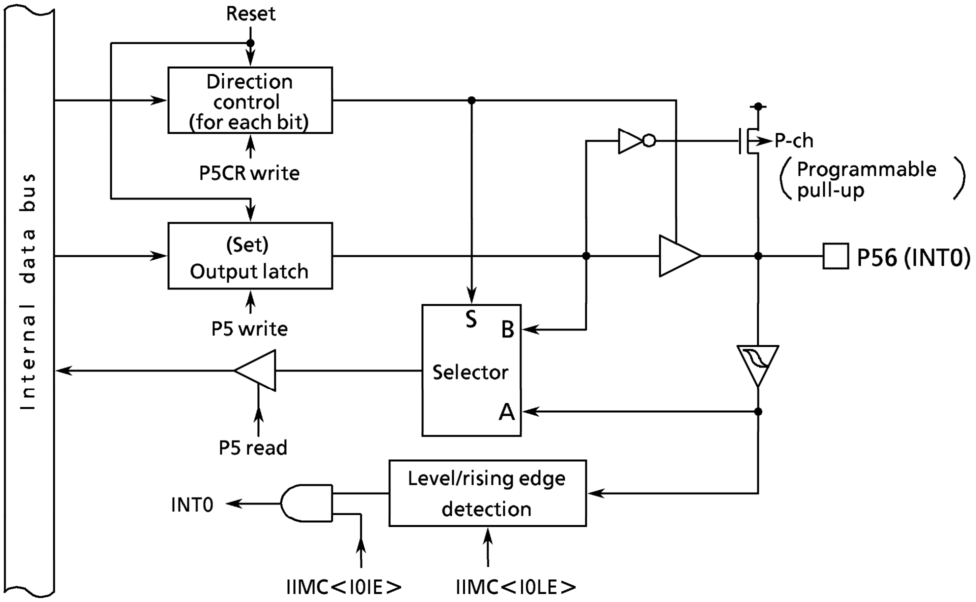


Figure 3.5 (16) Port 5 (P56)

(7) Port 57 (SCLK2/$\overline{\text{CTS2}}$)

In addition to being a general-purpose input/output port, port 57 also functions as the serial channel 2 SCLK2 input/output pin, or the $\overline{\text{CTS2}}$ input pin.

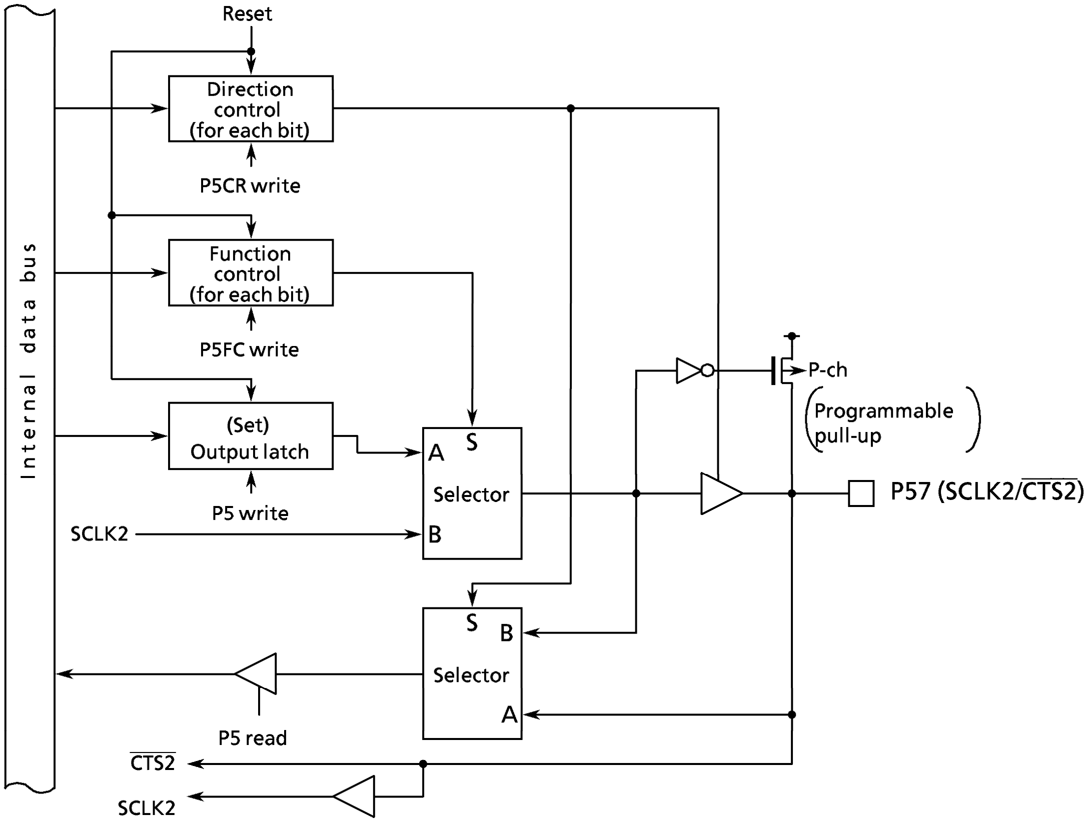


Figure 3.5 (17) Port 5 (P57)

Port 5 Register

		7	6	5	4	3	2	1	0
P5 (000DH)	bit Symbol	P57	P56	P55	P54	P53	P52	P51	P50
	Read/Write	R/W							
	After reset	Input mode (Set to 1 / pulled up)						Output only (set to 1)	
	Function	Also functions as SCLK2/$\overline{\text{CTS2}}$	Also functions as INT0	Also functions as $\overline{\text{WAIT}}$	Also functions as $\overline{\text{BUSAK}}$	Also functions as $\overline{\text{BUSRQ}}$	Also functions as $\overline{\text{HWR}}$	Also functions as $\overline{\text{WR}}$	Also functions as $\overline{\text{RD}}$

Note: When port 5 is in input mode, the internal pull-up resistor is controlled by the P5 register. When using port 5 in input mode or in both input and output modes (if just one bit is set to input mode), read-modify-write instructions cannot be executed. The internal pull-up resistor setting may change depending on the state of the input pin.

Port 5 Control Register

		7	6	5	4	3	2	1	0
P5CR (0010H)	bit Symbol	P57C	P56C	P55C	P54C	P53C	P52C		
Read-modify-write instructions prohibited.	Read/Write	W							
	After reset	0	0	0	0	0	0		
	Function	Port 57 to 52 input/output settings 0: Input 1: Output							

Port 56 function settings (Note 2)

<P56> <P56C>	0	1
0	Input port / INT0 input	Input port / INT0 input (pull-up)
1	Output port	

Port 55 function settings (Note 1)

<P55> <P55C>	0	1
0	Input port / $\overline{\text{WAIT}}$ input	Input port / $\overline{\text{WAIT}}$ input (pull-up)
1	Output port	

Note 1: When using port 55 as the $\overline{\text{WAIT}}$ pin, set P5CR<P55C> to 0 and set the chip select/wait control register BxCS<BxW2:0> to 010 (1 WAIT + N) or 100 (0 + N WAIT).

Note 2: When using port 56 as the INT0 pin, set P5CR<P56C> to 0 and set the interrupt input mode control register IIMC<I0IE> to 1.

Figure 3.5 (18)-1 Port 5 Related Registers

Port 5 Function Register

P5FC
(0011H)

Read-modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol	P57F			P54F	P53F	P52F	P51F	P50F
Read/Write	W			W				
After reset	0			0	0	0	0	0
Function	0: Port 1: SCLK2 /$\overline{\text{CTS2}}$			0: Port 1: $\overline{\text{BUSAK}}$	0: Port 1: $\overline{\text{BUSRQ}}$	0: Port 1: $\overline{\text{HWR}}$	0: Port 1: $\overline{\text{WR}}$	0: Port 1: $\overline{\text{RD}}$

Port 53 function settings

<P53F> <P53C>	0	1
0	Input port (When <P53> = 0, not pulled up; when <P53> = 1, pulled up)	$\overline{\text{BUSRQ}}$ input (When <P53> = 0, not pulled up; when <P53> = 1, pulled up)
1	Output port	

Port 54 function settings

<P54F> <P54C>	0	1
0	Input port (When <P54> = 0, not pulled up; when <P54> = 1, pulled up)	Do not set
1	Output port	$\overline{\text{BUSAK}}$ output

Port 57 function settings

<P57F> <P57C>	0	1
0	Input port SCLK2/$\overline{\text{CTS2}}$ (When <P57> = 0, not pulled up; when <P57> = 1, pulled up)	Do not set
1	Output port	SCLK2 output

Port 50 function settings

<P50> <P50F>	0	1
0	Output port	
1	Normally $\overline{\text{RD}}$ output (for PSRAM)	$\overline{\text{RD}}$ output only at external access

Port 51 function settings

0	Output port
1	$\overline{\text{WR}}$ output only at external access

Port 52 function settings

<P52F> <P52C>	0	1
0	Input port (When <P52> = 0, not pulled up; when <P52> = 1, pulled up)	Do not set
1	Output port	$\overline{\text{HWR}}$ output

Figure 3.5 (18)-2 Port 5 Related Register

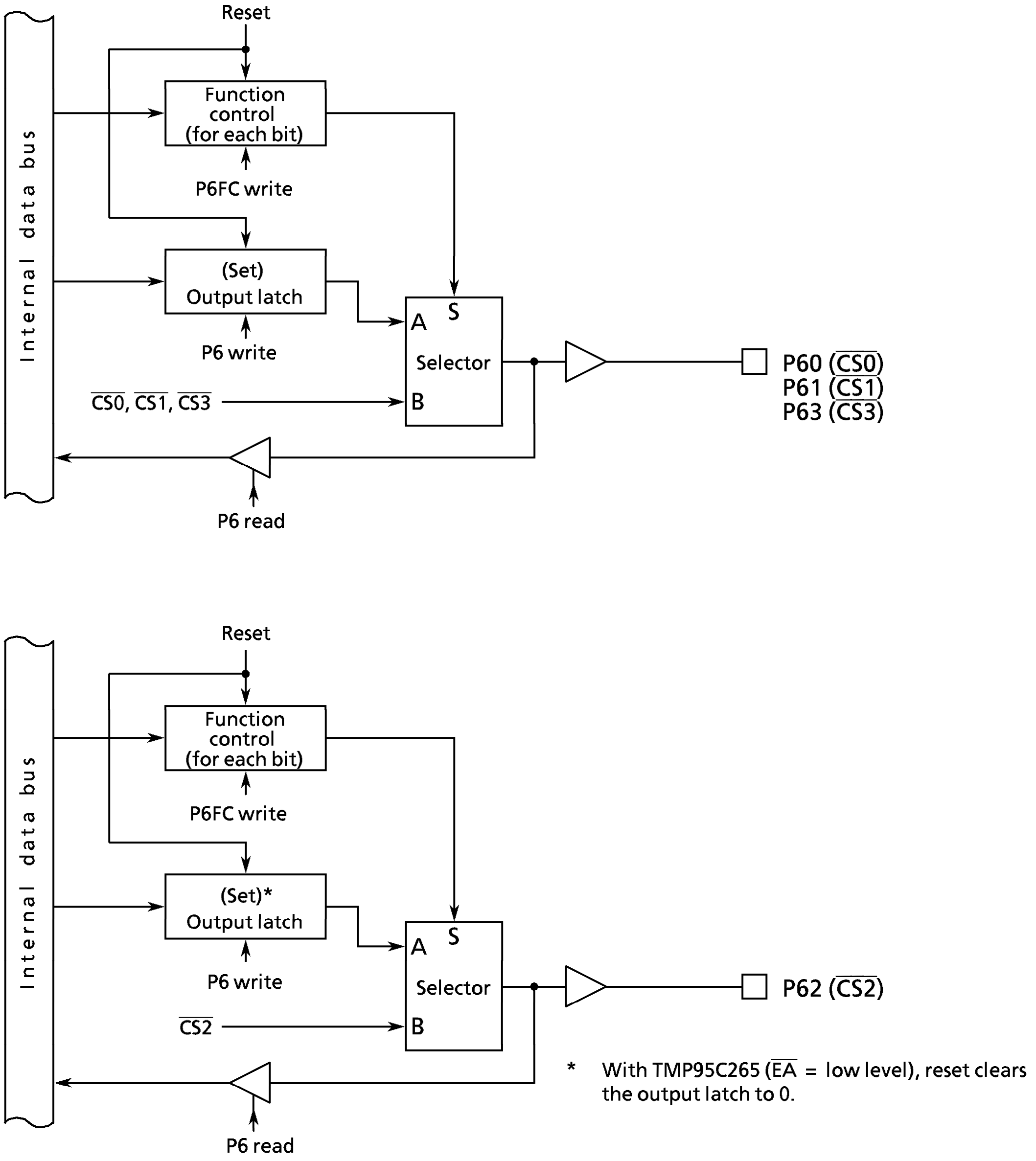
3.5.7 Port 6 (P60 to P63)

Port 6 is a 4-bit general-purpose output-only port.

In addition to functioning as a general-purpose output port, port 6 also has a chip select signal output function ($\overline{\text{CS0}}$ to $\overline{\text{CS3}}$). The port 6 function register P6FC sets the functions.

Reset sets the P60 to P63 output latch to 1. Reset also clears all bits of the P6FC register to 0, setting port 6 to a general-purpose output port.

In the external ROM version of TMP95C265, after a reset, the P62 ($\overline{\text{CS2}}$) output latch is cleared to 0 and the CS2 area is selected.



Reset
Function control (for each bit)
P6FC write
(Set) Output latch
P6 write
$\overline{\text{CS0}}$, $\overline{\text{CS1}}$, $\overline{\text{CS3}}$
A
S
Selector
B
P60 ($\overline{\text{CS0}}$)
P61 ($\overline{\text{CS1}}$)
P63 ($\overline{\text{CS3}}$)
Internal data bus
P6 read
Reset
Function control (for each bit)
P6FC write
(Set)*
Output latch
P6 write
$\overline{\text{CS2}}$
A
S
Selector
B
P62 ($\overline{\text{CS2}}$)
Internal data bus
P6 read

* With TMP95C265 ($\overline{\text{EA}}$ = low level), reset clears the output latch to 0.

Figure 3.5 (19) Port 6 (P60 to P63)

Port 6 Register

P6
(0012H)

	7	6	5	4	3	2	1	0
bit Symbol					P63	P62	P61	P60
Read/Write					R/W			
After reset					Output mode (set to 1)			
Function					Also functions as $\overline{\text{CS3}}$.	Also functions as $\overline{\text{CS2}}$.	Also functions as $\overline{\text{CS1}}$.	Also functions as $\overline{\text{CS0}}$.

<P62> initial value

TMP95CS64 ($\overline{\text{EA}}$ = high level)	1
TMP95C265 ($\overline{\text{EA}}$ = low level)	0

Note: After reset, the initial value for <P62> only depends on the setting of the $\overline{\text{EA}}$ pin.

Port 6 function register

P6FC
(0015H)

Read-modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol					P63F	P62F	P61F	P60F
Read/Write					W			
After reset					0	0	0	0
Function					0: Port 1: $\overline{\text{CS3}}$	0: Port 1: $\overline{\text{CS2}}$	0: Port 1: $\overline{\text{CS1}}$	0: Port 1: $\overline{\text{CS0}}$

Note: The chip select/wait control register (B0CS, B1CS, B2CS, B3CS) sets the $\overline{\text{CS}}$ area operations.

Figure 3.5 (20) Port 6 Related Registers

3.5.8 Port 7 (P70 to P75)

Port 7 is a 6-bit general-purpose input/output port with each port bit settable as an input or output.
In addition to functioning as general-purpose input/output port pins, port 6 pins also function as event count inputs for the 8-bit timer, outputs for the 8-bit timer, and INT1 to 4 inputs for the external interrupt function.
Port 7 control register P7CR and port 7 function register P7FC set the port 7 functions.
Reset clears all bits of the output latch register and P7CR to 0, setting all pins to input mode.
To enable the timer output function, write 1 to the corresponding bits in both P7CR and P7FC.

(1) Port 70, 73 (TI0/INT1, T14/INT3)

In addition to functioning as a general-purpose input/output port, port 70 can also function as the event count input TI0 for timer 0 and as the external interrupt request input INT1.
In addition to functioning as a general-purpose input/output port, port 73 can also function as the event count input TI4 for timer 4 and as the external interrupt request input INT3.

Cautions when using INT1 and INT3 interrupts

Input is always enabled for the INT1 and INT3 external interrupt requests.
Caution is required if port 70 or 73 is used as a general-purpose input/output port or a timer event count input while the INT1 and INT3 interrupt functions are in use. This is because rising edges on these input/output signals generate interrupt requests.

Cautions when using timer event count inputs TI0 and TI4

Input is always enabled for the timer event count inputs TI0 and TI4.
Caution is required if port 70 or 73 is used as a general-purpose input/output port or an INT1 or INT3 interrupt during event counting based on TI0 or TI4. This is because these input/output signals trigger an event count on the timer.

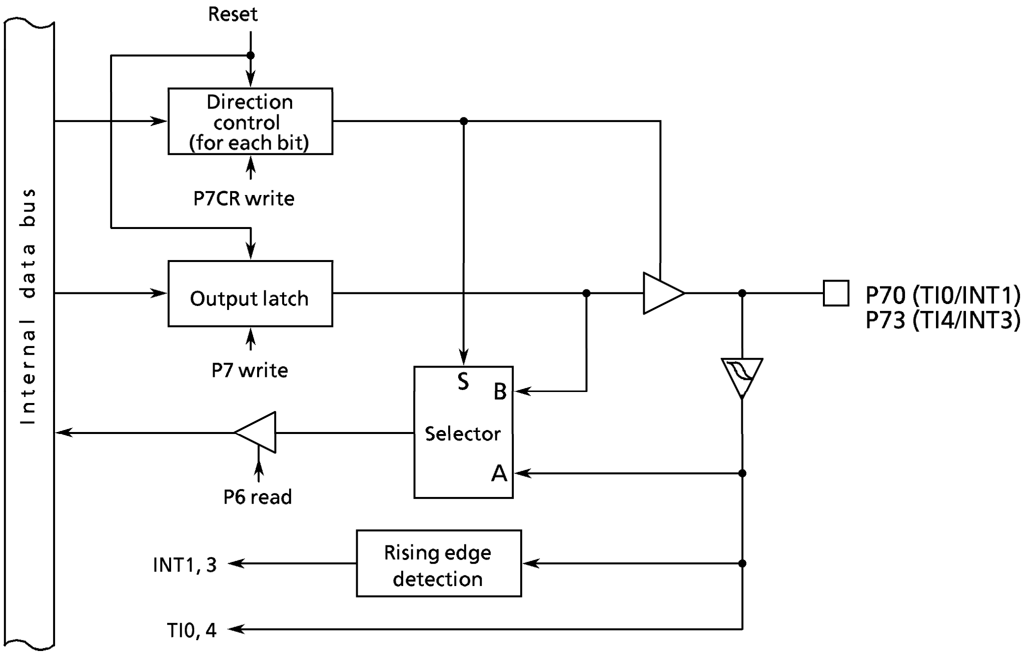


Figure 3.5 (21) Port 7 (P70, P73)

(2) Port 71, 74 (TO1, TO5)

In addition to functioning as a general-purpose input/output port, port 71 also functions as TO1 for output of timers 0 and 1. In addition to functioning as a general-purpose input/output port, port 74 also functions as TO5 for output of timers 4 and 5.

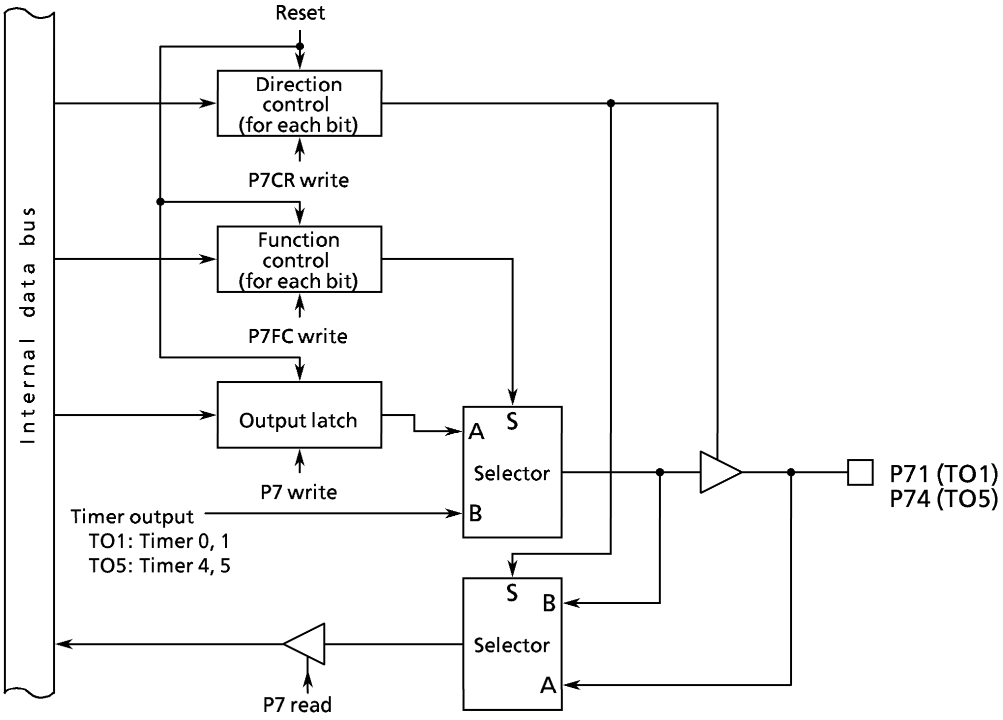


Figure 3.5 (22) Port 7 (P71, P74)

(3) Port 72, 75 (TO3/INT2, TO7/INT4)

In addition to functioning as a general-purpose input/output port, port 72 also functions as TO3 for output of timers 2 and 3 and as the external interrupt request input INT2.
In addition to functioning as a general-purpose input/output port, port 75 also functions as TO7 for output of timers 6 and 7 and as the external interrupt request input INT4.

Cautions when using INT2 or INT4 interrupts

Input is always enabled for the INT2 and INT4 external interrupt requests.
Caution is required if port 72 or 75 is used as a general-purpose input/output port or timer event count input port while the INT2 and INT4 interrupt functions are in use. This is because rising edges on these input/output signals generate interrupt requests.

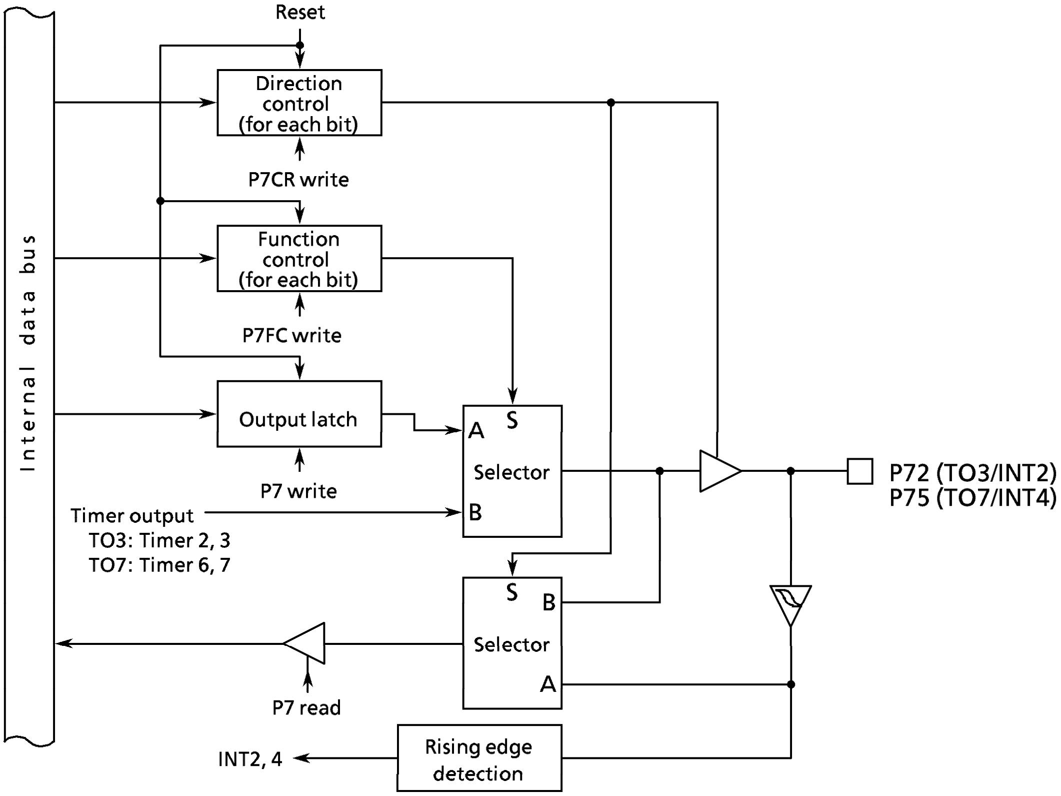


Figure 3.5 (23) Port 7 (P72, P75)

Port 7 Register

P7
(0013H)

	7	6	5	4	3	2	1	0
bit Symbol			P75	P74	P73	P72	P71	P70
Read/Write			R/W					
After reset			Input mode (output latch register cleared to 0)					
Function			Also functions as TO7/INT4 (Note)	Also functions as TO5	Also functions as TI4/INT3 (Note)	Also functions as TO3/INT2 (Note)	Also functions as TO1	Also functions as TI0/INT1 (Note)

Port 7 Control Register

P7CR
(0016H)

Read-modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol			P75C	P74C	P73C	P72C	P71C	P70C
Read/Write			W					
After reset			0	0	0	0	0	0
Function			Port 7 input/output settings 0: Input 1: Output					

Port 7 Function Register

P7FC
(0017H)

Read-modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol			P75F	P74F		P72F	P71F	
Read/Write			W			W		
After reset			0	0		0	0	
Function			0: Port 1: TO7	0: Port 1: TO5		0: Port 1: TO3	0: Port 1: TO1	

Port 71 function settings

<P71F> <P71C>	0	1
0	Input port	
1	Output port	TO1 output

Port 72 function settings

<P72F> <P72C>	0	1
0	Input port/INT2 input	
1	Output port	TO3 output

Port 74 function settings

<P74F> <P74C>	0	1
0	Input port	
1	Output port	TO5 output

Port 75 function settings

<P75F> <P75C>	0	1
0	Input port/INT4 input	
1	Output port	TO7 output

Note: No register is provided to switch between the external interrupt input, event counter input, and port input/output functions. Therefore, even when port 7 is used as a port, the interrupt signal input and event counter input are enabled. When using port 7 exclusively as a port, disable the interrupt and event counter input.

Figure 3.5 (24) Port 7 Related Registers

3.5.9 Port 8 (P80 to P87)

Port 8 is an 8-bit general-purpose input/output port with each port bit settable as an input or output.

In addition to being a general-purpose input/output port, port 8 also functions as a serial channel TxD output, RxD input, and SCLK input/output.

Port 8 control register P8CR and port 8 function register P8FC set the functions.

Reset sets all bits of the output latch to 1. It also clears all bits of the P8CR and P8FC registers to 0, setting port 8 to input mode using pull-up resistors.

Port pins 80, 83, and 86 have a programmable open drain function.

(1) Ports 80, 83, 86 (TxD0, 1, 2)

Ports 80, 83, and 86 function as the serial channel TxD0 to 2 outputs as well as input/output ports. These ports have a programmable drain function. Setting open drain disables pull-up.

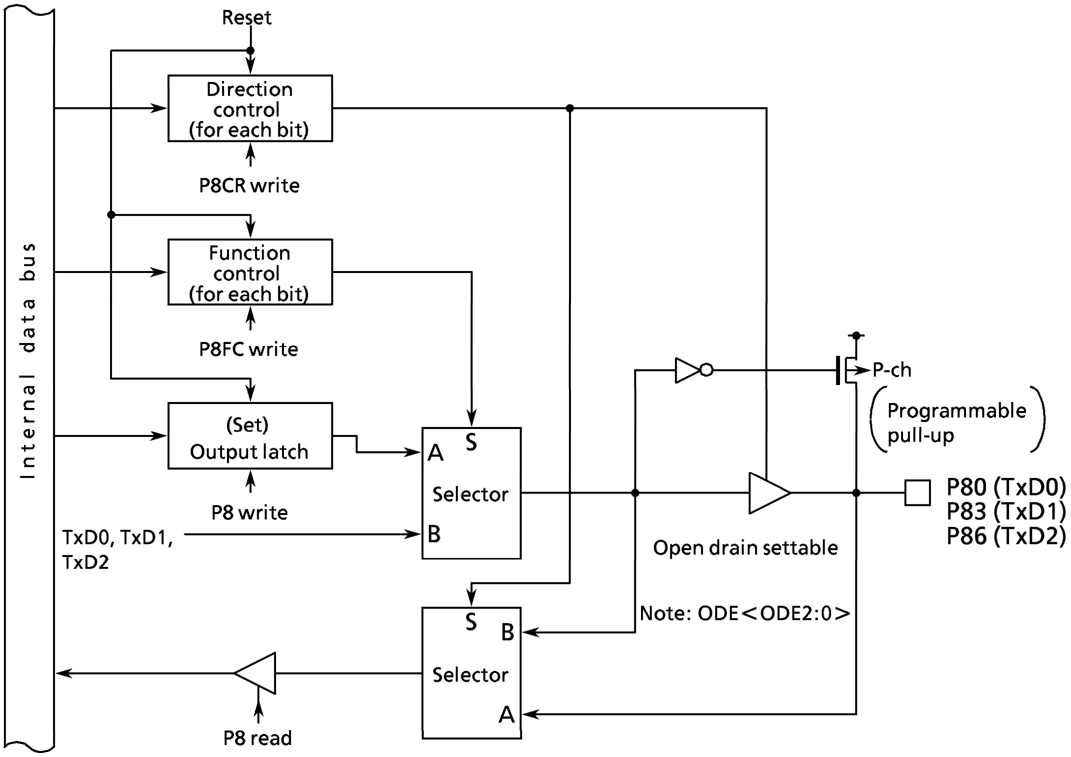


Figure 3.5 (25) Port 8 (P80, P83, P86)

(2) Port 81, 84, 87 (RxD0, 1, 2)

Ports 81, 84, and 87 function as serial channel RxD0 to 2 inputs as well as input/output ports.

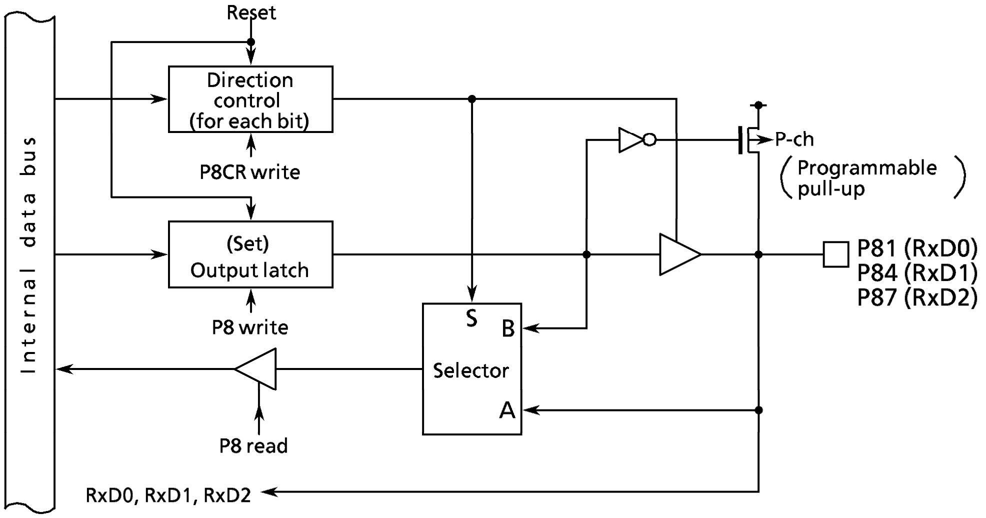


Figure 3.5 (26) Port 8 (P81, P84, P87)

(3) Port 82 (SCLK0/$\overline{\text{CTS0}}$)

Port 82 functions as the SCLK0 input/output for serial channel 0 as well as an input/output port. The port also functions as the $\overline{\text{CTS0}}$ input.

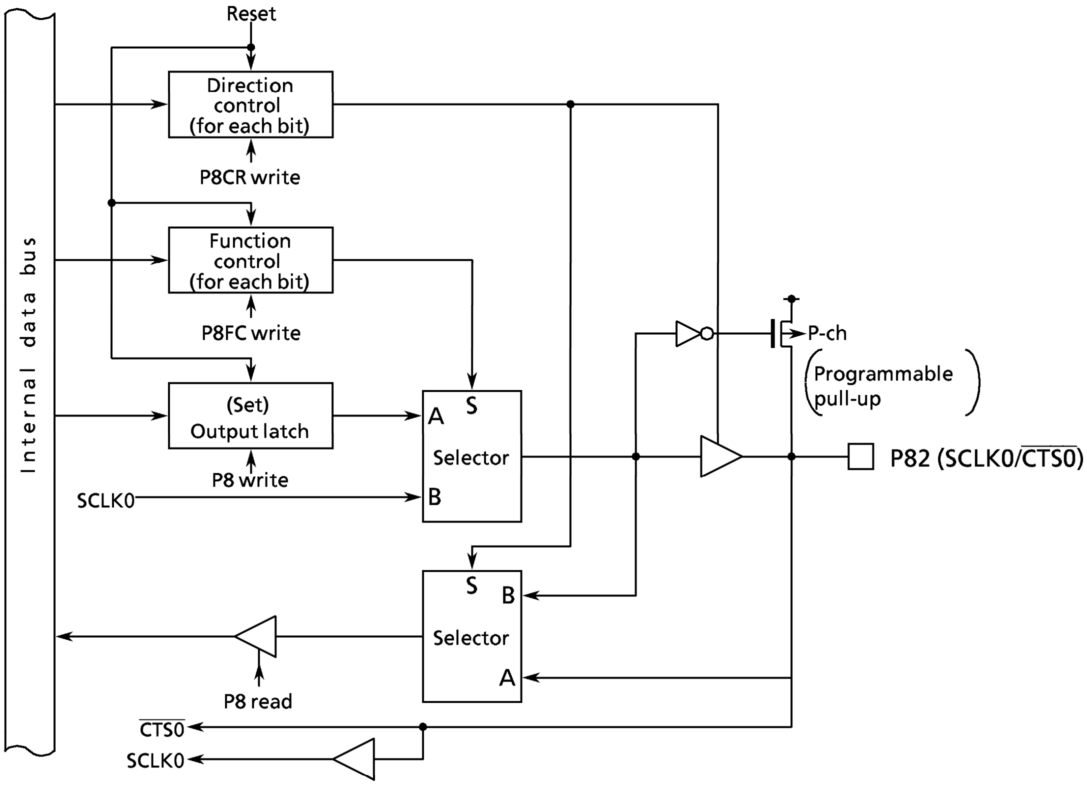


Figure 3.5 (27) Port 8 (P82)

(4) Port 85 (SCLK1/$\overline{\text{CTS1}}$)

Port 85 functions as the SCLK1 input/output for serial channel 1 as well as an input/output port. The port also functions as the $\overline{\text{CTS1}}$ input.

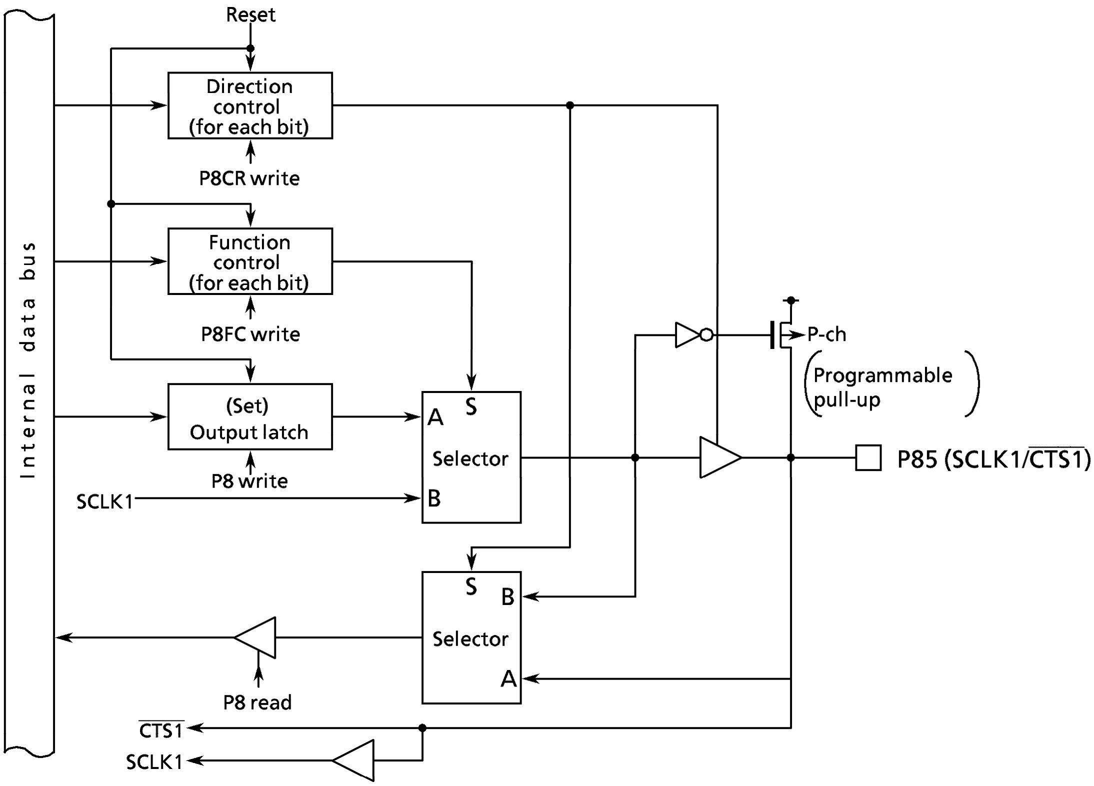


Figure 3.5 (28) Port 8 (P85)

Port 8 Register

P8
(0018H)

	7	6	5	4	3	2	1	0
bit Symbol	P87	P86	P85	P84	P83	P82	P81	P80
Read/Write	R/W							
After reset	Input mode (Set to 1/pulled up)							
Function	Also functions as RxD2	Also functions as TxD2	Also functions as SCLK1/$\overline{\text{CTS1}}$	Also functions as RxD1	Also functions as TxD1	Also functions as SCLK0/$\overline{\text{CTS0}}$	Also functions as RxD0	Also functions as TxD0

Note: When port 8 is in input mode, the P8 register controls the internal pull-up resistor. When using port 8 in input mode or in both input and output modes (if a bit is set to input), do not execute read-modify-write instructions. The internal pull-up resistor setting may change depending on the state of the input pin.

Open Drain Enable Register

ODE
(0058H)

	7	6	5	4	3	2	1	0
bit Symbol						ODE2	ODE1	ODE0
Read/Write						R/W		
After reset						0	0	0
Function						Port 86 output settings 0: CMOS 1: Open drain	Port 83 output settings 0: CMOS 1: Open drain	Port 80 output settings 0: CMOS 1: Open drain

Port 80 output settings

0	CMOS output
1	Open drain output

Port 83 output settings

0	CMOS output
1	Open drain output

Port 86 output settings

0	CMOS output
1	Open drain output

Figure 3.5 (29)-1 Port 8 Related Registers

Port 8 Control Register

P8CR (001AH)

Read-modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol	P87C	P86C	P85C	P84C	P83C	P82C	P81C	P80C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Port 8 input / output setting 0: Input 1: Output							

Port 8 Function Register

P8FC (001BH)

Read-modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol		P86F	P85F		P83F	P82F		P80F
Read/Write		W			W			W
After reset		0	0		0	0		0
Function		0: Port 1: TxD2	0: Port 1: SCLK1 /$\overline{\text{CTS1}}$		0: Port 1: TxD1	0: Port 1: SCLK0 /$\overline{\text{CTS0}}$		0: Port 1: TxD0

Port 85 function settings

<P85F> <P85C>	0	1
0	Input port SCLK1/$\overline{\text{CTS1}}$ (When <P85> = 0, not pulled up; when <P85> = 1, pulled up)	Do not set
1	Output port	SCLK1 output

Port 86 function settings

<P86F> <P86C>	0	1
0	Input port (When <P86> = 0, not pulled up; when <P86> = 1, pulled up)	Do not set
1	Output port	TxD2 output

Port 80 function settings

<P80F> <P80C>	0	1
0	Input port (When <P80> = 0, not pulled up; when <P80> = 1, pulled up)	Do not set
1	Output port	TxD0 output

Port 82 function settings

<P82F> <P82C>	0	1
0	Input port SCLK0/$\overline{\text{CTS0}}$ (When <P82> = 0, not pulled up; when <P82> = 1, pulled up)	Do not set
1	Output port	SCLK0 output

Port 83 function settings

<P83F> <P83C>	0	1
0	Input port (When <P83> = 0, not pulled up; when <P83> = 1, pulled up)	Do not set
1	Output port	TxD1 output

Note: When setting pins TxD0 to 2 to open drain output, set the open drain enable register ODE<ODE0:2> to 1. Pins P81/RxD0, P84/RxD1, and P87/RXD2 do not have port/function switching registers. Therefore, even when the pins are used as input ports, data are still input to SIO as serial receive data.

Figure 3.5 (29)-2 Port 8 Related Registers

3.5.10 Port 9 (P90 to P96)

Port 9 is a 7-bit general-purpose input/output port with each port bit settable as an input or output.
In addition to its input/output port functions, port 9 also functions as a 16-bit timer input clock pin, a 16-bit timer output pin, and inputs for INT5 to 8. Port 9 control register P9CR and port 9 function register P9FC set the port 9 functions.
A reset clears all bits of the P9 output latch and all bits of the P9CR and P9FC registers to 0, setting port 9 to input mode.
To enable the timer output function, write 1 to the corresponding bit in P9FC.

(1) Ports 90, 91, 94, 95 (TI8/INT5, TI9/INT6, TIA/INT7, TIB/INT8)

In addition to functioning as general-purpose input/output ports, ports 90 and 91 can also function as timer 8 event count inputs TI8 and TI9, and as external interrupt request inputs INT5 and INT6. Ports 94 and 95, in addition to being general-purpose input/output ports, can also function as the timer 9 event count inputs TIA and TIB, and as the external interrupt request inputs INT7 and INT8.

Cautions when using INT5 to INT8 interrupts

Input is always enabled for the INT5 to INT8 external interrupt requests.
Caution is required if ports 90, 91, 94, or 95 are used as general-purpose input/output ports or timer event count inputs while the INT5 to INT8 interrupt functions are in use. This is because rising or falling edges on these input/output signals generate interrupt requests.

Cautions when using timer event count inputs TI8 to TIB

Input is always enabled for timer event count inputs TI8 to TIB.
Caution is required if ports 90, 91, 94, or 95 are used as general-purpose input/output ports or INT5 to INT8 interrupts during event counting based on TI8 to TIB. This is because these input/output signals trigger an event count on the timer.

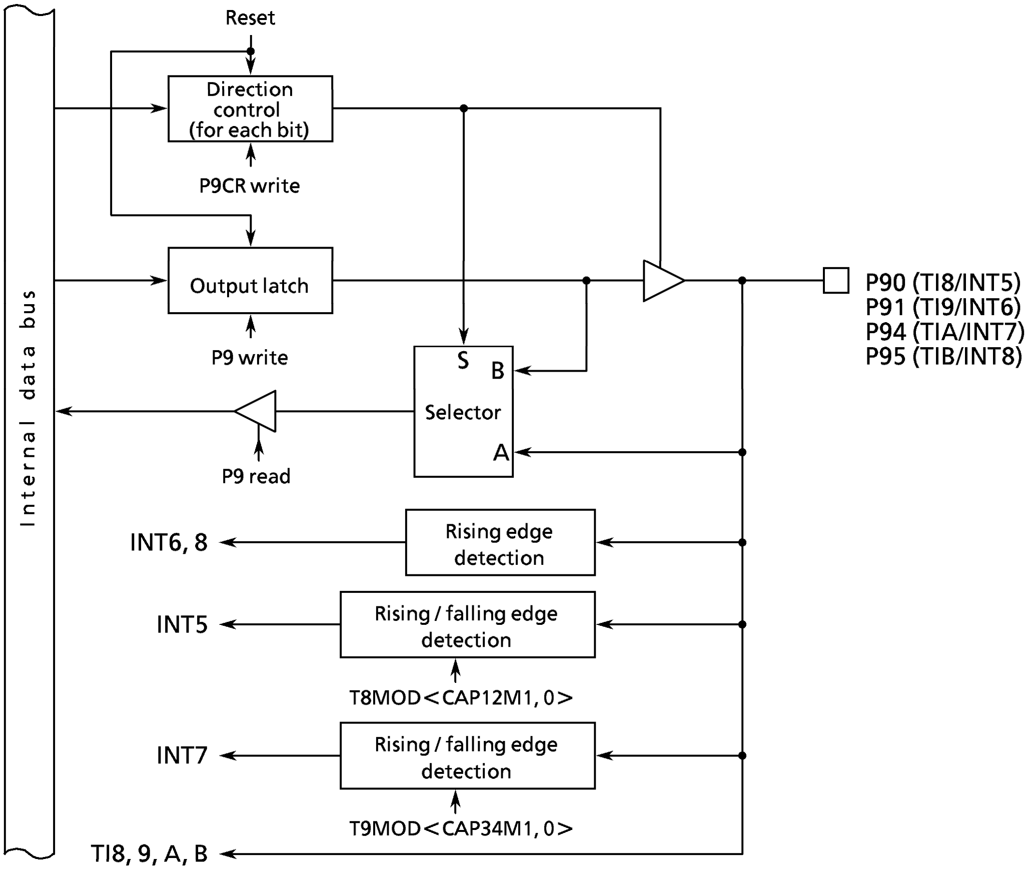


Figure 3.5 (30) Port 9 (P90, P91, P94, P95)

(2) Ports 92, 93 (TO8, TO9)

In addition to operating as a general-purpose input/output port, port 92 also functions as the TO8 output for timer 8. Port 93 operates as the TO9 output for timer 8 as well as functioning as a general-purpose input/output port.

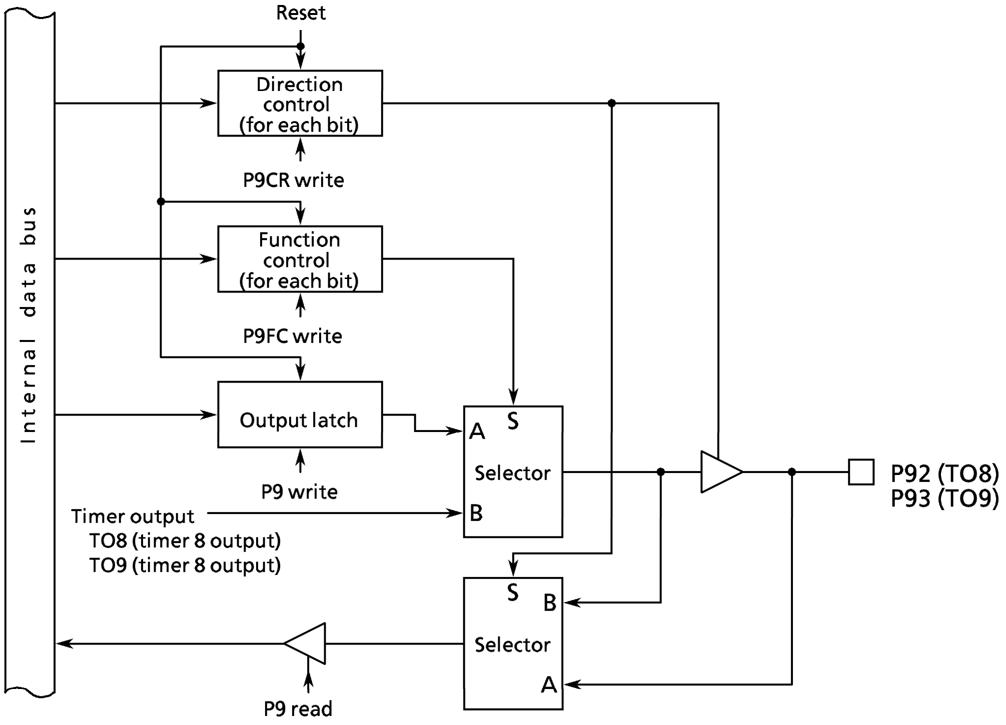


Figure 3.5 (31) Port 9 (P92, P93)

(3) Port 96 (TOA/TOB)

In addition to functioning as a general-purpose input/output port, port 96 also functions as the TOA and TOB outputs for timer 9.

