TOSHIBA

TOSHIBA Original CMOS 32-Bit Microcontroller

TLCS-900/H2 Series

TMP94C251A

TOSHIBA CORPORATION

Semiconductor Company

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 IDLE or STOP mode (RUN 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 32-Bit Microcontroller

TMP94C251AF

1. Outline and Device Characteristics

TMP94C251A is high-speed advanced 32-bit microcontroller developed for controlling equipment which processes mass data.

TMP94C251A is a microcontroller which has a high-performance CPU (900/H2 CPU) and various built-in I/Os. And TMP94C251A is enhanced memory interface functions. TMP94C251AF is housed in an 144-pin mini flat package.

Device characteristics are as follows:

(1) CPU: 32-bit CPU (900/H2 CPU)
- Compatible with TLCS-900, 900/L, 900/L1, 900/H's instruction code
- 16 Mbytes of linear address space
- General-purpose registers and register banks
- Micro DMA: 8 channels (250 ns/4 bytes at 20 MHz)

(2) Minimum instruction execution time: 50 ns (at 20 MHz)

(3) Internal memory
- Internal RAM: 2 Kbytes (can use for code section)
- Internal ROM: None

(4) External memory expansion
- Expandable up to 16 Mbytes (shared program/data area)
- Can simultaneously support 8-/16-bit width external data bus

(5) Memory controller
- Chip select output: 6 channels

(6) DRAM Controller: 2 channels
- Direct interface (supported 8-/16-bit external data bus)

(7) 8-bit timer: 4 channels

(8) 16-bit timer: 4 channels

(9) Serial interface: 2 channels

(10) 10-bit AD converter: 8 channels (with sample hold circuit)

(11) 8-bit DA converter: 2 channels (with CMOS-AMP)

(12) Watchdog timer

(13) Interrupt controller

- 18 internal interrupts
- 10 external interrupts

(14) I/O port: 64 pins

(15) Package: 144-pin QFP (P-QFP144-2020-0.50)