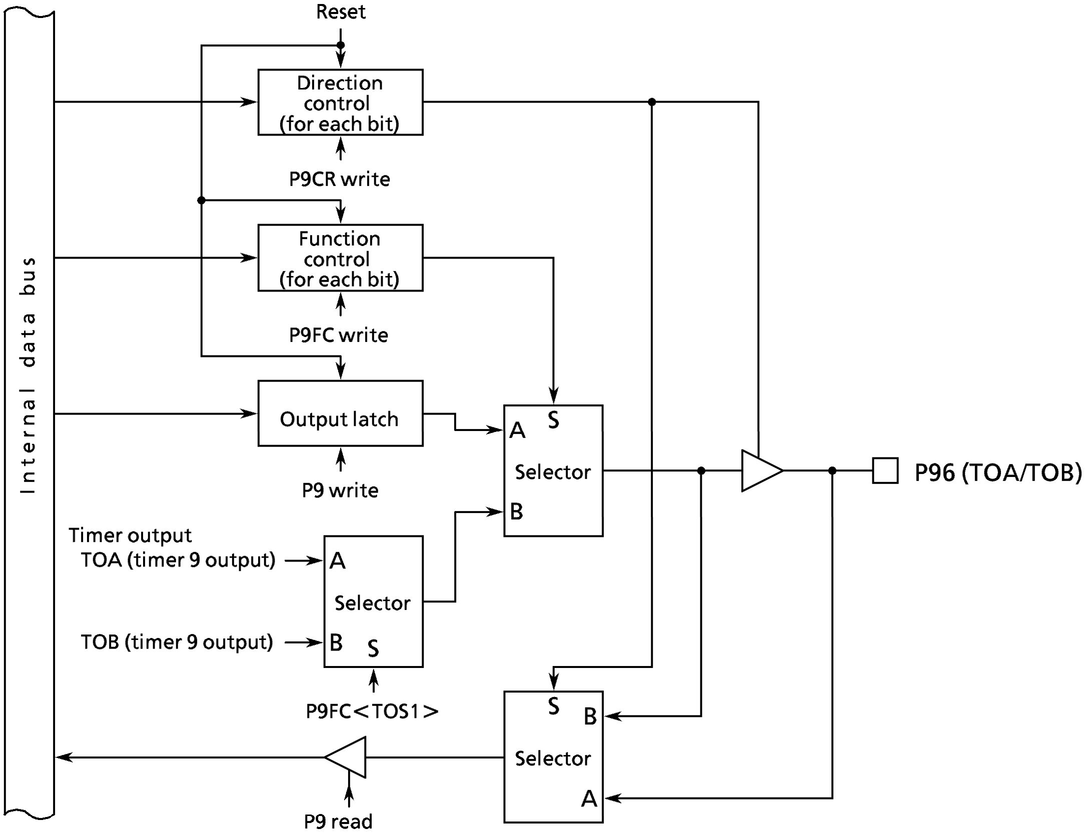


Figure 3.5 (32) Port 9 (P96)

Port 9 Register

P9
(0019H)

	7	6	5	4	3	2	1	0
bit Symbol		P96	P95	P94	P93	P92	P91	P90
Read/Write		R/W						
After reset		Input mode (output latch register cleared to 0)						
Function		Also functions as TOA/TOB	Also functions as TIB/INT8 (Note)	Also functions as TIA/INT7 (Note)	Also functions as TO9	Also functions as TO8	Also functions as TI9/INT6 (Note)	Also functions as TI8/INT5 (Note)

Port 9 Control Register

P9CR
(001CH)

Read-modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol		P96C	P95C	P94C	P93C	P92C	P91C	P90C
Read/Write		W						
After reset		0	0	0	0	0	0	0
Function		Port 9 input / output settings 0: Input 1: Output						

Port 9 Function Register

P9FC
(001DH)

Read-modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol	TOS1	P96F			P93F	P92F		
Read/Write	W				W			
After reset	0	0			0	0		
Function	TOA/TOB output selection 0: TOA 1: TOB	0:Port 1: TOA /TOB			0:Port 1: TO9	0:Port 1: TO8		

Port 96 function settings

<P96F> <P96C>	0	1
0	Input port	
1	Output port	TOA/TOB output

Timer out A/B output selection

0	Timer out A output
1	Timer out B output

Port 92 function settings

<P92F> <P92C>	0	1
0	Input port	
1	Output port	TO8 output

Port 93 function settings

<P93F> <P93C>	0	1
0	Input port	
1	Output port	TO9 output

Note: No register is provided to switch between the external interrupt input, event counter input, and port input/output functions. Therefore, even when port 9 is used as a port, the interrupt signal input and event counter input are enabled.
When using port 9 exclusively as a port, disable the external interrupts (INT5 to 8) and event count inputs (TI8 to B).

Figure 3.5 (33) Port 9 Related Registers

3.5.11 Port A (PA0 to PA7)

Port A is an 8-bit input-only port with analog input pins (AN0 to AN7). The PA3 pin also functions as the external trigger input for analog conversion ($\overline{\text{ADTRG}}$).

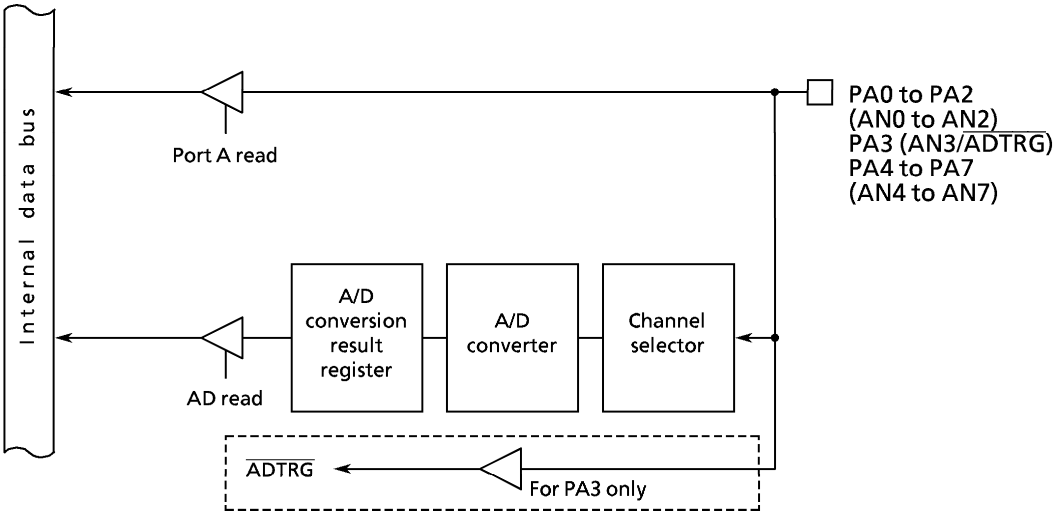


Figure 3.5 (34) Port A (PA0 to PA7)

Port A Register

		7	6	5	4	3	2	1	0
PA (001EH)	bit Symbol	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
	Read/Write	R							
	After reset	Input only							
	Function	Also functions as AN7	Also functions as AN6	Also functions as AN5	Also functions as AN4	Also functions as AN3 /$\overline{\text{ADTRG}}$	Also functions as AN2	Also functions as AN1	Also functions as AN0

Note: A/D mode register 1, ADMOD1, selects the A/D converter input channel.

Figure 3.5 (35) Port A Related Registers

3.6 Chip Select/Wait Controller

In TMP95CS64/265, four user-specifiable address area blocks (CS0 to CS3) can be set. The data bus width and number of waits can be set independently for each address area (CS0 to CS3 and others).

The $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ pins (which also function as P60 to P63) are the output pins for the CS0 to CS3 areas. When the CPU specifies an address in one of these areas, these pins output the chip select signal for that address area (ROM/SRAM or PSRAM signal). However, to output the chip select signal, the port 6 function register P6FC must be set. TMP95CS64/265 supports connection of external PSRAM as well as ROM and SRAM.

The CS0 to CS3 areas are set by a combination of memory start address registers MSAR0 to MSAR3 and memory address mask registers MAMR0 to MAMR3.

Use chip select/wait control registers B0CS to B3CS and BEXCS to specify the master enable, data bus width, and number of waits for each address area.

The input pins controlling these states are the bus wait request pin ($\overline{\text{WAIT}}$), the external data bus selection pin (AM8/$\overline{\text{16}}$), and the external memory access pin ($\overline{\text{EA}}$). (See 3.1.2, External Data Bus Width Selection Function.)

3.6.1 Specifying Address Areas

The CS0 to CS3 address areas are specified using the start address registers (MSAR0 to MSAR3) and memory address mask registers (MAMR0 to MAMR3).

At each bus cycle, a compare operation is performed to determine if the address on the bus specifies a location in the CS0 to CS3 areas. If the result of the comparison is a match, this indicates an access to the corresponding CS area. In this case, the $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ pin outputs the chip select signal and the bus cycle operates in accordance with the settings in chip select/wait control register B0CS to B3CS. (See 3.6.2, Chip Select/Wait Control Register.)

(1) Memory Start Address Registers

Figure 3.6 (1) shows the memory start address registers. Memory start address registers MSAR0 to MSAR3 set the start address for the CS0 to CS3 areas. Set the upper eight bits (A23 to A16) of the start address in <S23:16>. The lower 16 bits of the start address (A15 to A0) are permanently set to 0. Accordingly, the start address can only be set in 64 K-byte increments, starting from 000000H. Figure 3.6 (2) shows the relationship between the start address and the start address register value.

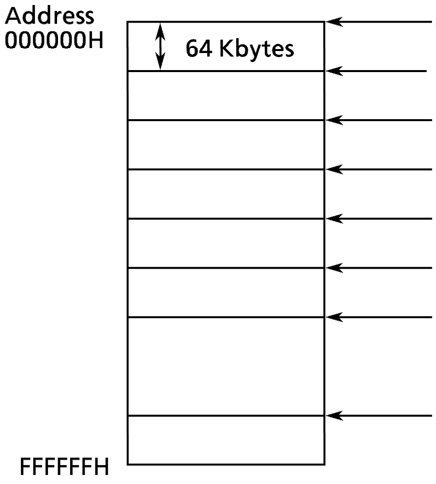
Memory start address register (CS0 to CS2 areas)

MSAR0 (0094H) / MSAR1 (0096H)
MSAR2 (0098H) / MSAR3 (009AH)

	7	6	5	4	3	2	1	0
bit Symbol	S23	S22	S21	S20	S19	S18	S17	S16
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	Sets start address A23 to A16							

Sets start address of CS0 to CS2 area

Figure 3.6 (1) Memory Start Address Register



Start address	Start address register value (MSAR0 to 3)
000000H	00H
010000H	01H
020000H	02H
030000H	03H
040000H	04H
050000H	05H
060000H	06H
to	to
FF0000H	FFH

Figure 3.6 (2) Relationship Between Start Address and Start Address Register Value

(2) Memory Address Mask Registers

Figure 3.6 (3) shows the memory address mask registers. Memory address mask registers MAMR0 to MAMR3 are used to set the size of the CS0 to CS3 areas by specifying a mask for each bit of the start address set in memory start address registers MSAR0 to MSAR3. The compare operation used to determine if an address is in the CS0 to CS3 areas is only performed for bus address bits corresponding to bits set to 0 in these registers.
Also, the address bits that can be masked by MAMR0 to MAMR3 differ between CS0 to CS3 areas. Accordingly, the size that can be set for each area is different.

Memory address mask register (CS0)

MAMR0
(0095H)

	7	6	5	4	3	2	1	0
bit Symbol	V20	V19	V18	V17	V16	V15	V14 to 9	V8
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	Sets size of CS0 area 0: Used for address compare							

The CS0 area can be set within the following range: 256 bytes to 2 Mbytes.

Memory address mask register (CS1)

MAMR1
(0097H)

	7	6	5	4	3	2	1	0
bit Symbol	V21	V20	V19	V18	V17	V16	V15 to 9	V8
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	Sets size of CS1 area 0: Used for address compare							

The CS1 area can be set within the following range: 256 bytes to 4 Mbytes.

Memory address mask register (CS2, CS3)

MAMR2 / MAMR3
(0099H) / (009BH)

	7	6	5	4	3	2	1	0
bit Symbol	V22	V21	V20	V19	V18	V17	V16	V15
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	Sets size of CS2, CS3 area 0: Used for address compare							

The CS2 and CS3 areas can be set within the following range: 32 Kbytes to 8 Mbytes.

Figure 3.6 (3) Memory Address Mask Registers

(3) How to Set Memory Start Addresses and Address Areas

Figure 3.6 (4) shows an example of specifying a 64 K-byte address area starting from 010000H using the CS0 area.
Set 01H in memory start address register MSAR0<S23:16> (corresponding to the upper 8 bits of the start address). Next, calculate the difference between the start address and the anticipated end address (01FFFFH) based on the size of the CS0 area. Bits 20 to 8 of the result correspond to the mask value to be set for the CS0 area. Setting this value in memory address mask register MAMR0<V20:8> sets the area size. This example sets 07H in MAMR0 to specify a 64 K-byte area.

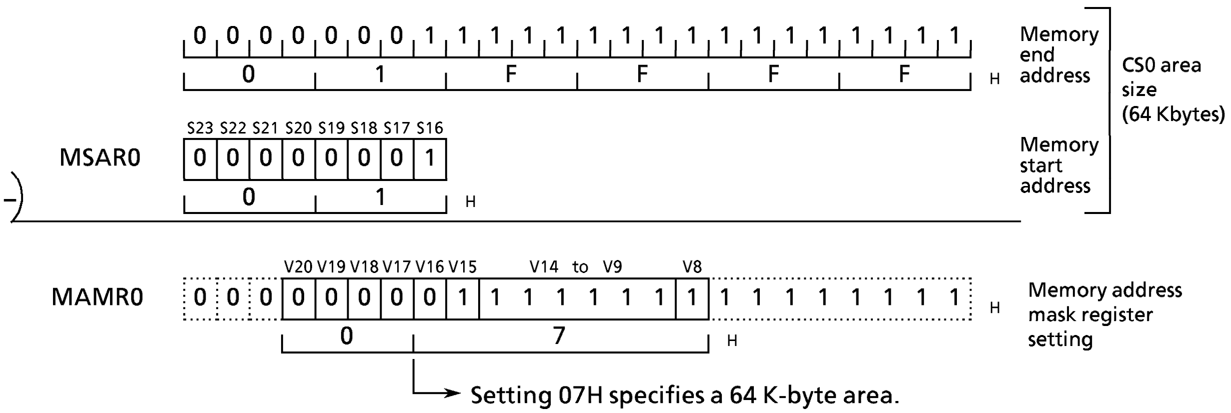


Figure 3.6 (4) CS0 Area Setting Example

After a reset, MSAR0 to MSAR3 and MAMR0 to MAMR3 are set to FFH. B0CS<B0E>, B1CS<B1E>, and B3CS<B3E> are reset to 0. This disables the CS0, CS1, and CS3 areas. However, as B2CS<B2M> is reset to 0 and B2CS<B2E> to 1, CS2 is enabled from 0008A0H to FEFFFFH in TMP95CS64, and from 0008A0H to FFFFFFH in TMP95C265. Also, the bus width and number of waits specified in BEXCS are used for accessing addresses outside the specified CS0 to CS3 areas. (See 3.6.2, Chip Select/Wait Control Register.)

(4) Address Area Size Specifications

Table 3.6 (1) shows the relationship between CS area and area size. △ indicates areas that cannot be set by memory start address register and memory address mask register combinations. When setting an area size using a combination indicated by △, set the start address in the desired steps starting from 000000H.
If the CS2 area is set to 16 M-byte or if two or more areas overlap, the smaller CS area number has the higher priority.

Example: When setting CS0 as a 128 K-byte area:

① Available start addresses

000000H
128 Kbytes
020000H
128 Kbytes
040000H
128 Kbytes
060000H
⋮

Any of these start addresses can be set.

② Unavailable start addresses

000000H
64 Kbytes
010000H
128 Kbytes
030000H
128 Kbytes
050000H
⋮

← This exceeds the size of the steps that can be set. In this case, the following start addresses cannot set the desired area size.

Table 3.6 (1) CS Area and Area Size

Size (bytes) CS area	256	512	32 K	64 K	128 K	256 K	512 K	1 M	2 M	4 M	8 M
CS0	○	○	○	○	△	△	△	△	△		
CS1	○	○		○	△	△	△	△	△	△	
CS2			○	○	△	△	△	△	△	△	△
CS3			○	○	△	△	△	△	△	△	△

3.6.2 Chip Select/Wait Control Registers

Table 3.6 (5) lists the chip select/wait control registers. The master enable/disable, chip select output waveform, data bus width, and number of wait states for each address area (CS0 to CS3 and others) are set in their respective chip select/wait control registers, B0CS to B3CS and BEXCS.

Chip Select/Wait Control Register

		7	6	5	4	3	2	1	0
B0CS (0090H)	bit Symbol	B0E		B0OM1	B0OM0	B0BUS	B0W2	B0W1	B0W0
	Read/Write	W		W					
	After reset	0		0	0	0	0	0	0
Read-modify-write instructions prohibited.	Function	0: Disable 1: Enable		Chip select output waveform selection 00: For ROM/SRAM 01: For PSRAM 10: } Don't care 11:		Data bus width 0: 16-bit 1: 8-bit	Number of Waits Do not set 000: 2 WAIT 100: 0 + N WAIT 001: 1 WAIT 101 010: 1 WAIT + N 110 } Do not set 011: 0 WAIT 111		
B1CS (0091H)	bit Symbol	B1E		B1OM1	B1OM0	B1BUS	B1W2	B1W1	B1W0
	Read/Write	W		W					
		0		0	0	0	0	0	0
Read-modify-write instructions prohibited.	Function	0: Disable 1: Enable		Chip select output waveform selection 00: For ROM/SRAM 01: For PSRAM 10: } Don't care 11:		Data bus width 0: 16-bit 1: 8-bit	Number of Waits Do not set 000: 2 WAIT 100: 0 + N WAIT 001: 1 WAIT 101 010: 1 WAIT + N 110 } Do not set 011: 0 WAIT 111		
B2CS (0092H)	bit Symbol	B2E	B2M	B2OM1	B2OM0	B2BUS	B2W2	B2W1	B2W0
	Read/Write	W							
	After reset	1	0	0	0	0	0	0	0
Read-modify-write instructions prohibited.	Function	0: Disable 1: Enable	CS2 area selection 0: 16 M-byte area 1: CS area	Chip select output waveform selection 00: For ROM/SRAM 01: For PSRAM 10: } Don't care 11:		Data bus width 0: 16-bit 1: 8-bit	Number of Waits Do not set 000: 2 WAIT 100: 0 + N WAIT 001: 1 WAIT 101 010: 1 WAIT + N 110 } Do not set 011: 0 WAIT 111		
B3CS (0093H)	bit Symbol	B3E		B3OM1	B3OM0	B3BUS	B3W2	B3W1	B3W0
	Read/Write	W		W					
		0		0	0	0	0	0	0
Read-modify-write instructions prohibited.	Function	0: Disable 1: Enable		Chip select output waveform selection 00: For ROM/SRAM 01: For PSRAM 10: } Don't care 11:		Data bus width 0: 16-bit 1: 8-bit	Number of Waits Do not set 000: 2 WAIT 100: 0 + N WAIT 001: 1 WAIT 101 010: 1 WAIT + N 110 } Do not set 011: 0 WAIT 111		
BEXCS (009CH)	bit Symbol					BEXBUS	BEXW2	BEXW1	BEXW0
	Read/Write					W			
	After reset					0	0	0	0
Read-modify-write instructions prohibited.	Function					Data bus width 0: 16-bit 1: 8-bit	Number of Waits Do not set 000: 2 WAIT 100: 0 + N WAIT 001: 1 WAIT 101 010: 1 WAIT + N 110 } Do not set 011: 0 WAIT 111		

Master enable bit

0	CS area disable
1	CS area enable

CS2 area selection

0	16 M-byte area
1	Address specification area

Chip select output waveform selection

00	For ROM/SRAM
01	For PSRAM
10 11	Don't care

Number of address area waits setting
(See 3.6.2, (4) Wait Control.)

Data bus width selection

0	16-bit data bus
1	8-bit data bus

Figure 3.6 (5) Chip Select/Wait Control Registers

(1) Master Enable Bits

Bit 7 (<B0E>, <B1E>, <B2E>, and <B3E>) of the chip select/wait control registers is the master bit used to enable or disable settings for the address area. Writing 1 to the bit enables the settings. Reset disables (sets to 0) <B0E>, <B1E>, and <B3E>, and enables (sets to 1) <B2E>. This enables area CS2 only.

(2) Selection of Chip Select Output Waveform

Bits 5 and 4 (<B0OM1, 0>, <B1OM1, 0>, <B2OM1, 0>, and <B3OM1, 0>) of the chip select/wait control registers specify the chip select output waveform for external memory access. Setting the bits to 00 outputs the chip select signal for selecting ROM and SRAM from the $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ pins. Setting the bits to 01 outputs the chip select signal for selecting PSRAM from the $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ pins.
For details on the waveform of the chip select signal during external memory access, see Figure 3.6 (6)

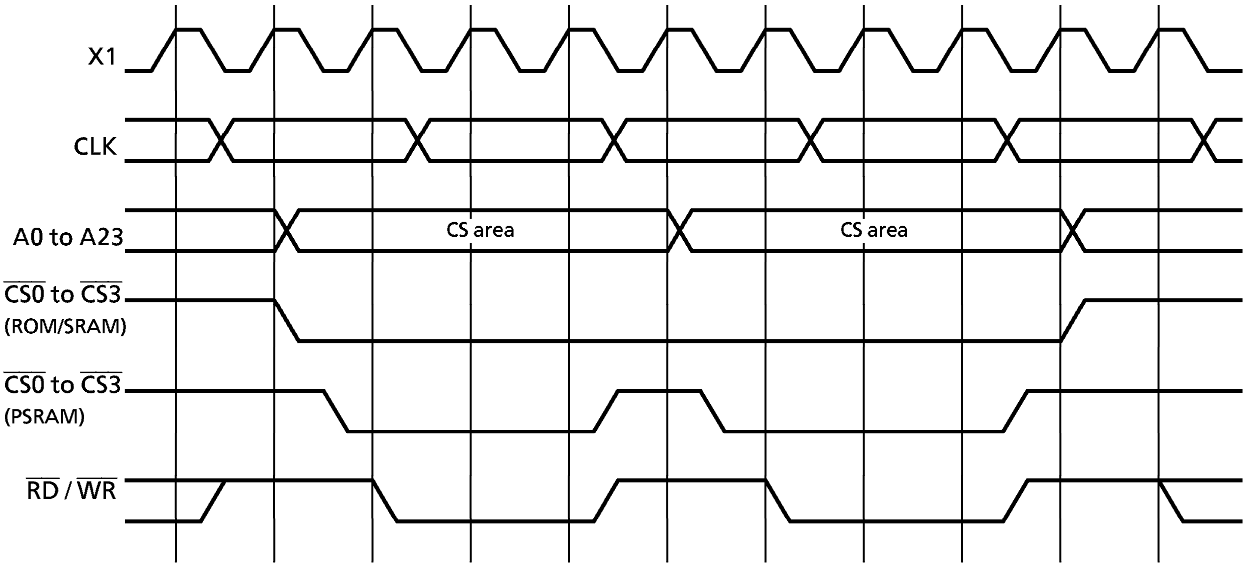

X1
CLK
A0 to A23
CS area
CS area
$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$
(ROM/SRAM)
$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$
(PSRAM)
$\overline{\text{RD}}$ / $\overline{\text{WR}}$

Figure 3.6 (6) Waveform for Chip Select Signal Operation at External Memory Access ($\overline{\text{CS0}}$ to $\overline{\text{CS3}}$)

(3) Selection of Data Bus Width

Bit 3 (<B0BUS>, <B1BUS>, <B2BUS>, <B3BUS>, and <BEXBUS>) of the chip select/wait control registers specifies the width of the data bus. Set 0 to access memory when using a 16-bit data bus. Set 1 when using an 8-bit data bus.

- TMP95CS64
  Connect the $\overline{\text{EA}}$ and AM8/$\overline{\text{16}}$ pins to VCC. This enables external memory to be accessed using the data bus width set in the data bus width select bit.
- TMP95C265
  Connect the $\overline{\text{EA}}$ pin to GND. This enables external memory to be accessed using the data bus width set in the data bus width select bit only when the AM8/$\overline{\text{16}}$ pin is at low level.
  If the AM8/$\overline{\text{16}}$ pin is at high level, external address areas are accessed using an 8-bit data bus.

This method of changing the data bus width depending on the address being accessed is called dynamic bus sizing. For details of this bus operation, see Table 3.6 (2).

Table 3.6 (2) Dynamic Bus Sizing

Operand Data Bus Width	Operand Start Address	Memory Data Bus Width	CPU Address	CPU Data	
				D15 to D8	D7 to D0
8-bit	2n + 0 (Even number)	8 bits	2n + 0	xxxxx	b7 to b0
		16 bits	2n + 0	xxxxx	b7 to b0
	2n + 1 (Odd number)	8 bits	2n + 1	xxxxx	b7 to b0
		16 bits	2n + 1	b7 to b0	xxxxx
16-bit	2n + 0 (Even number)	8 bits	2n + 0	xxxxx	b7 to b0
			2n + 1	xxxxx	b15 to b8
		16 bits	2n + 0	b15 to b8	b7 to b0
	2n + 1 (Odd number)	8 bits	2n + 1	xxxxx	b7 to b0
			2n + 2	xxxxx	b15 to b8
		16 bits	2n + 1	b7 to b0	xxxxx
			2n + 2	xxxxx	b15 to b8
32-bit	2n + 0 (Even number)	8 bits	2n + 0	xxxxx	b7 to b0
			2n + 1	xxxxx	b15 to b8
			2n + 2	xxxxx	b23 to b16
			2n + 3	xxxxx	b31 to b24
		16 bits	2n + 0	b15 to b8	b7 to b0
			2n + 2	b31 to b24	b23 to b16
	2n + 1 (Odd number)	8 bits	2n + 1	xxxxx	b7 to b0
			2n + 2	xxxxx	b15 to b8
			2n + 3	xxxxx	b23 to b16
			2n + 4	xxxxx	b31 to b24
		16 bits	2n + 1	b7 to b0	xxxxx
			2n + 2	b23 to b16	b15 to b8
			2n + 4	xxxxx	b31 to b24

xxxxx: Indicates that the input data from these bits are ignored during a read. During a write, indicates that the bus for these bits goes to high impedance; also, that the write strobe signal for the bus remains inactive.

(4) Wait Control

Bits 2 to 0 (<B0W2, 0>, <B1W2, 0>, <B2W2, 0>, <B3W2, 0>, and <BEXW2, 0>) of the chip select/wait control registers specify the number of waits to insert.
The following types of wait operation can be specified using combinations of these bits. Do not set combinations other than those listed in the table.

Table 3.6 (3) Wait Operation Settings

<BxW2:0>	No. of Waits	Wait Operation
000	2WAIT	Inserts a wait of two states, irrespective of the $\overline{\text{WAIT}}$ pin state.
001	1WAIT	Inserts a wait of one state, irrespective of the $\overline{\text{WAIT}}$ pin state.
010	1WAIT + N	Samples the state of the $\overline{\text{WAIT}}$ pin after inserting a wait of one state. If the $\overline{\text{WAIT}}$ pin is low, the waits continue and the bus cycle is extended until the pin goes high.
011	0WAIT	Ends the bus cycle without a wait, regardless of the $\overline{\text{WAIT}}$ pin state.
100	0 + NWAIT	Continuously samples the $\overline{\text{WAIT}}$ pin state and inserts waits if the pin is low, extending the bus cycle until the pin goes high.

Figures 3.6 (7) and (8) show the timing for N = 0, 1 when the setting is 0 + NWAIT.
For the timings for settings other than 0 + NWAIT, see Figures 7 (1) to (5) in 7, Basic Timing, Chapter 3, TLCS-900/H CPU.
Reset sets these bits to 000 (2 WAIT).

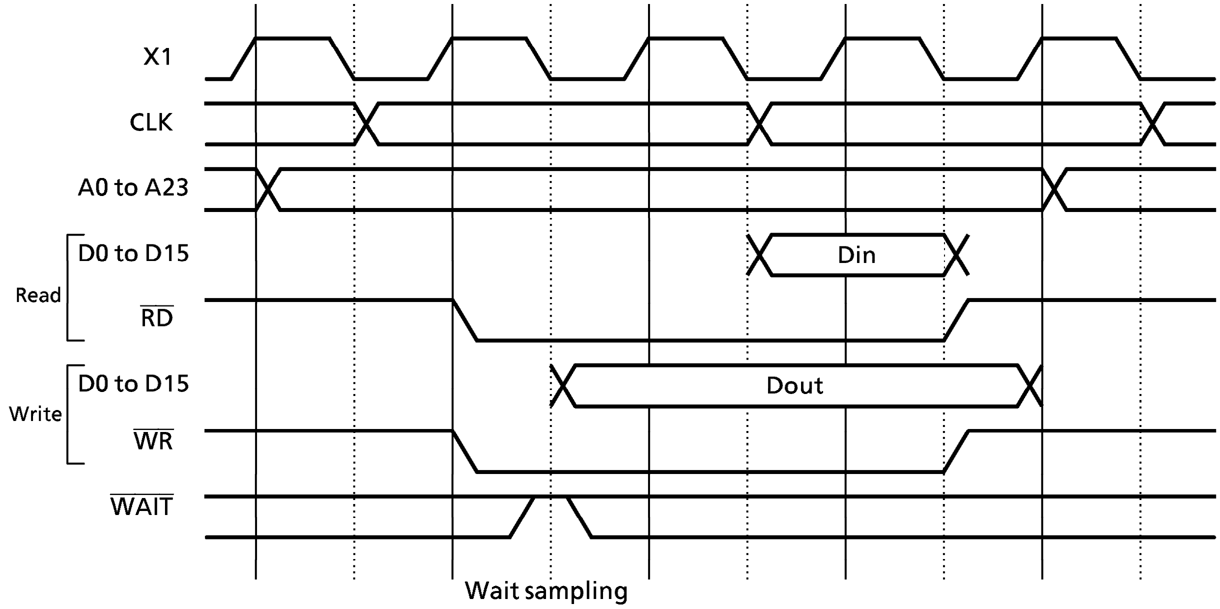


Figure 3.6 (7) 0 + NWAIT Read/Write Cycle (When N = 0)

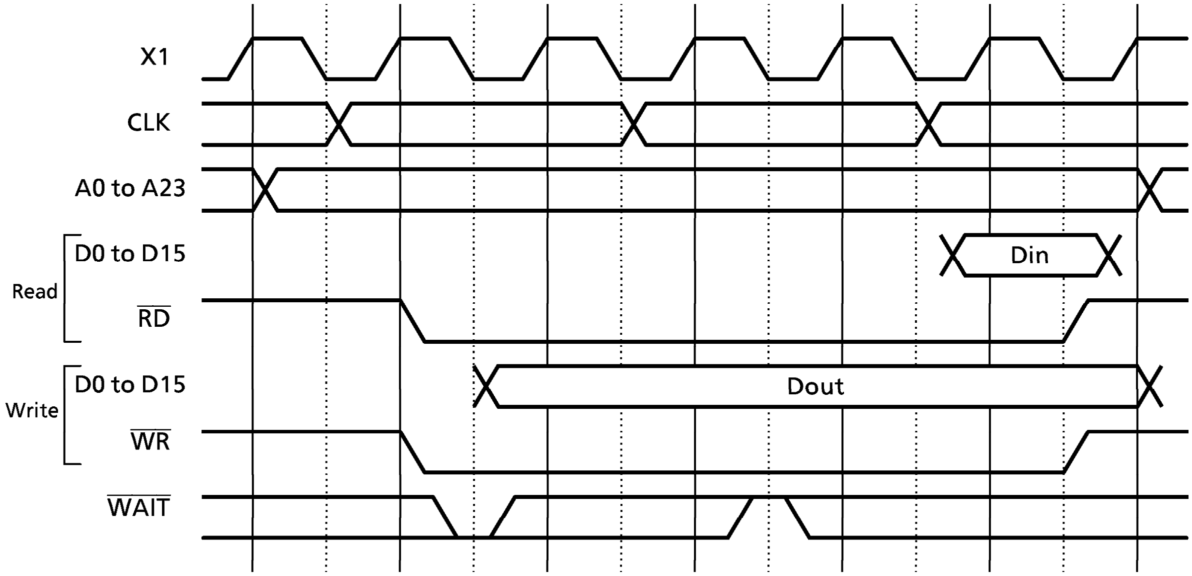


Figure 3.6 (8) 0 + NWAIT Read/Write Cycle (When N = 1)

(5) Bus Width and Wait Control Outside CS0 to CS3 Areas

The chip select/wait control register BEXCS controls the bus width and number of waits when locations outside the four user-specified address area blocks (CS0 to CS3) are accessed. The BEXCS register settings are always enabled for areas other than CS0 to CS3.

(6) 16 M-byte Area / Address Setting Area Selection

Setting the chip select/wait control register B2CS<B2M> to 0 selects a 16 M-byte address area (0008A0H to FEFFFFH) for CS2. (In TMP95C265, 0008A0H to FFFFFFH is a 16 M-byte area.) Setting B2CS<B2M> to 1 selects the address area specified by start address register MSAR2 and address mask register MAMR2 for CS2, and likewise for CS0, CS1, and CS3. Reset clears this bit to 0 and selects a 16 M-byte address area.

(7) Chip Select / Wait Control Setting Procedure

When using the chip select/wait control function, set the registers as follows:

① Set memory start address registers MSAR0 to MSAR3.
Set the CS0 to CS3 start addresses.

② Set memory address mask registers MAMR0 to MAMR3.
Set the size of CS0 to CS3.

③ Set control registers B0CS to B3CS.
Set the chip select output waveform, data bus width, number of waits, and master enable/disable for CS0 to CS3.

The $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ pins also function as pins P60 to P63. To output the chip select signal from these pins, set the corresponding bits of port 6 function register P6FC to 1.
In the case of addresses set for one of the CS0 to CS3 areas but which specify an internal I/O, RAM, or ROM area, the $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ pins do not output a chip select signal and the CPU accesses the internal area.

Setting example:

This example sets the CS0 area as 010000H to 01FFFFH (64 K-byte area) with a 16-bit bus and zero waits:

MSAR0 = 01H	Start address: 010000H
MAMR0 = 07H	Address area: 64 Kbytes
B0CS = 83H	ROM/SRAM, 16-bit data bus, zero waits, CS0 area settings enabled

3.6.3 How to Connect External Memory

Figure 3.6 (9) shows an example of connecting external memory to TMP95C265.
In the example, ROM is connected using a 16-bit bus. RAM and I/O are connected using an 8-bit bus.

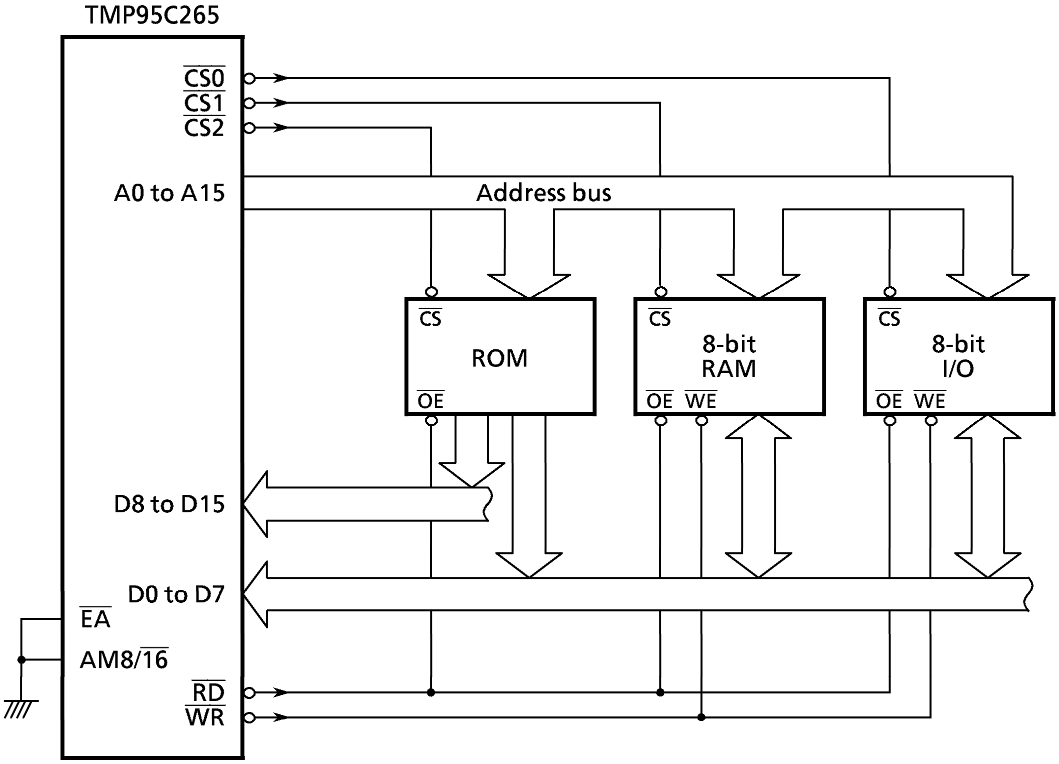


Figure 3.6 (9) External Memory Connection Example
(ROM = 16-bit bus, RAM and I/O = 8-bit bus)

After resetting TMP95C265, the <P62> bit of the port 6 register is cleared to 0 and pin P62 ($\overline{CS2}$) outputs a low signal. This enables the CS2 area.
However, as port 6 function register P6FC is cleared to 0, the CS signal output is disabled. When outputting the CS signal, set the necessary P6FC bit to 1.

3.7 8-Bit Timers

TMP95CS64/265 incorporates eight 8-bit timers (timers 0 to 7).
Each timer can operate independently or be cascaded to form four 16-bit timers. The 8-bit timers have the following four operating modes.

- 8-bit interval timer mode (8 channels)
- 16-bit interval timer mode (4 channels)

These modes can be combined (for example, four 8-bit timers and two 16-bit timers).

- 8-bit programmable square wave pulse generation (PPG: variable cycle, variable duty) output mode (4 channels)
- 8-bit PWM (pulse width modulation: variable duty at fixed cycle) output mode (4 channels)

Figure 3.7 (1) shows the block diagram of 8-bit timers (timers 0, 1). Other 8-bit timers (timers 2 and 3, 4 and 5, and 6 and 7) have the same circuit configuration as timers 0 and 1.
Each 8-bit timer consists of an 8-bit up-counter, an 8-bit comparator, and an 8-bit timer register. One timer flip-flop each (TFF1, TFF3, TFF5, and TFF7) is provided for the timer pairs, consisting of timers 0 and 1, timers 2 and 3, timers 4 and 5, and timers 6 and 7.
Of the input clock sources for the 8-bit timers, the ϕT1, ϕT4, ϕT16, and ϕT256 internal clocks are obtained from the 9-bit internal prescaler.
The 8-bit timers are controlled by nine control registers (T01MOD, T23MOD, T45MOD, T67MOD, T02FFCR, T46FFCR, T8RUN, T16RUN, and TRDC).

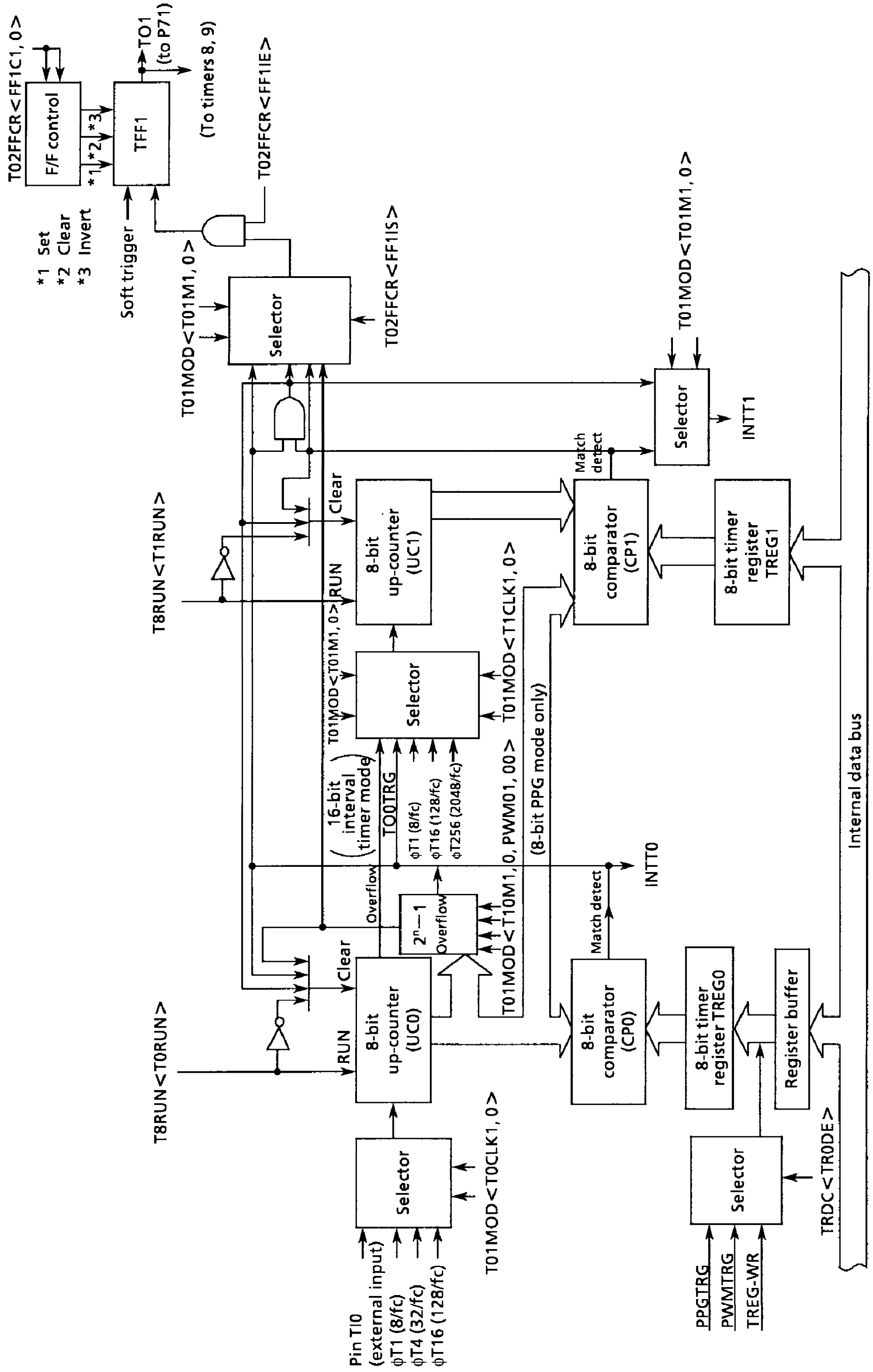


Figure 3.7 (1) 8-Bit Timer Block Diagram (Timers 0,1)

3.7.1 8-Bit Timer Registers

Figure 3.7 (2) shows the 8-bit timer registers. Setting these registers controls the operation of the 8-bit timers.

8-bit Timer Operation Control Register

T8RUN (0020H)

	7	6	5	4	3	2	1	0
bit Symbol	T7RUN	T6RUN	T5RUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Timer 7 operation control 0: Stop and clear 1: Run	Timer 6 operation control 0: Stop and clear 1: Run	Timer 5 operation control 0: Stop and clear 1: Run	Timer 4 operation control 0: Stop and clear 1: Run	Timer 3 operation control 0: Stop and clear 1: Run	Timer 2 operation control 0: Stop and clear 1: Run	Timer 1 operation control 0: Stop and clear 1: Run	Timer 0 operation control 0: Stop and clear 1: Run

8-bit timer up-counter operation control

0	Stop and clear
1	Run

16-Bit Timer Operation Control Register

T16RUN (003BH)

	7	6	5	4	3	2	1	0
bit Symbol	PRRUN		T9RUN	T8RUN				
Read/Write	R/W		R/W					
After reset	0		0	0				
Function	Prescaler operation control 0: Stop and clear 1: Run		Timer 9 operation control 0: Stop and clear 1: Run	Timer 8 operation control 0: Stop and clear 1: Run				

16-bit timer up-counter operation control (See 3.8, 16-bit Timers/Event Counters.)

Prescaler operation control (Note)

0	Stop and clear
1	Run

Note: Set T16RUN<PRRUN> to 1 when using an 8-bit timer.

Figure 3.7 (2)-1 8-Bit Timer Related Registers

Timer Register Double Buffer Control Register

TRDC (0021H)

	7	6	5	4	3	2	1	0
bit Symbol					TR6DE	TR4DE	TR2DE	TR0DE
Read/Write					R/W			
After reset					0	0	0	0
Function					TREG6 double buffer control 0: Disable 1: Enable	TREG4 double buffer control 0: Disable 1: Enable	TREG2 double buffer control 0: Disable 1: Enable	TREG0 double buffer control 0: Disable 1: Enable

Timer register 0 double buffer control

0	Disable
1	Enable

Timer register 2 double buffer control

0	Disable
1	Enable

Timer register 4 double buffer control

0	Disable
1	Enable

Timer register 6 double buffer control

0	Disable
1	Enable

Figure 3.7 (2)-2 8-Bit Timer Related Register

Timers 0, 1 Mode Control Register

T01MOD
(0024H)

	7	6	5	4	3	2	1	0
bit Symbol	T01M1	T01M0	PWM01	PWM00	T1CLK1	T1CLK0	T0CLK1	T0CLK0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Set operating mode for timers 0, 1 00: 8-bit interval timer 01: 16-bit interval timer 10: 8-bit PPG 11: 8-bit PWM		PWM0 cycle selection 00: Don't care 01: $2^6 - 1$ 10: $2^7 - 1$ 11: $2^8 - 1$		Timer 1 input clock selection 00: TO0TRG 01: ϕT1 10: ϕT16 11: ϕT256		Timer 0 input clock selection 00: TI0 01: ϕT1 10: ϕT4 11: ϕT16	

Timer 0 input clock selection

00	External clock TI0
01	Internal clock ϕT1 (8/fc)
10	Internal clock ϕT4 (32/fc)
11	Internal clock ϕT16 (128/fc)

Timer 1 input clock selection

	<T01M1, 0> ≠ 01	<T01M1, 0> = 01
00	Timer 0 comparator output	Timer 0 overflow output (16-bit interval timer mode)
01	Internal clock ϕT1 (8/fc)	
10	Internal clock ϕT16 (128/fc)	
11	Internal clock ϕT256 (2048/fc)	

PWM0 cycle selection
(Except for PWM mode (<T01M1, 0> = 11), don't care)

00	Don't care
01	$(2^6 - 1) \times$ timer 0 input clock cycle
10	$(2^7 - 1) \times$ timer 0 input clock cycle
11	$(2^8 - 1) \times$ timer 0 input clock cycle

Timers 0, 1 operating mode settings

00	8-bit interval timer x 2 channels (timers 0, 1)
01	16-bit interval timer
10	8-bit PPG output
11	8-bit PWM output (timer 0) + 8-bit interval timer (timer 1)

Figure 3.7 (2)-3 8-Bit Timer Related Register

Timers 2, 3 Mode Control Register

T23MOD
(0028H)

	7	6	5	4	3	2	1	0
bit Symbol	T23M1	T23M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Set operating mode for timers 2, 3 00: 8-bit interval timer 01: 16-bit interval timer 10: 8-bit PPG 11: 8-bit PWM		PWM2 cycle selection 00: Don’t care 01: $2^6 - 1$ 10: $2^7 - 1$ 11: $2^8 - 1$		Timer 3 input clock selection 00: TO2TRG 01: ϕT1 10: ϕT16 11: ϕT256		Timer 2 input clock selection 00: Don’t care 01: ϕT1 10: ϕT4 11: ϕT16	

→ Timer 2 input clock selection

00	Don’t care
01	Internal clock ϕT1 (8/fc)
10	Internal clock ϕT4 (32/fc)
11	Internal clock ϕT16 (128/fc)

→ Timer 3 input clock selection

	<T23M1, 0> ≠ 01	<T23M1, 0> = 01
00	Timer 2 comparator output	Timer 2 overflow output (16-bit interval timer mode)
01	Internal clock ϕT1 (8/fc)	
10	Internal clock ϕT16 (128/fc)	
11	Internal clock ϕT256 (2048/fc)	

→ PWM2 cycle selection
(Except for PWM mode (<T23M1, 0> = 11), don’t care)

00	Don’t care
01	$(2^6 - 1)$ × timer 2 input clock cycle
10	$(2^7 - 1)$ × timer 2 input clock cycle
11	$(2^8 - 1)$ × timer 2 input clock cycle

→ Timers 2, 3 operating mode settings

00	8-bit interval timer × 2 channels (timers 2, 3)
01	16-bit interval timer
10	8-bit PPG output
11	8-bit PWM output (timer 2) + 8-bit interval timer (timer 3)

Figure 3.7 (2)-4 8-Bit Timer Related Register

Timers 4, 5 Mode Control Register

T45MOD
(002BH)

	7	6	5	4	3	2	1	0
bit Symbol	T45M1	T45M0	PWM41	PWM40	T5CLK1	T5CLK0	T4CLK1	T4CLK0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Set operating mode for timers 4, 5 00: 8-bit interval timer 01: 16-bit interval timer 10: 8-bit PPG 11: 8-bit PWM		PWM4 cycle selection 00: Don't care 01: $2^6 - 1$ 10: $2^7 - 1$ 11: $2^8 - 1$		Timer 5 input clock selection 00: TO4TRG 01: ϕT1 10: ϕT16 11: ϕT256		Timer 4 input clock selection 00: TI4 01: ϕT1 10: ϕT4 11: ϕT16	

Timer 4 input clock selection

00	External clock TI4
01	Internal clock ϕT1 (8/fc)
10	Internal clock ϕT4 (32/fc)
11	Internal clock ϕT16 (128/fc)

Timer 5 input clock selection

	<T45M1, 0> ≠ 01	<T45M1, 0> = 01
00	Timer 4 comparator output	Timer 4 overflow output (16-bit interval timer mode)
01	Internal clock ϕT1 (8/fc)	
10	Internal clock ϕT16 (128/fc)	
11	Internal clock ϕT256 (2048/fc)	

PWM4 cycle selection
(Except for PWM mode (<T45M1, 0> = 11), don't care)

00	Don't care
01	$(2^6 - 1) \times$ timer 4 input clock cycle
10	$(2^7 - 1) \times$ timer 4 input clock cycle
11	$(2^8 - 1) \times$ timer 4 input clock cycle

Timers 4, 5 operating mode settings

00	8-bit interval timer × 2 channels (timers 4, 5)
01	16-bit interval timer
10	8-bit PPG output
11	8-bit PWM output (timer 4) + 8-bit interval timer (timer 5)

Figure 3.7 (2)-5 8-Bit Timer Related Register

Timers 6, 7 Mode Control Register

T67MOD
(002FH)

	7	6	5	4	3	2	1	0
bit Symbol	T67M1	T67M0	PWM61	PWM60	T7CLK1	T7CLK0	T6CLK1	T6CLK0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Set operating mode for timers 6, 7 00: 8-bit interval timer 01: 16-bit interval timer 10: 8-bit PPG 11: 8-bit PWM		PWM6 cycle selection 00: Don't care 01: $2^6 - 1$ 10: $2^7 - 1$ 11: $2^8 - 1$		Timer 7 input clock selection 00: TO6TRG 01: ϕT1 10: ϕT16 11: ϕT256		Timer 6 input clock selection 00: Don't Care 01: ϕT1 10: ϕT4 11: ϕT16	

Timer 6 input clock selection

00	Don't care
01	Internal clock ϕT1 (8/fc)
10	Internal clock ϕT4 (32/fc)
11	Internal clock ϕT16 (128/fc)

Timer 7 input clock selection

	<T67M1, 0> ≠ 01	<T67M1, 0> = 01
00	Timer 6 comparator output	Timer 6 overflow output (16-bit interval timer mode)
01	Internal clock ϕT1 (8/fc)	
10	Internal clock ϕT16 (128/fc)	
11	Internal clock ϕT256 (2048/fc)	

PWM6 cycle selection
(Except for PWM mode (<T67M1, 0> = 11), don't care)

00	Don't care
01	$(2^6 - 1) \times$ timer 6 input clock cycle
10	$(2^7 - 1) \times$ timer 6 input clock cycle
11	$(2^8 - 1) \times$ timer 6 input clock cycle

Timers 6, 7 operating mode settings

00	8-bit interval timer × 2 channels (timers 6, 7)
01	16-bit interval timer
10	8-bit PPG output
11	8-bit PWM output (timer 6) + 8-bit interval timer (timer 7)

Figure 3.7 (2)-6 8-Bit Timer Related Register

Timers 0, 2 Flip-Flop Control Register

T02FFCR
(0025H)

		TFF3				TFF1		
	7	6	5	4	3	2	1	0
bit Symbol	FF3C1	FF3C0	FF3IE	FF3IS	FF1C1	FF1C0	FF1IE	FF1IS
Read/Write	W		R/W		W		R/W	
After reset	1	1	0	0	1	1	0	0
Function	TFF3 control 00: Invert TFF3 01: Set TFF3 10: Clear TFF3 11: Don't care		TFF3 inversion control 0: Disable 1: Enable	TFF3 inversion signal selection 0: Inversion by timer 2 1: Inversion by timer 3	TFF1 control 00: Invert TFF1 01: Set TFF1 10: Clear TFF1 11: Don't care		TFF1 inversion control 0: Disable 1: Enable	TFF1 inversion signal selection 0: Inversion by timer 0 1: Inversion by timer 1

Timer flip-flop TFF1 inversion signal selection
(Except for 8-bit interval timer mode, don't care)

0	Inversion by timer 0
1	Inversion by timer 1

Timer flip-flop TFF1 inversion control

0	Disable
1	Enable

Timer flip-flop TFF1 control

00	Invert TFF1 value (software inversion)
01	Set TFF1 to 1
10	Clear TFF1 to 0
11	Don't care (11 when read)

Timer flip-flop TFF3 inversion signal selection
(Except for 8-bit interval timer mode, don't care)

0	Inversion by timer 2
1	Inversion by timer 3

Timer flip-flop TFF3 inversion control

0	Disable
1	Enable

Timer flip-flop TFF3 control

00	Invert TFF3 value (software inversion)
01	Set TFF3 to 1
10	Clear TFF3 to 0
11	Don't care (11 when read)

Figure 3.7 (2)-7 8-Bit Timer Related Register

Timers 4, 6 Flip-Flop Control Register

T46FFCR
(002CH)

	TFF7				TFF5			
	7	6	5	4	3	2	1	0
bit Symbol	FF7C1	FF7C0	FF7IE	FF7IS	FF5C1	FF5C0	FF5IE	FF5IS
Read/Write	W		R/W		W		R/W	
After reset	1	1	0	0	1	1	0	0
Function	TFF7 control 00: Invert TFF7 01: Set TFF7 10: Clear TFF7 11: Don't care		TFF7 inversion control 0: Disable 1: Enable	TFF7 inversion signal selection 0: Inversion by timer 6 1: Inversion by timer 7	TFF5 control 00: Invert TFF5 01: Set TFF5 10: Clear TFF5 11: Don't care		TFF5 inversion control 0: Disable 1: Enable	TFF5 inversion signal selection 0: Inversion by timer 4 1: Inversion by timer 5

Timer flip-flop TFF5 inversion signal selection
(Except for 8-bit interval timer mode, don't care)

0	Inversion by timer 4
1	Inversion by timer 5

Timer flip-flop TFF5 inversion control

0	Disable
1	Enable

Timer flip-flop TFF5 control

00	Invert TFF5 value (software inversion)
01	Set TFF5 to 1
10	Clear TFF5 to 0
11	Don't care (11 when read)

Timer flip-flop TFF7 inversion signal selection
(Except for 8-bit interval timer mode, don't care)

0	Inversion by timer 6
1	Inversion by timer 7

Timer flip-flop TFF7 inversion control

0	Disable
1	Enable

Timer flip-flop TFF7 control

00	Invert TFF7 value (software inversion)
01	Set TFF7 to 1
10	Clear TFF7 to 0
11	Don't care (11 when read)

Figure 3.7 (2)-8 8-Bit Timer Related Register

3.7.2 Block Structure

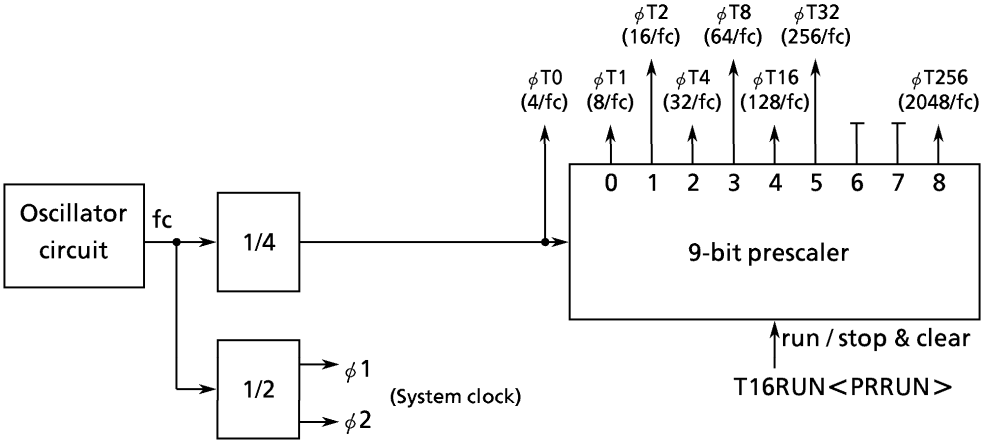
(1) Prescaler

The prescaler is a 9-bit divider circuit that divides its supplied clock (4/fc) by 2^n (n = 1, ..., 6, 9). The clock supplied to the prescaler is the CPU clock (fc) divided by four (4/fc). The divided clock is used as the input clock for such functions as the 8-bit timers, 16-bit timer/event counters, and baud rate generator.

The prescaler count can be turned on and off using timer operation control register T16RUN<PRRUN>. Setting T16RUN<PRRUN> to 1 starts the count.

Setting 0 clears the divided clock to zero and stops the prescaler. A reset clears <PRRUN> to 0, clearing and stopping the prescaler.

fc = 25 MHz Input clock	Cycle
φT1 (8/fc)	320 ns
φT4 (32/fc)	1.28 μs
φT16 (128/fc)	5.12 μs
φT256 (2048/fc)	81.92 μs



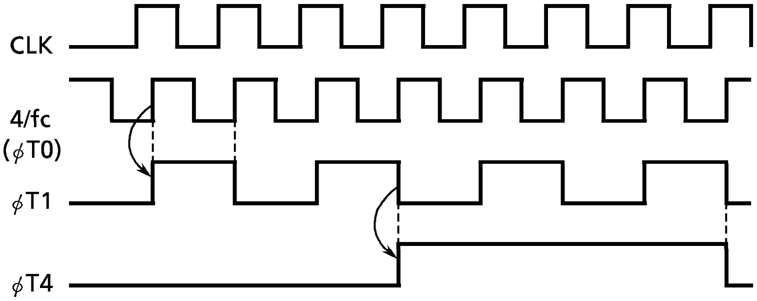


Figure 3.7 (3) Prescaler

(2) 8-bit Up-Counters

The 8-bit up-counters UC0 to 7 are the 8-bit binary counters for timers 0 to 7. The up-counters count up on the internal or external clock selected by 8-bit timer mode control registers T01MOD, T23MOD, T45MOD, and T67MOD. The 8-bit timer operation control register T8RUN settings control the up-counter operation.

The available input clocks for UC0, 2, 4, 6 are the internal clocks ϕT1, ϕT4, or ϕ16. UC0 and 4 can use the external clocks input from the timer input pin (TI0 and TI4) signals.

The input clocks for UC1, 3, 5, 7 vary according to the operating mode.

In 16-bit timer mode, the overflow output signals of timer 0, 2, 4, 6 are used as the input clocks.

In other than 16-bit timer mode, the available input clocks are internal clocks ϕT1, ϕT16, ϕT256 or TOxTRG (timer 0, 2, 4, 6 match detect signals).

A reset clears T8RUN and stops UC0 to 7.

(3) 8-bit Timer Registers

The 8-bit timer registers are 8-bit registers for setting count values.

The comparator outputs a match detect signal when the value set in 8-bit timer register TREG0 to 7 matches the 8-bit up-counter UC0 to 7 value. If 00H is set, the match detect signal is output when the 8-bit up-counter overflows.

8-bit timer registers TREG0, 2, 4, 6 have a double-buffer configuration (each has a dedicated register buffer).

Timer register double-buffer control registers TRDC<TR0/2/4/6DE> enable or disable the double buffer. Setting <TR0/2/4/6DE> to 0 disables the double-buffer; setting <TR0/2/4/6DE> to 1 enables the double buffer.

When the double buffer is enabled, data are transferred from the register buffer to the timer register at a $2^n - 1$ overflow in pulse width modulation (PWM) mode, or at a cycle compare match in programmable pulse generation (PPG) mode.

Always disable the double buffer in 8-bit and 16-bit interval timer modes.

A reset clears TRDC to 0 and disables the double buffer. When using the double buffer, first write data to TREG0, 2, 4, 6 and set TRDC<TR0/2/4/6DE> to 1, then write the next settings.

As TREG0 to 7 are undefined after a reset, set the registers before using the 8-bit timers.

Figure 3.7 (4) shows the configuration of timer registers 0, 2, 4, 6.

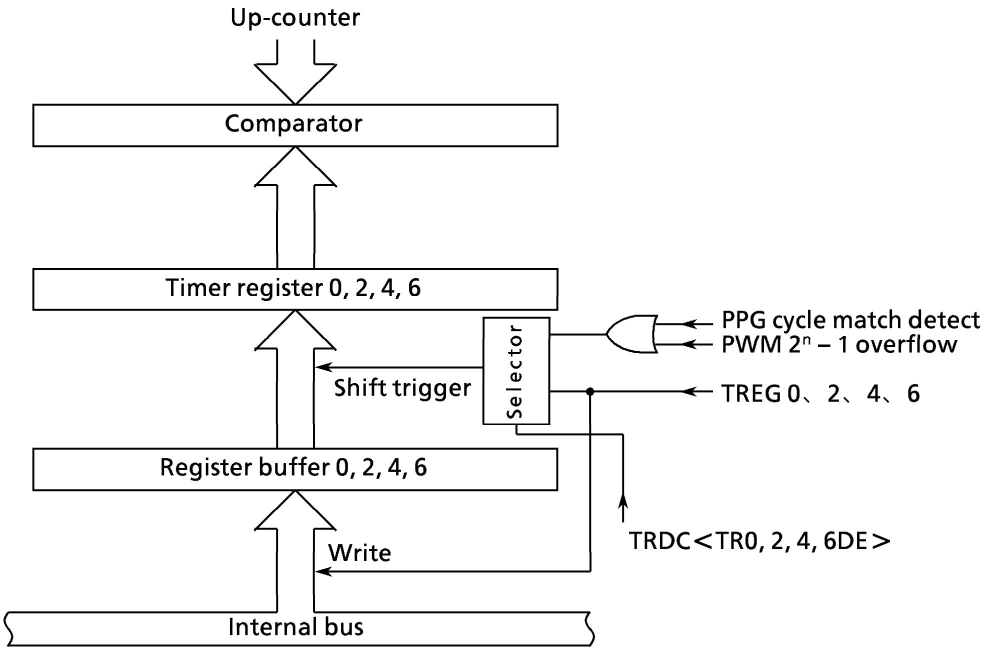


Figure 3.7 (4) Configuration of Timer Registers 0, 2, 4, 6

Note: The timer register and register buffer are allocated to the same address in memory. When TRDC<TR0/2/4/6DE> is set to 0, the same value is written to both the register buffer and the timer register. When TRDC<TR0/2/4/6DE> is set to 1, the value is written to the register buffer only. Accordingly, when writing the initial values to the timer registers, first disable the register buffers.

The timer registers are located in memory as follows.

TREG0	TREG1
8 bits	8 bits
000022H	000023H

TREG2	TREG3
8 bits	8 bits
000026H	000027H

TREG4	TREG5
8 bits	8 bits
000029H	00002AH

TREG6	TREG7
8 bits	8 bits
00002DH	00002EH

All registers are write-only and therefore cannot be read.

(4) 8-bit Comparator

The 8-bit comparator compares the 8-bit up-counter value with the 8-bit timer register value and detects when the values are equal (match). If the values match, a match detect signal is output, the 8-bit up-counter is cleared to zero, and an interrupt is generated (INTT0 to 7).

(5) Timer Flip-Flops

The timer flip-flops (TFF1, TFF3, TFF5, TFF7) are inverted by a match detect signal from the 8-bit comparator.
Timer flip-flop control registers T02FFCR<FF3IE>, <FF1IE>, and T46FFCR <FF7IE>, <FF5IE> enable or disable inversion. Setting these bits to 0 disables inversion; setting to 1 enables inversion.
The timer flip-flop values after a reset are undefined. Writing 01 or 10 to T02FFCR<FF3C1, 0>, <FF1C1, 0>, or T46FFCR<FF7C1, 0>, <FF5C1, 0> sets the timer flip-flop to 0 or 1. Writing 00 to the bits inverts the timer flip-flop value (software inversion).
The TFF1, TFF3, TFF5, and TFF7 values can be output to the timer output pins TO1 (shared with P71), TO3 (shared with P72), TO5 (shared with P74), and TO7 (shared with P75) respectively.
As the timer output pins also function as P71, P72, P74, and P75, be sure to set the port 7 function register (P7FC) before performing timer output.
(See Figure 3.5 (24) Port 7 Registers)

3.7.3 Operation Description for Each Mode

(1) 8-bit Interval Timer Mode

The eight interval timers 0 to 7 can be used independently. When setting the functions and count data, first stop timers 0 to 7.
The following describes the example of timer 1 only.

① Generate interrupts at fixed intervals

Use T01MOD to select the operating mode and input clock. Set the interval time (cycle) in TREG1. Enable interrupt INTT1 such that INTT1 is generated when a match occurs between UC1 and TREG1. After setting the registers, start the timer counting.
Table 3.7 (1) shows the input clock selection.

Example: To generate a timer 1 interrupt every 32 μs (at fc = 25 MHz), set the registers in the following order:

```
              MSB        LSB
               7 6 5 4 3 2 1 0
 ┌T8RUN    ←   - - - - - - 0 -     Stop timer 1 and clear to zero.
 │T01MOD   ←   0 0 X X 0 1 - -     Set 8-bit interval timer mode and set input clock to ϕT1
 │                                 (0.32 μs @fc = 25 MHz).
 │TREG1    ←   0 1 1 0 0 1 0 0     Set 32 μs ÷ ϕT1 = 100 (64H) in timer register.
 │INTET01  ←   1 1 0 1 - - - -     Enable INTT1 and set interrupt level to 5.
 │T16RUN   ←   1 X - - X X X X
 └T8RUN    ←   - - - - - - 1 -     Start timer 1 counting.
```

Note: X: Don't care −: No change

Table 3.7 (1) Selecting Interval and Input Clock for 8-Bit Timer Interrupt

Input Clock	Interrupt Interval (@fc = 25 MHz)	Resolution
ϕT1 (8/fc)	0.32 μs to 81.92 μs	0.32 μs
ϕT4 (32/fc)	1.28 μs to 327.7 μs	1.28 μs
ϕT16 (128/fc)	5.12 μs to 1.311 ms	5.12 μs
ϕT256 (2048/fc)	81.92 μs to 20.97 ms	81.92 μs

② Generate square wave with 50%-duty cycle

To output a square wave with a duty cycle of 50%, set a count value equivalent to half the desired cycle and TFF1 to invert on a match detect signal from timer 1 (T02FFCR<FF1IE, FF1IS> = 11).
Also, set P71 as a timer output (P7CR<P71C> = 1, P7FC<P71F> = 1)

Example: To output a square wave from pin TO1 with an interval of 1.92 μs (at fc = 25 MHz), set the registers in the following order:

		MSB LSB	
		7 6 5 4 3 2 1 0	
T8RUN	←	– X – – – – 0 –	Stop timer 1 and clear to zero.
T01MOD	←	0 0 X X 0 1 – –	Set 8-bit interval timer mode and set input clock to φT1.
TREG1	←	0 0 0 0 0 0 1 1	Set 1.92 μs ÷ φT1 (0.32 μs) ÷ 2 = 3 in timer register.
T02FFCR	←	– – – – 1 0 1 1	Clear TFF1 to 0 and set to invert on match detect signal from timer 1.
P7CR	←	X X – – – – 1 –	Set P71 as TO1 pin.
P7FC	←	X X – – X – 1 X	
T16RUN	←	1 X – – X X X X	
T8RUN	←	– – – – – – 1 –	Start timer 1 counting.

Note: X: Don't care –: No change

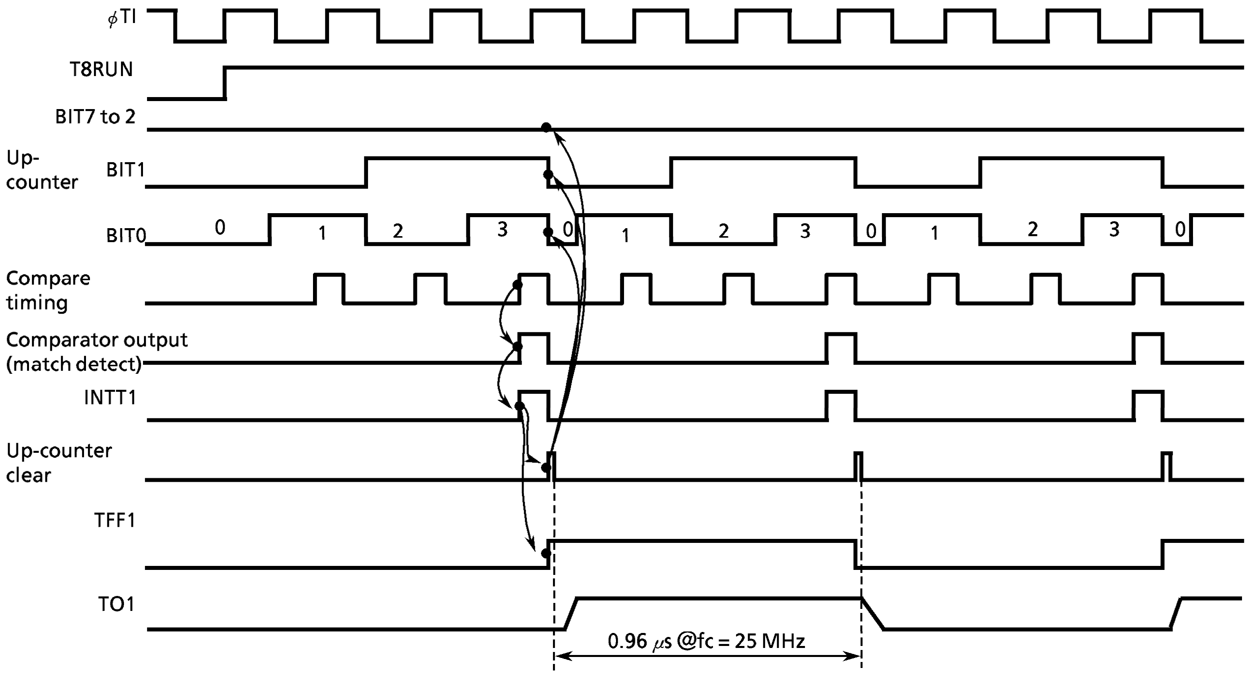


Figure 3.7 (5) Square Wave (50% Duty Cycle) Output Timing Chart

③ To count up at each timer 0 match output, set timer 1

Set 8-bit timer mode and the timer 0 comparator output as the timer 1 input clock (T01MOD<T1CLK1, 0> =00).

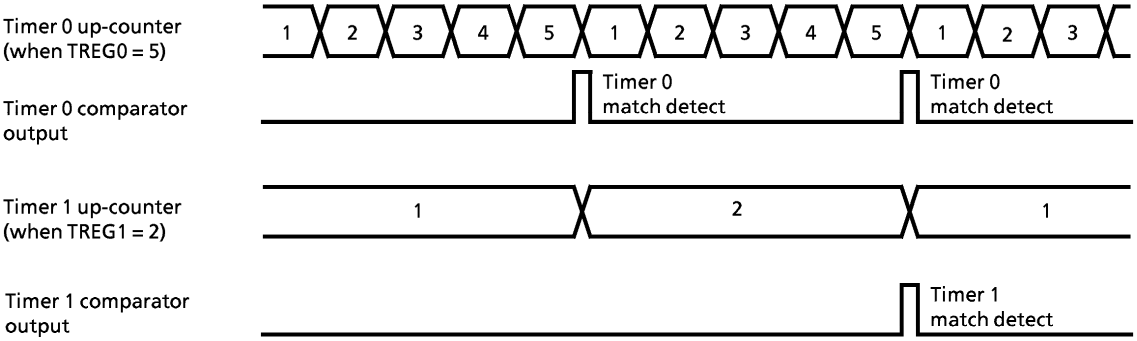


Figure 3.7 (6) Using Timer 0 to Drive Timer 1 Count

(2) 16-Bit Interval Timer Mode

The 8-bit timers can be cascaded in pairs (timers 0 and 1, 2 and 3, 4 and 5, 6 and 7) to create 16-bit interval timers.
Timers 0 and 1, 2 and 3, 4 and 5, 6 and 7 operate the same. Each pair can be used independently.
The following describes the example of timers 0 and 1.
To cascade timers 0 and 1 to form a 16-bit interval timer, set the timer 0, 1 mode control register T01MOD<T01M1, 0> to 01.
When 16-bit interval timer mode is set, the T01MOD<T1CLK1, 0> setting is ignored and the timer 0 overflow output is forcibly set as the timer 1 input clock.
Figure 3.7 (2) shows the relationship between the timer (interrupt) interval and the input clock selection.

Table 3.7 (2) 16-Bit Timer (Interrupt) Interval and Input Clock Selection

Input Clock	Interrupt Interval (fc = 25 MHz)	Resolution
φT1 (8/fc)	0.32 μs to 20.971 ms	0.32 μs
φT4 (32/fc)	1.28 μs to 83.885 ms	1.28 μs
φT16 (128/fc)	5.12 μs to 335.539 ms	5.12 μs

To set the timer interrupt interval, set the lower eight bits in timer register TREG0 and the upper eight bits in TREG1. Be sure to set TREG0 first (as entering data in TREG0 temporarily disables compare, while entering data in TREG1 starts compare).

Example: To generate interrupt INTT1 every 0.32s at fc = 25 MHz, set the following values in timer registers TREG0 and TREG1:
Using ϕT16 (= 5.12 μs @ 25 MHz) as a timer input clock
0.32 s ÷ 5.12 μs = 62500 = F424H
Therefore, set TREG1 to F4H, and TREG0 to 24H.

Whenever 8-bit up-counter UC0 and TREG0 match, the timer 0 comparator outputs a match detect signal, but up-counter UC0 is not cleared. No INTT0 interrupt is generated.

When up-counter UC1 and TREG1 match, at comparator timing the timer 1 comparator outputs a match detect signal.
When comparator match detect signals for both timer 0 and timer 1 are output at the same time, up-counter 0 and up-counter 1 are cleared to 0 and interrupt INTT1 only is generated. When the timer flip-flop inversion is enabled, the value of timer flip-flop TFF1 is inverted.

Table 3.7 (3) Differences Between 16-Bit Timer Mode and 8-Bit Timer Mode (Timer 1 Input Clock: TO0TRG)

	Timer 0			Timer 1		
	INTT0 interrupt	TO1 output	Counter operation when match detected	INTT1 interrupt	TO1 output	Counter operation when match detected
16-bit timer mode (count-up timer 1 on each timer 0 overflow)	No interrupt generated	Output disabled (of a signal indicating a match with TREG0 is disabled)	TREG0 (count-up even when a match occurs. Clear at match with TREG1)	Interrupt generated	Output enabled (can output a match signal for both timers 0 and 1)	TREG1 × 2^8 + TREG0 : Full 16 bits (clear at match)
8-bit timer mode (count up timer 1 on each timer 0 match)	Interrupt generated	Output enabled (either from timer 0 or timer 1)	TREG0 (clear at match)	Interrupt generated	Output enabled (either from timer 0 or timer 1)	TREG1 × TREG0 : Multiplication value (clear at match)

Example: When TREG1 = 04H and TREG0 = 80H:

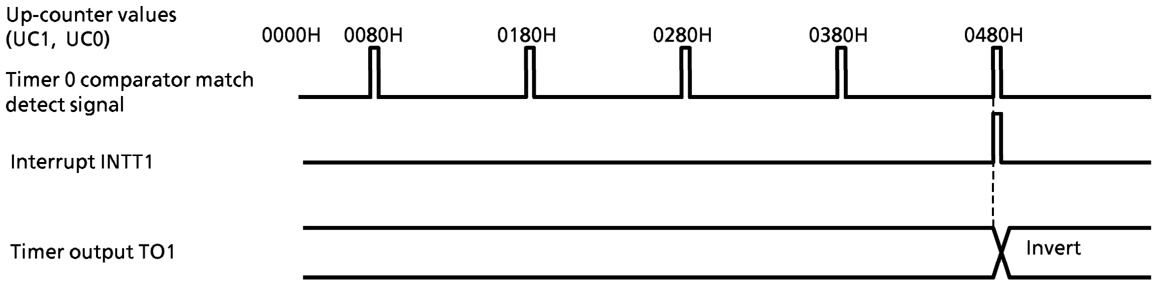


Figure 3.7 (7) Timer Output for 16-Bit Timer Mode

(3) 8-Bit Programmable Pulse Generation (PPG) Output Mode

Timers 0, 2, 4, or 6 can output square waves with variable frequencies and variable duty (programmable pulse generation). The output pulse can be set to either active low or active high. Timers 1, 3, 5, and 7 cannot be used in this mode.

Timer 0 outputs from pin TO1 (shared with pin P71), timer 2 outputs from pin TO3 (shared with pin P72), timer 4 outputs from TO5 (shared with pin P74), and timer 6 outputs from TO7 (shared with pin P75).

The following describes the example of timer 0. (Timers 2, 4, 6 operate the same.)

A programmable square wave can be output from pin TO1 by setting 8-bit programmable square wave output mode and enabling inversion of the timer flip-flop TFF1.

The TFF1 value is inverted by a match between 8-bit up-counter UC0 and TREG0, and by a match with TREG1. UC0 is cleared by a match with TREG1.

In PPG mode, timer 1 cannot be used, but timer 1 up-counter UC1 must be run (T8RUN<T1RUN>=1).

Also, the TREG0 and TREG1 settings in PPG mode must satisfy the following condition.

(TREG0 setting value) < (TREG1 setting value)

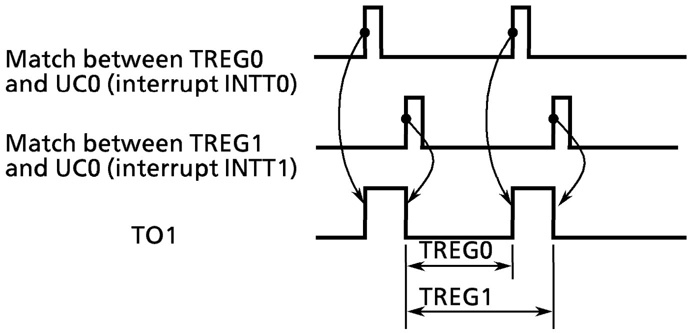


Figure 3.7 (8) 8-Bit PPG Output Waveform

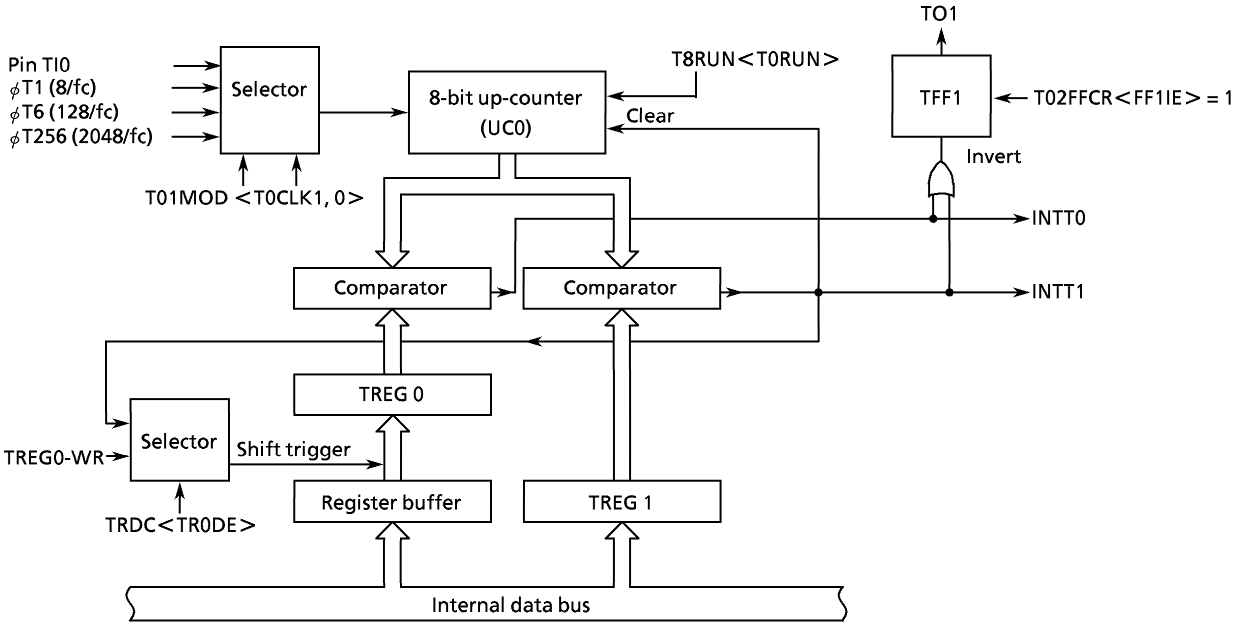


Figure 3.7 (9) Block Diagram of 8-Bit PPG Output Mode

Enabling the timer register TREG0 double buffer in this mode shifts the register buffer value to TREG0 when timer register TREG1 matches 8-bit up-counter UC0.
Using the double buffer facilitates handling of small duty waves (when changing the duty).

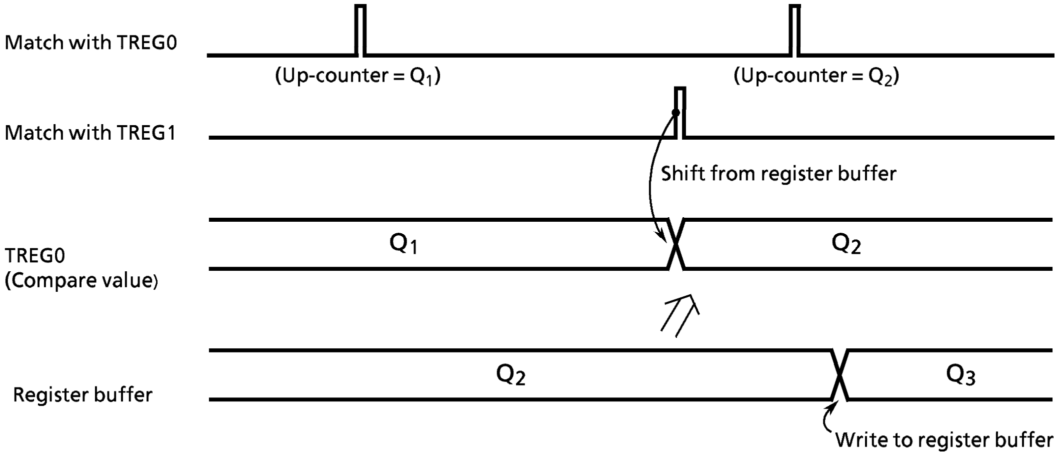
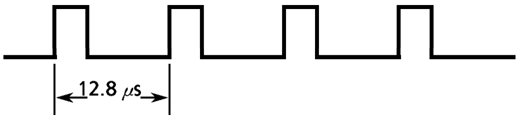


Figure 3.7 (10) Register Buffer Operation

Example: Output 1/4-duty 78.125 kHz-pulse (@fc = 25 MHz)

Calculate the setting of the timer register.
Setting the frequency to 78.125 kHz creates a square wave with a cycle of t = 1/78.125 kHz = 12.8 μs.



Using ϕT1 = 0.32 μs (@fc = 25 MHz) results in
12.8 μs ÷ 0.32 μs = 40
Accordingly, set TREG1 = 40 = 28H.
Next, set the duty to 1/4 as follows:
t × 1/4 = 12.8 μs × 1/4 = 3.2 μs
As with TREG1,
3.2 μs ÷ 0.32 μs = 10
Accordingly, set TREG0 = 10 = 0AH.

```
              MSB           LSB
              7 6 5 4 3 2 1 0
 T8RUN   ←    - - - - - - 0 0     Stop timers 0 and 1, and clear to 0.
 T16RUN  ←    0 X - - X X X X
 T01MOD  ←    1 0 X X 0 1 0 1     Set 8-bit PPG mode and set input clock to φT1.

 T02FFCR ←    - - - - 0 1 1 x     Set TFF1 and enable inversion.
                      └─┘
                       └────────→ Setting to 10 obtains negative logic output wave.
 TREG0   ←    0 0 0 0 1 0 1 0     Write 0AH.
 TREG1   ←    0 0 1 0 1 0 0 0     Write 28H.
 P7CR    ←    X X - - - - 1 -     }
 P7FC    ←    X X - - X - 1 X     } Set P71 to TO1 pin.
 T16RUN  ←    1 X - - X X X X
 T8RUN   ←    - - - - - - 1 1     Start timers 0 and 1 counting.
```

Note: X: Don't care -: No change

(4) 8-Bit Pulse Width Modulation (PWM) Output Mode (PWM: Pulse Width Modulation)

Only timers 0, 2, 4, 6 can be set to this mode, which allows up to four pulse width modulation outputs with 8-bit resolution. Timers 1, 3, 5, and 7 can be used as 8-bit timers.
In the case of timer 0, PWM is output to pin TO1 (shared with P71). In the case of timers 2, 4, 6, PWM is output to pins TO3 (shared with P72), TO5 (shared with P74), and TO7 (shared with P75) respectively.
Here, the example of timer 0 is used. (Timers 2, 4, 6 operate the same as timer 0.)
Timer output inversion occurs when the 8-bit up-counter UC0 setting and the timer register TREG0 setting match, or when $2^n - 1$ (T01MOD specifies one of n = 6, n = 7, or n = 8) counter overflow occurs. UC0 is cleared by the $2^n - 1$ counter overflow.
In addition, the following conditions must be satisfied when using 8-bit PWM output mode:

(Timer register setting) < ($2^n - 1$ counter overflow setting)
(Timer register setting) ≠ 0

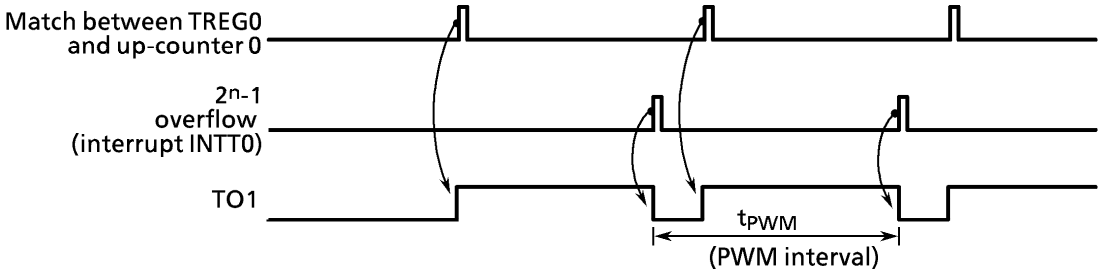

Match between TREG0 and up-counter 0
2n-1 overflow (interrupt INTT0)
TO1
tPWM
(PWM interval)

Figure 3.7 (11) 8-Bit PWM Output Waveform

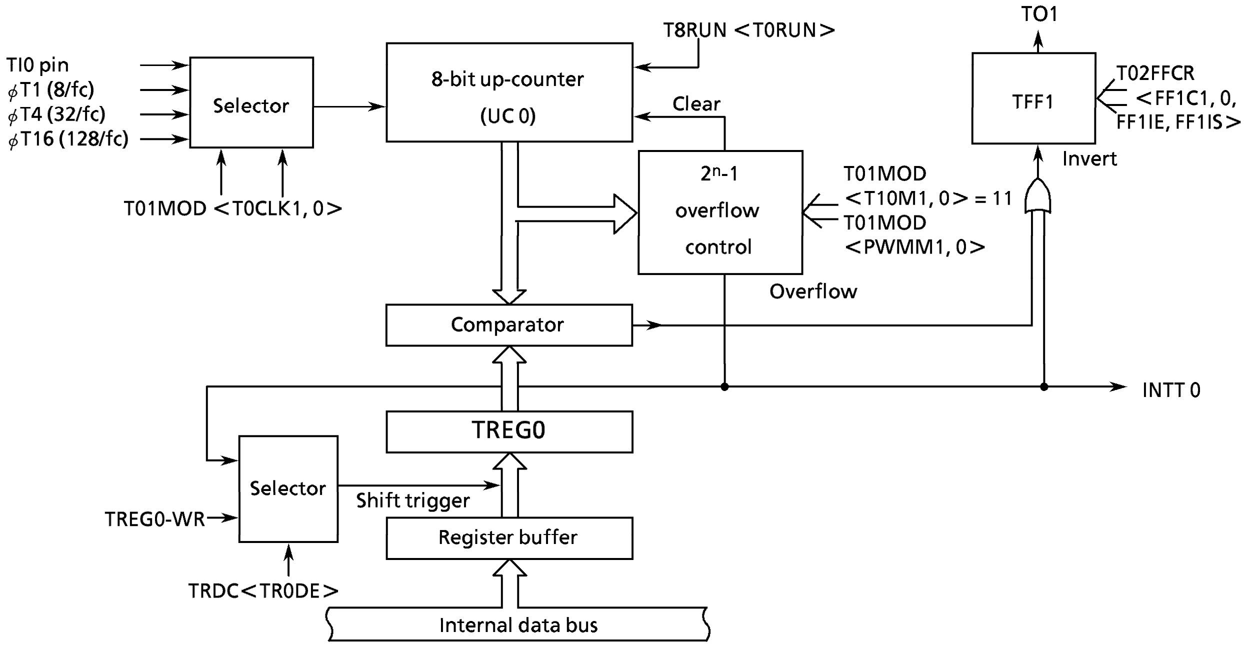


Figure 3.7 (12) Block Diagram of 8-Bit PWM Output Mode

Enabling the TREG0 double-buffer in this mode shifts the register buffer value to TREG0 when $2^n - 1$ counter overflow is detected.
Using the double buffer facilitates handling of small duty waves.

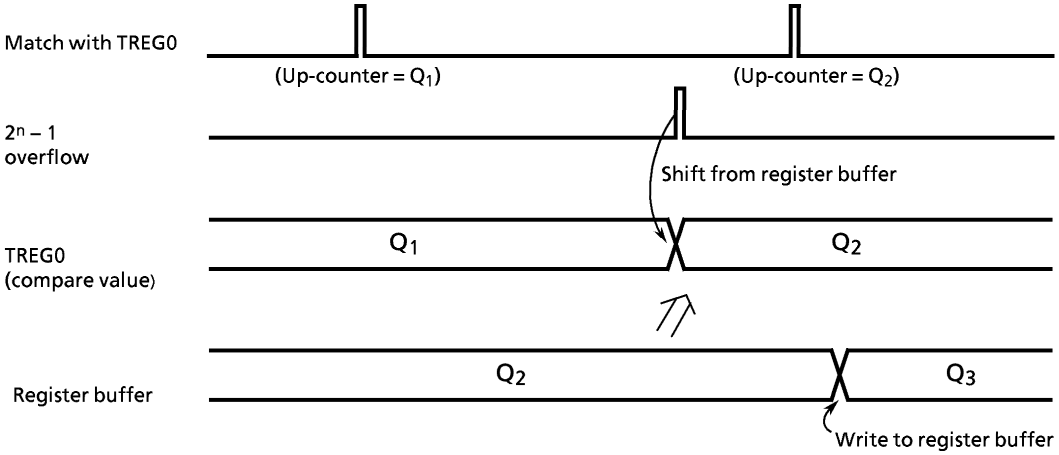
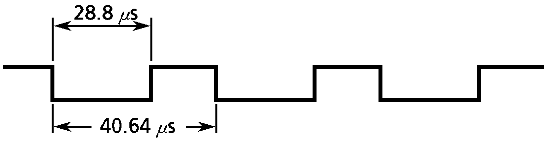


Figure 3.7 (13) Register Buffer Operation

Example: Output following PWM waveform to pin TO1 (@fc = 25 MHz)



To realize a PWM interval of 40.64 μs using ϕT1 = 0.32 μs (@fc = 25 MHz):

$$40.64\ \mu s \div 0.32\ \mu s = 127 = 2^{n} - 1$$

Accordingly, set n = 7.

As the low level cycle is 28.8 μs, at ϕT1 = 0.32 μs,

$$28.8\ \mu s \div 0.32\ \mu s = 90$$

Accordingly, set TREG0 = 90 = 5AH.

```
                MSB              LSB
                 7  6  5  4  3  2  1  0
┌ T8RUN     ←   -  -  -  -  -  -  -  0     Stop timer 0 and clear to 0.
│ T01MOD    ←   1  1  1  0  -  -  0  1     Set 8-bit PWM mode (interval = 2^7 -1) and set input
│                                          clock to φT1.
│ T02FFCR   ←   -  -  -  -  1  0  1  X     Clear TFF1 and enable inversion.
│ TREG0     ←   0  1  0  1  1  0  1  0     Write 5AH.
│ P7CR      ←   X  X  -  -  -  -  1  -     }
│ P7FC      ←   X  X  -  -  X  -  1  X     }  Set P71 to pin TO1.
│ T16RUN    ←   1  X  -  -  X  X  X  X
└ T8RUN     ←   -  -  -  -  -  -  -  1     Start timer 0 counting.
  Note: X: Don't care    -: No change
```

Table 3.7 (4) shows the timer input clock source and the PWM interval determined by the $(2^{n} - 1)$ counter.

Table 3.7 (4) Setting of PWM Interval (@fc = 25 MHz)

Input Clock / $(2^{n} - 1)$ Counter	ϕT1	ϕT4	ϕT16
$2^{6} - 1$	20.2 μs (49.6 kHz)	80.6 μs (12.4 kHz)	322.6 μs (3.1 kHz)
$2^{7} - 1$	40.6 μs (24.6 kHz)	162.6 μs (6.2 kHz)	650.2 μs (1.5 kHz)
$2^{8} - 1$	81.6 μs (12.3 kHz)	326.4 μs (3.1 kHz)	1.31 ms (0.8 kHz)

(5) Timer Mode List

The 8-bit timers 0 to 7 can be set to 8-bit timer mode, 16-bit timer mode, 8-bit PPG mode, or 8-bit PWM mode. Table 3.7 (5) lists settings for the timer modes.

Table 3.7 (5) Settings for All Timer Modes

Register Name	TxxMOD				TxxFFCR
bit Symbol Timer mode (for 8-bit timer channels × 2)	Timer mode <T01M1, 0> <T23M1, 0> <T45M1, 0> <T67M1, 0>	PWM interval <PWM01, 00> <PWM21, 20> <PWM41, 40> <PWM61, 60>	Upper timer input clock <T1CLK1, 0> <T3CLK1, 0> <T5CLK1, 0> <T7CLK1, 0>	Lower timer input clock <T0CLK1, 0> <T2CLK1, 0>(note) <T4CLK1, 0> <T6CLK1, 0>(note)	Inversion select <FF1IS> <FF3IS> <FF5IS> <FF7IS>
16-bit timer (full 16 bits) × 1ch	01	–	–	00: External input 01: φT1 10: φT4 11: φT16	–
8-bit timer (8-bit × 8-bit mode) × 1ch (Inputs lower timer comparator output to upper timer)	00	–	00	00: External input 01: φT1 10: φT4 11: φT16	0: Lower timer 1: Upper timer
8-bit timer × 2ch	00	–	00: Don't care 01: φT1 10: φT16 11: φT256	00: External input 01: φT1 10: φT4 11: φT16	0: Lower timer 1: Upper timer
8-bit PPG × 1ch	10	–	–	00: External input 01: φT1 10: φT4 11: φT16	–
8-bit PWM × 1ch (lower) 8-bit timer × 1ch (upper)	11	00: Don't care 01: $2^6 - 1$ 10: $2^7 - 1$ 11: $2^8 - 1$	00: Don't care 01: φT1 10: φT16 11: φT256	00: External input 01: φT1 10: φT4 11: φT16	–

Note: External clock is not input to timer 2 or timer 6.

3.8 16-Bit Timers / Event Counters

TMP95CS64/265 incorporates two multi-function 16-bit timer/event counters (timers 8 and 9). Timers 8 and 9 have the same functions and can operate independently. The 16-bit timers have the following three operating modes.

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable pulse generation (PPG) output mode

The capture function can also be used to perform the following operations.

- One-shot pulse output from the external trigger pulse
- Frequency measurement
- Pulse width measurement
- Time differential measurement

Also, the 16-bit timers can be used to output a signal with any phase difference.
Figure 3.8 (1) is a block diagram of the 16-bit timer/event counters (timer 8). Timer 9 also has the same circuit configuration.
Each 16-bit timer / event counter consists of a 16-bit up-counter, a 16-bit comparator, a 16-bit timer register, and a 16-bit capture register. Timers 8 and 9 each have two timer flip-flops (TFF8/9 and TFFA/B).
Clock sources ϕT1, ϕT4, and ϕT16 input to the 16-bit timers are obtained from the internal 9-bit prescaler (see 3.7.2 (1), Prescaler).
The 16-bit timer/event counters are controlled by six control registers (T8MOD, T9MOD, T8FFCR, T9FFCR, T16RUN, and T89CR).

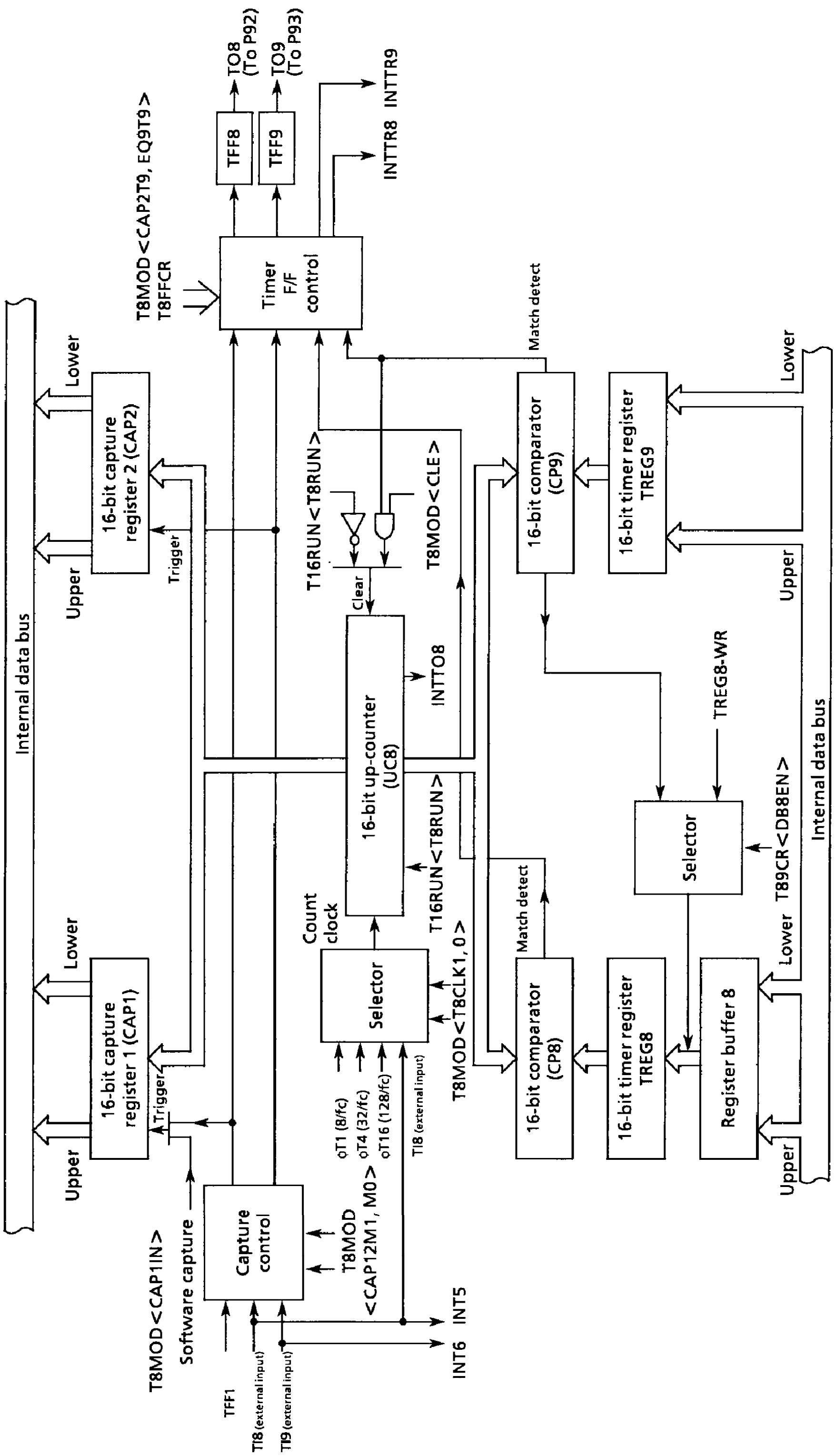


Figure 3.8 (1) 16-Bit Timer Block Diagram (Timer 8)

3.8.1 16-Bit Timer / Event Counter Registers

Figure 3.7 (2) shows the 16-bit timer/event counter related registers.
These register settings control the 16-bit timer/event counter operations.

Timer 8, 9 Control Register

T89CR
(003AH)

	7	6	5	4	3	2	1	0
bit Symbol	–					–	DBAEN	DB8EN
Read/Write	R/W					R/W		
After reset	0					0	0	0
Function	Note: Always fixed to 0					Note: Always fixed to 0	TREGA double buffer control 0: Disable 1: Enable	TREG8 double buffer control 0: Disable 1: Enable

Timer register A double buffer control

0	Disable
1	Enable

Timer register 8 double buffer control

0	Disable
1	Enable

16-bit Timer Operation Control Register

T16RUN
(003BH)

	7	6	5	4	3	2	1	0
bit Symbol	PRRUN		T9RUN	T8RUN				
Read/Write	R/W		R/W					
After reset	0		0	0				
Function	Prescaler operation control 0: Stop and clear 1: Run		Timer 9 operation control 0: Stop and clear 1: Run	Timer 8 operation control 0: Stop and clear 1: Run				

Timer 8 up-counter operation control

0	Stop and clear
1	Run

Prescaler operation control (Note)

0	Stop and clear
1	Run

Timer 9 up-counter operation control

0	Stop and clear
1	Run

Note: When running a 16-bit timer, set T16RUN<PRRUN> to 1.

Figure 3.8 (2)-1 16-Bit Timer/Event Counter Related Registers

Timer 8 Mode Control Register

T8MOD
(0038H)

<table>
<tr><td></td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>bit Symbol</td><td>CAP2T9</td><td>EQ9T9</td><td>CAP1IN</td><td>CAP12M1</td><td>CAP12M0</td><td>CLE</td><td>T8CLK1</td><td>T8CLK0</td></tr>
<tr><td>Read/Write</td><td colspan="2">R/W</td><td>W</td><td colspan="5">R/W</td></tr>
<tr><td>After reset</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td rowspan="2">Function</td><td colspan="2">TFF9 inversion trigger
0: Trigger disable
1: Trigger enable</td><td rowspan="2">Software capture control
0: Software capture
1: Don't care</td><td colspan="2" rowspan="2">Capture timing
00: Disable
INT5 at rising edge
01: TI8 ↑ TI9 ↑
INT5 on rising edge
10: TI8 ↑ TI8 ↓
INT5 on falling edge
11: TFF1 ↑ TFF1 ↓
INT5 on rising edge</td><td rowspan="2">Timer 8 up-counter control
0: Clear disable
1: Clear enable</td><td colspan="2" rowspan="2">Timer 8 source clock selection
00: TI8
01: φT1
10: φT4
11: φT16</td></tr>
<tr><td>At loading of up-counter value to CAP2</td><td>At match between up-counter and TREG9</td></tr>
</table>

Timer 8 capture timing

	Capture control	INT5 control
00	Capture disable	INT5 generated on TI8 rising edge ↑
01	CAP1 on TI8 rising edge CAP2 on TI9 rising edge	
10	CAP1 on TI8 rising edge CAP2 on TI8 falling edge	INT5 generated on TI8 falling edge ↓
11	CAP1 on TFF1 rising edge CAP2 on TFF1 falling edge	INT5 generated on TI8 rising edge ↑

Timer 8 input clock

00	External input clock (TI8)
01	φT1 (8/fc)
10	φT4 (32/fc)
11	φT16 (128/fc)

Up-counter (UC8) clear

0	Disable up-counter clear
1	Clear at match with TREG9

Software capture control
(always read as 1)

0	At loading of up-counter 8 value to CAP1
1	Don't care

At match between up-counter and TREG9
Invert trigger of timer flip-flop 9 (TFF9)

0	Trigger disable (inversion disabled)
1	Trigger enable (inversion enabled)

At loading of up-counter value to CAP2
Invert trigger of timer flip-flop 9 (TFF9)

0	Trigger disable (inversion disabled)
1	Trigger enable (inversion enabled)

Figure 3.8 (2)-2 16-Bit Timer/Event Counter Related Register

Timer 8 Flip-Flop Control Register

		7	6	5	4	3	2	1	0
T8FFCR (0039H)	bit Symbol	TFF9C1	TFF9C0	CAP2T8	CAP1T8	EQ9T8	EQ8T8	TFF8C1	TFF8C0
	Read/Write	W		R/W				W	
	After reset	1	1	0	0	0	0	1	1
	Function	TFF9 control 00: Invert TFF9 01: Set TFF9 10: Clear TFF9 11: Don't care		TFF8 inversion trigger 0: Trigger disable 1: Trigger enable				TFF8 control 00: Invert TFF8 01: Set TFF8 10: Clear TFF8 11: Don't care	
				At loading of up-counter value to CAP2	At loading of up-counter value to CAP1	At match between up-counter and TREG9	At match between up-counter and TREG8		

Timer flip-flop 8 (TFF8) control

00	Invert TFF8 value (software inversion)
01	Set TFF8 to 1
10	Clear TFF8 to 0
11	Don't care (always read as 11)

At match between up-counter and TREG8
Invert trigger of timer flip-flop 8 (TFF8)

0	Trigger disable (inversion disabled)
1	Trigger enable (inversion enabled)

At match between up-counter and TREG9
Invert trigger of timer flip-flop 8 (TFF8)

0	Trigger disable (inversion disabled)
1	Trigger enable (inversion enabled)

At loading of up-counter value to CAP1
Invert trigger of timer flip-flop 8 (TFF8)

0	Trigger disable (inversion disabled)
1	Trigger enable (inversion enabled)

At loading of up-counter value to CAP2
Invert trigger of timer flip-flop 8 (TFF8)

0	Trigger disable (inversion disabled)
1	Trigger enable (inversion enabled)

Timer flip-flop 9 (TFF9) control

00	Invert TFF9 value (software inversion)
01	Set TFF9 to 1
10	Clear TFF9 to 0
11	Don't care (always read as 11)

Figure 3.8 (2)-3 16-Bit Timer/Event Counter Related Register

Timer 9 Mode Control Register

T9MOD
(0048H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP4TB	EQBTB	CAP3IN	CAP34M1	CAP34M0	CLE	T9CLK1	T9CLK0
Read/Write	R/W		W	R/W				
After reset	0	0	1	0	0	0	0	0
Function	TFFB inversion trigger 0: Trigger disable 1: Trigger enable		Software capture control 0: Software capture 1: Don't care	Capture timing 00: Disable INT7 is rising edge 01: TIA ↑ TIB ↑ INT7 on rising edge 10: TIA ↑ TIA ↓ INT7 on falling edge 11: TFF1 ↑ TFF1 ↓ INT7 on rising edge		Timer 9 up-counter control 0: Clear disable 1: Clear enable	Timer 9 source clock selection 00: TIA 01: φT1 10: φT4 11: φT16	
	At loading of up-counter value to CAP4	At match between up-counter and TREGB						

Timer 9 capture timing

	Capture control	INT7 control
00	Capture disable	INT7 generated on TIA rising edge
01	CAP3 on TIA rising edge CAP4 on TIB rising edge	
10	CAP3 on TIA rising edge CAP4 on TIA falling edge	INT7 generated on TIA falling edge
11	CAP3 on TFF1 rising edge CAP4 on TFF1 falling edge	INT7 generated on TIA rising edge

Timer 9 input clock

00	External input clock (TIA)
01	φT1 (8/fc)
10	φT4 (32/fc)
11	φT16 (128/fc)

Up-counter (UC9) clear

0	Disable up-counter clear
1	Clear at match with TREGB

Software capture control
(always read as 1)

0	At loading of up-counter 9 value to CAP3
1	Don't care

At match between up-counter and TREGB
Invert trigger of timer flip-flop B (TFFB)

0	Trigger disable (inversion disabled)
1	Trigger enable (inversion enabled)

At loading of up-counter value to CAP2
Timer flip-flop B (TFFB) inversion trigger

0	Trigger disable (inversion disabled)
1)	Trigger enable (inversion enabled)

Figure 3.8 (2)-4 16-Bit Timer/Event Counter Related Register

Timer 9 Flip-Flop Control Register

T9FFCR
(0049H)

	7	6	5	4	3	2	1	0
bit Symbol	TFFBC1	TFFBC0	CAP4TA	CAP3TA	EQBTA	EQATA	TFFAC1	TFFAC0
Read/Write	W		R/W				W	
After reset	1	1	0	0	0	0	1	1
Function	TFFB control 00: Invert TFFB 01: Set TFFB 10: Clear TFFB 11: Don't care		TFFA inversion trigger 0: Trigger disable 1: Trigger enable				TFFA control 00: Invert TFFA 01: Set TFFA 10: Clear TFFA 11: Don't care	
			At loading of up-counter value to CAP4	At loading of up-counter value to CAP3	At match between up-counter and TREG9	At match between up-counter and TREG8		

Timer flip-flop A (TFFA) control

00	Invert TFFA value (software inversion)
01	Set TFFA to 1
10	Clear TFFA to 0
11	Don't care (always read as 11)

At match between up-counter and TREGA
Timer flip-flop A (TFFA) inversion trigger

0	Trigger disable (inversion disabled)
1	Trigger enable (inversion enabled)

At match between up-counter and TREGB
Timer flip-flop A (TFFA) inversion trigger

0	Trigger disable (inversion disabled)
1	Trigger enable (inversion enabled)

At loading of up-counter value to CAP3
Timer flip-flop A (TFFA) inversion trigger

0	Trigger disable (inversion disabled)
1	Trigger enable (inversion enabled)

At loading of up-counter value to CAP4
Timer flip-flop A (TFFA) inversion trigger

0	Trigger disable (inversion disabled)
1	Trigger enable (inversion enabled)

Timer flip-flop B (TFFB) control

00	Invert TFFB value (software inversion)
01	Set TFFB to 1
10	Clear TFFB to 0
11	Don't care (always read as 11)

Figure 3.8 (2)-5 16-Bit Timer/Event Counter Related Register

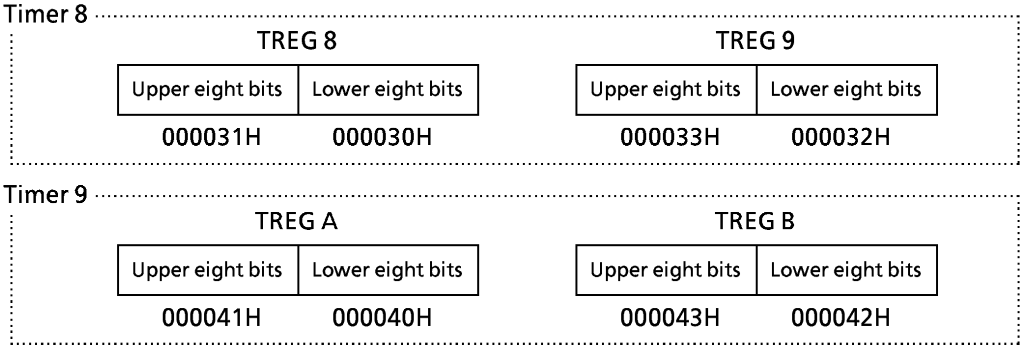
3.8.2 Block Structure

(1) 16-bit Up-Counters

16-bit up-counters UC8 and 9 are 16-bit binary counters for timers 8 and 9.
These up-counters count up on the external and internal clocks selected by 16-bit timer mode control registers T8MOD and T9MOD. To control the up-counter operations, use 16-bit timer operation control register T16RUN.
The UC8, 9 input clock is selected from either internal clocks ϕT1, ϕT4, and ϕT16, or the external clocks input from the timer input pin (TI8 and TI9).
Any overflow from UC8 or 9 triggers interrupt request INTTO8 or INTTO9.
At a reset, T16RUN is cleared, and the prescaler and UC8, 9 are stopped.

(2) 16-Bit Timer Registers

Each timer has two internal 16-bit timer registers for setting counters. A match between these timer register settings and the value of the 16-bit up-counter UC8, 9 outputs a comparator match detect signal.
Data set to 16-bit timer registers TREG8, TREG9 and TREGA, TREGB use a 2-byte data transfer instruction, or two 1-byte data transfer instructions; first for the lower eight bits, then for the upper eight bits.



Timer 8

TREG 8		TREG 9	
Upper eight bits	Lower eight bits	Upper eight bits	Lower eight bits
000031H	000030H	000033H	000032H

Timer 9

TREG A		TREG B	
Upper eight bits	Lower eight bits	Upper eight bits	Lower eight bits
000041H	000040H	000043H	000042H

TREG8 to TREGB are write-only registers and therefore cannot be read.

Of the 16-bit timer registers, TREG8 and TREGA have a double-buffer configuration (each has a register buffer).
Timer 8, 9 control register T89CR<DB8EN, DBAEN> enables/disables the double buffer. Setting <DB8EN, DBAEN> to 0 disables the double buffer; setting <DB8EN, DBAEN> to 1 enables the double buffer.
With the double buffer enabled, data are transmitted from the register buffer to the timer register at a match between up-counter UC8 and TREG9, or between UC9 and timer register TREGB.
As TREG8 to TREGB are undefined after a reset, when using a 16-bit timer write the data first.
A reset clears T89CR to 0 and disables the double buffer. When using the double buffer, write data to TREG8, TREGA, set T89CR<DB8EN, DBAEN> to 1, then write the next data to the register buffer.
The 16-bit timer registers and register buffers are allocated to the same addresses in memory. When T89CR<DB8EN, DBAEN> is set to 0, the same value is written to the timer register and register buffer.

When <DB8EN, DBAEN> is set to 1, the value is written to the register buffer only. Therefore, the register buffer must be disabled before writing the initial value to the timer register.

(3) Capture Register

The capture register is a 16-bit register for latching the 16-bit up-counter UC8, 9 value.
When reading the capture register, use a 2-byte data load instruction, or two 1-byte data load instructions; first to read the lower eight bits, then to read the upper eight bits.

Timer 8

CAP 1		CAP 2	
Upper eight bits	Lower eight bits	Upper eight bits	Lower eight bits
000035H	000034H	000037H	000036H

Timer 9

CAP 3		CAP 4	
Upper eight bits	Lower eight bits	Upper eight bits	Lower eight bits
000045H	000044H	000047H	000046H

CAP1 to CAP4 are read-only registers and cannot be written by software.

(4) Capture Input Control

The capture input control circuit controls the timing of the latching of the 16-bit up-counter UC8, 9 value to capture registers CAP1, CAP2, CAP3, and CAP4. Set the capture register latch timing with the timer 8, 9 mode control registers T8MOD<CAP12M1, 0>, T9MOD<CAP34M1, 0>.
The following describes the latch timing setting and operation.

- When T8MOD<CAP12M1, 0>, T9MOD<CAP34M1, 0> are set to 00:
  The capture function is disabled. A reset also disables the capture function.
- When T8MOD<CAP12M1, 0>, T9MOD<CAP34M1, 0> are set to 01:
  On the external input rising edge of TI8 (shared with P90/INT5) and TIA (shared with P94/INT7), capture register CAP1, CAP3 loads the up-counter value. On the external input rising edge of TI9 (shared with P91/INT6) and TIB (shared with P95/INT8), capture register CAP2, CAP4 loads the up-counter value. (Time differential measurement)
- When T8MOD<CAP12M1, 0>, T9MOD<CAP34M1, 0> are set to 10:
  On the TI8, TIA external input rising edge, capture register CAP1, CAP3 loads the up-counter value. On the input falling edge, capture register CAP2, CAP4 loads the up-counter value. Interrupt INT4, INT6 is generated on a falling edge in this mode only. (Pulse width measurement)
- When T8MOD<CAP12M1, 0>, T9MOD<CAP34M1, 0> are set to 11:
  On the timer flip-flop TFF1 rising edge, capture register CAP1, CAP3 loads the up-counter value. On the falling edge, capture register CAP2, CAP4 loads the up-counter value.
  The UC8, 9 up-counter value can also be loaded to a capture register on a software request. When 0 is written to T8MOD<CAP1IN>, T9MOD<CAP3IN>, the UC8, 9 up-counter value at that time is loaded to capture register CAP1, 3.
  The prescaler must first be set to RUN (set T16RUN<PRRUN> = 1).

(5) Comparator

To detect a match, the 16-bit comparator compares the 16-bit up-counter UC8, 9 with the 16-bit timer register TREG8, 9 and TREGA, B settings.
On detection of a match, the comparator outputs a match detect signal and generates interrupts INTTR8, 9 or INTTRA, B from the respective 16-bit timer.
UC8 is cleared by a match between the UC8 value and the TREG9 value. UC9 is cleared by a match between the UC9 value and the TREGB value. UC8, 9 clearing can be disabled by setting the timer 8, 9 mode control registers T8MOD<CLE>, T9MOD<CLE> to 0.

(6) Timer Flip-Flops

Timers 8 and 9 have two timer flip-flops each. The flip-flops of each timer have different functions.

① TFF8, TFFA

Flip-flops TFF8 and TFFA are inverted by a match signal from the comparator and a latch signal to the capture register.
In timer 8 and timer 9, two different capture operations and two types of match detection can be specified as inversion triggers. Use bits 2 to 5 of the T8FFCR and T9FFCR registers to set the inversion triggers.

② TFF9, TFFB

Timer flip-flops TFF9 and TFFB are inverted by a match signal from the comparator and a latch signal to the capture register.
In timers 8 and 9, one type of capture operation and one type of match detection can be specified as inversion triggers. Use bits 6 and 7 of the T8MOD and T9MOD registers to set the inversion triggers.

After a reset the timer flip-flop values are undefined. Writing 01 to T8FFCR <TFF8C1, 0>, <TFF9C1, 0> or T9FFCR <TFFAC1, 0>, <TFFBC1, 0> sets the timer flip-flop to 0; writing 10 to the bits sets the timer flip-flop to 1. Writing 00 to the bits inverts the timer flip-flop value (software inversion).
The TFF8, TFF9, TFFA, and TFFB values can be output to timer output pins TO8 (shared with P92), TO9 (shared with P93), TOA (shared with P96), and TOB (shared with P96) respectively.
As the timer output pins also function as P92, P93, and P96, set port 9 function register P9FC before performing timer output. (See Figure 3.5 (33), Port 9 Related Registers)

3.8.3 Operation Description for Each Mode

(1) 16-bit Interval Timer Mode

Interval timers 8 and 9 can be used independently as 16-bit interval timers. The following describes the example of timer 8 only.

Example: Generate interrupts at fixed intervals

To generate timer interrupts at fixed intervals, set the interval time (cycle) in 16-bit timer register TREG9 and use interrupt INTTR9.
Set the registers as follows.

```
           7 6 5 4 3 2 1 0
T16RUN   ← - X - 0 X X X X      Stop timer 8.
INTET89  ← 1 1 0 0 1 0 0 0      Enable INTTR9, set interrupt level to 4, and disable
                                INTTR8.
T8FFCR   ← 1 1 0 0 0 0 1 1      Disable trigger.
T8MOD    ← 0 0 1 0 0 1 * *      Set internal clock to input clock, disable capture
             (** = 01, 10, 11)  function, clear and enable up-counter.
TREG9    ← * * * * * * * *      Set interval time. (16 bits)
           * * * * * * * *
T16RUN   ← 1 X - 1 X X X X      Start timer 8.
```

Note: X: Don't care -: No change

(2) 16-Bit Event Counter Mode

Timers 8 and 9 can be set to operate as event counters by setting external inputs TI8 and TIA as the timer clock sources. The following describes timer 8 only.
The 16-bit up-counter UC8 counts up on the rising edge of the TI8 input. The count value can be read by performing a software capture and reading the capture value.
Timer input pin TI8 is shared with P90. However, there is no selection function. Therefore, event counter operation can be performed at any time by setting timer 8 to operating state. Set the registers as follows.

```
           7 6 5 4 3 2 1 0
T16RUN   ← - X - 0 X X X X      Stop timer 8.
P9CR     ← - - - - - - - 0      Set P90 to input mode.
INTET89  ← 1 1 0 0 1 0 0 0      Enable INTTR9 (level 4) and disable INTTR8.
T8FFCR   ← 1 1 0 0 0 0 1 1      Disable trigger.
T8MOD    ← 0 0 1 0 0 1 0 0      Set input clock to TI8.
TREG9    ← * * * * * * * *      Set number of counts (16 bits).
           * * * * * * * *
T16RUN   ← 1 X - 1 X X X X      Start timer 8.
```

Note 1: X: Don't care -: No change
Note 2: The prescaler must also be running when using a 16-bit timer as an event counter (T16RUN<PRRUN> =1).

(3) 16-Bit Programmable Pulse Generation (PPG) Output Mode

Timers 8 and 9 can output a square wave with a user-specified frequency and duty (programmable square wave). The output pulse can be either active-low or active-high.
Timer 8 outputs a square wave from pin TO8 (shared with P92); timer 9, from TOA (shared with P96).
The following describes timer 8 only.
A programmable pulse (square wave) can be output from pin TO8 by triggering inversion of timer flip-flop TFF8 when a match occurs between the 16-bit up-counter UC8 and TREG8, or between UC8 and TREG9. The TREG8 and TREG9 settings must satisfy the following condition:

(TREG8 setting) < (TREG9 setting)

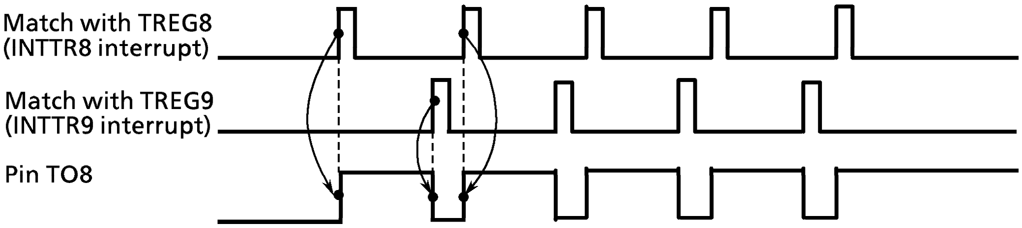


Figure 3.8 (3) 16-Bit Programmable Pulse Generation (PPG) Output Waveform

Enabling the TREG8 double-buffer in this mode shifts the value of register buffer 8 to TREG8 when TREG9 matches UC8. Using the double-buffer facilitates handling of small duty waves.

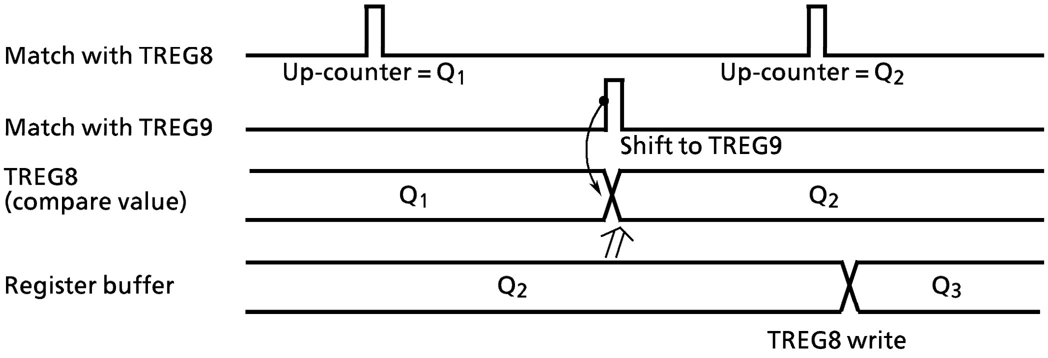


Figure 3.8 (4) Register Buffer Operation

Figure 3.8 (5) is a block diagram of 16-bit PPG output mode.

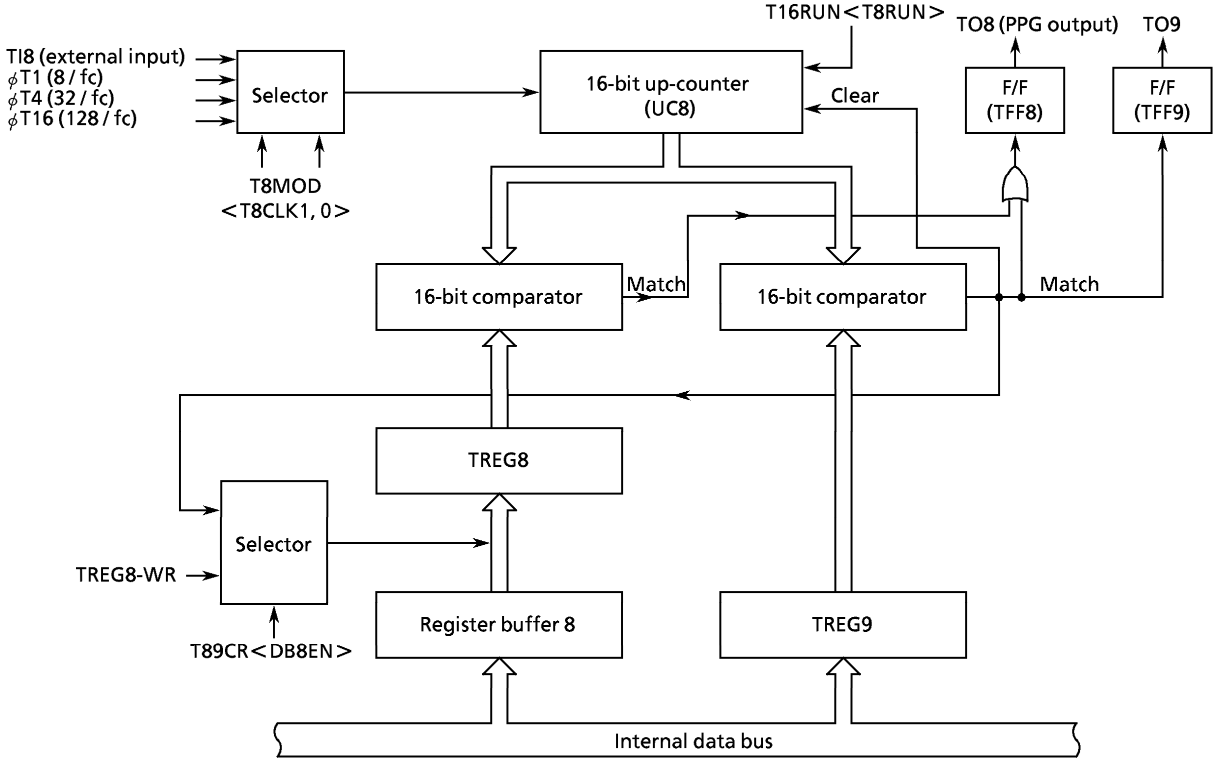


Figure 3.8 (5) 16-Bit PPG Output Mode Block Diagram

In 16-bit PPG output mode, set the registers as follows.

```
            7 6 5 4 3 2 1 0
┌
│ T16RUN  ← - X - 0 X X X X     Stop timer 8.
│ TREG8   ← * * * * * * * *     Set the duty.  (16 bits)
│           * * * * * * * *
│ TREG9   ← * * * * * * * *     Set the interval.  (16 bits)
│           * * * * * * * *
│ T89CR   ← 0 X X X X 0 - 1     Enable TREG8 double-buffer
│                               (Duty/interval modified by INTTR9 interrupt)
│ T8FFCR  ← 1 1 0 0 1 1 1 0     Set TFF8 to invert at detection of match with TREG8
│                               or TREG9.  Set TFF8 initial value to 0.
│ T8MOD   ← 0 0 1 0 0 1 * *     Set input clock to internal clock, and disable capture
│             (** = 01, 10, 11)  function.
│ P9CR    ← - - - - - - 1 - -  ┐
│ P9FC    ← X - X X - 1 X X    } Set P92 as TO8.
└ T16RUN  ← 1 X - 1 X X X X     Start timer 8.
```

Note: X: Don't care   -: No change

(4) Example of Capture Function Application

Use the capture function to realize many applications, including the following examples.

① One-shot pulse output from the external trigger pulse

② Frequency measurement

③ Pulse width measurement

④ Time differential measurement

The following describes these applications based on timer 8.

① One-shot pulse output from external trigger pulse

Obtain one-shot pulse output from the external trigger pulse as follows.
Set 16-bit up-counter UC8 to free-running count-up using an internal clock.
Input the external trigger pulse from pin TI8. Load the up-counter value to capture register CAP1 on the rising edge of the external trigger pulse using the capture function.
Interrupt INT5 is generated on the rising edge of the external trigger pulse. Add the value of capture register CAP1 at this interrupt (c) to the delay time (d), and set timer register TREG8 to the sum of these values (c + d). Add the pulse width of the one-shot pulse (p) to TREG8, and set timer register TREG9 to the result (c + d + p).
In addition, set the timer 8 flip-flop control register T8FFCR<EQ9T8, EQ8T8> to 11 and enable the trigger to invert timer flip-flop TFF8 when a match occurs between UC8 and TREG8 or UC8 and TREG9. Then, after output of the one-shot pulse, set the trigger back to disabled state during INTTR9 interrupt processing.
The (c), (d), and (p) notation above corresponds to c, d, and p in Figure 3.8 (6), One-Shot Pulse Output from External Trigger Pulse (With Delay).

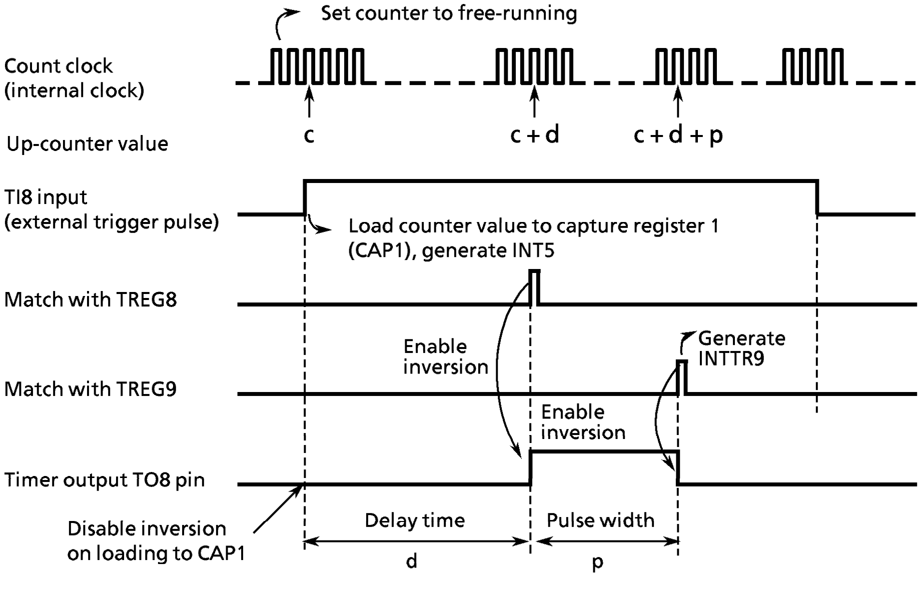

Set counter to free-running
Count clock (internal clock)
Up-counter value
c
c + d
c + d + p
TI8 input (external trigger pulse)
Load counter value to capture register 1 (CAP1), generate INT5
Match with TREG8
Enable inversion
Generate INTTR9
Match with TREG9
Enable inversion
Timer output TO8 pin
Disable inversion on loading to CAP1
Delay time
d
Pulse width
p

Figure 3.8 (6) One-Shot Pulse Output from External Trigger Pulse (With Delay)

Example: On pin TI8, output 2ms one-shot pulse with 3ms-delay after external trigger pulse.

```
Main settings                                Set to free-running.
                                             Set source clock to φT1.
T8MOD   ←  - - 1 0 1 0 0 1
                                             Load counter value to CAP1 at TI8 input rising edge.
T8FFCR  ←  1 1 0 0 0 0 1 0
                                             Zero clear TFF8 output.
                                             Disable TFF8 inversion.
P9CR    ←  - - - - - 1 - -
P9FC    ←  X - X X - 1 X X                 } Set P92 as TO8.

INTE45  ←  - - - - 1 1 0 0                   Enable INT5 and disable INTTR8 and INTTR9.
INTET89 ←  1 0 0 0 1 0 0 0
T16RUN  ←  1 X - 1 X X X X                   Start timer 8.

Settings at INT5

TREG8   ←  * * * * * * * *                   CAP1+3ms/φT1
TREG9   ←  * * * * * * * *                   TREG8+2ms/φT1
T8FFCR  ←  - - - - 1 1 - -
                                             Enable TFF8 inversion on match with TREG8 or TREG9.
INTET89 ←  1 1 0 0 - - - -                   Enable INTTR9.

Settings at INTTR9

T8FFCR  ←  - - - - 0 0 - -
                                             Disable TFF8 inversion on match with TREG8 or TREG9.

INTET89 ←  1 0 0 0 - - - -                   Disable INTTR9.
```

Note: X: Don't care -: No change

If delay time is not required, invert timer flip-flop TFF8 by loading capture register 1 (CAP1). Set timer register TREG9 to the sum of the one-shot pulse width (p) and the value of CAP1 at interrupt INT5 (c) (c + p). Set the TFF8 inversion on a match between TREG9 and UC8, and select inversion enable. On interrupt INTTR9, disable the TFF8 inversion.

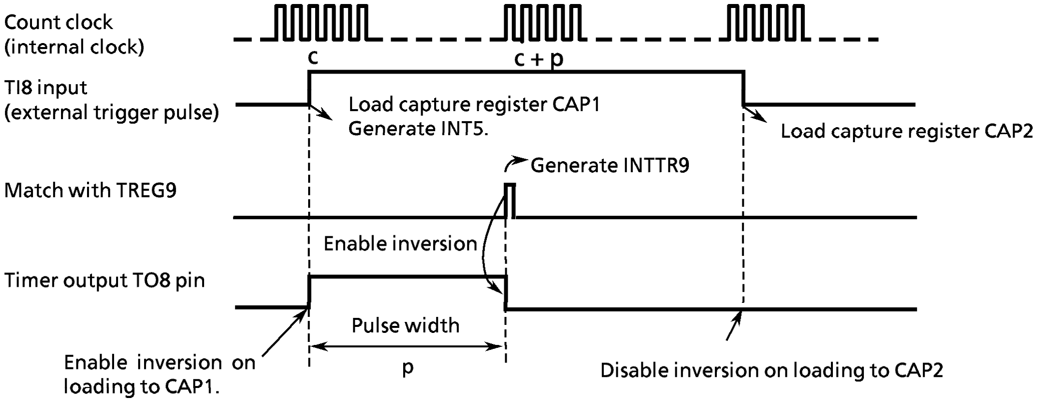


Figure 3.8 (7) External Trigger Pulse One-Shot Pulse Output (No Delay)

② Frequency measurement

The frequency of an external clock can be measured by the capture function.
The frequency is measured by combining the 8-bit timers (timers 0, 1) in 16-bit event counter mode. (Timers 0 and 1 are used to set the measuring time by inverting TFF1.)
Select the TI8 input as the timer 8 count clock and count timer 8 on the external clock input. Set timer 8 mode control register T8MOD<CAP12M1, 0> to 11. This setting loads the counter value of 16-bit up-counter UC8 into capture register CAP1 on the rising edge of timer flip-flop TFF1. It also loads the counter value into capture register CAP2 on the falling edge of timer flip-flop TFF1. TFF1 is the timer flip-flop of the 8-bit timers (timers 0, 1).
Based on the measuring time, the frequency is calculated from the difference between capture registers CAP1 and CAP2 at the 8-bit timer interrupts (INTT0 or INTT1).

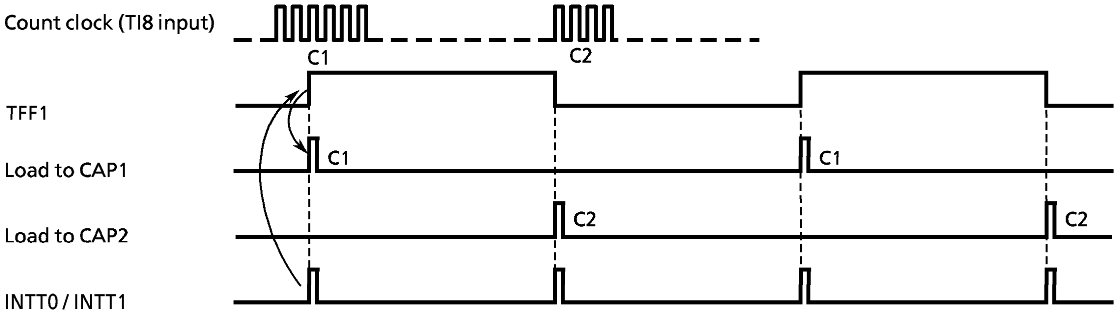


Figure 3.8 (8) Frequency Measurement

For example, if TFF1 (8-bit timer flip-flop) is set to 1 for 0.5 s, and the difference between CAP1 and CAP2 is 100, the frequency is 100 ÷ 0.5 s = 200 Hz.

③ Pulse width measurement

The high-level width of an external pulse can be measured using the 16-bit timer capture function.
To measure the pulse width, first set 16-bit up-counter UC8 to operate as a free-running up-counter driven by an internal clock. Using the capture function, load the up-counter value into capture registers CAP1 and CAP2 on the rising and falling edges respectively of the external pulse being measured on the TI8 pin.
Using these settings, the high-level pulse width can be calculated during INT5 interrupt processing by multiplying the difference between CAP1 and CAP2 by the internal clock cycle.

For example, if the difference between CAP1 and CAP2 is 100 and the internal clock cycle is 0.8μs, the pulse width is $100 \times 0.8\mu s = 80\mu s$.

Caution is required for the case when the width of the pulse being measured exceeds the maximum UC8 count time (which is determined by the clock source). Software processing is required for this case.

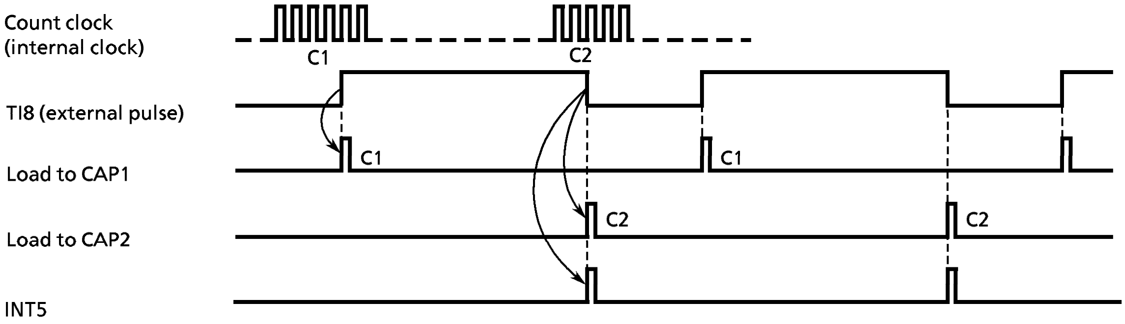


Figure 3.8 (9) Pulse Width Measurement

Note: Measure pulse width by setting the timer 8 mode control register T8MOD<CAP12M1, 0> to 10. External interrupt INT5 is generated on the falling edge of the TI8 input pin. At other settings, INT5 is generated on the rising edge of TI8.

The width of low level external pulses can also be measured. In this case, the pulse width is calculated during the interrupt processing for the second INT5 interrupt by multiplying the internal clock cycle by the difference between the value of C2 at the first INT5 interrupt and the value of C1 at the second INT5 interrupt. However, as the first C2 value has been overwritten by the time of the second INT5 interrupt, the C2 value must be saved during the first INT5 interrupt processing.

④ Time difference measurement

The time difference between two events can be measured using the 16-bit timer capture function.
To measure time difference, first set the 16-bit up-counter UC8 to operate as a free-running up-counter driven by an internal clock. Load the value of up-counter UC8 is into capture register CAP1 on a rising edge detected on the TI8 pin input pulse. At this time, interrupt INT5 is generated.
Similarly, on a rising edge detected on the TI9 pin input pulse, load the value of up-counter UC8 value into capture register CAP2. At this time, interrupt INT6 is generated.
When both values have been loaded into the capture registers, calculate the time difference by multiplying the difference between CAP2 and CAP1 by the internal clock cycle.

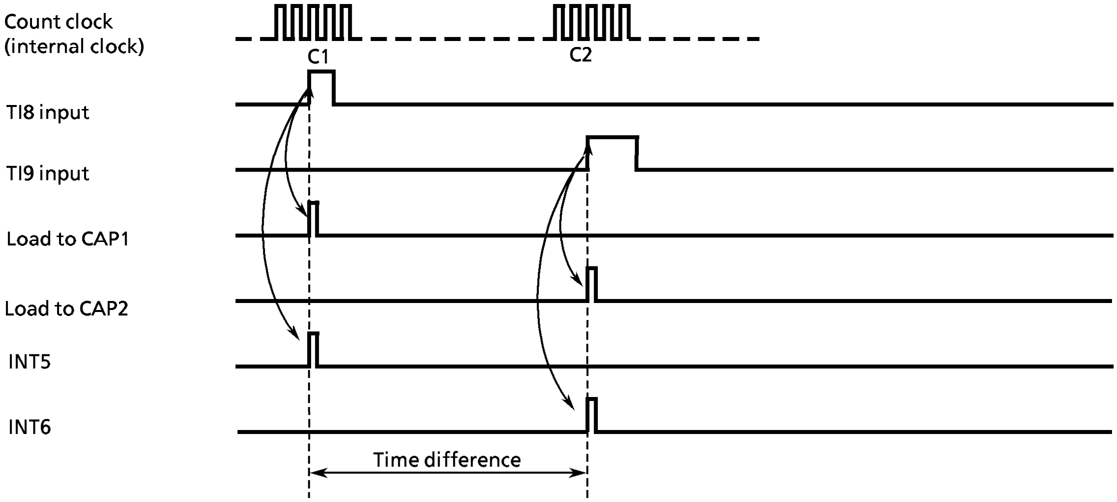


Figure 3.8 (10) Time Difference Measurement

(5) Phase Output (Only available on timer 8)

Signals with a user-specified phase difference can be output using the 16-bit timer.
Select an internal clock as the clock source and set the 16-bit up-counter UC8 to free-running. Set the phase difference in 16-bit timer registers TREG8 and TREG9, set timer flip-flops TFF8 and TFF9 to invert when a match is detected for TREG8 and TREG9, and output the flip-flop values from TO8 and TO9.

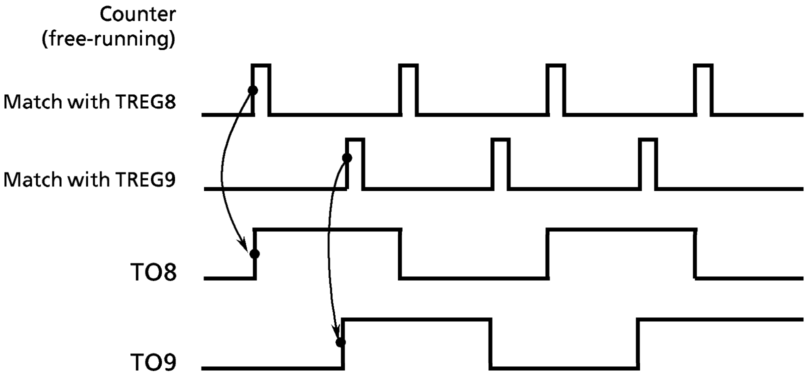


Figure 3.8 (11) Phase Output

Table 3.8 (1) lists the cycles (counter overflow times) that can be set for each clock source.

Table 3.8 (1) 16-Bit Up-Counter Overflow Times

	20 MHz	25 MHz
φT1	26.214 ms	20.97 ms
φT4	104.856 ms	83.88 ms
φT16	419.424 ms	335.54 ms

3.9 Serial Channels

TMP95CS64/265 has three internal serial input/output channels. The serial channels have the following four operating modes.

- I/O interface mode
  - Mode 0: Can be used to expand the I/O by sending and receiving I/O data and the associated synchronizing signal (SCLK).
- Universal asynchronous receiver transmitter (UART) mode
  - Mode 1: Send/receive data length: 7 bits
  - Mode 2: Send/receive data length: 8 bits
  - Mode 3: Send/receive data length: 9 bits

A parity bit can be added in modes 1 and 2. Mode 3 has a wake-up function that allows a master controller to activate slave controllers via a serial link (multi-controller system).

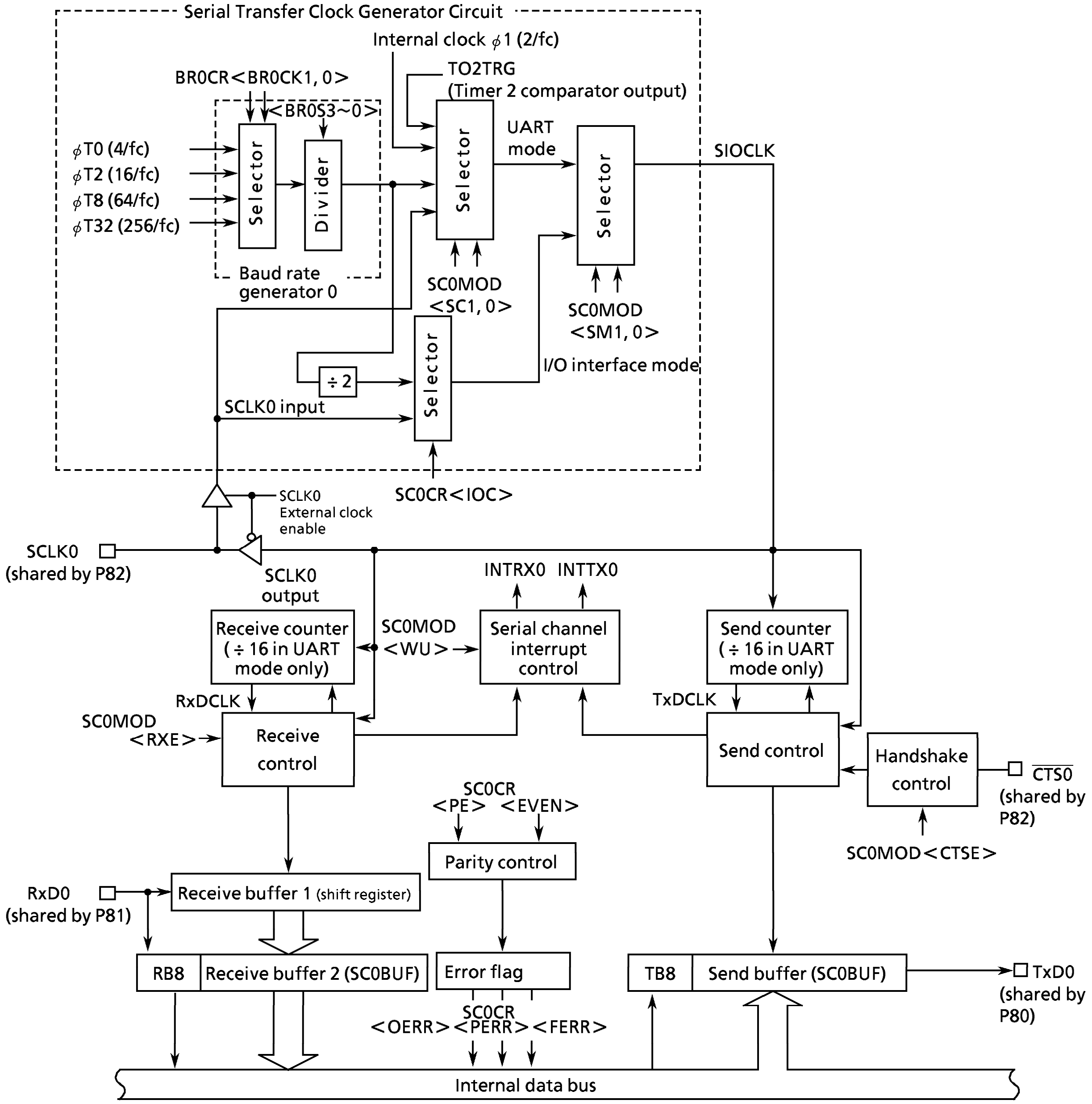


Figure 3.9 (1) Block Diagram of Serial Channel 0

3.9.1 Serial Channel Registers

Each serial channel is controlled by three control registers (SC0CR, SC0MOD, and BR0CR in the case of channel 0). Data sent and received are stored in the serial send/receive buffer register in each channel (SC0BUF in the case of channel 0).

(1) Serial Channel 0

Serial Channel 0 Mode Control Register

SC0MOD
(004EH)

	7	6	5	4	3	2	1	0
bit Symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	Undefined	0	0	0	0	0	0	0
Function	Bit 8 of send data	Handshake function control 0: $\overline{\text{CTS0}}$ disable 1: $\overline{\text{CTS0}}$ enable	Receive control 0: Receive disable 1: Receive enable	Wake-up function 0: Disable 1: Enable	Serial transfer mode selection 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial transfer clock selection (UART mode) 00: TO2 trigger 01: Baud rate generator 0 10: Internal clock ϕ1 11: SCLK0 pin input (external clock)	

Serial transfer clock selection (UART mode)

00	Timer 2 comparator output
01	Baud rate generator 0
10	Internal clock ϕ1 (2/fc)
11	SCLK0 pin input (external clock)

Serial transfer mode selection

00	I/O interface mode
01	7-bit UART mode
10	8-bit UART mode
11	9-bit UART mode

Wake-up function
(Other than 9-bit UART mode, don't care)

0	Disable
1	Enable

Receive control

0	Receive disable
1	Receive enable

Store bit 8 of send data

8-bit UART mode (with parity)	Store send parity bit
9-bit UART mode	Store bit 8 of receive data

Handshake function ($\overline{\text{CTS0}}$ pin) enable

0	Disable (send always enabled)
1	Enable

Figure 3.9 (2)-1 Serial Channel 0 Related Register

Serial Channel 0 Control Register

SC0CR
(004DH)

	7	6	5	4	3	2	1	0
bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (Cleared to 0 when read)			R/W	
After reset	Undefined	0	0	0	0	0	0	0
Function	Bit 8 of receive data	Odd/even parity selection 0: Odd 1: Even	Parity bit addition control 0: No parity bit 1: Parity bit	1: Error Overrun error flag	Parity error flag	Framing error flag	SCLK0 pin input edge selection 0: Rising edge 1: Falling edge	Serial transfer clock selection (I/O interface mode) 0: Baud rate generator 0 1: SCLK0 pin input

Serial transfer clock selection (I/O interface mode)

0	Baud rate generator 0 (Note 1)
1	SCLK0 pin input (external clock)

Edge selection in SCLK0 pin input operations

0	Data send/receive at SCLK0 rising edge
1	Data send/receive at SCLK0 falling edge

Framing error flag
Parity error flag
Overrun error flag

Cleared to 0 when read. (Note 2)

Parity bit addition control

0	No parity bit
1	Parity bit

Even/odd parity selection

0	Odd parity
1	Even parity

Receive data store bit 8

8-bit UART mode (with parity)	Store receive parity bit
9-bit UART mode	Store bit 8 of receive data

Note 1: To use the baud rate generator, set T16RUN<PRRUN> to 1 and run the prescaler.
Note 2: As the error flags are all cleared to 0 after reading, don’t test only one bit with a bit test instruction.

Figure 3.9 (2)-2 Serial Channel 0 Related Register

Baud Rate Generator 0 Control Register

BR0CR
(004FH)

	7	6	5	4	3	2	1	0
bit Symbol	–		BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write	R/W		R/W					
After reset	0		0	0	0	0	0	0
Function	Note: Always fixed to 0.		Baud rate generator 0 input clock selection 00: ϕT0 (4/fc) 01: ϕT2 (16/fc) 10: ϕT8 (64/fc) 11: ϕT32 (256/fc)		Divisor setting for baud rate generator 0			

Baud rate generator 0 divisor setting

0000	Divide by 16
0001 ~ 1111	Divide by 1 (no division) (Note 1) to 15

Baud rate generator 0 input clock selection

00	Internal clock ϕT0 (4/fc)
01	Internal clock ϕT2 (16/fc)
10	Internal clock ϕT8 (64/fc)
11	Internal clock ϕT32 (256/fc)

Note 1: The baud rate generator can be divided by 1 in UART mode only. Do not use this setting in I/O interface mode.

Note 2: Don't read from or write to BR0CR register during sending or receiving.

Serial Channel 0 Buffer Register

SC0BUF
(004CH)
Read-modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol	RB07	RB06	RB05	RB04	RB03	RB02	RB01	RB00
	TB07	TB06	TB05	TB04	TB03	TB02	TB01	TB00
Read/Write	R (receive) / W (send)							
After reset	Undefined							

Figure 3.9 (2)-3 Serial Channel 0 Related Registers

(2) Serial Channel 1

Serial Channel 1 Mode Control Register

SC1MOD (0052H)

	7	6	5	4	3	2	1	0
bit Symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	Undefined	0	0	0	0	0	0	0
Function	Bit 8 of send data	Handshake function control 0: $\overline{\text{CTS1}}$ disable 1: $\overline{\text{CTS1}}$ enable	Receive control 0: Receive disable 1: Receive enable	Wake-up function 0: Disable 1: Enable	Serial transfer mode selection 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial transfer clock selection (UART mode) 00: TO2 trigger 01: Baud rate generator 1 10: Internal clock ϕ1 11: SCLK1 pin input (external clock)	

→ Serial transfer clock selection (UART mode)

00	Timer 2 comparator output
01	Baud rate generator 1 output
10	Internal clock ϕ1
11	SCLK1 pin input (external clock)

→ Serial transfer mode selection

00	I/O interface mode
01	7-bit UART mode
10	8-bit UART mode
11	9-bit UART mode

→ Wake-up function (Other than 9-bit UART mode, don't care)

0	Disable
1	Enable

→ Receive control

0	Receive disable
1	Receive enable

→ Handshake function ($\overline{\text{CTS1}}$ pin) enable

0	Disable (send always enabled)
1	Enable

Store bit 8 of send data

8-bit UART mode (with parity)	Store send parity bit
9-bit UART mode	Store bit 8 of receive data

Figure 3.9 (2)-4 Serial Channel 1 Related Register

Serial Channel 1 Control Register

SC1CR
(0051H)

	7	6	5	4	3	2	1	0
bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (Cleared to 0 when read.)			R/W	
After reset	Undefined	0	0	0	0	0	0	0
Function	Bit 8 of receive data	Even/odd parity selection 0: Odd 1: Even	Parity bit addition control 0: No parity bit 1: Parity bit	1: Error Overrun error flag	Parity error flag	Framing error flag	SCLK1 pin input edge selection 0: Rising edge 1: Falling edge	Serial transfer clock selection (I/O interface mode) 0: Baud rate generator 1 1: SCLK1 pin input

Serial transfer clock selection
(I/O interface mode)

0	Baud rate generator 1 (Note 1)
1	SCLK1 pin input (external clock)

Edge selection in SCLK1 pin input operations

0	Data send/receive at SCLK1 rising edge
1	Data send/receive at SCLK1 falling edge

Framing error flag
Parity error flag
Overrun error flag
Cleared to 0 when read. (Note 2)

Parity bit addition control

0	No parity bit
1	Parity bit

Even/odd parity selection

0	Odd parity
1	Even parity

Store bit 8 of receive data

8-bit UART mode (with parity)	Store receive parity bit
9-bit UART mode	Store bit 8 of receive data

Note 1: To use the baud rate generator, set T16RUN<PRRUN> to 1 and run the prescaler.
Note 2: As the error flags are all cleared to 0 after reading, don't test only one bit with a bit test instruction.

Figure 3.9 (2)-5 Serial Channel 1 Related Register

Baud Rate Generator 1 Control Register

BR1CR
(0053H)

	7	6	5	4	3	2	1	0
bit Symbol	–		BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
Read/Write	R/W		R/W					
After reset	0		0	0	0	0	0	0
Function	Note: Always fixed to 0.		Baud rate generator 1 input clock selection 00: φT0 (4/fc) 01: φT2 (16/fc) 10: φT8 (64/fc) 11: φT32 (256/fc)		Divisor setting for baud rate generator 1			

Baud rate generator 1 divisor setting

0000	Divide by 16
0001 to 1111	Divide by 1 (no division) (Note 1) to 15

Baud rate generator 1 input clock selection

00	Internal clock φT0 (4/fc)
01	Internal clock φT2 (16/fc)
10	Internal clock φT8 (64/fc)
11	Internal clock φT32 (256/fc)

Note 1: The baud rate generator can be divided by 1 in UART mode only. Do not use this setting in I/O interface mode.

Note 2: Don't read from or write to BR1CR register during sending or receiving.

Serial Channel 1 Buffer Register

SC1BUF
(0050H)
Read-modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol	RB17	RB16	RB15	RB14	RB13	RB12	RB11	RB10
	TB17	TB16	TB15	TB14	TB13	TB12	TB11	TB10
Read/Write	R (receive) / W (send)							
After reset	Undefined							

Figure 3.9 (2)-6 Serial Channel 1 Related Registers

(3) Serial Channel 2

Serial Channel 2 Mode Control Register

SC2MOD
(0056H)

	7	6	5	4	3	2	1	0
bit Symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	Undefined	0	0	0	0	0	0	0
Function	Bit 8 of send data	Handshake function control 0: $\overline{\text{CTS2}}$ disable 1: $\overline{\text{CTS2}}$ enable	Receive control 0: Receive disable 1: Receive enable	Wake-up function 0: Disable 1: Enable	Serial transfer mode selection 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial transfer clock selection (UART mode) 00: TO2 trigger 01: Baud rate generator 10: Internal clock ϕ1 11: SCLK2 pin input (external clock)	

Serial transfer clock selection (UART mode)

00	Timer 2 match detect signal
01	Baud rate generator
10	Internal clock ϕ1
11	SCLK2 pin input (external clock)

Serial transfer mode selection

00	I/O interface mode
01	7-bit UART mode
10	8-bit UART mode
11	9-bit UART mode

Wake-up function
(Other than 9-bit UART mode, don't care)

0	Disable
1	Enable

Receive control

0	Receive disable
1	Receive enable

Handshake function ($\overline{\text{CTS2}}$ pin) enable

0	Disable (send always enabled)
1	Enable

Store bit 8 of send data

8-bit UART mode (with parity)	Store send parity bit
9-bit UART mode	Store bit 8 of receive data

Figure 3.9 (2)-7 Serial Channel 2 Related Register

Serial Channel 2 Control Register

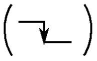
SC2CR
(0055H)

	7	6	5	4	3	2	1	0
bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (Cleared to 0 when read.)			R/W	
After reset	Undefined	0	0	0	0	0	0	0
Function	Bit 8 of receive data	Even/odd parity selection 0: Odd 1: Even	Parity bit addition control 0: No parity bit 1: Parity bit	1: Error Overrun error flag	Parity error flag	Framing error flag	SCLK2 pin input edge selection 0: Rising edge 1: Falling edge	Serial transfer clock selection (I/O interface mode) 0: Baud rate generator 2 1: SCLK2 pin input

Serial transfer clock selection (I/O interface mode)

0	Baud rate generator 2 (Note 1)
1	SCLK2 pin input (external clock)

Edge selection in SCLK2 pin input operations

0	Data send/receive at SCLK2 rising edge	
1	Data send/receive at SCLK2 falling edge	

Framing error flag
Parity error flag
Overrun error flag
Cleared to 0 when read. (Note 2)

Parity bit addition control

0	No parity bit
1	Parity bit

Even/odd parity selection

0	Odd parity
1	Even parity

Store bit 8 of receive data

8-bit UART mode (with parity)	Store receive parity bit
9-bit UART mode	Store Bit 8 of receive data

Note 1: To use the baud rate generator, set T16RUN<PRRUN> to 1 and run the prescaler.
Note 2: As the error flags are all cleared to 0 after reading, don't test only one bit with a bit test instruction.

Figure 3.9 (2)-8 Serial Channel 2 Related Register

Baud Rate Generator 2 Control Register

BR2CR
(0057H)

	7	6	5	4	3	2	1	0
bit Symbol	–		BR2CK1	BR2CK0	BR2S3	BR2S2	BR2S1	BR2S0
Read/Write	R/W		R/W					
After reset	0		0	0	0	0	0	0
Function	Note: Always fixed to 0.		Baud rate generator 2 input clock selection 00: ϕT0 (4/fc) 01: ϕT2 (16/fc) 10: ϕT8 (64/fc) 11: ϕT32 (256/fc)		Divisor setting for baud rate generator 2			

Baud rate generator 2 divisor setting

0000	Divide by 16
0001 to 1111	Divide by 1 (no division) (note 1) to 15

Baud rate generator 2 input clock selection

00	Internal clock ϕT0 (4/fc)
01	Internal clock ϕT2 (16/fc)
10	Internal clock ϕT8 (64/fc)
11	Internal clock ϕT32 (256/fc)

Note 1: The baud rate generator can be divided by 1 in UART mode only. Do not use this setting in I/O interface mode.

Note 2: Don’t read from or write to BR2CR register during sending or receiving.

Serial Channel 2 Buffer Register

SC2BUF
(0054H)
Read-modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol	RB27	RB26	RB25	RB24	RB23	RB22	RB21	RB20
	TB27	TB26	TB25	TB24	TB23	TB22	TB21	TB20
Read/Write	R (receive) / W (send)							
After reset	Undefined							

Figure 3.9 (2)-9 Serial Channel 2 Related Registers

3.9.2 Block Structure

As serial channels 0 to 2 operate identically, the following uses channel 0 as an example.

(1) Serial Transfer Clock Generator Circuit

The serial transfer clock generator circuit generates SIOCLK (internal signal), which is the send/receive basic clock. To generate SIOCLK, select the clock source required for the generation.

① I/O interface mode

As the clock source, select either baud rate generator 0, or SCLK0 from an external source. Set the clock source in bit 0 (<IOC>) of serial channel 0 control register SC0CR.
When baud rate generator 0 is selected (<IOC> = 0), this circuit generates SIOCLK by dividing the output of the baud rate generator by 2.
When external SCLK0 is selected (<IOC> = 1), SIOCLK is set to the same value as the external source.

② UART mode

In addition to the clock sources in I/O interface mode, the comparator output of timer 2 and internal clock ϕ1 (2/fc) can also be selected as clock sources.
Bits 1 and 0 of serial channel 0 mode control register SC0MOD<SC1,0> select the clock source. SIOCLK is set to the same value as the selected clock source.

(2) Receive Counter

The receive counter is a 4-bit binary counter used in UART mode.
The receive counter uses SIOCLK as the count clock to generate receive sampling clock RxDCLK (internal signal).

(3) Receive Control

① I/O Interface mode

In I/O interface mode, the receive data input to the RxD0 pin are sampled synchronously with transfer clock SCLK0.
Setting serial channel 0 control register SC0CR<IOC> to 0 samples the received data on the rising edge of SCLK0. Setting SC0CR<IOC> to 1 samples the data on the rising or the falling edge of SCLK0 as determined by the setting of SC0CR<SCLKS>.

② UART mode

The receive data are sampled bit by bit using RxDCLK, which is generated with the receive counter. Each bit of data is sampled three times, using majority rule. If two or more instances of the same value are detected among three samples, the circuit recognizes the data as receive data. If the sampled data are 1, 0, 1, for example, the data are evaluated as 1. If 0, 0 , 1, the data are evaluated as 0.

(4) Receive Buffer

The receive buffer has a double-buffer configuration to prevent overrun error. Receive buffer 1 stores the data received bit by bit.
When receive buffer 1 contains seven or eight bits of data, the data are transferred to receive buffer 2 (SC0BUF), generating interrupt INTRX0.
Reading the data in receive buffer 2 clears the interrupt request flag INTRX0<IRX0C>.
Even before the CPU reads the data in receive buffer 2, the next data can be received and stored in receive buffer 1.
However, receive buffer 2 must be read before all bits of the next data frame are received by buffer 1. If not, an overrun error occurs and the contents of receive buffer 1 are lost, although the contents of receive buffer 2 and the serial channel 0 control register SC0CR<RB8> are preserved.
In 8-bit UART mode (mode 2) with parity added, the parity bit is stored in SC0CR<RB8>. In 9-bit UART mode (mode 3), the MSB is stored in SC0CR<RB8>.

(5) Send Counter

The send counter is a 4-bit binary counter used in UART mode.
The send counter uses SIOCLK as its count clock, generating send clock TxDCLK (internal signals).

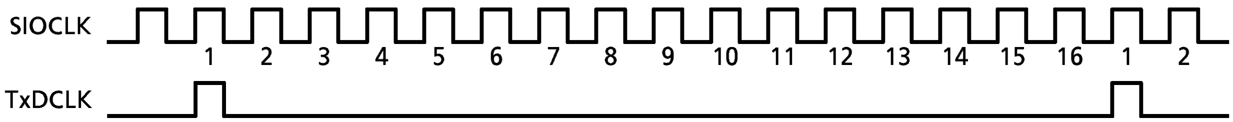


Figure 3.9 (3) Send Clock Generation

(6) Send Control

① I/O interface mode

In I/O interface mode, TMP95CS64/265 outputs send data from the TxD0 pin synchronously with transfer clock SCLK0.
Setting serial channel 0 control register SC0CR<IOC> to 0 outputs send data on the rising edge of transfer clock SCLK0.
Setting SC0CR<IOC> to 1 outputs the send data on the rising or falling edge of SCLK0 as determined by the setting of SC0CR<SCLKS>.

② UART mode

In UART mode, the send data are output synchronously with the rising edge of the TxDCLK send clock generated by the send counter.

(7) Send Buffer

Send buffer (SC0BUF) outputs the send data written by the CPU, beginning with the least significant bit.
When all bits are output, the empty send buffer generates interrupt request INTTX0.

(8) Parity Control

Parity bit addition can only be set in 7-bit UART mode (mode 1) and 8-bit UART mode (mode 2).
When serial channel 0 control register SC0CR<PE> is set to 1, data can be sent with a parity bit added. SC0CR<EVEN> selects even parity or odd parity.
A send operation automatically generates the parity bit determined by the send data. In mode 1, SC0BUF<TB7> stores the parity bit; in mode 2, serial channel 0 mode control register SC0MOD<TB8> stores the parity bit.
Set both <PE> and <EVEN> before writing the send data in SC0BUF.
When receiving, parity is calculated from the received data and compared with the received parity bit. If the parities differ, a parity error occurs and parity error flag SC0CR<PERR> is set to 1.

(9) Error Flags

To improve the reliability of data reception, serial channel 0 control register SC0CR contains the following three error flags.

① Overrun error <OERR>

When all bits of the next data frame have been received in receive buffer 1 while valid data are stored in receive buffer 2 (SC0BUF), an overrun error occurs.
At an overrun error, the data received in buffer 1 are lost.

② Parity error <PERR>

The parity bit determined by the data stored in receive buffer 2 (SC0BUF) is compared with the received parity bit. If the parities differ, a parity error occurs.

③ Framing error <FERR>

The stop bit of data received is sampled three times. If the majority of samples are 0, a framing error occurs.

If an error occurs, these error flags are set to 1. Reading the SC0CR register clears the error flags to 0. If an error occurs, fix by software.

(10) Handshake Function Control (only supported in UART mode)

The serial channels use the $\overline{\text{CTS0}}$ input pin to send data in one-frame units, thus preventing an overrun error. The serial channel 0 mode control register SC0MOD<CTSE> enables or disables the handshake function.
In send operations, sending starts when a low level signal is input to the $\overline{\text{CTS0}}$ pin.
When $\overline{\text{CTS0}}$ goes high, data sending is halted when sending of the current data completes and the pin is set to wait state. Sending is not restarted until $\overline{\text{CTS0}}$ goes low again.
Although an $\overline{\text{RTS0}}$ pin is not provided, any port can be assigned to the $\overline{\text{RTS0}}$ function. When the receiving side has completed reception, the receiving interrupt processing routine outputs a high-level signal from the port assigned to the $\overline{\text{RTS0}}$ function. A handshake function can be easily configured by connecting the sending side $\overline{\text{CTS0}}$ pin and the receiving side $\overline{\text{RTS0}}$ pin.

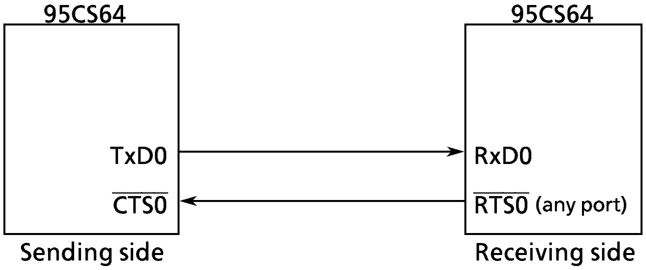
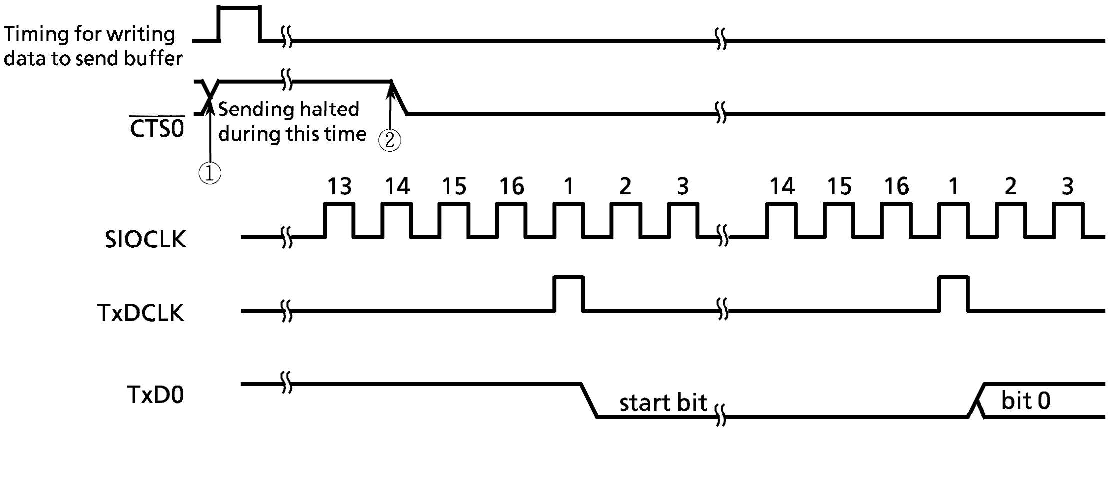


Figure 3.9 (4) Handshake Function


Timing for writing data to send buffer
CTS0
Sending halted during this time
①
②
13 14 15 16 1 2 3 14 15 16 1 2 3
SIOCLK
TxDCLK
TxD0
start bit
bit 0

① When the $\overline{\text{CTS0}}$ signal rises during sending, sending of the current data frame completes and sending of the next data frame halts.
② Sending begins at the first TxDCLK clock falling edge after the $\overline{\text{CTS0}}$ signal drops.

Figure 3.9 (5) $\overline{\text{CTS0}}$ (Clear to Send) Signal Timing

3.9.3 Description of Operation

As serial channels 0 to 2 operate identically, the following uses channel 0 as an example.

(1) Setting Send/Receive Clock Transfer Rate

① Transfer rate setting with baud rate generator selected

The baud rate generator is a circuit used to generate a clock source for the send/receive clock that controls the serial channel transfer rate.
The input clock for generating the clock source can be selected among ϕT0 (4/fc), ϕT2 (16/fc), ϕT8 (64/fc), or ϕT32 (256/fc) from the 9-bit prescaler (see 3.7.2 (1), Prescaler). The 8-bit and 16-bit timers share the prescaler. Bits 5, 4 of baud rate generator control register BR0CR<BR0CK1:0> select the input clock. The selected input clock is divided by the 4-bit divider performing 1 to 16 divisions. Bits 3 to 0 of BR0CR<BR0S3:0> set the divider. The divided clock is the output clock for the baud rate generator.

The following are the transfer rate calculation formulas when the baud rate generator is selected:

- I/O interface mode

$$\text{Transfer rate [bps]} = \frac{\text{Baud rate generator input clock [Hz]}}{\text{Baud rate generator divisor (2 to 16)}} \div 2$$

Note: In I/O interface mode, do not set divisor to 1.

- UART mode

$$\text{Transfer rate [bps]} = \frac{\text{Baud rate generator input clock [Hz]}}{\text{Baud rate generator divisor (1 to 16)}} \div 16$$

The relationship between the input clock and the source clock (fc) is:

ϕT0 = 4/fc
ϕT2 = 16/fc
ϕT8 = 64/fc
ϕT32 = 256/fc

Accordingly, with the source clock set to 12.288MHz, when ϕT2 (16/fc) is selected as the input clock and the divisor is 5, the transfer rate in UART mode is:

$$\text{Transfer rate} = \frac{\text{fc/16}}{5} \div 16 = 12.288 \times 10^{6} \div 16 \div 5 \div 16 = 9600\text{[bps]}$$

Table 3.9 (1) shows examples of transfer rate settings in UART mode.

② Transfer rate settings with the timer 2 comparator output selected (UART mode only)

The following are the transfer rate calculation formulas when the timer 2 comparator output is selected:

$$\text{Transfer rate [bps]} = \frac{\text{Timer 2 input clock [Hz]}}{\text{TREG2 (1 to 256)}} \div 16$$

The relationship between the timer 2 input clock and the source clock (fc) is:

$\phi T1 = 8/fc$
$\phi T4 = 32/fc$
$\phi T16 = 128/fc$

Accordingly, with the source clock set to 25MHz, when the timer 2 input clock is set to ϕT1 and TREG2 is set to 1, the transfer rate is:

$$\text{Transfer rate} = \frac{fc/8}{\text{TREG2}} \div 16 = 25 \times 10^6 \div 8 \div 1 \div 16 \doteqdot 195312 \text{ [bps]}$$

Table 3.9 (2) shows examples of the transfer rate settings.

③ Transfer rate settings with external SCLK input selected

The following are the transfer rate calculation formulas when the external SCLK input is selected:

- I/O interface mode
  Transfer rate [bps] = external SCLK input [Hz] ÷ 2
- UART mode
  Transfer rate [bps] = external SCLK input [Hz] ÷ 16

Table 3.9 (1) UART Mode Transfer Rate Setting Example (1) (Using Baud Rate Generator)

Unit: Kbps

fc [MHz]	Input clock Divisor	φT0 (4/fc)	φT2 (16/fc)	φT8 (64/fc)	φT32 (256/fc)
9.830400	1	153.600	38.400	9.600	2.400
	2	76.800	19.200	4.800	1.200
	4	38.400	9.600	2.400	0.600
	8	19.200	4.800	1.200	0.300
	16	9.600	2.400	0.600	0.150
12.288000	5	38.400	9.600	2.400	0.600
	10	19.200	4.800	1.200	0.300
14.745600	1	230.400	57.600	14.400	3.600
	3	76.800	19.200	4.800	1.200
	6	38.400	9.600	2.400	0.600
	12	19.200	4.800	1.200	0.300
17.2032	7	38.400	9.600	2.400	0.600
	14	19.200	4.800	1.200	0.300
19.6608	2	153.600	38.400	9.600	2.400
	4	76.800	19.200	4.800	1.200
	8	38.400	9.600	2.400	0.600
	16	19.200	4.800	1.200	0.300
22.1184	9	38.400	9.600	2.400	0.600
24.5760	5	76.800	19.200	4.800	1.200
	10	38.400	9.600	2.400	0.600

Note: In I/O interface mode, the transfer rates are 8 times the values in this table.
In I/O interface mode, do not set the baud rate generator divisor to 1.

Table 3.9 (2) UART Mode Transfer Rate Setting Example (2) (Using Timer 2 Input Clock φT1)

Unit: Kbps

fc TREG2	24.576 MHz	12.288 MHz	12 MHz	9.8304 MHz	8 MHz	6.144 MHz
1H	192	96		76.8	62.5	48
2H	96	48		38.4	31.25	24
3H	64	32	31.25			16
4H	48	24		19.2		12
5H	38.4	19.2				9.6
8H	24	12		9.6		6
AH	19.2	9.6				4.8
10H	12	6		4.8		3
14H	9.6	4.8				2.4

(2) Data Format

Figure 3.9 (6) shows the data format for each mode.

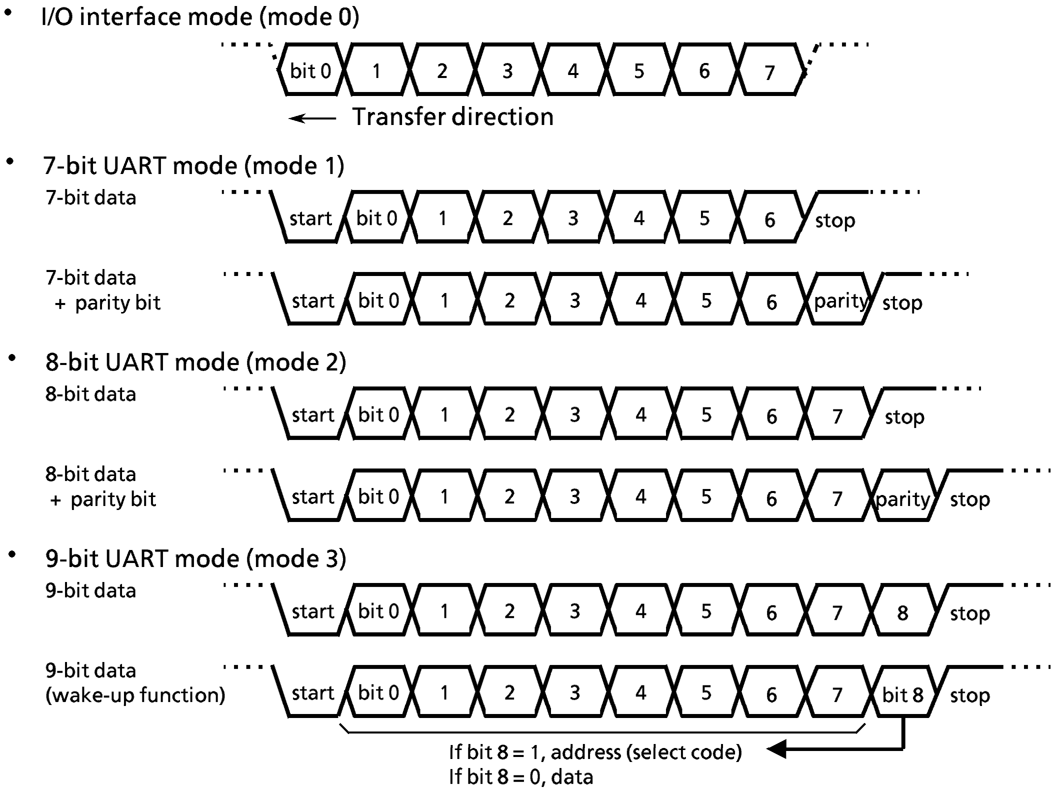


Figure 3.9 (6) Data Formats

(3) I/O interface mode (Mode 0)

In this mode, data transfer to an external device is synchronous with the transfer clock.
This mode is used to increase the number of I/O pins for sending or receiving data to an external shift register or other external destinations.
This mode consists of SCLK0 output mode, which outputs a synchronous clock (SCLK0), and SCLK0 input mode, which inputs a synchronous clock (SCLK0) from an external source.

Figures 3.9 (7) and (8) show connection examples of SCLK0 output and input modes.

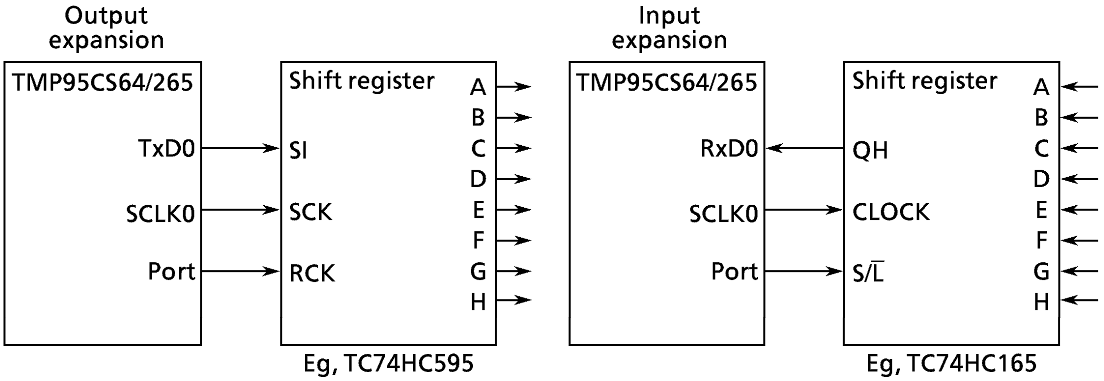


Figure 3.9 (7) Example of SCLK0 Output Mode Connection

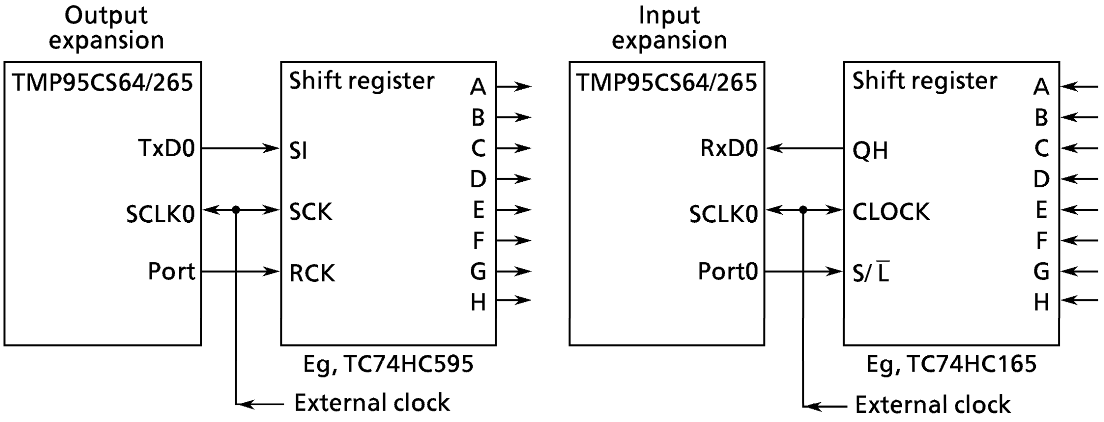


Figure 3.9 (8) Example of SCLK0 Input Mode Connection

① Sending

In SCLK0 output mode, each time the CPU writes data in the send buffer, eight data bits are output from the TxD0 pin, and a transfer clock signal is output from the SCLK0 pin. When all data have been sent, INTES0<ITX0C> is set, triggering an INTTX0 interrupt request.

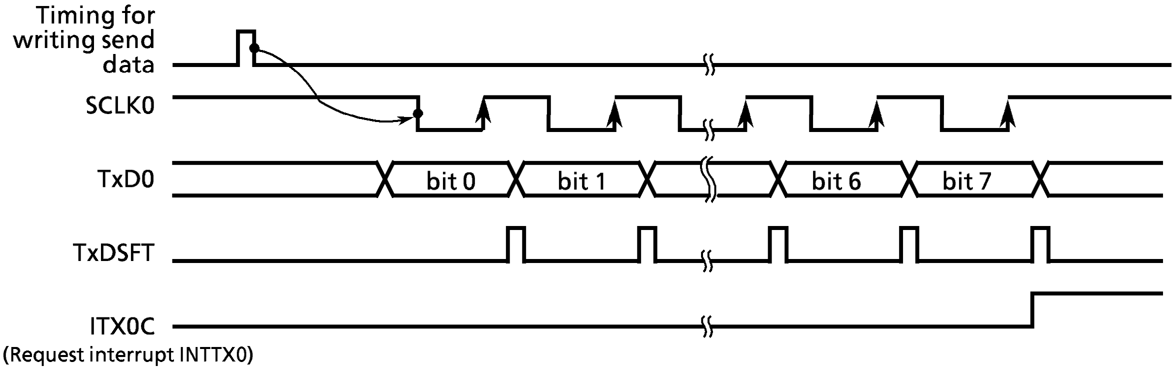


Figure 3.9 (9) Sending in I/O Interface Mode (SCLK0 Output Mode)

In SCLK0 input mode, pin TxD0 outputs eight transfer data bits when SCLK0 input is supplied and data are written to the send buffer by the CPU.
When all data have been sent, INTES0<ITX0C> is set, triggering an INTTX0 interrupt request.

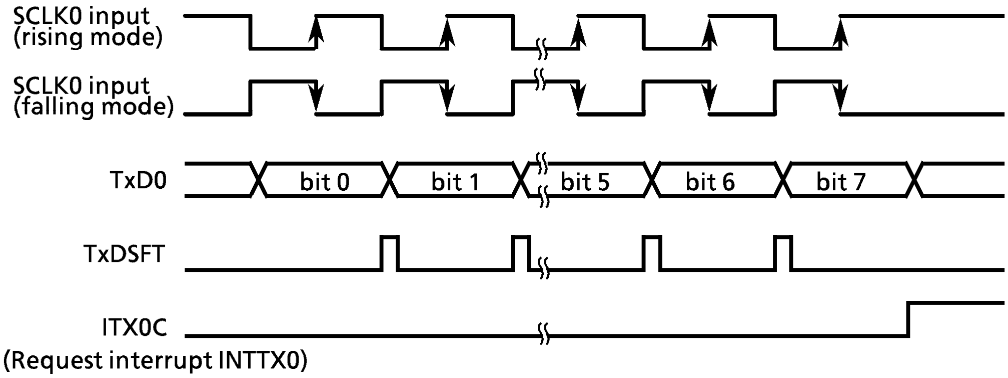

SCLK0 input (rising mode)
SCLK0 input (falling mode)
TxD0
bit 0
bit 1
bit 5
bit 6
bit 7
TxDSFT
ITX0C
(Request interrupt INTTX0)

Figure 3.9 (10) Sending in I/O Interface Mode (SCLK0 Input Mode)

② Receiving

In SCLK0 output mode, whenever the receive interrupt flag INTES0<IRX0C> is cleared by the CPU reading the received data, a synchronous clock is output from the SCLK0 pin and the next data frame is shifted to receive buffer 1. When an 8-bit data frame is received, it is transferred to receive buffer 2 (SC0BUF), and INTES0<IRX0C> is set again, triggering an INTRX0 interrupt request.

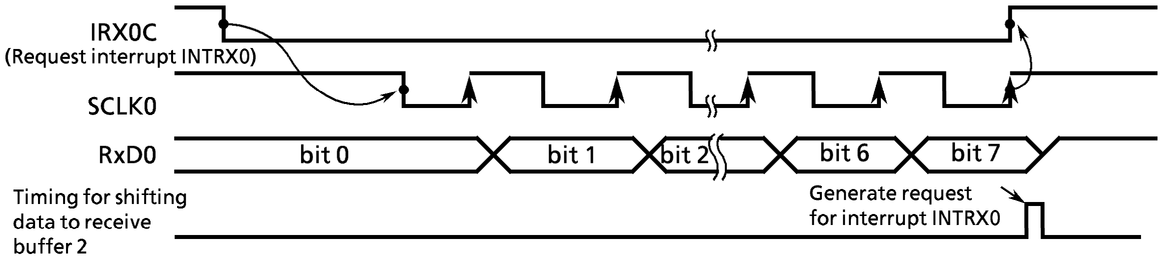

IRX0C
(Request interrupt INTRX0)
SCLK0
RxD0
bit 0
bit 1
bit 2
bit 6
bit 7
Timing for shifting data to receive buffer 2
Generate request for interrupt INTRX0

Figure 3.9 (11) Receiving in I/O Interface Mode (SCLK0 Output Mode)

In SCLK0 input mode, if SCLK0 input is supplied when received data are read by the CPU, thus clearing receive interrupt flag INTES0<IRX0C>, the next data frame is shifted into receive buffer 1. When an 8-bit data frame is received, it is shifted to receive buffer 2 (SC0BUF) and INTES0<IRX0C> is set again, triggering an INTRX0 interrupt request.

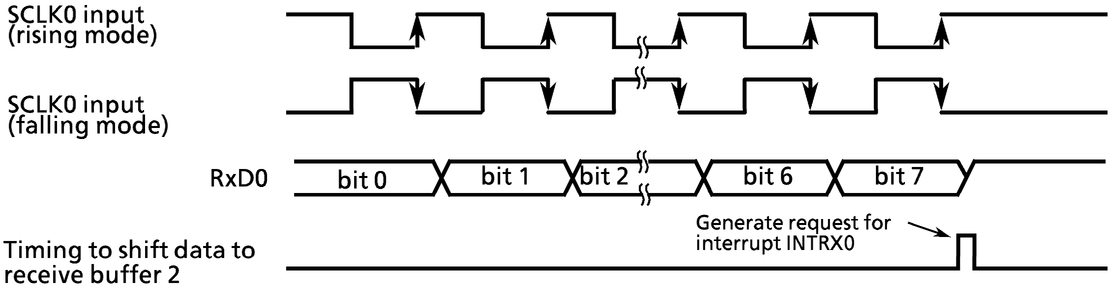

SCLK0 input (rising mode)
SCLK0 input (falling mode)
RxD0
bit 0
bit 1
bit 2
bit 6
bit 7
Generate request for interrupt INTRX0
Timing to shift data to receive buffer 2

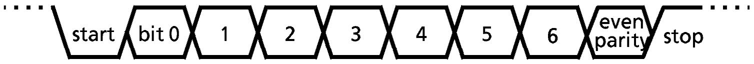
Figure 3.9 (12) Receiving in I/O Interface Mode (SCLK0 Input Mode)

Note: To receive data, first enable reception (set SC0MOD<RXE> to 1) for either SCLK0 input mode or output mode.

(4) 7-bit UART Mode (Mode 1)

Setting serial channel 0 mode control register SC0MOD<SM1:0> to 01 specifies 7-bit UART mode.
A parity bit may be added in this mode. Enable or disable the addition of a parity bit by serial channel 0 control register SC0CR<PE>.
With <PE> set to 1 (parity bit added), SC0CR<EVEN> selects even or odd parity.

Setting example: send 7-bit data with an even parity bit added:



← Transfer direction (transfer rate: 2400 bps @fc = 12.288 MHz)

```
         7 6 5 4 3 2 1 0
┌P8CR   ← - - - - - - - 1   } Select P80 as TxD0 pin.
 P8FC   ← X - - X - - X 1   }
 SC0MOD ← X 0 - X 0 1 0 1     Set 7-bit UART mode.
 SC0CR  ← X 1 1 X X X 0 0     Add even parity.
 BR0CR  ← 0 X 1 0 0 1 0 1     Set transfer rate to 2400bps.
 T16RUN ← 1 X - - - - - -     Start prescaler for baud rate generator.
 INTES0 ← 1 1 0 0 - - - -     Enable interrupt INTTX0 and set interrupt level to 4.
└SC0BUF ← * * * * * * * *     Set send data.
```

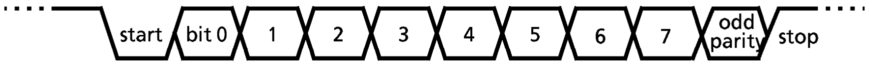
Note: X: Don't care   -: No change

(5) 8-bit UART Mode (Mode 2)

Setting serial channel 0 mode control register SC0MOD<SM1:0> to 10 selects 8-bit UART mode.

A parity bit may be added in this mode. Enable or disable the addition of a parity bit by serial channel 0 control register SC0CR<PE>. With <PE> set to 1 (parity bit added), SC0CR<EVEN> selects even or odd parity.

Setting example: send 8-bit data with an odd parity bit added:



← Transfer direction (transfer rate: 9600 bps @fc = 12.288 MHz)

Main routine settings:

```
            7 6 5 4 3 2 1 0
 P8CR    ←  - - - - - - 0 -     Select P81 (RxD0) as input pin.
 SC0MOD  ←  - 0 1 X 1 0 0 1     Set 8-bit UART mode and enable reception.
 SC0CR   ←  X 0 1 X X X 0 0     Add odd parity.
 BR0CR   ←  0 X 0 1 0 1 0 1     Set transfer rate to 9600 bps.
 T16RUN  ←  1 X - - - - - -     Start the prescaler for baud rate generator.
 INTES0  ←  - - - - 1 1 0 0     Enable interrupt INTRX0 and set interrupt level 4.
```

Note: X: Don't care -: No change

Interrupt routine processing example:

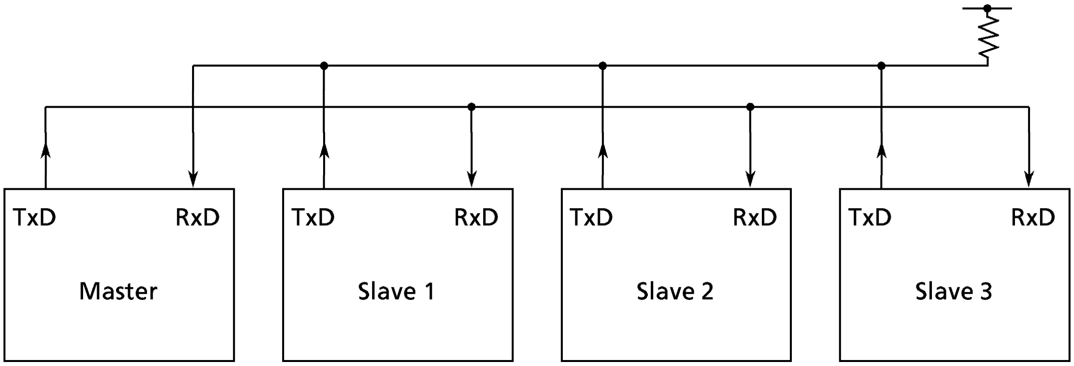
Check for errors with SC0CR error flags (<OERR>, <PERR>, <FERR>). If there are no errors, read the data received.

(6) 9-bit UART Mode (Mode 3)

Setting the serial channel 0 mode control register SC0MOD<SM1:0> to 11 selects 9-bit UART mode.
A parity bit cannot be added in this mode.
When sending, the most significant bit (bit 9) is written to SC0MOD<TB8>.
When receiving, the most significant bit is saved in serial channel control register SC0CR<RB8>.
When the buffer is written to or read from, the most significant bit is always read or written first.

Wake-Up Function

In 9-bit UART mode, select the slave controller wake-up function by setting SC0MOD<WU> to 1. When SC0CR<RB8> = 1, received data are interpreted as select code, and an INTRX0 interrupt request occurs.



Note: The TxD pin of the slave controller must always be set to open-drain output mode using the ODE register.

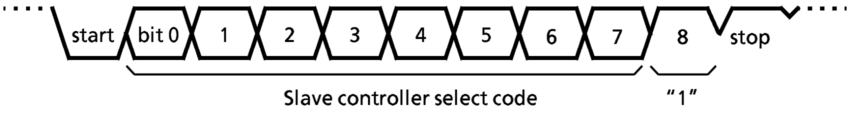
Figure 3.9 (13) Serial Link with Wake-Up Function

Protocol

① Set the master controller and all slave controllers to 9-bit UART mode.

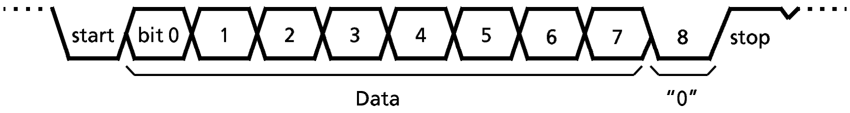
② Set the serial channel 0 mode control register SC0MOD<WU> of each slave controller to 1 to enable data reception.

③ The master controller sends one frame with the most significant bit (bit 8) SC0MOD<TB8> set to 1. This frame contains the 8-bit select code of a slave controller.



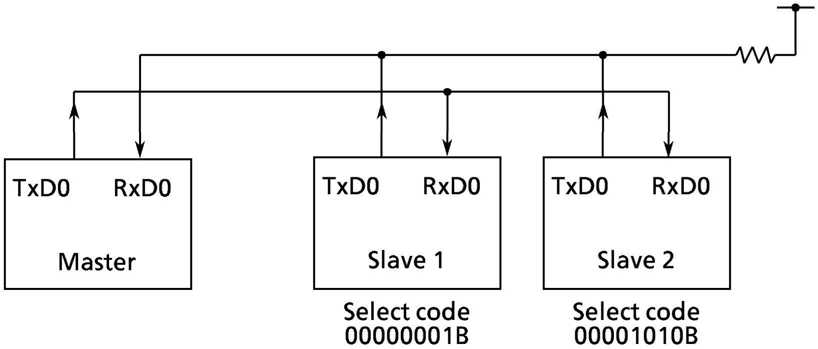
④ The slave controllers receive the above data frame. The slave controller whose select code matches the select code in the data frame received clears its SC0MOD<WU> bit to 0.

⑤ The master controller sends data frames with their most significant bit (bit 8) SC0MOD<TB8> set to 0 to the specified slave controller (the controller whose SC0MOD<WU> bit is cleared to 0).



⑥ The slave controllers whose SC0MOD<WU> bit is 1 ignore the received data as interrupt INTRX0 is not generated when the most significant bit (bit 8) SC0CR<RB8> remains cleared to 0 (when data are sent).
The slave controller whose SC0MOD<WU> bit is cleared to 0 can inform the master controller of the termination of a send it received by sending data to the master controller.

Setting example: When linking two slave controllers serially with the master controller using internal clock ϕ1 as the transfer clock.



As serial channels 0, 1, and 2 operate identically in this mode, the following describes channel 0 only.

- Setting the master controller

Main routine:

```
P8CR   ← - - - - - - 0 1  }  Select P80 as TxD0 pin, and P81 as RxD0 pin.
P8FC   ← X - - X - - X 1
INTES0 ← 1 1 0 0 1 1 0 1    Enable interrupt INTTX0 and set interrupt level to 4.
                            Enable interrupt INTRX0 and set interrupt level to 5.
SC0MOD ← 1 0 1 0 1 1 1 0    Set ϕ1 as transfer clock and set 9-bit UART mode.
SC0BUF ← 0 0 0 0 0 0 0 1    Set select code for slave controller 1.
```

INTTX0 interrupt routine:

```
SC0MOD ← 0 - - - - - - -    Set SC0MOD<TB8> to 0.
SC0BUF ← * * * * * * * *    Set send data.
```

Note: X: Don't care -: No change

- Setting slave controller 2

Main routine:

```
P8CR   ← - - - - - - 0 1  \
P8FC   ← X - - X - - X 1   > Select P80 as TxD0 pin (open-drain output), and P81 as RxD0
ODE    ← X X X X X - - 1  /  pin.
INTES0 ← 1 1 0 1 1 1 1 0     Enable interrupts INTTX0 and INTRX0.
SC0MOD ← 0 0 1 1 1 1 1 0     Set 9-bit UART mode using transfer clock ϕ1 (2/fc), and enable
                             wake-up mode (set <WU> to 1).
```

INTRX0 interrupt routine:

Compare SC0BUF and select code (00001010B). If these match, clear SC0MOD<WU> to 0.

Note: X: Don't care -: No change

(7) Signal Generation Timing

① In I/O Interface mode

Timing for send interrupt generation	SCLK0 output mode	Immediately after rise of last SCLK0 signal (See Figure 3.9 (9))
	SCLK0 input mode	Immediately after rise (rising mode) or fall (falling mode) of last SCLK0 signal (See Figure 3.9 (10).)
Timing for receive interrupt generation	SCLK0 output mode	Immediately after final SCLK0 (When received data are transferred to receive buffer 2 (SC0BUF)) (See Figure 3.9 (11).)
	SCLK0 input mode	Immediately after final SCLK0 (When received data are transferred to receive buffer 2 (SC0BUF)) (See Figure 3.9 (12).)

② In UART mode

Receive

Mode	9-Bit	8-Bit + Parity	8-Bit, 7-Bit + Parity, 7-Bit
Timing for interrupt generation	Around center of bit 8	Around center of parity bit	Around center of stop bit
Timing for framing error generation	Around center of stop bit	Around center of stop bit	Around center of stop bit
Timing for parity error generation	————	Around center of parity bit	←
Timing for overrun error generation	Around center of bit 8	Around center of parity bit	Around center of stop bit

Send

Mode	9-Bit	8-Bit + Parity	8-Bit, 7-Bit + Parity, 7-Bit
Timing for interrupt generation	Immediately before stop bit sent	←	←

3.10 Analog / Digital Converter

TMP95CS64/265 incorporates a high-speed, high-precision 10-bit successive approximation-type analog/digital converter (A/D converter) with 8-channel analog input.
Figure 3.10 (1) is a block diagram of the A/D converter. The 8-channel analog input pins (AN0 to AN7) are shared by input-only port A and can thus be used as an input port.

Note: When the power is reduced by setting IDLE2, IDLE1, or STOP mode, with some timings, the system may enter standby mode even though the internal comparator is still enabled. Therefore, be sure to check that A/D converter operations are halted before executing a HALT instruction.

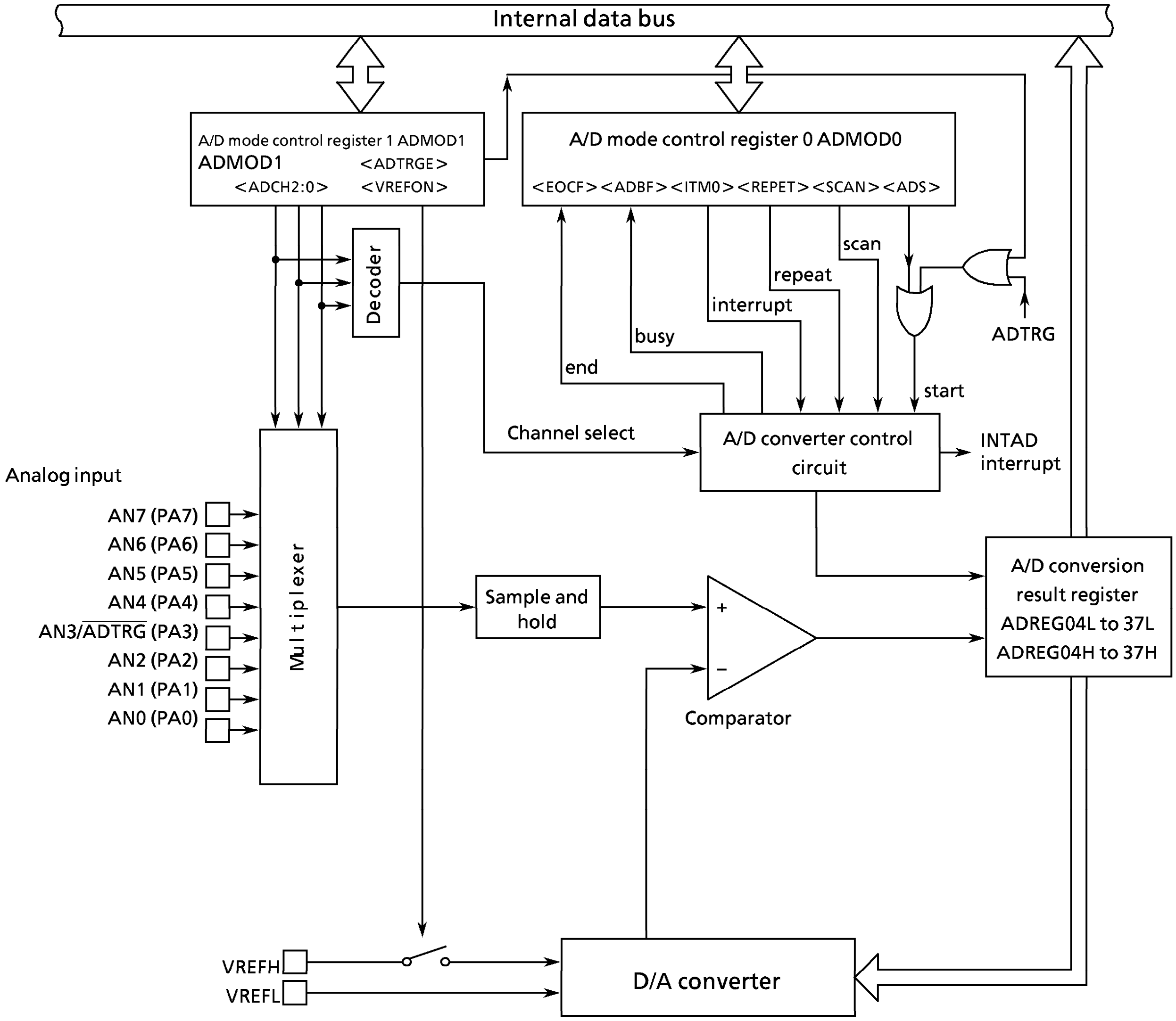


Figure 3.10 (1) A/D Converter Block Diagram

3.10.1 Analog / Digital Converter Registers

The A/D converter is controlled by two A/D mode control registers: ADMOD0 and ADMOD1. Eight A/D conversion data upper and lower registers (ADREG04H/L, ADREG15H/L, ADREG26H/L, and ADREG37H/L) store the A/D conversion results.
Figures 3.10 (2) shows registers related to the A/D converter.

A/D Mode Control Register 0

ADMOD0
(005EH)

	7	6	5	4	3	2	1	0
bit Symbol	EOCF	ADBF	–	–	ITM0	REPET	SCAN	ADS
Read/Write	R		R/W					
After reset	0	0	0	0	0	0	0	0
Function	A/D conversion end flag 0: Conversion in progress 1: Conversion complete	A/D conversion busy flag 0: Conversion stopped 1: Conversion in progress	Note: Always fixed to 0.	Note: Always fixed to 0.	Interrupt specification in conversion channel fixed repeat mode 0: Every conversion 1: Every fourth conversion	Repeat mode specification 0: Single conversion mode 1: Repeat conversion mode	Scan mode specification 0: Conversion channel fixed mode 1: Conversion channel scan mode	A/D conversion start 0: Don't care 1: Start conversion Always read as 0.

A/D conversion start

0	Don't care
1	Start A/D conversion

Note: Always read as 0.

A/D scan mode setting

0	A/D conversion channel fixed mode
1	A/D conversion channel scan mode

A/D repeat mode setting

0	A/D single conversion mode
1	A/D repeat conversion mode

Specify A/D conversion interrupt for channel fixed repeat conversion mode

	Channel fixed repeat conversion mode <SCAN> = 0, <REPET> = 1
0	Generate interrupt every conversion
1	Generate interrupt every fourth conversion

A/D conversion busy flag

0	A/D Conversion stopped
1	A/D conversion in progress

A/D conversion end flag

0	Before or during A/D conversion
1	A/D conversion complete

Figure 3.10 (2)-1 A/D Converter Related Register

A/D Mode Control Register 1

ADMOD1
(005FH)

	7	6	5	4	3	2	1	0
bit Symbol	VREFON				ADTRGE	ADCH2	ADCH1	ADCH0
Read/Write	R/W				R/W			
After reset	1				0	0	0	0
Function	VREF application control 0: OFF 1: ON				A/D external trigger start control 0: Disable 1: Enable	Analog input channel selection		

Analog input channel selection

<SCAN> <ADCH2,1,0>	0 (Channel fixed)	1 (Channel scan)
000	AN0	AN0
001	AN1	AN0→AN1
010	AN2	AN0→AN1→AN2
011 (Note)	AN3	AN0→AN1→AN2→AN3
100	AN4	AN4
101	AN5	AN4→AN5
110	AN6	AN4→AN5→AN6
111	AN7	AN4→AN5→AN6→AN7

A/D conversion start control by external trigger ($\overline{\text{ADTRG}}$ input)

0	Disable
1	Enable

Control of application of reference voltage to A/D converter

0	OFF
1	ON

Before starting conversion (before writing 1 to ADMOD0<ADS>), set the <VREFON> bit to 1.

Note: As pin AN3 also functions as the $\overline{\text{ADTRG}}$ input pin, do not set <ADCH2 to 0> = 011 when using $\overline{\text{ADTRG}}$ with <ADTRGE> set to 1.

Figure 3.10 (2)-2 A/D Converter Related Register

A/D Conversion Data Lower Register 0/4

		7	6	5	4	3	2	1	0
ADREG04L (0060H)	bit Symbol	ADR01	ADR00						ADR0RF
	Read/Write	R							R
	After reset	Undefined							0
	Function	Stores lower 2 bits of A/D conversion result							A/D conversion data storage flag 1: Conversion result stored

A/D Conversion Data Upper Register 0/4

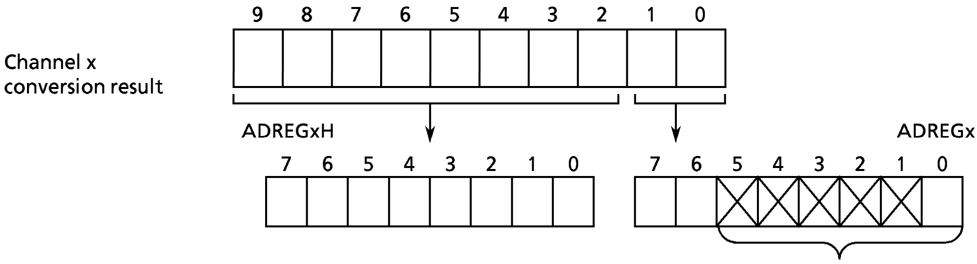
		7	6	5	4	3	2	1	0
ADREG04H (0061H)	bit Symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
	Read/Write	R							
	After reset	Undefined							
	Function	Stores upper eight bits of A/D conversion result.							

A/D Conversion Data Lower Register 1/5

		7	6	5	4	3	2	1	0
ADREG15L (0062H)	bit Symbol	ADR11	ADR10						ADR1RF
	Read/Write	R							R
	After reset	Undefined							0
	Function	Stores lower 2 bits of A/D conversion result							A/D conversion result flag 1: Conversion result stored

A/D Conversion Data Upper Register 1/5

		7	6	5	4	3	2	1	0
ADREG15H (0063H)	bit Symbol	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
	Read/Write	R							
	After reset	Undefined							
	Function	Stores upper eight bits of A/D conversion result.							



- Bits 5 to 1 are always read as 1.
- Bit 0 is the A/D conversion data storage flag <ADRxRF>. When the A/D conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.10 (2)-3 A/D Converter Related Registers

A/D Conversion Result Lower Register 2/6

		7	6	5	4	3	2	1	0
ADREG26L (0064H)	bit Symbol	ADR21	ADR20						ADR2RF
	Read/Write	R							R
	After reset	Undefined							0
	Function	Stores lower 2 bits of A/D conversion result							A/D conversion data storage flag 1: Conversion result stored

A/D Conversion Data Upper Register 2/6

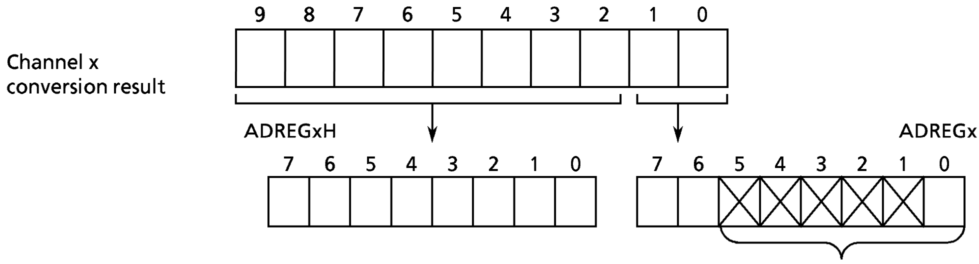
		7	6	5	4	3	2	1	0
ADREG26H (0065H)	bit Symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
	Read/Write	R							
	After reset	Undefined							
	Function	Stores upper eight bits of A/D conversion result.							

A/D Conversion Data Lower Register 3/7

		7	6	5	4	3	2	1	0
ADREG37L (0066H)	bit Symbol	ADR31	ADR30						ADR3RF
	Read/Write	R							R
	After reset	Undefined							0
	Function	Stores lower 2 bits of A/D conversion result							A/D conversion data storage flag 1: Conversion result stored

A/D Conversion Result Upper Register 3/7

		7	6	5	4	3	2	1	0
ADREG37H (0067H)	bit Symbol	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
	Read/Write	R							
	After reset	Undefined							
	Function	Stores upper eight bits of A/D conversion result.							



- Bits 5 to 1 are always read as 1.
- Bit 0 is the A/D conversion data storage flag <ADRxRF>. When the A/D conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.10 (2)-4 A/D Converter Related Registers

3.10.2 Description of Operation

(1) Analog Reference Voltage

A high level analog reference voltage is applied to the VREFH pin; a low level analog reference voltage to the VREFL pin. To perform A/D conversion, the reference voltage, the difference between VREFH and VREFL, is divided by 1024 using string resistance. Then, the result of the division is compared with the analog input voltage.
To turn off the switch between VREFH and VREFL, write 0 to A/D mode control register 1 ADMOD1<VREFON>. To start A/D conversion from the off state, first write 1 to <VREFON>, wait 3 μs until the internal reference voltage stabilizes (not related to the fc), then write 1 to A/D mode register ADMOD0<ADS>.

(2) Analog input channel selection

The analog input channel selection varies according to the operating mode of the A/D converter.

- In analog input channel fixed mode (ADMOD0<SCAN>=0)
  Setting ADMOD1<ADCH2 to 0> selects one channel among analog input pins AN0 to AN7.
- In analog input channel scan mode (ADMOD0<SCAN>=1)
  Setting ADMOD1<ADCH2 to 0> selects one scan mode from among eight scan modes.

Table 3.10 (1) shows the analog input channel selection for each operating mode.

After a reset, ADMOD0<SCAN> is set to 0 and ADMOD1<ADCH2 to 0> is initialized to 000, thus selecting pin AN0 as the channel fixed input. Pins not used as analog input channels can be used as standard input ports.

Table 3.10 (1) Analog Input Channel Selection

<ADCH2~0>	Channel fixed <SCAN> = "0"	Channel scan <SCAN> = "1"
000	AN0	AN0
001	AN1	AN0→AN1
010	AN2	AN0→AN1→AN2
011	AN3	AN0→AN1→AN2→AN3
100	AN4	AN4
101	AN5	AN4→AN5
110	AN6	AN4→AN5→AN6
111	AN7	AN4→AN5→AN6→AN7

(3) Starting A/D Conversion

To start A/D conversion, write 1 to A/D mode control register 0 ADMOD0<ADS> or A/D mode control register 1 ADMOD1<ADTRGE> and input a falling edge on the $\overline{\text{ADTRG}}$ pin. When A/D conversion starts, the A/D conversion busy flag ADMOD0<ADBF> is set to 1, indicating A/D conversion is in progress.
Writing 1 to <ADS> during A/D conversion restarts conversion. At that time, to determine whether the A/D conversion results are preserved, check the conversion data storage flag ADREGxL<ADRxRF>.
During A/D conversion, inputting a falling edge to the $\overline{\text{ADTRG}}$ pin is ignored.

(4) A/D conversion modes and A/D conversion end interrupt

The four A/D conversion modes are:

- Channel fixed single conversion mode
- Channel scan single conversion mode
- Channel fixed repeat conversion mode
- Channel scan repeat conversion mode

A/D mode control register 0 ADMOD0<REPET>, <SCAN> selects the A/D mode.
Completion of A/D conversion triggers the A/D conversion end INTAD interrupt request. Also, ADMOD0<EOCF> is set to 1 to indicate that A/D conversion is complete.

① Channel fixed single conversion mode

Setting ADMOD0<REPET>, <SCAN> to 00 sets conversion channel fixed single conversion mode.
In this mode, one specified channel is converted once only. When the conversion is complete, the ADMOD0<EOCF> flag is set to 1, ADMOD0<ADBF> is cleared to 0, and an INTAD interrupt request is generated.

② Channel scan single conversion mode

Setting ADMOD0<REPET>, <SCAN> to 01 sets conversion channel scan single conversion mode.
In this mode, the specified scan channels are converted once only. When scan conversion is complete, ADMOD0<EOCF> is set to 1, ADMOD0<ADBF> is cleared to 0, and an INTAD interrupt request is generated.

③ Channel fixed repeat conversion mode

Setting ADMOD0<REPET>, <SCAN> to 10 sets conversion channel fixed repeat conversion mode. In this mode, one specified channel is converted repeatedly. When conversion is complete, ADMOD0<EOCF> is set to 1 and ADMOD0<ADBF> is not cleared to 0 but held at 1. The INTAD interrupt request generation timing is selected by ADMOD0<ITM0>.
Setting <ITM0> to 0 generates an interrupt request when every A/D conversion completes.
Setting <ITM0> to 1 generates an interrupt request when every fourth conversion completes.

④ Channel scan repeat conversion mode

Setting ADMOD0<REPET>, <SCAN> to 11 sets conversion channel scan repeat conversion mode. In this mode, the specified scan channels are converted repeatedly. When each scan conversion completes, ADMOD0<EOCF> is set to 1 and an INTAD interrupt request is generated. ADMOD0<ADBF> is not cleared to 0 but held at 1.
To stop conversion in a repeat conversion mode (mode ③ or ④), write 0 to ADMOD0<REPET>. After the current conversion is complete, the repeat conversion mode terminates and ADMOD0<ADBF> is cleared to 0.
Switching to a halt state (IDLE2, IDLE1, or STOP) immediately stops the A/D converter even with A/D conversion still in progress. In repeat conversion modes (modes ③ and ④), after the halt is released, conversion restarts from the beginning. In single conversion modes (modes ① and ②), conversion does not restart (the converter remains stopped).

Table 3.10 (2) shows the relationship between A/D conversion modes and interrupt requests.

Table 3.10 (2) Relationship Between A/D Conversion Modes and Interrupt Requests

Mode	Interrupt request generation	ADMOD0		
		<ITM0>	<REPET>	<SCAN>
Channel fixed single conversion mode	After completion of conversion	X	0	0
Channel scan single conversion mode	After completion of scan conversion	X	0	1
Channel fixed repeat conversion mode	Every conversion	0	1	0
	Every fourth conversion	1		
Channel scan repeat conversion mode	After completion of every scan conversion	X	1	1

X: Don't care

(5) A/D conversion time

84 states (6.72 μs @fc = 25 MHz) are required for A/D conversion of one channel.

(6) Storing and reading A/D conversion result

The A/D conversion data upper and lower registers (ADREG04H/L to ADREG37H/L) store the A/D conversion results. (ADREG04H/L to ADRG37H/L are read-only registers.)
In channel fixed repeat conversion mode, the conversion results are stored successively in registers ADREG04H/L to ADRG37H/L. In other modes, the AN0 and AN4, AN1 and AN5, AN2 and AN6, and AN3 and AN7 conversion results are stored in ADREG04H/L, ADREG15H/L, ADREG26H/L, and ADREG37H/L respectively.
Table 3.10 (3) shows the correspondence between analog input channels and A/D conversion result registers.

Table 3.10 (3) Correspondence Between Analog Input Channels and A/D Conversion Result Registers

Analog input channel (port A)	A/D conversion result register	
	Conversion modes other than at right	Channel fixed repeat conversion mode (every 4th conversion)
AN0	ADREG04H/L	ADREG04H/L ↓ ADREG15H/L ↓ ADREG26H/L ↓ ADREG37H/L
AN1	ADREG15H/L	
AN2	ADREG26H/L	
AN3	ADREG37H/L	
AN4	ADREG04H/L	
AN5	ADREG15H/L	
AN6	ADREG26H/L	
AN7	ADREG37H/L	

The A/D conversion data storage flag <ADRxRF> uses bit 0 of the A/D conversion data lower register. The storage flag indicates whether the A/D conversion result register was read or not. When a conversion result is stored in the A/D conversion result register the flag is set to 1. When either of the A/D conversion result registers (ADREGxH or ADREGxL) is read the flag is cleared to 0.
Reading the A/D conversion result also clears the A/D conversion end flag ADMOD0<EOCF> to 0.

Setting example:

① Convert the analog input voltage at the AN3 pin and write the result, using the A/D interrupt (INTAD) processing routine, to memory address 0800H.

Main routine setting:

```
           7 6 5 4 3 2 1 0
INTE0AD ←  1 1 0 0 0 0 0 0     Enable INTAD and set level to 4.
ADMOD1  ←  1 X X X 0 0 1 1     Set analog input channel to pin AN3.
ADMOD0  ←  X X 0 0 0 0 0 1     Start conversion in channel fixed single conversion mode.
```

Interrupt routine processing example:

```
WA      ← ADREG37              Read value of ADREG37L and ADREG37H to general-
                               purpose register WA (16 bits).
WA      >> 6                   Shift contents read in WA six times to right and zero-fill
                               upper bits.
(0800H) ← WA                   Write contents of WA to memory address 0800H.
```

② This example repeatedly converts the analog input voltages at the three pins AN0 to AN2, using channel scan repeat conversion mode.

```
INTE0AD ←  1 0 0 0 0 0 0 0     Disable INTAD.
ADMOD1  ←  1 X X X 0 0 1 0     Set pins AN0 to AN2 as analog input channels.
ADMOD0  ←  X X 0 0 0 1 1 1     Start conversion in channel scan repeat conversion mode.
```

Note: X: Don’t care -: No change

3.11 Digital/Analog Converter

TMP95CS64/265 incorporates a 2-channel, 8-bit resolution digital/analog converter with the following features.

- An R-2R-type 8-bit resolution D/A converter with two internal channels.
- The output analog voltage is determined by the potential difference between AVCC and AVSS and by the value set in D/A conversion registers DAREG0 and DAREG1.

Figure 3.11 (1) is a block diagram of the D/A converter.

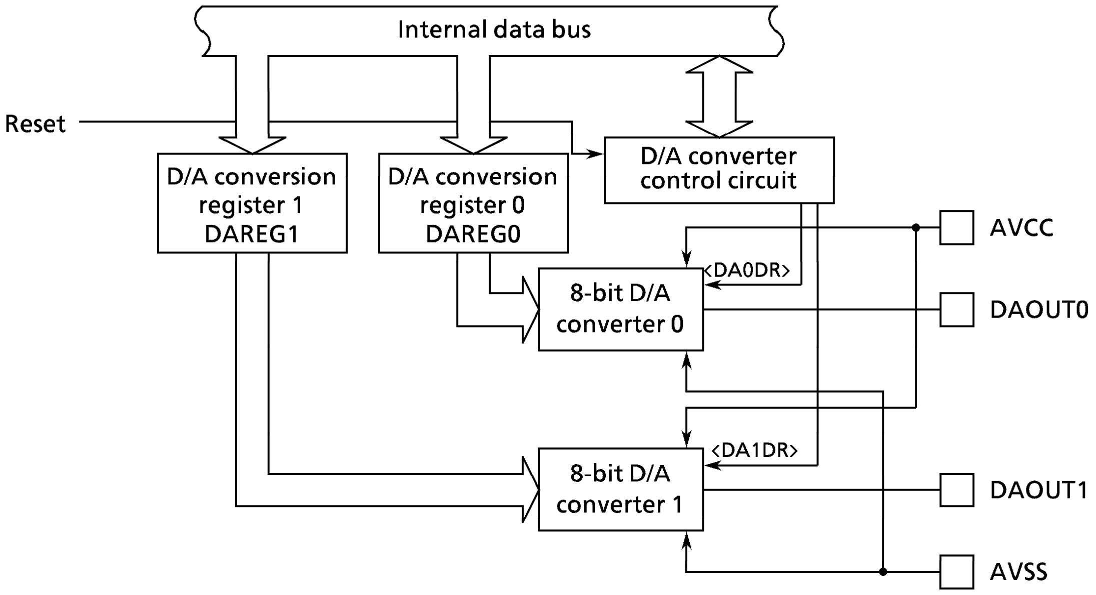


Figure 3.11 (1) D/A Converter Block Diagram

3.11.1 Digital/Analog Converter Registers

The D/A converter is controlled by the D/A conversion drive register DADRV and two D/A conversion value setting registers DAREG1 and DAREG2.
Figure 3.11 (2) shows the D/A converter related registers.

D/A Conversion Drive Register

DADRV
(009DH)

	7	6	5	4	3	2	1	0
bit Symbol							DA1DR	DA0DR
Read/Write							R/W	
After reset							0	0
Function							Output pin DAOUT1 drive specification	Output pin DAOUT0 drive specification
							0: Fixed to 0 V output 1: D/A conversion result output	

D / A Conversion Value Setting Register 0

DAREG0
(009EH)

Read-modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol	—							
Read/Write	W							
After reset	Undefined							
Function	Set D/A converter 0 conversion value N Output voltage V = (AV_{CC} - AV_{SS}) × N/256							

D / A Conversion Value Setting Register 1

DAREG1
(009FH)

Read-modify-write instructions prohibited.

	7	6	5	4	3	2	1	0
bit Symbol	—							
Read/Write	W							
After reset	Undefined							
Function	Set D/A converter 1 conversion value N Output voltage V = (AV_{CC} - AV_{SS}) × N/256							

Figure 3.11 (2) D/A Converter Related Registers

3.11.2 Description of Operation

The analog voltage output by the D/A converter is expressed by the following formula:

Analog voltage = (AVCC - AVSS) x N/256

Here, "N" is the value (0 to 255) set in the D/A conversion value setting register DAREG0 or DAREG1. The channel 0 and 1 D/A conversion results are output from the DAOUT0 and DAOUT1 pins respectively.
Bits 1 and 0 of the D/A conversion drive register DADRV<DA1DR>, <DA0DR> are the drive bits of the DAOUT1 and DAOUT0 pins respectively. Setting <DA1DR>, <DA0DR> to 0 fixes the DAOUT1:0 pins to 0 voltage. Setting to 1 sets the DAOUT1:0 pins to D/A conversion result output pins.
As a reset clears the D/A conversion drive register DADRV<DA1DR>, <DA0DR> to 0, the DAOUT1:0 pins output 0V. When performing D/A conversion following a reset, the contents of DAREG0 and DAREG1 are undefined. Therefore, be sure to set "N" first then <DA1DR>, <DA0DR> to 1.
Also, once D/A conversion has started, write "N" as required to output the desired analog voltage. There is no need to clear <DA1DR>, <DA0DR> when rewriting "N".
Also, in STOP mode, the DAOUT0:1 pins output 0V regardless of the DADRV or DAREG setting.

Setting example: After a reset, output from the DAOUT1 pin VCC and VCC/2 consecutively (set to AVCC = VCC, AVSS = GND):

		7	6	5	4	3	2	1	0		
DAREG1	←	1	1	1	1	1	1	1	1	Write FFH.	$\text{DAOUT1} = \text{Vcc} \times \frac{255}{256} \fallingdotseq \text{Vcc}$
DADRV	←	X	X	X	X	X	X	1	-	Output DAOUT1.	
DAREG1	←	1	0	0	0	0	0	0	0	Write 80H.	Output $\frac{\text{Vcc}}{2}$ on DAOUT1.

Note: X: Don't care -: No change

3.12 Watchdog Timer (Runaway Detection Timer)

TMP95CS64/265 incorporates a watchdog timer for detecting a runaway (out-of-control) condition.
The watchdog timer (WDT) returns the CPU to its normal state when it detects the start of a CPU runaway due to, for example, noise. When the watchdog timer detects a runaway, it generates an INTWD (non-maskable) interrupt to notify the CPU of the condition.
In addition, the runaway detection result can be used for a forcible reset of the microcontroller itself.
The watchdog timer consists of a 22-step binary counter with 2/fc as the input clock, and a control block.
Figure 3.12 (1) is a block diagram of the watchdog timer (WDT).

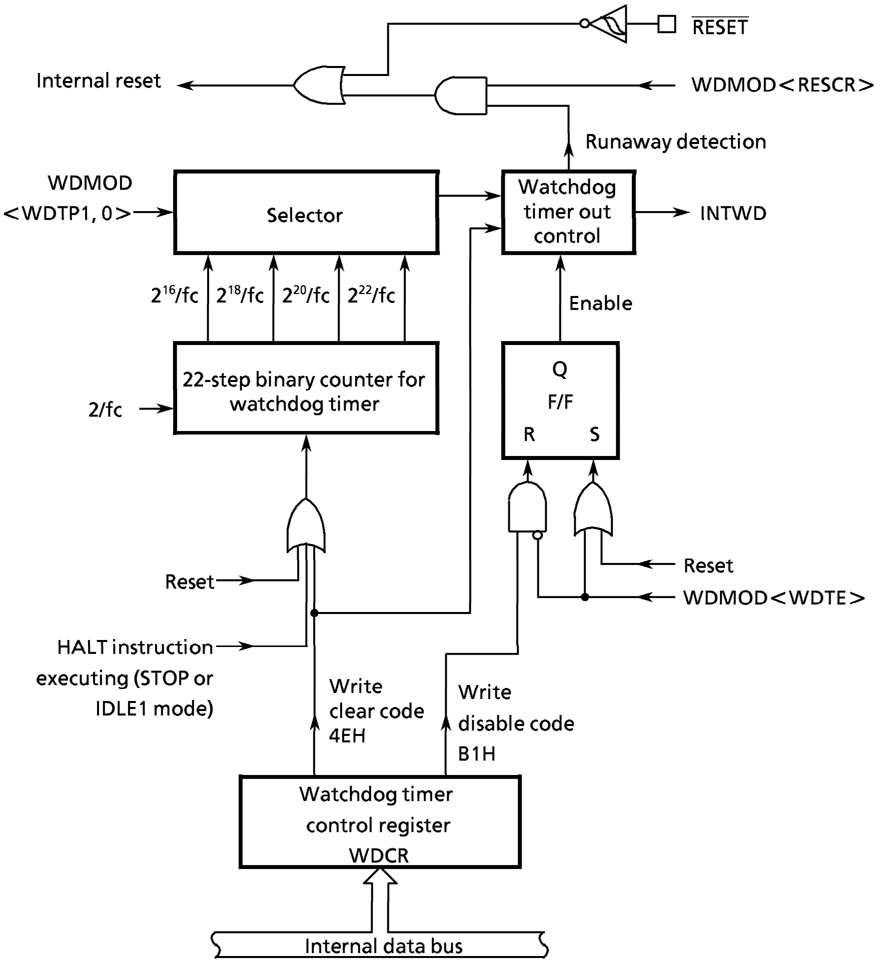


Figure 3.12 (1) Watchdog Timer Block Diagram

3.12.1 Watchdog Timer Registers

The watchdog timer (WDT) is controlled by two control registers. Figure 3.12 (2) shows watchdog timer mode control register WDMOD and watchdog timer control register WDCR.

Watchdog Timer Mode Control Register

WDMOD
(006EH)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
Read/Write	R/W							
After reset	1	0	0	0	0	0	0	0
Function	Watchdog timer control 0: Disable 1: Enable	Watchdog timer detection time selection 00: 2^{16}/fc 01: 2^{18}/fc 10: 2^{20}/fc 11: 2^{22}/fc		Warm-up time 0: 2^{14}/fc 1: 2^{16}/fc	HALT mode selection 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		Internal reset control at runaway detection 1: Perform internal reset by runaway detection	Pin control in STOP mode 1: Drive pins in STOP mode

(See 3.4, Standby Function)

Internal reset control at runaway detection

0	Don't care
1	Perform internal reset by runaway detection

Watchdog timer detection time selection

00	2^{16}/fc (approximately 2.6 ms @ 25 MHz)
01	2^{18}/fc (approximately 10.5 ms @ 25 MHz)
10	2^{20}/fc (approximately 41.9 ms @ 25 MHz)
11	2^{22}/fc (approximately 168 ms @ 25 MHz)

Watchdog timer enable/disable control

0	Disable
1	Enable

Watchdog Timer Control Register

WDCR
(006FH)

	7	6	5	4	3	2	1	0
bit Symbol	–							
Read/Write	W							
After reset	–							
Function	B1H: WDT disable code 4EH: WDT clear code							

Watchdog timer disable and clear

B1H	Disable code
4EH	Clear code
Others	Don't care

Figure 3.12 (2) Watchdog Timer Related Registers

(1) Watchdog timer mode control register (WDMOD)

① Setting watchdog timer detection time <WDTP1: 0>

This 2-bit register is used to set the watchdog timer interrupt time for detecting a runaway. After a reset, WDMOD <WDTP1: 0> is set to 00, which sets a detection time of 2^{16}/fc [s]. (The number of states is approximately 32,768.)

② Watchdog timer enable/disable control <WDTE>

After a reset, WDMOD<WDTE> is initialized to 1, enabling the watchdog timer.
Disabling the watchdog timer requires both clearing this bit to 0 and writing the disable code B1H in watchdog timer control register WDCR. This two-step process is an insurance against an out-of-control system disabling the watchdog timer.
To return from disable state to enable state, simply set <WDTE> to 1.

③ Runaway detection time internal reset control <RESCR>

This register determines whether or not the watchdog timer resets itself on detection of a runaway. Setting WDMOD <RESCR> to 1 forcibly resets the microcontroller after detection of a runaway. On reset, <RESCR> is initialized to 0. Therefore, detection of a runaway will not trigger an internal reset. In such a case, the watchdog timer holds the runaway detection state until the clear code is written to WDCR.

(2) Watchdog timer control register WDCR

This register is used to disable the watchdog timer functions and to clear the binary counter.

- Disable control

  After clearing WDMOD<WDTE> to 0, write the disable code B1H to WDCR to disable the watchdog timer.

```
          7 6 5 4 3 2 1 0
┌WDMOD  ← 0 - - - - - X X      Clear <WDTE> to 0.
└WDCR   ← 1 0 1 1 0 0 0 1      Write disable code B1H.
```

  Note: X: Don't care – : No change

- Watchdog timer clear control

  Writing clear code 4EH to WDCR clears the binary counter and resumes the count.

```
 WDCR   ← 0 1 0 0 1 1 1 0      Write clear code 4EH.
```

3.12.2 Description of Operation

After the detection time set by the watchdog timer mode register WDMOD <WDTP1: 0> is reached, the watchdog timer generates interrupt INTWD. The watchdog timer detection time can be selected from 2^{16}f/c, 2^{18}f/c, 2^{20}f/c, and 2^{22}f/c. The binary counter for the watchdog timer must be cleared to 0 by software (by instruction) before the INTWD interrupt is generated. If the CPU malfunctions (is out of control) due to factors such as noise and does not execute an instruction to clear the binary counter, the binary counter overflows and generates interrupt INTWD. The CPU interprets the INTWD interrupt as a malfunction (runaway condition) detection signal, which can be used to start program-based anti-malfunction measures to return the system to normal (normal mode).

Runaway detection can also be used for an internal reset (reset mode). To perform an internal reset by runaway detection, first set WDMOD <RESCR> to 1.

The INTWD interrupt generation cycle is twice the watchdog timer detection time selected by <WDTP1: 0>.

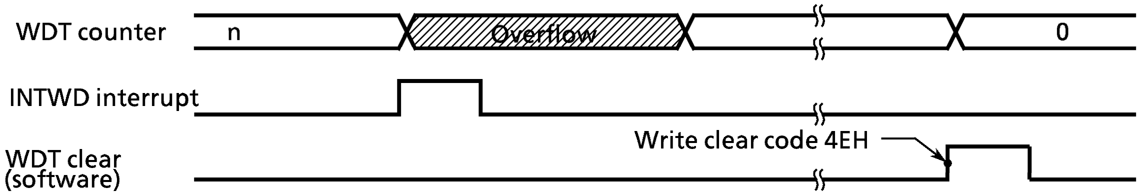


Figure 3.12 (3) Normal Mode

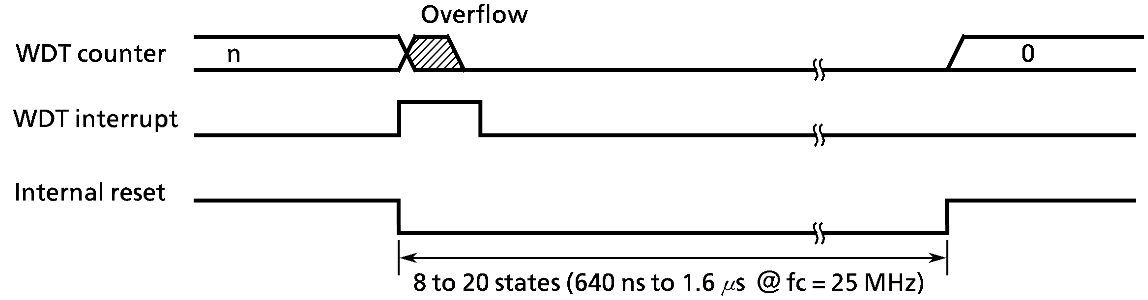


Figure 3.12 (4) Reset Mode

The watchdog timer operates during RUN and IDLE2 modes. While an INTWD interrupt does not occur during IDLE2 mode, to prevent an INTWD interrupt being triggered immediately after the halt release, disable the watchdog timer. The watchdog timer is halted in IDLE1 and STOP modes.

As the binary counter continues counting during bus release (when $\overline{\text{BUSAK}}$ goes low), set the runaway detection time in accordance with the bus release time. If the watchdog timer detects a runaway condition during bus release, the watchdog timer generates an INTWD interrupt immediately after the bus release.

The watchdog timer starts operating immediately after reset release.

Examples: ① Clear the binary counter.

```
WDCR  ← 0 1 0 0 1 1 1 0     Write clear code 4EH.
```

② Set the watchdog timer detection time to 2^{18}/fc.

```
WDMOD ← 1 0 1 – – – X X
```

③ Disable the watchdog timer.

```
WDMOD ← 0 – – – – – X X     Clear <WDTE> to 0.
WDCR  ← 1 0 1 1 0 0 0 1     Write disable code B1H.
```

④ Select IDLE1 mode.

```
WDMOD ← 0 – – – 1 0 X X     Disable WDT and set IDLE1 mode.
WDCR  ← 1 0 1 1 0 0 0 1
Execute HALT instruction.   Set HALT mode.
```

⑤ Select IDLE2 mode.

```
WDMOD ← 0 – – – 1 1 X X     Disable WDT and set IDLE2 mode.
WDCR  ← 1 0 1 1 0 0 0 1
Execute HALT instruction.   Set HALT mode.
```

⑥ Select STOP mode. (Warm-up time 2^{16}/fc)

```
WDMOD ← – – – 1 0 1 X X     Set STOP mode.
Execute HALT instruction.   Set HALT mode.
```

Note: X: Don’t care –: No change

3.13 Bus Release Function

TMP95CS64/265 has a bus request pin ($\overline{\text{BUSRQ}}$, shared with P53) for releasing the bus, and a bus acknowledge pin ($\overline{\text{BUSAK}}$, shared with P54). These pins are set by the P5CR and P5FC registers.

3.13 .1 Description of Operation

When a low level signal is input to the $\overline{\text{BUSRQ}}$ pin, TMP95CS64/265 recognizes a bus release request. When the current bus cycle terminates, the address bus (A23 to A0) and the bus control signals ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{HWR}}$, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$) first go high. Then these signals and the data bus (D15 to D0) output buffer are set to off and the $\overline{\text{BUSAK}}$ pin outputs a low signal. This sequence indicates that the bus is released.

During bus release, TMP95CS64/265 disables all access to the internal I/O registers, although internal I/O functions are not affected. Accordingly, the watchdog timer continues to count up during bus release. When using the bus release function, set the runaway detection time in accordance with the bus release time.

3.13.2 Pin States When Bus is Released

Table 3.13 shows the pin states when the bus is released.

Table 3.13 Pin States at Bus Release

Pin Name	Pin State at Bus Release	
	Port Mode	Function Mode
P07 to P00 (D7 to D0) P17 to P10 (D15 to D8)	No change	Goes to high impedance.
P27 to P20 (A23 to A16) P37 to P30 (A15 to A8) P47toP40 (A7 to A0) P50 ($\overline{\text{RD}}$) P51 ($\overline{\text{WR}}$)	No change	Goes to high impedance. (Goes high immediately before bus release.)
P52 ($\overline{\text{HWR}}$)	No change	Turns output buffer off. Internal pull-up resistors are added regardless of the output latch value. (Goes high immediately before bus release.)
P63 ($\overline{\text{CS3}}$) P62 ($\overline{\text{CS2}}$) P61 ($\overline{\text{CS1}}$) P60 ($\overline{\text{CS0}}$)	No change	Goes to high impedance. (Goes high immediately before bus release.)

4. Electrical Characteristics

4.1 Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Power Supply Voltage	V_{cc}	− 0.5 to + 6.5	V
Input Voltage	V_{IN}	− 0.5 to Vcc + 0.5	V
Output current (total)	ΣI_{OL}	+ 120	mA
Output current (total)	ΣI_{OH}	− 120	mA
Power Dissipation (Ta = + 70°C)	P_D	600	mW
Soldering Temperature (10 s)	T_{SOLDER}	+ 260	°C
Storage Temperature	T_{STG}	− 65 to + 150	°C
Operating Temperature	T_{OPR}	− 20 to + 70	°C

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

4.2 DC Electrical Characteristics

(1) Vcc = + 5 V ± 10%, Ta = − 20 to + 70°C (fc = 8 to 25 MHz)

(Typical values are for Ta = + 25°C, VCC = + 5 V.)

Parameter	Symbol	Test Condition	Min	Max	Unit
Input Low Voltage (D0 to 15)	V_{IL}		−0.3	0.8	V
Port 2 to A	V_{IL1}		−0.3	0.3 Vcc	V
(except P56, P70, P72, P73, P75)					
RESET, NMI, INT0 to 4	V_{IL2}		−0.3	0.25 Vcc	V
EA, AM8/16	V_{IL3}		−0.3	0.3	V
X1	V_{IL4}		−0.3	0.2 Vcc	V
Input High Voltage (D0 to 15)	V_{IH}		2.2	Vcc + 0.3	V
Port 2 to A	V_{IH1}		0.7 Vcc	Vcc + 0.3	V
(except P56, P70, P72, P73, P75)					
RESET, NMI, INT0 to 4	V_{IH2}		0.75 Vcc	Vcc + 0.3	V
EA, AM8/16	V_{IH3}		Vcc − 0.3	Vcc + 0.3	V
X1	V_{IH4}		0.8 Vcc	Vcc + 0.3	V
Output Low Voltage	V_{OL}	I_{OL} = 1.6 mA		0.45	V
Output High Voltage	V_{OH}	I_{OH} = − 400 μA	2.4		V
	V_{OH1}	I_{OH} = − 100 μA	0.75 Vcc		V
	V_{OH2}	I_{OH} = − 20 μA	0.9 Vcc		V
Darlington Drive Current	I_{DAR}	V_{EXT} = 1.5 V	− 1.0	− 3.5	mA
(8 Output Pins max.)		R_{EXT} = 1.1 kΩ			
Input Leakage Current	I_{LI}	0.0 ≦ Vin ≦ Vcc	0.02 (Typ)	± 5	μA
Output Leakage Current	I_{LO}	0.2 ≦ Vin ≦ Vcc − 0.2	0.05 (Typ)	± 10	μA
Operating Current (RUN)	I_{cc}	fc = 25 MHz	40 (Typ)	50	mA
IDLE2			30 (Typ)	40	mA
IDLE1			3.5 (Typ)	10	mA
STOP (Ta = − 20 to + 70°C)		0.2 ≦ Vin ≦ Vcc − 0.2	0.5 (Typ)	50	μA
STOP (Ta = 0 to + 50°C)		0.2 ≦ Vin ≦ Vcc − 0.2		10	μA
Power Down Voltage	V_{STOP}	V_{IL2} = 0.2 Vcc,	2.0	6.0	V
(@STOP, RAM Back up)		V_{IH2} = 0.8 Vcc			
Pull Up Registance	R_{RP}		45	160	kΩ
Pin Capacitance	C_{IO}	fc = 1 MHz		10	pF
Schmitt Width	V_{TH}		0.4	1.0 (Typ)	V
RESET, NMI, INT0 to 4					

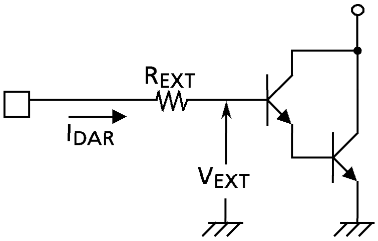
Note: I_{DAR} guarantees up to eight pins from any output port.

(2) Vcc = +3 V ± 10%, Ta = −20 to +70°C (fc = 4 to 10 MHz)

(Typical values are for Ta = +25°C, VCC = +3 V.)

Parameter	Symbol	Test Condition	Min	Max	Unit
Input Low Voltage (D0 to 15)	V_{IL}		−0.3	0.6	V
Port 2 to A	V_{IL1}		−0.3	0.3 Vcc	V
(except P56, P70, P72, P73, P75)					
$\overline{RESET}$, $\overline{NMI}$, INT0 to 4	V_{IL2}		−0.3	0.25 Vcc	V
$\overline{EA}$, AM8/$\overline{16}$	V_{IL3}		−0.3	0.3	V
X1	V_{IL4}		−0.3	0.2 Vcc	V
Input High Voltage (D0 to 15)	V_{IH}		2.0	Vcc + 0.3	V
Port 2 to A	V_{IH1}		0.7 Vcc	Vcc + 0.3	V
(except P56, P70, P72, P73, P75)					
$\overline{RESET}$, $\overline{NMI}$, INT0 to 4	V_{IH2}		0.75 Vcc	Vcc + 0.3	V
$\overline{EA}$, AM8/$\overline{16}$	V_{IH3}		Vcc − 0.3	Vcc + 0.3	V
X1	V_{IH4}		0.8 Vcc	Vcc + 0.3	V
Output Low Voltage	V_{OL}	I_{OL} = 1.6 mA		0.45	V
Output High Voltage	V_{OH}	I_{OH} = −400 μA	2.4		V
Input Leakage Current	I_{LI}	0.0 ≦ Vin ≦ Vcc	0.02 (Typ)	± 5	μA
Output Leakage Current	I_{LO}	0.2 ≦ Vin ≦ Vcc − 0.2	0.05 (Typ)	± 10	μA
Operating Current (RUN)	I_{cc}	fc = 10 MHz	12 (Typ)	25	mA
IDLE2			4.5 (Typ)	17	mA
IDLE1			0.8 (Typ)	5	mA
STOP (Ta = −20 to +70°C)		0.2 ≦ Vin ≦ Vcc − 0.2	0.5 (Typ)	50	μA
STOP (Ta = 0 to +50°C)		0.2 ≦ Vin ≦ Vcc − 0.2		10	μA
Power Down Voltage	V_{STOP}	V_{IL2} = 0.2 Vcc,	2.0	6.0	V
(@ STOP, RAM Back up)		V_{IH2} = 0.8 Vcc			
Pull Up Registance	R_{RP}		70	400	kΩ
Pin Capacitance	C_{IO}	fc = 1 MHz		10	pF
Schmitt Width	V_{TH}		0.4	1.0 (Typ)	V
$\overline{RESET}$, $\overline{NMI}$, INT0 to 4					

Refer: I_{DAR} definition diagram.



4.3 AC Electrical Characteristics

(1) Vcc = +5 V ± 10%, Ta = −20 to +70°C

(fc = 8 MHz to 25 MHz)

No.	Parameter	Symbol	Formula		20 MHz		25 MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	Oscillation cycle (= x)	t_{OSC}	40	125	50		40		ns
2	Clock pulse width	t_{CLK}	2.0x – 40		60		40		ns
3	A0 to 23 valid → Clock hold	t_{AK}	0.5x – 20		5		0		ns
4	Clock valid → A0 to 23 hold	t_{KA}	1.5x – 60		15		0		ns
5	A0 to 23 valid → $\overline{\text{RD}}$/$\overline{\text{WR}}$ fall	t_{AC}	1.0x – 20		30		20		ns
6	$\overline{\text{RD}}$/$\overline{\text{WR}}$ rise → A0 to 23 hold	t_{CA}	0.5x – 20		5		0		ns
7	A0 to 23 valid → D0 to 15 input	t_{AD}		3.5x – 40		135		100	ns
8	$\overline{\text{RD}}$ fall → D0 to 15 input	t_{RD}		2.5x – 45		80		55	ns
9	$\overline{\text{RD}}$ low pulse width	t_{RR}	2.5x – 40		85		60		ns
10	$\overline{\text{RD}}$ rise → D0 to 15 hold	t_{HR}	0		0		0		ns
11	$\overline{\text{WR}}$ low pulse width	t_{WW}	2.5x – 40		85		60		ns
12	D0 to 15 valid → $\overline{\text{WR}}$ rise	t_{DW}	2.0x – 40		60		40		ns
13	$\overline{\text{WR}}$ rise → D0 to 15 hold	t_{WD}	0.5x – 10		15		10		ns
14	A0 to 23 valid → $\overline{\text{WAIT}}$ input (1 WAIT + n mode)	t_{AW}		3.5x – 90		85		50	ns
	A0 to 23 valid → $\overline{\text{WAIT}}$ input (0 + n WAIT mode)	t_{AW}		1.5x – 40		35		20	ns
15	$\overline{\text{RD}}$/$\overline{\text{WR}}$ fall → $\overline{\text{WAIT}}$ hold (1 WAIT + n mode)	t_{CW}	2.5x + 0		125		100		ns
	$\overline{\text{RD}}$/$\overline{\text{WR}}$ fall → $\overline{\text{WAIT}}$ hold (0 + n WAIT mode)	t_{CW}	0.5x + 0		25		20		ns
16	$\overline{\text{WR}}$ rise → PORT valid	t_{CP}		200		200		200	ns
17	$\overline{\text{CS}}$ Low pulse width (PSRAM mode)	t_{CE}	3.0x – 40		110		80		ns
18	$\overline{\text{CS}}$ fall → D0 to 15 input (PSRAM mode)	t_{CEA}		3.0x – 60		90		60	ns
19	Address setup time (PSRAM mode)	t_{PASC}	0.5x – 15		10		5		ns
20	$\overline{\text{CS}}$ precharge time (PSRAM mode)	t_{PP}	1.0x – 10		40		30		ns

AC measuring conditions

- Output level: High 2.2 V/Low 0.8 V, CL = 50 pF
- Input level: High 2.4 V / Low 0.45 V (D0 to D15)
  High 0.8 Vcc / Low 0.2 Vcc (except for D0 to D15)

(2) Vcc = +3 V ± 10%, Ta = −20 to +70°C

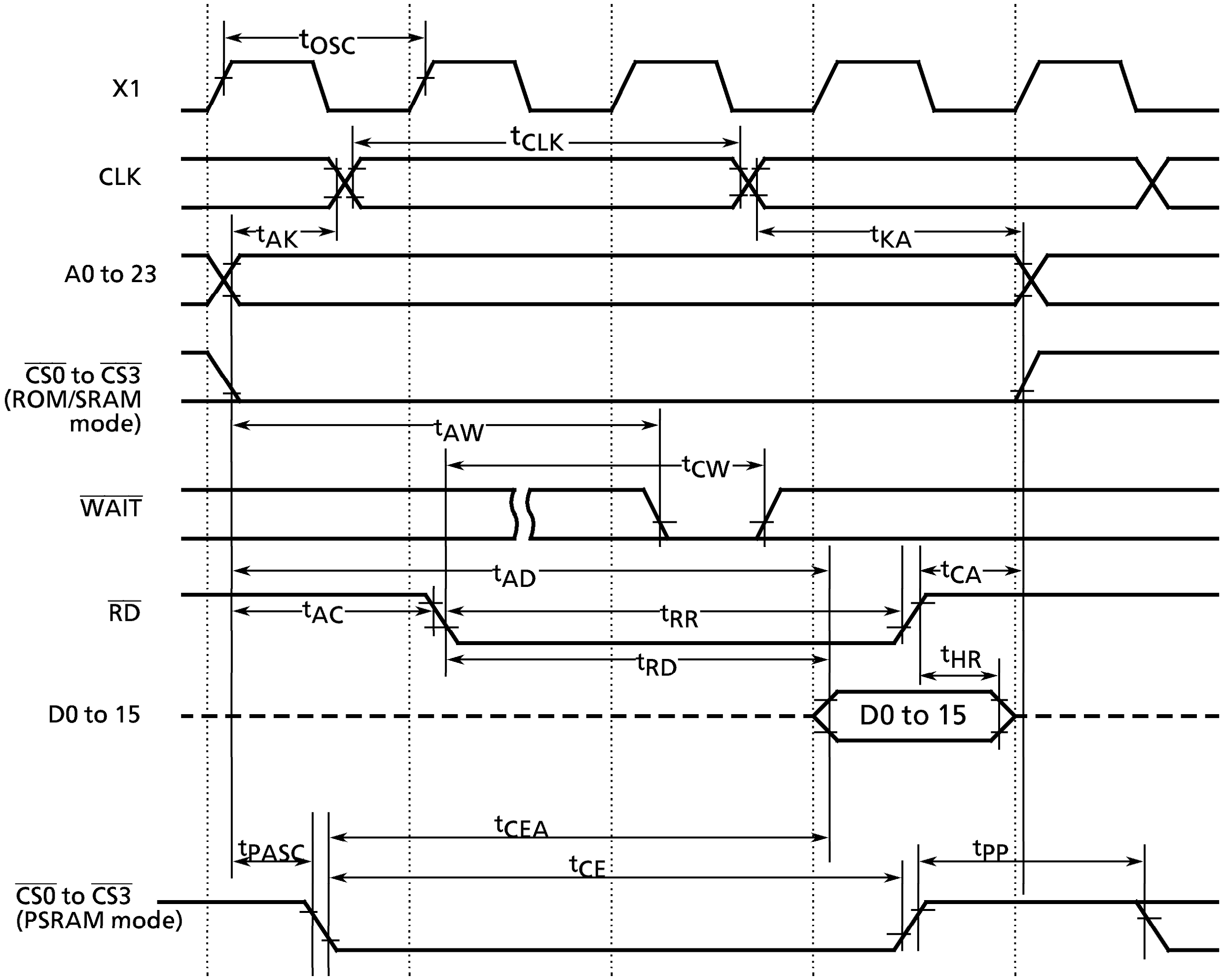
(fc = 4 MHz to 10 MHz)

No.	Parameter	Symbol	Formula		10 MHz		Unit
			Min	Max	Min	Max	
1	Oscillation cycle (= x)	t_{OSC}	100	250	100		ns
2	Clock pulse width	t_{CLK}	2.0x − 70		130		ns
3	A0 to 23 valid → $\overline{\text{RD}}$/$\overline{\text{WR}}$ fall	t_{AC}	1.0x − 60		40		ns
4	$\overline{\text{RD}}$/$\overline{\text{WR}}$ rise→ A0 to 23 hold	t_{CA}	0.5x − 40		10		ns
5	A0 to 23 valid → D0 to 15 input	t_{AD}		3.5x − 125		225	ns
6	$\overline{\text{RD}}$ fall → D0 to 15 input	t_{RD}		2.5x − 115		135	ns
7	$\overline{\text{RD}}$ Low pulse width	t_{RR}	2.5x − 40		210		ns
8	$\overline{\text{RD}}$rise→ D0 to 15 hold	t_{HR}	0		0		ns
9	$\overline{\text{WR}}$ Low pulse width	t_{WW}	2.5x − 40		210		ns
10	D0 to 15 valid → $\overline{\text{WR}}$ rise	t_{DW}	2.0x − 120		80		ns
11	$\overline{\text{WR}}$ rise →D0 to 15 hold	t_{WD}	0.5x − 40		10		ns
12	A0 to 23 valid → $\overline{\text{WAIT}}$ input (1 WAIT + n mode)	t_{AW}		3.5x − 130		220	ns
	A0 to 23 valid → $\overline{\text{WAIT}}$ input (0 + n WAIT mode)	t_{AW}		1.5x − 80		70	ns
13	$\overline{\text{RD}}$/$\overline{\text{WR}}$ fall → $\overline{\text{WAIT}}$ hold (1 WAIT + n mode)	t_{CW}	2.5x + 0		250		ns
	$\overline{\text{RD}}$/$\overline{\text{WR}}$ fall → $\overline{\text{WAIT}}$ hold (0 + n WAIT mode)	t_{CW}	0.5x + 0		50		ns
14	$\overline{\text{WR}}$ rise→ PORT valid	t_{CP}		200		200	ns
15	$\overline{\text{CS}}$ Low pulse width (PSRAM mode)	t_{CE}	3.0x − 70		230		ns
16	$\overline{\text{CS}}$ fall → D0 to 15 input (PSRAM mode)	t_{CEA}		3.0x − 160		140	ns
17	Address setup time (PSRAM mode)	t_{PASC}	0.5x − 30		20		ns
18	$\overline{\text{CS}}$ precharge time (PSRAM mode)	t_{PP}	1.0x − 40		60		ns

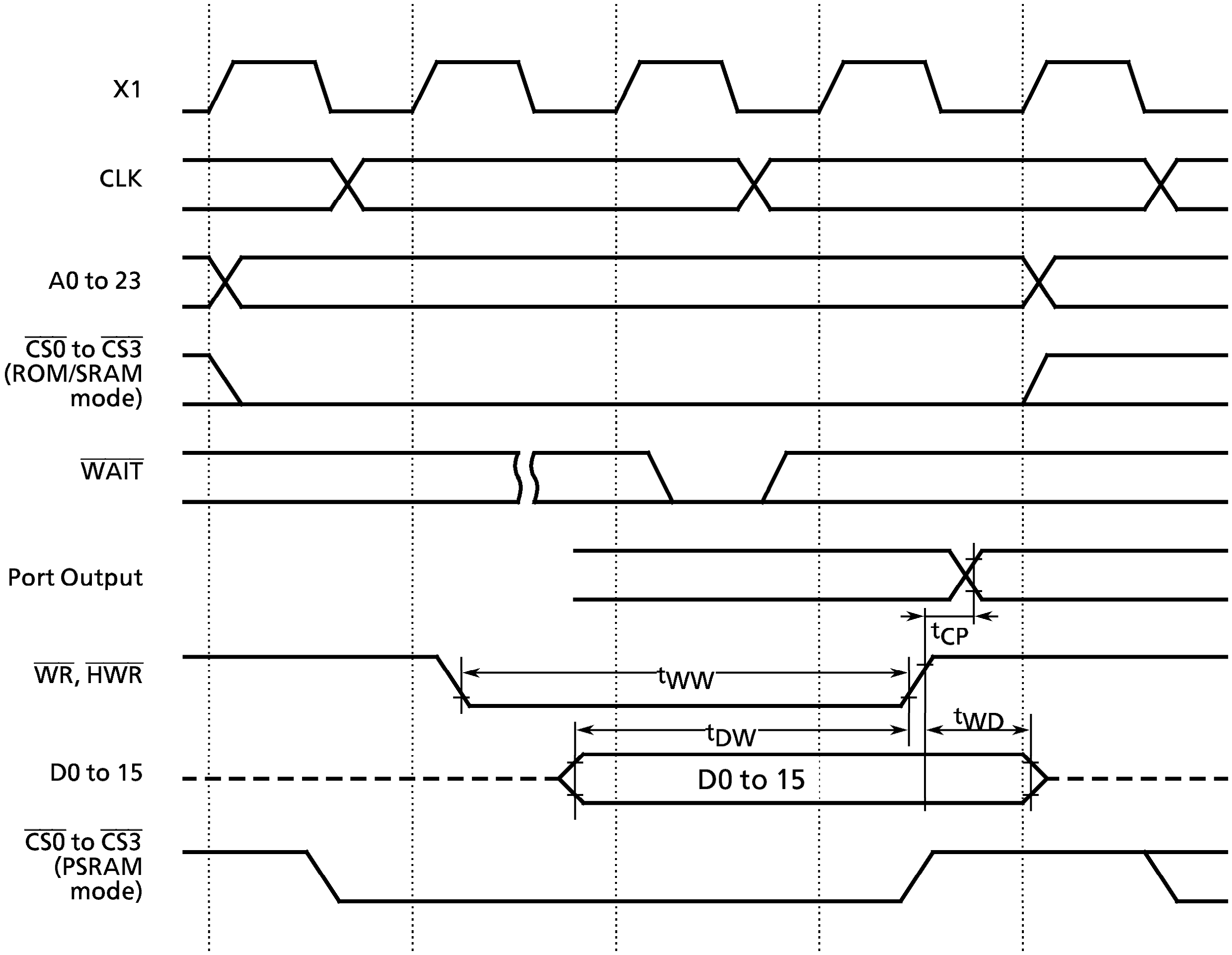
AC measuring conditions

- Output level: High 0.7x Vcc / Low 0.3x Vcc, CL = 50 pF
- Input level: High 0.9x Vcc / Low 0.1x Vcc

(3) Read Cycle


t_{OSC}
X1
t_{CLK}
CLK
t_{AK}
t_{KA}
A0 to 23
$\overline{CS0}$ to $\overline{CS3}$ (ROM/SRAM mode)
t_{AW}
t_{CW}
$\overline{WAIT}$
t_{AD}
t_{CA}
$\overline{RD}$
t_{AC}
t_{RR}
t_{RD}
t_{HR}
D0 to 15
D0 to 15
t_{CEA}
t_{PASC}
t_{PP}
t_{CE}
$\overline{CS0}$ to $\overline{CS3}$ (PSRAM mode)

(4) Write Cycle


X1
CLK
A0 to 23
$\overline{CS0}$ to $\overline{CS3}$ (ROM/SRAM mode)
$\overline{WAIT}$
Port Output
t_{CP}
$\overline{WR}$, $\overline{HWR}$
t_{WW}
t_{DW}
t_{WD}
D0 to 15
D0 to 15
$\overline{CS0}$ to $\overline{CS3}$ (PSRAM mode)

4.4 Serial Channel Timing

(1) I/O interface mode

① SCLK input mode

Vcc = +5 V ± 10%, Ta = −20 to +70°C (fc = 8 to 25 MHz)
Vcc = +3 V ± 10%, Ta = −20 to +70°C (fc = 4 to 10 MHz)

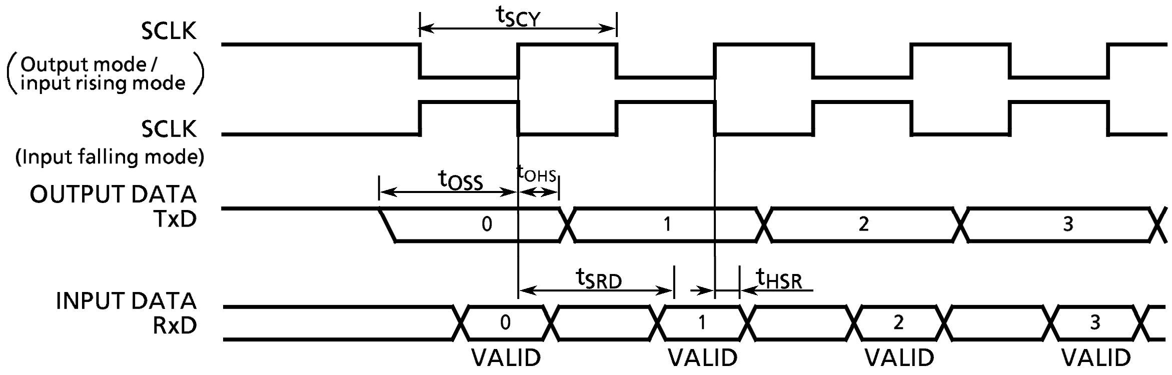
Parameter	Symbol	Formula		10 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK cycle	t_{SCY}	16x		1.6		0.64		μs
Output Data → SCLK rise/fall*	t_{OSS}	$t_{SCY}/2 - 5x - 50$		250		70		ns
SCLK rise/fall* → Output Data hold	t_{OHS}	5x − 100		400		100		ns
SCLK rise/fall* → input data hold	t_{HSR}	0		0		0		ns
SCLK rise/fall* → valid data input	t_{SRD}		$t_{SCY} - 5x - 100$		1000		340	ns

*) SCLK rise/fall: In SCLK rising edge mode, SCLK rising edge timing; in SCLK falling edge mode, SCLK falling edge timing

② SCLK output mode

Vcc = +5 V ± 10%, Ta = −20 to +70°C (fc = 8 to 25 MHz)
Vcc = +3 V ± 10%, Ta = −20 to +70°C (fc = 4 to 10 MHz)

Parameter	Symbol	Formula		10 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK cycle (programmable)	t_{SCY}	16x	8192x	1.6	819.2	0.64	327.6	μs
Output Data → SCLK rising edge	t_{OSS}	$t_{SCY} - 2x - 150$		1250		410		ns
SCLK rising edge → Output Data hold	t_{OHS}	2x − 80		120		0		ns
SCLK rising edge → Input Data hold	t_{HSR}	0		0		0		ns
SCLK rising edge → valid data input	t_{SRD}		$t_{SCY} - 2x - 150$		1250		410	ns



(2) UART Mode (SCLK0 to 2 External Input)

Vcc = +5 V ± 10%, Ta = −20 to +70°C (fc = 8 to 25 MHz)
Vcc = +3 V ± 10%, Ta = −20 to +70°C (fc = 4 to 10 MHz)

Parameter	Symbol	Formula		10 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK cycle	t_{SCY}	4x + 20		420		180		ns
Low-level SCLK pulse width	t_{SCYL}	2x + 5		205		85		ns
High-level SCLK pulse width	t_{SCYH}	2x + 5		205		85		ns

4.5 A/D Conversion Characteristics

Vcc = + 5 V ± 10%, Ta = – 20 to + 70°C (fc = 8 to 25 MHz)
Vcc = + 3 V ± 10%, Ta = – 20 to + 70°C (fc = 4 to 10 MHz)

Parameter		Symbol	Test Conditions	Min	Typ.	Max	Unit
A/D analog reference supply voltage (+)		V_{REFH}		Vcc – 0.2		Vcc	V
A/D analog reference supply voltage (–)		V_{REFL}		Vss		Vss + 0.2	
Analog reference voltage		AV_{CC}		Vcc – 0.2		Vcc	
Analog reference voltage		AV_{SS}		Vss		Vss + 0.2	
Analog input voltage		V_{AIN}		V_{REFL}		V_{REFH}	
Analog reference voltage supply current	<VREFON> = 1	I_{REF}	Vcc = 5 V ± 10%			3.7	mA
			Vcc = 3 V ± 10%			2.2	
	<VREFON> = 0		Vcc = 2.7 to 5.5 V		0.02	5.0	μA
Total tolerance (excludes quantization error)		E_T	Vcc = 5 V ± 10%		± 1	± 3	LSB
			Vcc = 3 V ± 10%		± 1	± 3	

Note 1: 1LSB = (VREFH – VREFL) / 2^{10} [V]

Note 2: Power supply current ICC from the VCC pin includes the power supply current from the AVCC pin.

4.6 D/A Conversion Characteristics

Vcc = + 5 V ± 10%, Ta = – 20 to + 70°C (fc = 8 to 25 MHz)
Vcc = + 3 V ± 10%, Ta = – 20 to + 70°C (fc = 4 to 10 MHz)

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Analog reference voltage	AV_{CC}		Vcc – 0.2		Vcc	V
Analog reference voltage	AV_{SS}		V_{SS}		Vss + 0.2	
Total tolerance		R = 1 MΩ (Note) R = 5 MΩ (Note) R = 10 MΩ (Note)			7.0 4.0 3.5	LSB LSB LSB
Differential linear error				2.0		LSB

Note: R is the external load resistance on the D/A converter output pin (DAOUT0, DAOUT1).

4.7 Event Counter (External Input Clocks: TI0, TI4, TI8, TI9, TIA, TIB)

Vcc = + 5 V ± 10%, Ta = – 20 to + 70°C (fc = 8 to 25 MHz)
Vcc = + 3 V ± 10%, Ta = – 20 to + 70°C (fc = 4 to 10 MHz)

Parameter	Symbol	Calculator		10 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
External input clock cycle	t_{VCK}	8x + 100		900		420		ns
External low-level input clock pulse width	t_{VCKL}	4x + 40		440		200		ns
External high-level input clock pulse width	t_{VCKH}	4x + 40		440		200		ns

4.8 Interrupt Operation

Vcc = +5 V ± 10%, Ta = − 20 to +70°C (fc = 8 to 25 MHz)
Vcc = +3 V ± 10%, Ta = − 20 to +70°C (fc = 4 to 10 MHz)

Parameter	Symbol	Calculator		10 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{\text{NMI}}$, INT0 to 4 low-level pulse width	t_{INTAL}	4x		400		160		ns
$\overline{\text{NMI}}$, INT0 to 4 high-level pulse width	t_{INTAH}	4x		400		160		ns
INT5 to INT8 low-level pulse width	t_{INTBL}	8x + 100		900		420		ns
INT5 to INT8 high-level pulse width	t_{INTBH}	8x + 100		900		420		ns

4.9 A/D External Start

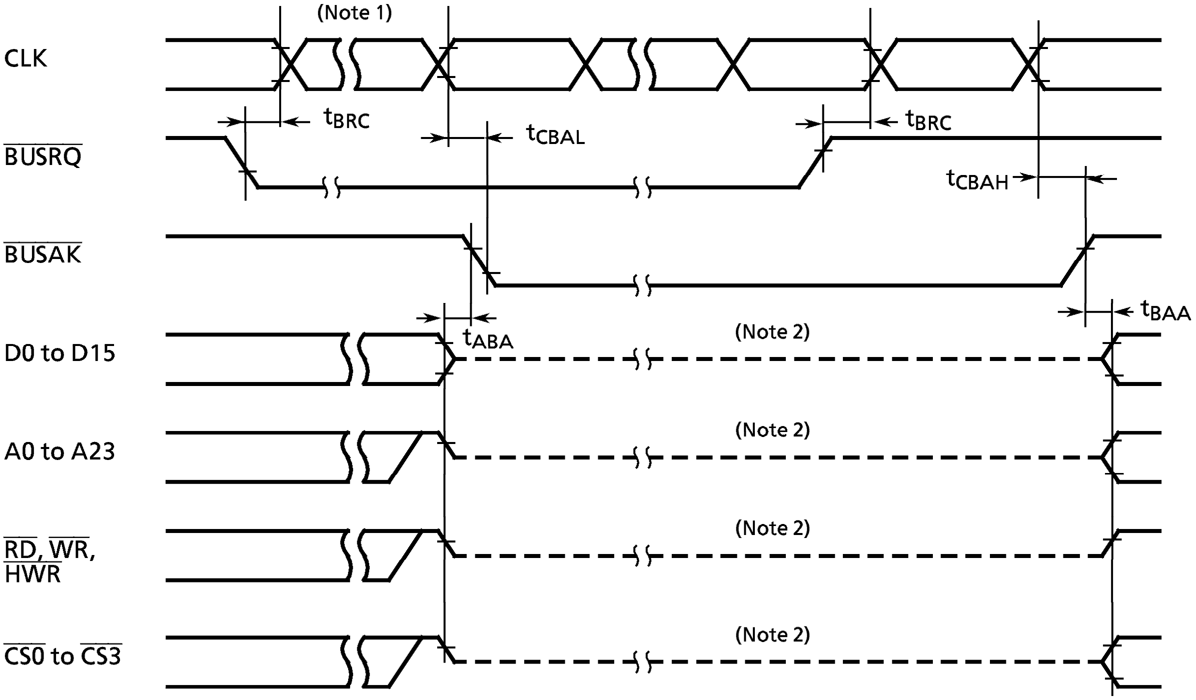
Vcc = +5 V ± 10%, Ta = − 20 to +70°C (fc = 8 to 25 MHz)
Vcc = +3 V ± 10%, Ta = − 20 to +70°C (fc = 4 to 10 MHz)

Parameter	Symbol	Calculator		10 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{\text{ADTRG}}$ low-level pulse width	t_{ADTG}	4x		400		160		ns

4.10 Bus Request/Bus Acknowledge Timing

Vcc = +5 V ± 10%, Ta = −20 to +70°C (fc = 8 to 25 MHz)
Vcc = +3 V ± 10%, Ta = −20 to +70°C (fc = 4 to 10 MHz)

Parameter	Symbol	Calculator		10 MHz		25 MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{\text{BUSRQ}}$ setup time for CLK	t_{BRC}	120		120		120		ns
CLK→$\overline{\text{BUSAK}}$ fall	t_{CBAL}		2.0x + 120		320		200	ns
CLK→$\overline{\text{BUSAK}}$ rise	t_{CBAH}		0.5x + 40		90		60	ns
Time from output buffer off until $\overline{\text{BUSAK}}$ falling edge	t_{ABA}	0	80	0	80	0	80	ns
Time from $\overline{\text{BUSAK}}$ rising edge until output buffer on	t_{BAA}	0	80	0	80	0	80	ns


CLK
(Note 1)
$\overline{\text{BUSRQ}}$
t_{BRC}
t_{CBAL}
t_{BRC}
t_{CBAH}
$\overline{\text{BUSAK}}$
t_{ABA}
t_{BAA}
D0 to D15
A0 to A23
$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{HWR}}$
$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$
(Note 2)
(Note 2)
(Note 2)
(Note 2)

Note 1: When $\overline{\text{BUSRQ}}$ goes to low level to request bus release, if the current bus cycle is yet complete due to a wait, the bus is not released until the wait completes.

Note 2: The dotted line indicates only that the output buffer is off, not that the signal is at middle level. Immediately after bus release, the signal level prior to the bus release is held dynamically by the external load capacitance. Therefore, designs should allow for the fact that when using an external resistor or similar to fix the signal level while the bus is released, after bus release a delay occurs before the signal goes to its fixed level (due to the CR time constant). The internal programmable pull-up resistor continues to function in accordance with the internal signal level.

5. List of Special Function Registers (SFR)

The special function registers (SFR), which control the input/output ports and peripheral components, are allocated 160 bytes within the 000000H to 00009FH address range.
The registers built into TMP95CS64/265 cannot be accessed from outside TMP95CS64/265.

(1) Input/output port

(2) Input/output port control

(3) Timer control

(4) Serial channel control

(5) Interrupt control

(6) Watchdog timer control

(7) Chip select/wait controller

(8) D/A converter control

(9) A/D converter control

Table structure

Symbol	Name	Address	7	6	1	0	
							→ bit Symbol
							→ Read / Write
							→ Initial value at reset
							→ Remarks

(Supplement for symbols used in Table)

① Read / Write

- R/W : Both readable and writable
- R : Readable
- W : Writable
- *R/W : Read-modify-write (RMW) instructions are prohibited for controlling ON/OFF of the pull-up resistors.

② RMW prohibited

- Cannot be read, modified, and written. (Cannot use the following instructions: EX, ADD, ADC, SUB, SBC, INC, DEC, AND, OR, XOR, STCF, RES, SET, CHG, TEST, RLC, RRC, RL, RR, SLA, SRA, SLL, SRL, RLD, RRD)

Table 5 List of TMP95CS64/265 Special Function Register Addresses

Address	Register Name	Address	Register Name	Address	Register Name	Address	Register Name
000000H	P0	30H	TREG8L	60H	ADREG04L	90H	B0CS
1H	P1	1H	TREG8H	1H	ADREG04H	1H	B1CS
2H	P0CR	2H	TREG9L	2H	ADREG15L	2H	B2CS
3H	(Reserved)	3H	TREG9H	3H	ADREG15H	3H	B3CS
4H	P1CR	4H	CAP1L	4H	ADREG26L	4H	MSAR0
5H	P1FC	5H	CAP1H	5H	ADREG26H	5H	MAMR0
6H	P2	6H	CAP2L	6H	ADREG37L	6H	MSAR1
7H	P3	7H	CAP2H	7H	ADREG37H	7H	MAMR1
8H	P2CR	8H	T8MOD	8H	(Reserved)	8H	MSAR2
9H	P2FC	9H	T8FFCR	9H	(Reserved)	9H	MAMR2
AH	P3CR	AH	T89CR	AH	SDMACR0	AH	MSAR3
BH	P3FC	BH	T16RUN	BH	SDMACR1	BH	MAMR3
CH	P4	CH	(Reserved)	CH	SDMACR2	CH	BEXCS
DH	P5	DH		DH	SDMACR3	DH	DADRV
EH	P4CR	EH		EH	WDMOD	EH	DAREG0
FH	P4FC	FH		FH	WDCR	FH	DAREG1
10H	P5CR	40H	TREGAL	70H	INTE0AD		
1H	P5FC	1H	TREGAH	1H	INTE12		
2H	P6	2H	TREGBL	2H	INTE34		
3H	P7	3H	TREGBH	3H	INTE56		
4H	(Reserved)	4H	CAP3L	4H	INTE78		
5H	P6FC	5H	CAP3H	5H	INTET01		
6H	P7CR	6H	CAP4L	6H	INTET23		
7H	P7FC	7H	CAP4H	7H	INTET45		
8H	P8	8H	T9MOD	8H	INTET67		
9H	P9	9H	T9FFCR	9H	INTET89		
AH	P8CR	AH	(Reserved)	AH	INTETAB		
BH	P8FC	BH	(Reserved)	BH	NTETOV		
CH	P9CR	CH	SC0BUF	CH	INTES0		
DH	P9FC	DH	SC0CR	DH	INTES1		
EH	PA	EH	SC0MOD	EH	INTES2		
FH	(Reserved)	FH	BR0CR	FH	INTETC01		
20H	T8RUN	50H	SC1BUF	80H	INTETC23		
1H	TRDC	1H	SC1CR	1H	(Reserved)		
2H	TREG0	2H	SC1MOD	2H			
3H	TREG1	3H	BR1CR	3H			
4H	T01MOD	4H	SC2BUF	4H			
5H	T02FFCR	5H	SC2CR	5H			
6H	TREG2	6H	SC2MOD	6H			
7H	TREG3	7H	BR2CR	7H			
8H	T23MOD	8H	ODE	8H			
9H	TREG4	9H	IIMC	9H			
AH	TREG5	AH	DMA0V	AH			
BH	T45MOD	BH	DMA1V	BH			
CH	T46FFCR	CH	DMA2V	CH			
DH	TREG6	DH	DMA3V	DH			
EH	TREG7	EH	ADMOD0	EH			
FH	T67MOD	FH	ADMOD1	FH			

(1) Input/Output Ports

Symbol	Name	Address	7	6	5	4	3	2	1	0
P0	Port 0 Register	00H	P07	P06	P05	P04	P03	P02	P01	P00
			R/W							
			Input mode (output latch register undefined)							
			shared with D7 to D0							
P1	Port 1 Register	01H	P17	P16	P15	P14	P13	P12	P11	P10
			R/W							
			Input mode (output latch register cleared to 0)							
			shared with D15 to D8							
P2	Port 2 Register	06H	P27	P26	P25	P24	P23	P22	P21	P20
			R/W							
			Input mode (output latch register cleared to 0)							
			shared with D23 to D16							
P3	Port 3 Register	07H	P37	P36	P35	P34	P33	P32	P31	P30
			R/W							
			Input mode (output latch register cleared to 0)							
			shared with A15 to A8							
P4	Port 4 Register	0CH	P47	P46	P45	P44	P43	P42	P41	P40
			R/W							
			Input mode (output latch register cleared to 0)							
			shared with A7 to A0							
P5	Port 5 Register	0DH	P57	P56	P55	P54	P53	P52	P51	P50
			*R/W							
			Input mode (set to 1 / Pull-up)						Output only (set to 1) (Note 1)	
			Shared with SCLK2/$\overline{\text{CTS2}}$	Shared with INT0	Shared with $\overline{\text{WAIT}}$	Shared with $\overline{\text{BUSAK}}$	Shared with $\overline{\text{BUSRQ}}$	Shared with $\overline{\text{HWR}}$	Shared with $\overline{\text{WR}}$	Shared with $\overline{\text{RD}}$
P6	Port 6 Register	12H					P63	P62	P61	P60
							R/W			
							Output mode (set to 1) (Note 2)			
							Shared with $\overline{\text{CS3}}$	Shared with $\overline{\text{CS2}}$	Shared with $\overline{\text{CS1}}$	Shared with $\overline{\text{CS0}}$
P7	Port 7 Register	13H			P75	P74	P73	P72	P71	P70
					R/W					
					Input mode (output latch register cleared to 0)					
					Shared with TO7/INT4	Shared with TO5	Shared with TI4/INT3	Shared with TO3/INT2	Shared with TO1	Shared with TI0/INT1
P8	Port 8 Register	18H	P87	P86	P85	P84	P83	P82	P81	P80
			*R/W							
			Input mode (set to 1/pulled up)							
			Shared with RxD2	Shared with TxD2	Shared with SCLK1/$\overline{\text{CTS1}}$	Shared with RxD1	Shared with TxD1	Shared with SCLK0/$\overline{\text{CTS0}}$	Shared with RxD0	Shared with TxD0
P9	Port 9 Register	19H		P96	P95	P94	P93	P92	P91	P90
				R/W						
				Input mode (output latch register cleared to 0)						
				Shared with TOA/TOB	Shared with TIB/INT8	Shared with TIA/INT7	Shared with TO9	Shared with TO8	Shared with TI9/INT6	Shared with TI8/INT5
PA	Port A Register	1EH	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
			R							
			Input-only							
			Shared with AN7	Shared with AN6	Shared with AN5	Shared with AN4	Shared with AN4/ $\overline{\text{ADTRG}}$	Shared with AN2	Shared with AN1	Shared with AN0

Note 1: When P5<P50> is cleared to 0 with P50 set as an $\overline{\text{RD}}$ pin (P5FC<P50> = 1 or TMP95C265), the P50 $\overline{\text{RD}}$ signal is still output even when the internal address area is accessed (for PSRAM).

Note 2: Only the <P62> post-reset initial value differs according to the $\overline{\text{EA}}$ pin setting.

	$\overline{\text{EA}}$ = low level	$\overline{\text{EA}}$ = high level
<P62> initial value	0	1

(2) Input/Output Port Control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P0CR	Port 0 Control Register	02H (RMW prohibited)	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
			W							
			0	0	0	0	0	0	0	0
			0: IN 1: OUT							
P1CR	Port 1 Control Register	04H (RMW prohibited)	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
			W							
			0	0	0	0	0	0	0	0
			0: IN 1: OUT							
P1FC	Port 1 Function Register	05H (RMW prohibited)	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
			W							
			0	0	0	0	0	0	0	0
			0: PORT 1: D15 to D8 (P1CR = 00H)							
P2CR	Port 2 Control Register	08H (RMW prohibited)	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
			W							
			0	0	0	0	0	0	0	0
			0: IN 1: OUT							
P2FC	Port 2 Function Register	09H (RMW prohibited)	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
			W							
			0	0	0	0	0	0	0	0
			0: PORT 1: A23 to A16 (P2CR = FFH)							
P3CR	Port 3 Control Register	0AH (RMW prohibited)	P37C	P36C	P35C	P34C	P33C	P32C	P31C	P30C
			W							
			0	0	0	0	0	0	0	0
			0: IN 1: OUT							
P3FC	Port 3 Function Register	0BH (RMW prohibited)	P37F	P36F	P35F	P34F	P33F	P32F	P31F	P30F
			W							
			0	0	0	0		0	0	0
			0: PORT 1: A15 to A8 (P3CR = FFH)							
P4CR	Port 4 Control Register	0EH (RMW prohibited)	P47C	P46C	P45C	P44C	P43C	P42C	P41C	P40C
			W							
			0	0	0	0	0	0	0	0
			0: IN 1: OUT							
P4FC	Port 4 Function Register	0FH (RMW prohibited)	P47F	P46F	P45F	P44F	P43F	P42F	P41F	P40F
			W							
			0	0	0	0	0	0	0	0
			0: PORT 1: A7 to A0 (P4CR = FFH)							
P5CR	Port 5 Control Register	10H (RMW prohibited)	P57C	P56C	P55C	P54C	P53C	P52C		
			W							
			0	0	0	0	0	0		
			0: IN 1: OUT							
P5FC	Port 5 Function Register	11H (RMW prohibited)	P57F			P54F	P53F	P52F	P51F	P50F
			W			W				
			0			0	0	0	0	0
			0: PORT 1: SCLK2 /$\overline{CTS2}$			0: PORT 1: $\overline{BUSAK}$	0: PORT 1: $\overline{BUSRQ}$	0: PORT 1: $\overline{HWR}$	0: PORT 1: $\overline{WR}$	0: PORT 1: $\overline{RD}$

Note: In the external ROM version of TMP95C265, port 0 functions as the data bus, port 3 and port 4 as the address bus, and pins P50 and P51 as the $\overline{RD}$ and $\overline{WR}$ signal output pins respectively, regardless of the P0CR, P3CR, P3FC, P4CR, P4FC, and P5FC<P50F>, <P51F> settings.

Input/Output Port Control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P6FC	Port 6 Function Register	15H (RMW prohibited)					P63F	P62F	P61F	P60F
							W			
							0	0	0	0
							0: PORT 1: $\overline{CS3}$	0: PORT 1: $\overline{CS2}$	0: PORT 1: $\overline{CS1}$	0: PORT 1: $\overline{CS0}$
P7CR	Port 7 Control Register	16H (RMW prohibited)			P75C	P74C	P73C	P72C	P71C	P70C
					W					
					0	0	0	0	0	0
					0: IN 1: OUT					
P7FC	Port 7 Function Register	17H (RMW prohibited)			P75F	P74F		P72F	P71F	
					W			W		
					0	0		0	0	
					0: PORT 1: TO7	0: PORT 1: TO5		0: PORT 1:TO3	0: PORT 1: TO1	
P8CR	Port 8 Control Register	1AH (RMW prohibited)	P87C	P86C	P85C	P84C	P83C	P82C	P81C	P80C
			W							
			0	0	0	0	0	0	0	0
			0: IN 1: OUT							
P8FC	Port 8 Function Register	1BH (RMW prohibited)		P86F	P85F		P83F	P82F		P80F
				W			W			W
				0	0		0	0		0
				0: PORT 1: TxD2	0: PORT 1: SCLK1 /$\overline{CTS1}$		0: PORT 1: TxD1	0: PORT 1: SCLK0 /$\overline{CTS0}$		0: PORT 1: TxD0
P9CR	Port 9 Control Register	1CH (RMW prohibited)		P96C	P95C	P94C	P93C	P92C	P91C	P90C
				W						
				0	0	0	0	0	0	0
				0: IN 1: OUT						
P9FC	Port 9 Function Register	1DH (RMW prohibited)	TOS1	P96F			P93F	P92F		
			W				W			
			0	0			0	0		
			0: TOA 1: TOB	0: PORT 1: TOA / TOB			0: PORT 1: TO9	0: PORT 1: TO8		

(3) Timer Control (1/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
T8RUN	8 bit Timer Run Control Register	20H	T7RUN	T6RUN	T5RUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
			R/W							
			0	0	0	0	0	0	0	0
			8-bit timer 7 0: Stop and clear 1: Count	8-bit timer 6 0: Stop and clear 1: Count	8-bit timer 5 0: Stop and clear 1: Count	8-bit timer 4 0: Stop and clear 1: Count	8-bit timer 3 0: Stop and clear 1: Count	8-bit timer 2 0: Stop and clear 1: Count	8-bit timer 1 0: Stop and clear 1: Count	8-bit timer 0 0: Stop and clear 1: Count
TRDC	Timer Register Double Buffer Control Register	21H					TR6DE	TR4DE	TR2DE	TR0DE
							R/W			
							0	0	0	0
							TREG6 double buffer 0: Disable 1: Enable	TREG4 double buffer 0: Disable 1: Enable	TREG2 double buffer 0: Disable 1: Enable	TREG0 double buffer 0: Disable 1: Enable
TREG0	8 bit Timer Register 0	22H (RMW prohibited)	–							
			W							
			Undefined							
TREG1	8 bit Timer Register 1	23H (RMW prohibited)	–							
			W							
			Undefined							
T01 MOD	8 bit Timer 0, 1 Mode Control Register	24H	T01M1	T01M0	PWM01	PWM00	T1CLK1	T1CLK0	T0CLK1	T0CLK0
			R/W							
			0	0	0	0	0	0	0	0
			Timer 0, 1 operating mode setting 00: 8 bit timer 01: 16 bit timer 10: 8 bit PPG 11: 8 bit PWM		PWM0 cycle selection 00: Don't care 01: $2^6 - 1$ 10: $2^7 - 1$ 11: $2^8 - 1$		Timer 1 input clock selection 00: TO0TRG 01: ϕT1 10: ϕT16 11: ϕT256		Timer 0 input clock selection 00: TI0 input 01: ϕT1 10: ϕT4 11: ϕT16	
T02 FFCR	8 bit Timer 0, 2 Flip-Flop Control Register	25H	FF3C1	FF3C0	FF3IE	FF3IS	FF1C1	FF1C0	FF1IE	FF1IS
			W		R/W		W		R/W	
			1	1	0	0	1	1	0	0
			00: Invert TFF3 01: Set TFF3 10: Clear TFF3 11: Don't care		TFF3 inversion control 0: Disable 1: Enable	0: Inversion by timer 2 1: Inversion by timer 3	00: Invert TFF1 01: Set TFF1 10: Clear TFF1 11: Don't care		TFF1 inversion control 0: Disable 1: Enable	0: Inversion by timer 0 1: Inversion by timer 1
TREG2	8 bit Timer Register 2	26H (RMW prohibited)	–							
			W							
			Undefined							
TREG3	8 bit Timer Register 3	27H (RMW prohibited)	–							
			W							
			Undefined							
T23 MOD	8 bit Timer 2, 3 Mode Control Register	28H	T23M1	T23M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
			R/W							
			0	0	0	0	0	0	0	0
			Timer 2, 3 operating mode setting 00: 8 bit timer 01: 16 bit timer 10: 8 bit PPG 11: 8 bit PWM		PWM2 cycle selection 00: Don't care 01: $2^6 - 1$ 10: $2^7 - 1$ 11: $2^8 - 1$		Timer 3 input clock selection 00: TO2TRG 01: ϕT1 10: ϕT16 11: ϕT256		Timer 2 input clock selection 00: Don't care 01: ϕT1 10: ϕT4 11: ϕT16	

Timer Control (2/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TREG4	8 bit Timer Register4	29H (RMW prohibited)	–							
			W							
			Undefined							
TREG5	8 bit Timer Register5	2AH (RMW prohibited)	–							
			W							
			Undefined							
T45 MOD	8 bit Timer 4, 5 Mode Control Register	2BH	T45M1	T45M0	PWM41	PWM40	T5CLK1	T5CLK0	T4CLK1	T4CLK0
			R/W							
			0	0	0	0	0	0	0	0
			Timer 4, 5 operating mode setting 00: 8 bit timer 01: 16 bit timer 10: 8 bit PPG 11: 8 bit PWM		PWM4 cycle selection 00: Don't care 01: $2^6 - 1$ 10: $2^7 - 1$ 11: $2^8 - 1$		Timer 5 input clock selection 00: TO4TRG 01: ϕT1 10: ϕT16 11: ϕT256		Timer 4 input clock selection 00: TI4 input 01: ϕT1 10: ϕT4 11: ϕT16	
T46 FFCR	8 bit Timer 4, 6 Flip-Flop Control Register	2CH	FF7C1	FF7C0	FF7IE	FF7IS	FF5C1	FF5C0	FF5IE	FF5IS
			W		R/W		W		R/W	
			1	1	0	0	1	1	0	0
			00: Invert TFF7 01: Set TFF7 10: Clear TFF7 11: Don't care		TFF7 inversion control 0: Disable 1: Enable	0: Inversion by timer 6 1: Inversion by timer 7	00: Invert TFF5 01: Set TFF5 10: Clear TFF5 11: Don't care		TFF5 inversion control 0: Disable 1: Enable	0: I inversion by timer 4 1: Inversion by timer 5
TREG6	8 bit Timer Register6	2DH (RMW prohibited)	–							
			W							
			Undefined							
TREG7	8 bit Timer Register7	2EH (RMW prohibited)	–							
			W							
			Undefined							
T67 MOD	8 bit Timer 6, 7 Mode Control Register	2FH	T67M1	T67M0	PWM61	PWM60	T7CLK1	T7CLK0	T6CLK1	T6CLK0
			R/W							
			0	0	0	0	0	0	0	0
			Timer 6, 7 operating mode setting 00: 8 bit timer 01: 16 bit timer 10: 8 bit PPG 11: 8 bit PWM		PWM6 cycle selection 00: Don't care 01: $2^6 - 1$ 10: $2^7 - 1$ 11: $2^8 - 1$		Timer 7 input clock selection 00: TO6TRG 01: ϕT1 10: ϕT16 11: ϕT256		Timer 6 input clock selection 00: Don't care 01: ϕT1 10: ϕT4 11: ϕT16	
TREG8L	16 bit Timer Register8L	30H (RMW prohibited)	–							
			W							
			Undefined							
TREG8H	16 bit Timer Register8H	31H (RMW prohibited)	–							
			W							
			Undefined							
TREG9L	16 bit Timer Register9L	32H (RMW prohibited)	–							
			W							
			Undefined							
TREG9H	16 bit Timer Register9H	33H (RMW prohibited)	–							
			W							
			Undefined							

Timer Control (3/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
CAP1L	Capture Register1L	34H	–							
			R							
			Undefined							
CAP1H	Capture Register1H	35H	–							
			R							
			Undefined							
CAP2L	Capture Register2L	36H	–							
			R							
			Undefined							
CAP2H	Capture Register2H	37H	–							
			R							
			Undefined							
T8MOD	16 bit Timer 8 Mode Control Register	38H	CAP2T9	EQ9T9	CAP1IN	CAP12M1	CAP12M0	CLE	T8CLK1	T8CLK0
			R/W		W	R/W				
			0	0	1	0	0	0	0	0
			TFF9 inversion trigger 0: Trigger Disable 1: Trigger Enable		0: Software capture 1: Don't care	Capture timing 00: Disable 01: TI8 ↑ TI9 ↑ 10: TI8 ↑ TI8 ↓ 11: TFF1 ↑ TFF1 ↓		Timer 8 up-counter control 0: Clear disabled 1: Clear at match with TREG9	Timer 8 input clock selection 00: TI8 input 01: φT1 10: φT4 11: φT16	
			At loading of up-counter value to CAP2	At match between up-counter and TREG9						
T8FFCR	16 bit Timer 8 Flip-Flop Control Register	39H	TFF9C1	TFF9C0	CAP2T8	CAP1T8	EQ9T8	EQ8T8	TFF8C1	TFF8C0
			W		R/W				W	
			1	1	0	0	0	0	1	1
			00: Invert TFF9 01: Set TFF9 10: Clear TFF9 11: Don't care		TFF8 inversion trigger 0: Trigger Disable 1: Trigger Enable				00: Invert TFF8 01: Set TFF8 10: Clear TFF8 11: Don't care	
					At loading of up-counter value to CAP2	At loading of up-counter value to CAP3	At match between up-counter and TREG9	At match between up-counter and TREG8		
T89CR	Timer 8/9 Control Register	3AH	–					–	DBAEN	DB8EN
			R/W					R/W		
			0					0	0	0
			Note: Always fixed to 0.					Note: Always fixed to 0.	TREGA double buffer 0: Disable 1: Enable	TREG8 double buffer 0: Disable 1: Enable
T16RUN	16 bit Timer Run Control Register	3BH	PRRUN		T9RUN	T8RUN				
			R/W		R/W					
			0		0	0				
			Prescaler 0: Stop and clear 1: Count		16-bit timer 9 0: Stop and clear 1: Count	16-bit timer 8 0: Stop and clear 1: Count				

Timer Control (4/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TREGAL	16 bit Timer RegisterAL	40H (RMW prohibited)	–							
			W							
			Undefined							
TREGAH	16 bit Timer RegisterAH	41H (RMW prohibited)	–							
			W							
			Undefined							
TREGBL	16 bit Timer RegisterBL	42H (RMW prohibited)	–							
			W							
			Undefined							
TREGBH	16 bit Timer RegisterBH	43H (RMW prohibited)	–							
			W							
			Undefined							
CAP3L	Capture Register3L	44H	–							
			R							
			Undefined							
CAP3H	Capture Register3H	45H	–							
			R							
			Undefined							
CAP4L	Capture Register4L	46H	–							
			R							
			Undefined							
CAP4H	Capture Register4H	47H	–							
			R							
			Undefined							
T9MOD	16 bit Timer 9 Mode Control Register	48H	CAP4TB	EQBTB	CAP3IN	CAP34M1	CAP34M0	CLE	T9CLK1	T9CLK0
			R/W		W	R/W				
			0	0	1	0	0	0	0	0
			TFFB inversion trigger 0: Trigger Disable 1: Trigger Enable		0: Software capture 1: Don't care	Capture timing 00: Disable 01: TIA ↑ TIB ↑ 10: TIA ↑ TIA ↓ 11: TFF1 ↑ TFF1 ↓		Timer 9 up-counter control 0: Clear disabled 1: Clear at match with TREGB	Timer 8 input clock selection 00: TIA input 01: ϕT1 10: ϕT4 11: ϕT16	
			At loading of up-counter value to CAP4	At match between up-counter and TREGB						
T9FFCR	16 bit Timer 9 Flip-Flop Control Register	49H	TFFBC1	TFFBC0	CAP4TA	CAP3TA	EQBTA	EQATA	TFFAC1	TFFAC0
			W		R/W				W	
			1	1	0	0	0	0	1	1
			00: Invert TFFB 01: Set TFFB 10: Clear TFFB 11: Don't care		TFFA inversion trigger 0: Trigger Disable 1: Trigger Enable				00: Invert TFFA 01: Set TFFA 10: Clear TFFA 11: Don't care	
					At loading of up-counter value to CAP4	At loading of up-counter value to CAP3	At match between up-counter and TREGB	At match between up-counter and TREGA		

(4) Serial Channel Control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC0BUF	Serial Channel 0 Buffer Register	4CH	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
			TB7	TB6	TB5	TB4	TB3	TB2	RB1	TB0
			R (receive) /W (send)							
			Undefined							
SC0CR	Serial Channel 0 Control Register	4DH	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (cleared to 0 when read)				
				0	0	0	0	0	0	0
			Bit 8 of receive data	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: Error Overrun	Parity	Framing	0: SCLK0 (↑) 1: SCLK0 (↓)	I/O interface mode clock selection 0: Baud rate generator 1 1: SCLK0 pin input
SC0-MOD	Serial Channel 0 Mode Control Register	4EH	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			Undefined	0	0	0	0	0	0	0
			Bit 8 of send data	Handshake function 0: CTS Disable 1: CTS Enable	Receive control 0: Disable 1: Enable	Wake-up function 0: Disable 1: Enable	Serial transfer mode selection 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		UART mode clock selection 00: TO2 trigger 01: Baud rate generator 0 10: Internal clock φ1 11: SCLK0 pin input (external clock)	
BR0CR	Baud Rate Generater 0 Control Register	4FH	–		BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
			R/W		R/W					
			0		0	0	0	0	0	0
			Note: Always fixed to 0.		Baud rate generator 0 input clock selection 00: φT0 (4/fc) 01: φT2 (16/fc) 10: φT8 (64/fc) 11: φT32 (256/fc)		Baud rate generator 0 divisor setting 0000: Divide by 16 0001: Divide by 1 (no division) to to 1111: Divide by 15			
SC1BUF	Serial Channel 1 Buffer Register	50H	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
			TB7	TB6	TB5	TB4	TB3	TB2	RB1	TB0
			R (receive) /W (send)							
			Undefined							
SC1CR	Serial Channel 1 Control Register	51H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (cleared to 0 when read)			R/W	
				0	0	0	0	0	0	0
			Bit 8 of receive data	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: Error Overrun	Parity	Framing	0: SCLK1 (↑) 1: SCLK1 (↓)	I/O interface mode clock selection 0: Baud rate generato r 1 1: SCLK1 pin input
SC1-MOD	Serial Channel 1 Mode Control Register	52H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			Undefined	0	0	0	0	0	0	0
			Bit 8 of send data	Handshake function 0: CTS Disable 1: CTS Enable	Receive control 0: Disable 1: Enable	Wake-up function 0: Disable 1: Enable	Serial transfer mode selection 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		UART mode clock selection 00: TO2 trigger 01: Baud rate generator 1 10: Internal clock φ1 11: SCLK1 pin input (external clock)	

Serial Channel Control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
BR1CR	Baud Rate Generater 1 Control Register	53H	–		BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
			R/W		R/W					
			0		0	0	0	0	0	0
			Always fixed to 0.		Baud rate generator 1 input clock selection 00: ϕT0 (4/fc) 01: ϕT2 (16/fc) 10: ϕT8 (64/fc) 11: ϕT32 (256/fc)		Baud rate generator 1 divisor setting 0000: Divide by 16 0001: Divide by 1 (no division) to 1111: Divide by 15			
SC2BUF	Serial Channel 2 Buffen Register	54H	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
			TB7	TB6	TB5	TB4	TB3	TB2	RB1	TB0
			R (receive) /W (send)							
			Undefined							
SC2CR	Serial Channel 2 Control Register	55H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (cleared to 0 when read)			R/W	
				0	0	0	0	0	0	0
			Bit 8 of receive data	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: Error Overrun	Parity	Framing	0: SCLK2 (↑) 1: SCLK2 (↓)	I/O interface mode clock selection 0: Baud rate generator 2 1: SCLK2 pin input
SC2-MOD	Serial Channel 2 Mode Control Register	56H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			Undefined	0	0	0	0	0	0	0
			Bit 8 of send data	Handshake function 0:CTS Disable 1:CTS Enable	Receive control 0: Disable 1: Enable	Wake-up function 0: Disable 1: Enable	Serial transfer mode selection 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		UART mode clock selection 00: TO2 trigger 01: Baud rate generator 2 10: Internal clock ϕ1 11: SCLK2 pin input (external clock)	
BR2CR	Baud Rate Generater 2 Control Register	57H	–		BR2CK1	BR2CK0	BR2S3	BR2S2	BR2S1	BR2S0
			R/W		R/W					
			0		0	0	0	0	0	0
			Note: Always fixed to 0.		Baud rate generator 2 input clock selection 00: ϕT0 (4/fc) 01: ϕT2 (16/fc) 10: ϕT8 (64/fc) 11: ϕT32 (256/fc)		Baud rate generator 2 divisor setting 0000: Divide by 16 0001: Divide by 1 (no division) to 1111: Divide by 15			
ODE	Serial Open Drain Enable Register	58H						ODE2	ODE1	ODE0
								R/W		
								0	0	0
								P86 output settings 0: CMOS 1: Open drain	P83 output settings 0: CMOS 1: Open drain	P80 output settings 0: CMOS 1: Open drain

(5) Interrupt Control (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE- 0AD	INT0/AD Enable Register	70H (RMW prohibited)	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R/W	W			R/W Note)	W		
			0	0	0	0	0	0	0	0
INTE12	INT1/2 Enable Register	71H (RMW prohibited)	INT2				INT1			
			I2C	I2M2	I2M1	I2M0	I1C	I1M2	I1M1	I1M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE34	INT3/4 Enable Register	72H (RMW prohibited)	INT4				INT3			
			I4C	I4M2	I4M1	I4M0	I3C	I3M2	I3M1	I3M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE56	INT5/6 Enable Register	73H (RMW prohibited)	INT6				INT5			
			I6C	I6M2	I6M1	I6M0	I5C	I5M2	I5M1	I5M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE78	INT7/8 Enable Register	74H (RMW prohibited)	INT8				INT7			
			I8C	I8M2	I8M1	I8M0	I7C	I7M2	I7M1	I7M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTET01	INTT0/1 Enable Register	75H (RMW prohibited)	INTT1 (timer 1)				INTT0 (timer 0)			
			IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTET23	INTT2/3 Enable Register	76H (RMW prohibited)	INTT3 (timer 3)				INTT2 (timer 2)			
			IT3C	IT3M2	IT3M1	IT3M0	IT2C	IT2M2	IT2M1	IT2M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	
INTET45	INTT4/5 Enable Register	77H (RMW prohibited)	INTT5 (timer 5)				INTT4 (timer 4)			
			IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTET67	INTT6/7 Enable Register	78H (RMW prohibited)	INTT7 (timer 7)				INTT6 (timer 6)			
			IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTET89	INTTR8/9 Enable Register	79H (RMW prohibited)	INTTR9 (timer 8)				INTTR8 (timer 8)			
			IT9C	IT9M2	IT9M1	IT9M0	IT8C	IT8M2	IT8M1	IT8M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTETAB	INTTRA/B Enable Register	7AH (RMW prohibited)	INTTRB (timer 9)				INTTRA (timer 9)			
			ITBC	ITBM2	ITBM1	ITBM0	ITAC	ITAM2	ITAM1	ITAM0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0

IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Disables interrupt request
0	0	1	Sets interrupt request level to 1
0	1	0	Sets interrupt request level to 2
0	1	1	Sets interrupt request level to 3
1	0	0	Sets interrupt request level to 4
1	0	1	Sets interrupt request level to 5
1	1	0	Sets interrupt request level to 6
1	1	1	Disables interrupt request

Note: In INT0 level mode, the interrupt request flag cannot be cleared by writing 0 to <I0C>.

IxxC	Function (Read)	Function (Write)
0	Indicates no interrupt request generated	Clears interrupt request flag
1	Indicates interrupt request generated	- - - - - Don’t care - - - - -

Interrupt Control (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTEOV	INTTO8/9 Enable Register	7BH (RMW prohibited)	INTTO9				INTTO8			
			ITO9C	ITO9M2	ITO9M1	ITO9M0	ITO8C	ITO8M2	ITO8M1	ITO8M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTES0	INTRX0/ TX0 Enable Register	7CH (RMW prohibited)	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R/W	W			R (Note)	W		
			0	0	0	0	0	0	0	0
INTES1	INTRX1/ TX1 Enable Register	7DH (RMW prohibited)	INTTX1				INTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R/W	W			R (Note)	W		
			0	0	0	0	0	0	0	0
INTES2	INTRX2/ TX2 Enable Register	7EH (RMW prohibited)	INTTX2				INTRX2			
			ITX2C	ITX2M2	ITX2M1	ITX2M0	IRX2C	IRX2M2	IRX2M1	IRX2M0
			R/W	W			R (Note)	W		
			0	0	0	0	0	0	0	0
INTETC 01	INTTC0/1 Enable Register	7FH (RMW prohibited)	INTTC1				INTTC0			
			ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC01C	ITC0M2	ITC0M1	ITC0M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTETC 23	INTTC2/3 Enable Register	80H (RMW prohibited)	INTTC3				INTTC2			
			ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
			R/W	W			R/W	W		
			0	0	0	0	0	0	0	0

IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Disables interrupt request
0	0	1	Sets interrupt request level to 1
0	1	0	Sets interrupt request level to 2
0	1	1	Sets interrupt request level to 3
1	0	0	Sets interrupt request level to 4
1	0	1	Sets interrupt request level to 5
1	1	0	Sets interrupt request level to 6
1	1	1	Disables interrupt request

Note: As <IRX0C>, <IRX1C>, and <IRX2C> are read-only, an interrupt request cannot be cleared by writing 0 to these flags.

IxxC	Function (Read)	Function (Write)
0	Indicates no interrupt request generated	Clears interrupt request flag
1	Indicates interrupt request generated	- - - - - Don’t care - - - - -

Interrupt Control (3/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
IIMC	Interrupt Input Mode Contorol Register	59H (RMW prohibited)			–			I0IE	I0LE	NMIREE
					W			W		
					0			0	0	0
					Note: Always set to 0			INT0 input 0: Disable 1: Enable	INT0 0: ↑ edge 1: level	$\overline{\text{NMI}}$ 0: ↓ edge 1: ↑ ↓ edge
DMA0V	Micro DMA 0 Start Vector Register	5AH (RMW prohibited)	DMA0V7	DMA0V6	DMA0V5	DMA0V4	DMA0V3	DMA0V2		
			W							
			0	0	0	0	0	0		
			Micro DMA0 start vector							
DMA1V	Micro DMA 1 Start Vector Register	5BH (RMW prohibited)	DMA1V7	DMA1V6	DMA1V5	DMA1V4	DMA1V3	DMA1V2		
			W							
			0	0	0	0	0	0		
			Micro DMA1 start vector							
DMA2V	Micro DMA 2 Start Vector Register	5CH (RMW prohibited)	DMA2V7	DMA2V6	DMA2V5	DMA2V4	DMA2V3	DMA2V2		
			W							
			0	0	0	0	0	0		
			Micro DMA2 start vector							
DMA3V	Micro DMA 3 Start Vector Register	5DH (RMW prohibited)	DMA3V7	DMA3V6	DMA3V5	DMA3V4	DMA3V3	DMA3V2		
			W							
			0	0	0	0	0	0		
			Micro DMA3 start vector							

Note: The micro DMA software start is activated in the write cycle of SDMACR0/1/2/3 (6AH/6BH/6CH/6DH). (Data values are not affected by a software start.)

(6) Watchdog Timer Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
WD-MOD	Watch Dog Timer Mode Control Register	6EH	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
			R/W							
			1	0	0	0	0	0	0	
			WDT control 0: Disable 1: Enable	WDT detection time selection 00: 2^{16}/fc 01: 2^{18}/fc 10: 2^{20}/fc 11: 2^{22}/fc		Warm-up time 0: 2^{14}/fc 1: 2^{16}/fc	HALT mode selection 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		1: Perform internal reset on runaway detection	1: Drive pins in STOP mode
WDCR	Watch Dog Timer Control Register	6FH (RMW prohibited)	–							
			W							
			–							
			B1H: WDT disable code 4EH: WDT clear code							

(7) Chip Select/Wait Controller (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
B0CS	Block 0 CS/WAIT Control Register	90H (RMW prohibited)	B0E		B0OM1	B0OM0	B0BUS	B0W2	B0W1	B0W0
			W		W					
			0		0	0	0	0	0	0
			0: Disable 1: Enable		00: ROM/SRAM 01: PSRAM 10: Don’t care 11: Don’t care		Data bus width selection 0: 16-bit 1: 8-bit	000: 2WAIT 001: 1WAIT 010: 1WAIT + N 011: 0WAIT	100: NWAIT 101 110 Do not set 111	
B1CS	Block 1 CS/WAIT Control Register	91H (RMW prohibited)	B1E		B1OM1	B1OM0	B1BUS	B1W2	B1W1	B1W0
			W		W					
			0		0	0	0	0	0	0
			0: Disable 1: Enable		00: ROM/SRAM 01: PSRAM 10: Don’t care 11: Don’t care		Data bus width selection 0: 16-bit 1: 8-bit	000: 2WAIT 001: 1WAIT 010: 1WAIT + N 011: 0WAIT	100: NWAIT 101 110 Do not set 111	
B2CS	Block 2 CS/WAIT Control Register	92H (RMW prohibited)	B2E	B2M	B2OM1	B2OM0	B2BUS	B2W2	B2W1	B2W0
			W							
			1	0	0	0	0	0	0	0
			0: Disable 1: Enable	0: 16M 1: CS area setting	00: ROM/SRAM 01: PSRAM 10: Don’t care 11: Don’t care		Data bus width selection 0: 16-bit 1: 8-bit	000: 2WAIT 001: 1WAIT 010: 1WAIT + N 011: 0WAIT	100: NWAIT 101 110 Do not set 111	
B3CS	Block 3 CS/WAIT Control Register	93H (RMW prohibited)	B3E		B3OM1	B3OM0	B3BUS	B3W2	B3W1	B3W0
			W		W					
			0		0	0	0	0	0	0
			0: Disable 1: Enable		00: ROM/SRAM 01: PSRAM 10: Don’t care 11: Don’t care		Data bus width selection 0: 16-bit 1: 8-bit	000: 2WAIT 001: 1WAIT 010: 1WAIT + N 011: 0WAIT	100: NWAIT 101 110 Do not set 111	
BEXCS	External CS/WAIT Control Register	9CH (RMW prohibited)					BEXBUS	BEXBUS	BEXW1	BEXW0
							W			
							0	0	0	0
							Data bus width selection 0: 16-bit 1: 8-bit	000: 2WAIT 001: 1WAIT 010: 1WAIT + N 011: 0WAIT	100: NWAIT 101 110 Do not set 111	
MSAR0	Memory Start Address Register 0	94H	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			Start address A23 to A16 setting							
MAMR0	Memory Address Mask Register 0	95H	V20	V19	V18	V17	V16	V15	V14 to 9	V8
			R/W							
			1	1	1	1	1	1	1	1
			CS0 area size setting 0: Used for address comparison							
MSAR1	Memory Start Address Register 1	96H	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			Start address A23 to A16 setting							
MAMR1	Memory Address Mask Register 1	97H	V21	V20	V19	V18	V17	V16	V15 to 9	V8
			R/W							
			1	1	1	1	1	1	1	1
			CS1 area size setting 0: Used for address comparison							

Chip Select/Wait Controller (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
MSAR2	Memory Start Address Register 2	98H	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			Start address A23 to A16 setting							
MAMR2	Memory Address Mask Register 2	99H	V22	V21	V20	V19	V18	V17	V16	V15
			R/W							
			1	1	1	1	1	1	1	1
			CS2 area size setting 0: Used for address comparison							
MSAR3	Memory Start Address Register 3	9AH	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			Start address A23 to A16 setting							
MAMR3	Memory Address Mask Register 3	9BH	V22	V21	V20	V19	V18	V17	V16	V15
			R/W							
			1	1	1	1	1	1	1	1
			CS3 area size setting 0: Used for address comparison							

(8) D/A Converter Control

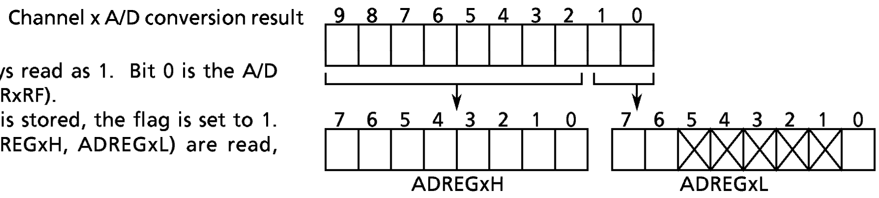
Symbol	Name	Address	7	6	5	4	3	2	1	0
DADRV	D/A Conversion Drive Register	9DH							DA1DR	DA0DR
									R/W	
									0	
									DAOUT1 drive specification	DAOUT0 drive specification
									0: Fixed to 0V output 1: D/A conversion result output	
DAREG0	D/A Conversion Register 0	9EH (RMW prohibited)	–							
			W							
			Undefined							
			D/A converter 0 input data “N” setting							
DAREG1	D/A Conversion Register 1	9FH (RMW prohibited)	–							
			W							
			Undefined							
			D/A converter 1 input data “N” setting							

(9) A/D Converter Control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
ADMOD 0	A/D Mode Control Register 0	5EH	EOCF	ADBF	–	–	ITM0	REPET	SCAN	ADS
			R		R/W					
			0	0	0	0	0	0	0	0
			A/D conversion end flag 0:Conversion in progress 1:Conversion complete	A/D conversion busy flag 0:Conversion idle 1:Conversion in progress	Note: Always fixed to 0.	Note: Always fixed to 0.	Interrupt specification in channel fixed repeat conversion mode 0:Every conversion 1:Every fourth conversion	Repeat mode specification 0:Single conversion mode 1:Repeat conversion mode	Scan mode specification 0:Conversion channel fixed mode 1:Conversion channel scan mode	A/D conversion start 0:Don’t Care 1:Conversion start Note: Always read as 0.

A/D Converter Control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
ADMOD 1	A/D Mode Control Register 1	5FH	VREFON				ADTRGE	ADCH2	ADCH1	ADCH0
			R/W				R/W			
			1				0	0	0	0
			VREF application control 0:OFF 1:ON				External trigger start control 0:Enable 1:Disable	Analog input channel selection		
AD REG04L	A/D Conversion Result Register 0/4 Low	60H	ADR01	ADR00						ADR0RF
			R							R
			Undefined							0
			Stores lower 2 bits of A/D conversion result							A/D conversion result storage flag
AD REG04H	A/D Conversion Result Register 0/4 High	61H	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
			R							
			Undefined							
			Stores upper 8 bits of A/D conversion result							
AD REG15L	A/D Conversion Result Register 1/5 Low	62H	ADR11	ADR10						ADR1RF
			R							R
			Undefined							0
			Stores lower 2 bits of A/D conversion result							A/D conversion result storage flag
AD REG15H	A/D Conversion Result Register 1/5 High	63H	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
			R							
			Undefined							
			Stores upper 8 bits of A/D conversion result							
AD REG26L	A/D Conversion Result Register 2/6 Low	64H	ADR21	ADR20						ADR2RF
			R							R
			Undefined							0
			Stores lower 2 bits of A/D conversion result							A/D conversion result storage flag
AD REG26H	A/D Conversion Result Register 2/6 High	65H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
			R							
			Undefined							
			Stores upper 8 bits of A/D conversion result							
AD REG37L	A/D Conversion Result Register 3/7 Low	66H	ADR31	ADR30						ADR3RF
			R							R
			Undefined							0
			Stores lower 2 bits of A/D conversion result							A/D conversion result storage flag
AD REG37H	A/D Conversion Result Register 3/7 High	67H	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
			R							
			Undefined							
			Stores upper 8 bits of A/D conversion result							



- Bits 5 to 1 of ADREGxL are always read as 1. Bit 0 is the A/D conversion result storage flag (ADRxRF).
 When the A/D conversion result is stored, the flag is set to 1. Wheneither of the registers (ADREGxH, ADREGxL) are read, the flag is cleared to 0.

6. Diagram of Equivalent Circuit in Port Block

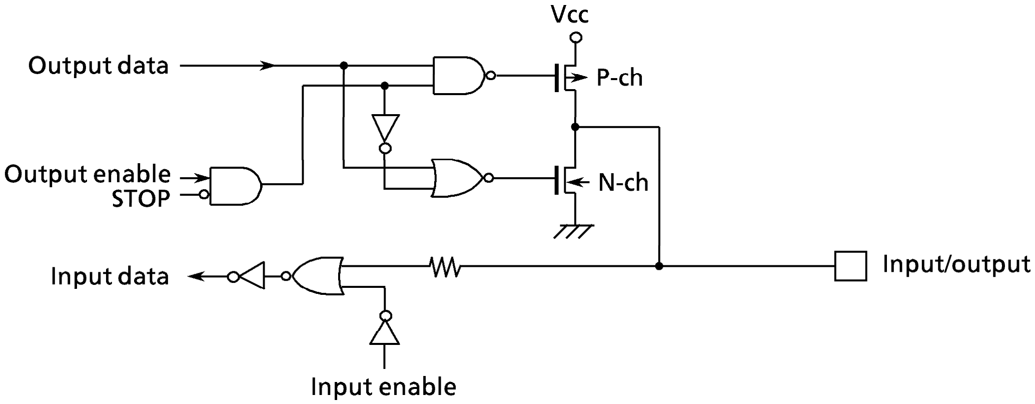
- Reading circuit diagrams

TMP95CS64/265 uses essentially the same gate symbols as the standard CMOS logic IC (74HCxxx) series.
The following lists the special symbols.

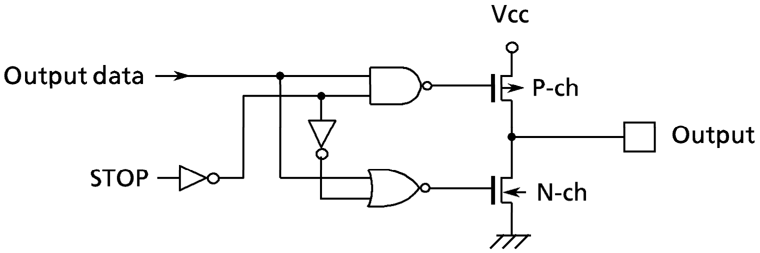
STOP: This symbol sets the HALT mode setting register to STOP mode (WDMOD<HALTM1:0>=0,1). When the CPU executes the HALT instruction, STOP is active 1.
Note that when the drive enable bit WDMOD<DRVE> is set to 1, STOP remains at 0.

- The input protection resistor operates in the range of tens to hundreds of Ω ms.

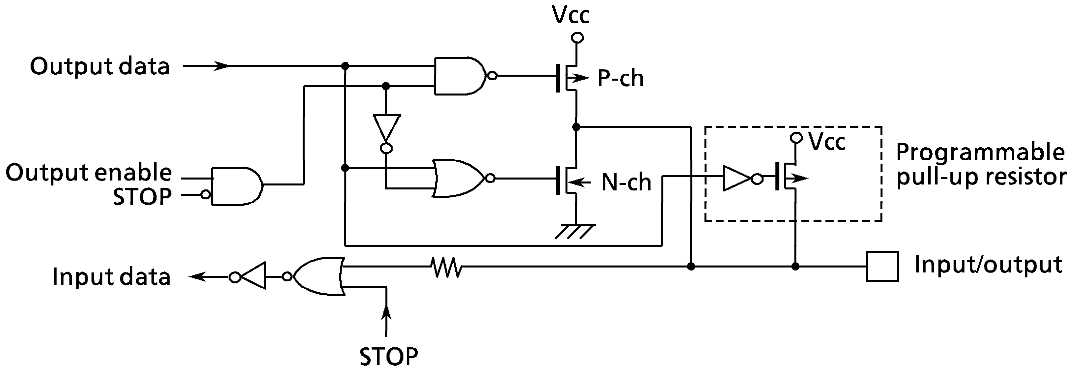
■ P0 (D0 to D7), P1 (D8 to 15), P2 (A16 to A23), P3 (A8 to A15), P4 (A0 to A7)



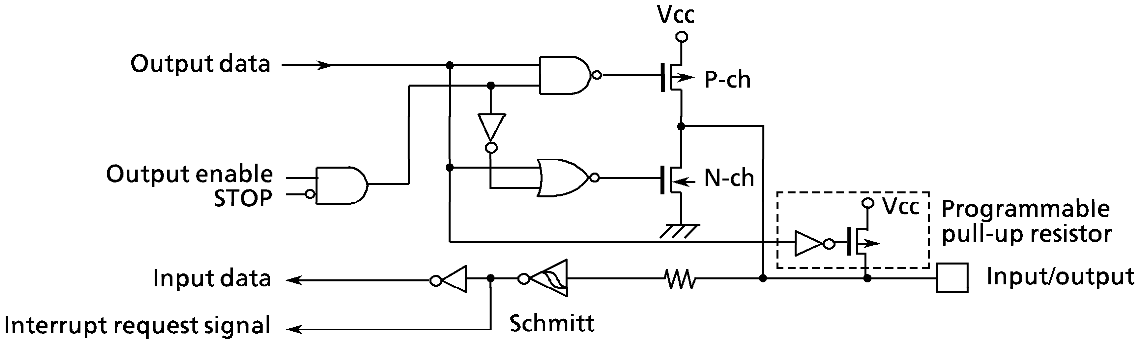
■ P50 ($\overline{\text{RD}}$), P51 ($\overline{\text{WR}}$), P6 ($\overline{\text{CS0}}$ to $\overline{\text{CS3}}$)



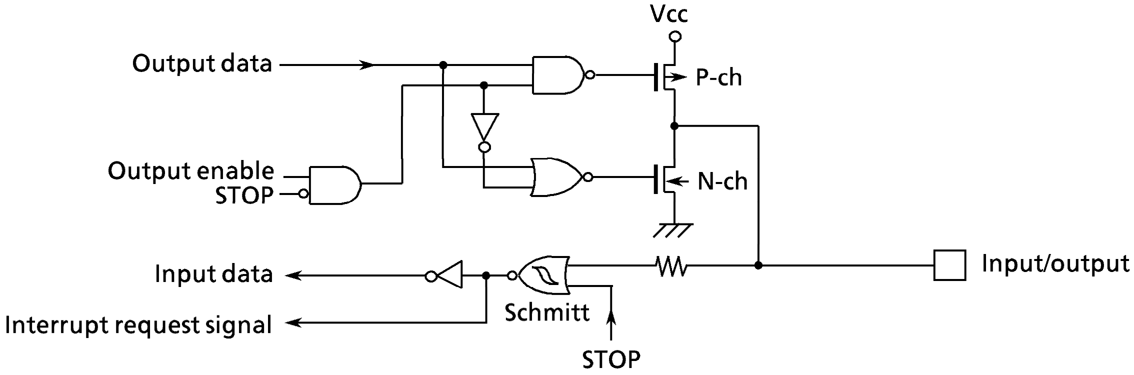
■ P52 to 55, P57, P81, P82, P84, P85, P87



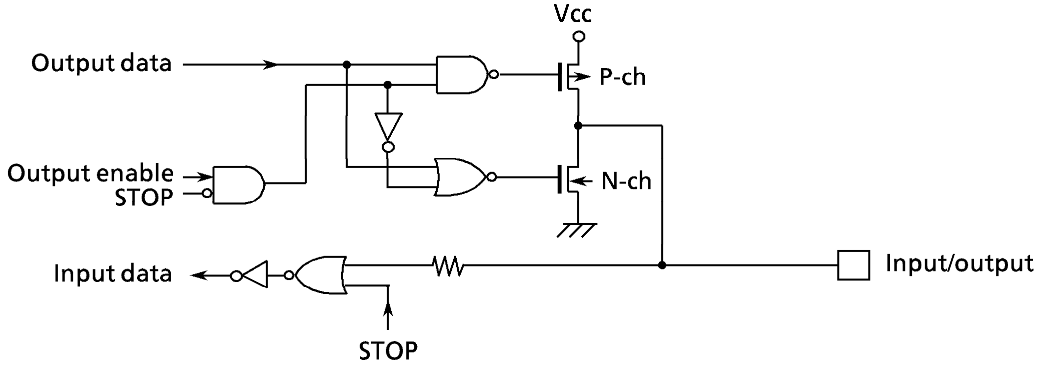
■ P56 (INT0)



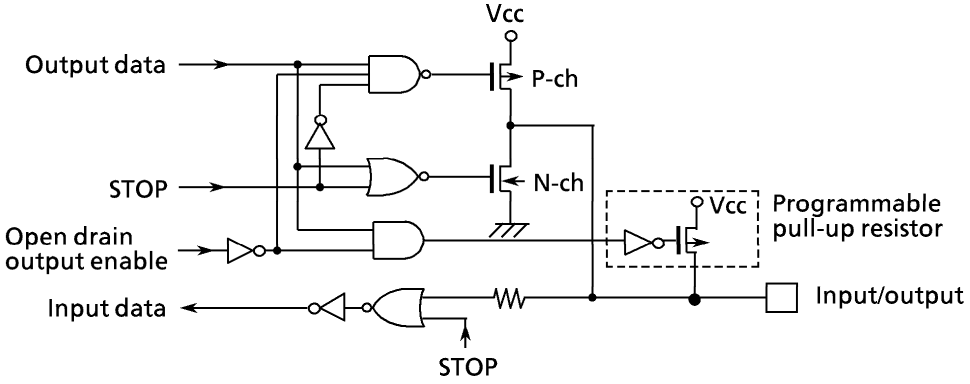
■ P70 (INT1), P72 (INT2), P73 (INT3), P75 (INT4)



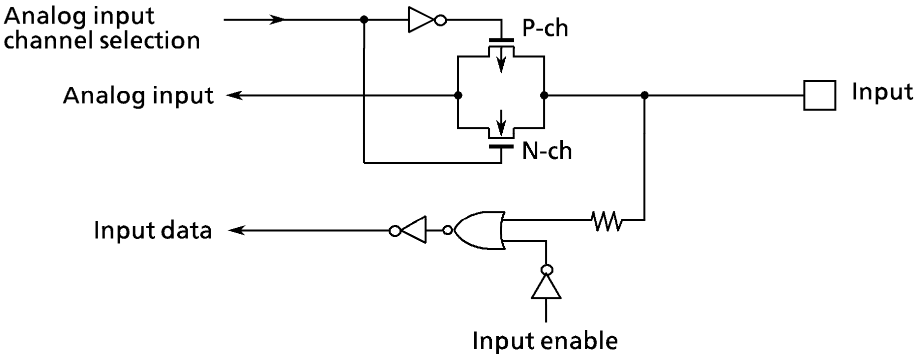
■ P71, P74, P9



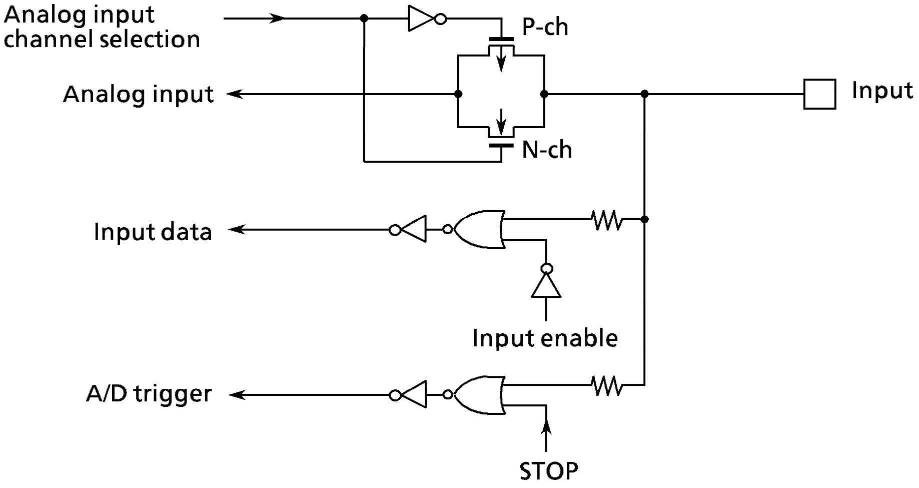
■ P80 (TxD0), P83 (TxD1), P86 (TxD2)



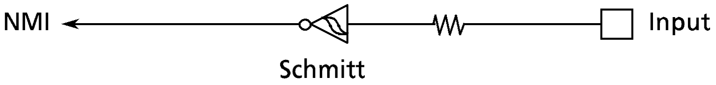
■ PA0 to 2 (AN0 to 2), PA4 to 7 (AN4 to 7)



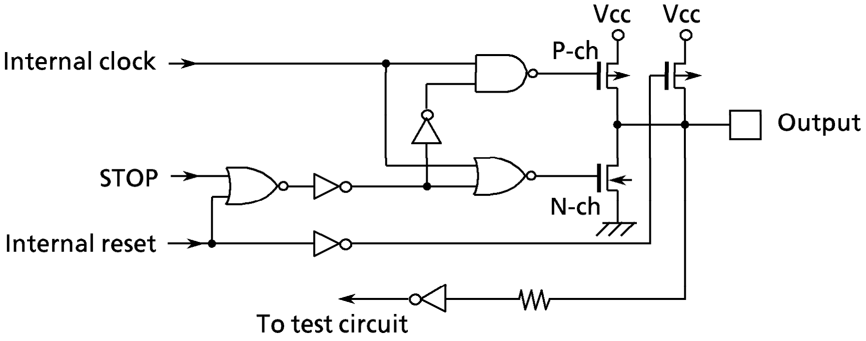
■ PA3 (AN3)



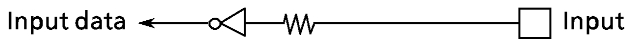
■ $\overline{\text{NMI}}$



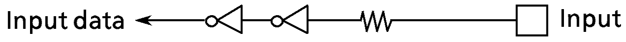
■ CLK



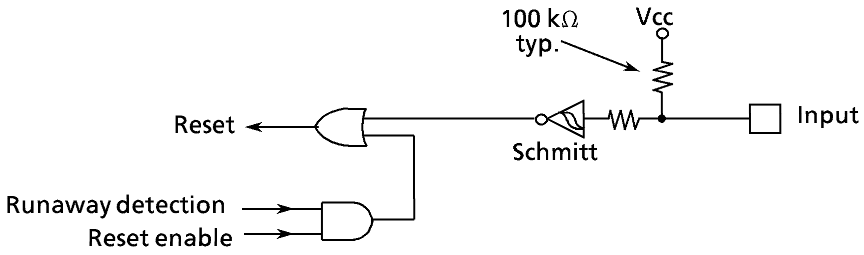
■ $\overline{\text{EA}}$



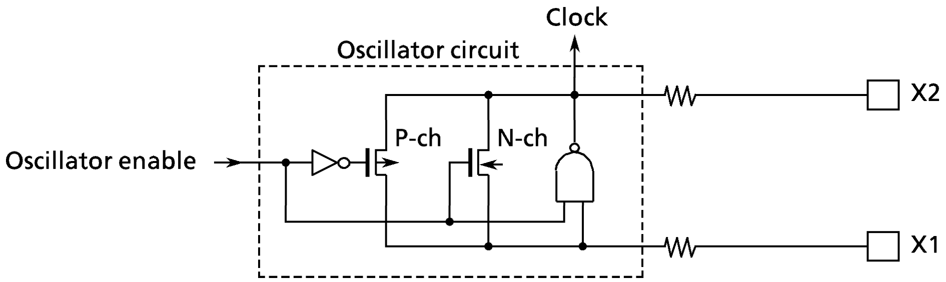
■ AM8/$\overline{\text{16}}$



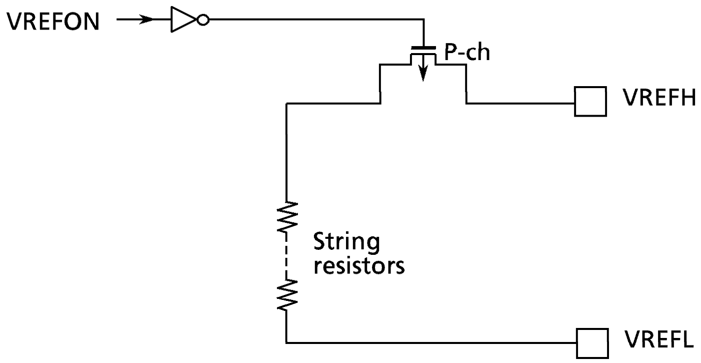
■ $\overline{\text{RESET}}$



■ X1, X2



■ VREFH, VREFL



7. Use Precautions and Restrictions

(1) Special Notations and Words

① Description of internal I/O registers: Register symbol<bit symbol>

Example: T8RUN<T0RUN> ··· The T0RUN bit of the T8RUN register

② Read-modify-write instructions

Instructions which tell the CPU to read the data in memory, manipulate them, then write them back to memory are called read-modify-write instructions.
Example 1) SET 3, (T8RUN) ··· Sets bit 3 of the T8RUN register.
Example 2) INC 1, (100H) ··· Adds 1 to the data at address 100H.

- TLCS-900 read-modify-write instructions

```
Conversion instruction
    EX     (mem), R
Arithmetic operations
    ADD  (mem), R/#       ADC   (mem), R/#
    SUB  (mem), R/#       SBC   (mem), R/#
    INC  #3, (mem)        DEC   #3, (mem)
Logic operations
    AND  (mem), R/#       OR    (mem), R/#
    XOR  (mem), R/#
Bit manipulation
    STCF #3/A, (mem)      SET   #3, (mem)
    RES  #3, (mem)        TEST #3, (mem)
    CHG  #3, (mem)
Rotate, shift
    RLC  (mem)            RRC   (mem)
    RL   (mem)            RR(mem)
    SLA  (mem)            SRA   (mem)
    SLL  (mem)            SRL   (mem)
    RLD  (mem)            RRD   (mem)
```

③ One state

The single cycle resulting from dividing the oscillation frequency by 2 is called “one state”.
Example: At oscillation frequency 25 MHz
2/25 MHz = 80 ns = 1 state

(2) Points of Note and Restrictions

① $\overline{\text{EA}}$ pin, AM8/$\overline{16}$ pin

This pin is connected to the VCC or the GND pin. Do not alter the level while the pin is active.

② Warm-up counter

When releasing STOP mode (by interrupt, for example) in a system that uses an external oscillator, a warm-up time is required until the system clock is output. The warm-up counter operates during the warm-up time.

③ Programmable pull-up resistor

The pull-up resistor of a port can only be set to programmable or non-programmable in input port mode. When using a port as an output port, its pull-up resistor cannot be set to programmable.

④ Watchdog timer

As the watchdog timer is enabled after a reset, disable the watchdog timer when it is not required.
Note that during bus release, the I/O block, including the watchdog timer, still operate.

⑤ CPU (Micro DMA)

Only "LDC cr, r" and "LDC r, cr" can write or read data to or from control registers (eg, transfer source register DMASx) in the CPU.

⑥ As this device does not support minimum mode, do not use the MIN instruction.

⑦ POP SR instruction

Please execute POP SR instruction during DI condition.

⑧ Releasing the HALT mode by requesting an interruption

Usually, interrupts can release all halts status. However, the interrupts = ($\overline{\text{NMI}}$, INT0), which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 3 clocks of X1) with IDLE1 or STOP mode (IDLE2 is not applicable to this case). (In this case, an interrupt request is kept on hold internally.)

If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficultly. The priority of this interrupt is compare with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.