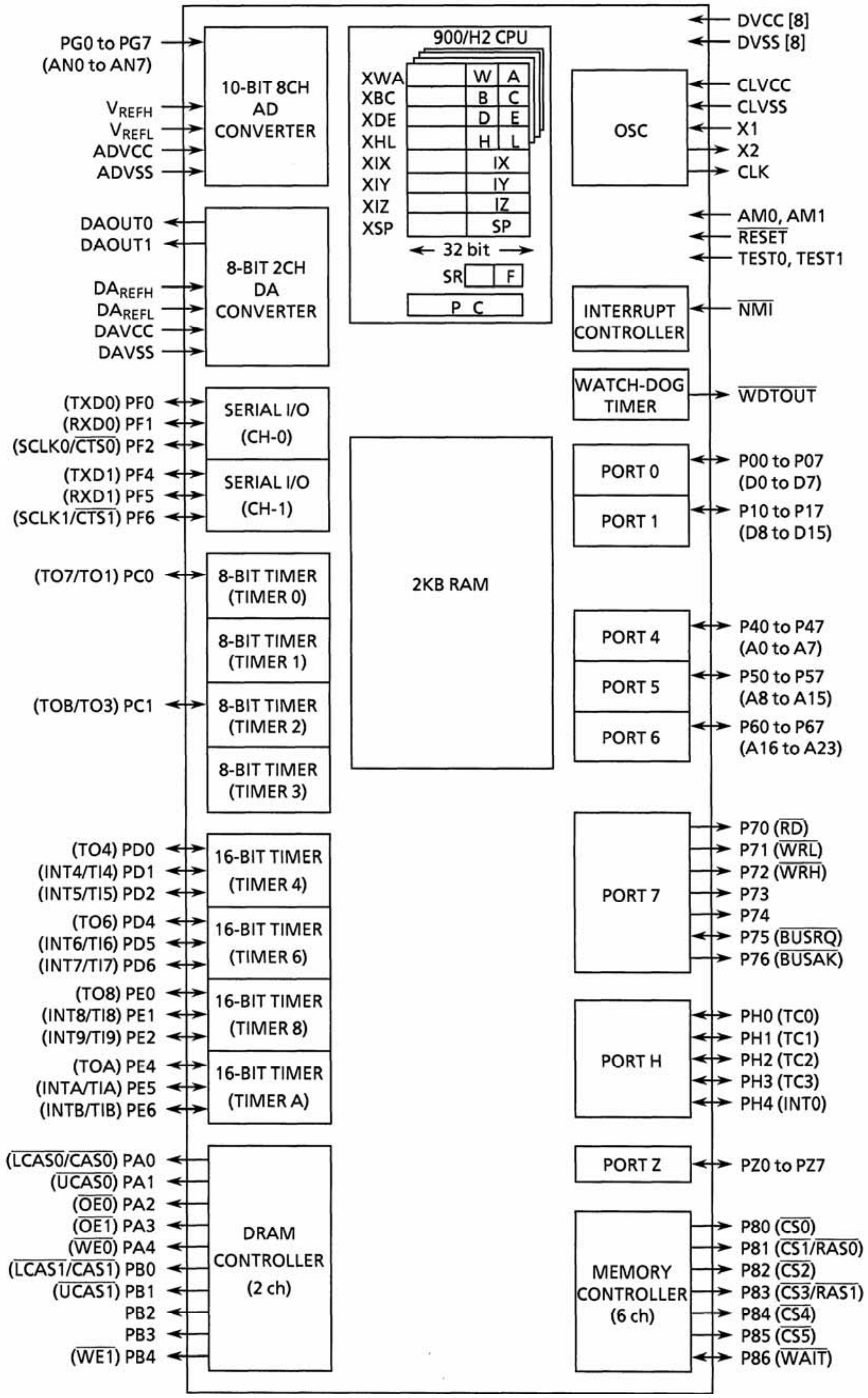


Figure 1.1 TMP94C251A Block Diagram

2. Pin Assignment and Functions

2.1 Pin Assignment (Top view)

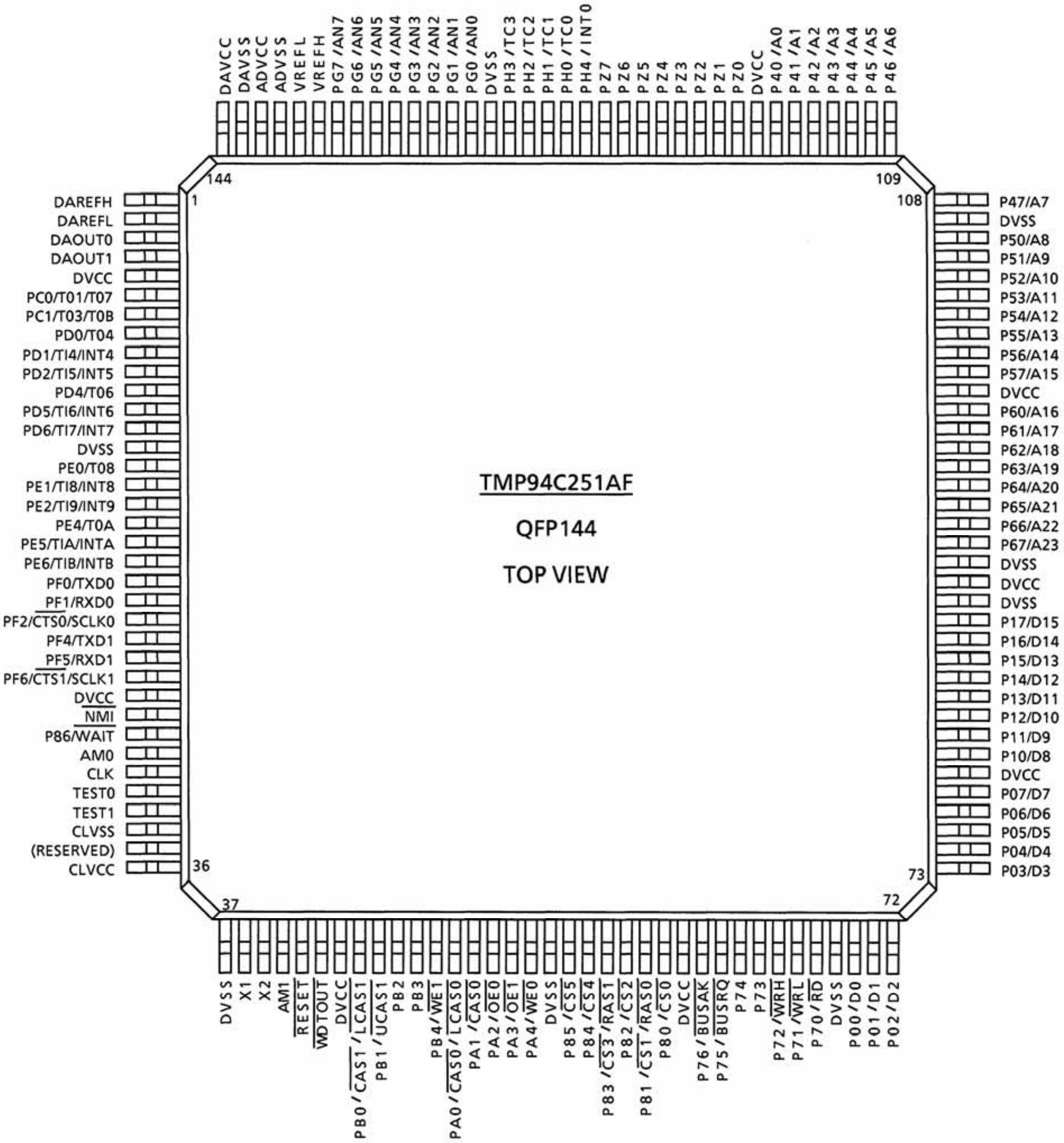


Figure 2.1.1 Pin Assignment

2.2 Pin Names and Functions

The names of input/output pins and their functions are described below.

Table 2.2.1 Pin Names and Functions (1/6)

Pin Name	Number of Pins	I/O	Functions
P00 to P07 D0 to D7	8 (TTL)	I/O I/O	Port 0: I/O port Data: 0 to 7 for data bus TMP94C251A is external ROM type, these pins are initialized to this function. When TMP94C251A doesn't access external memories, these pins are put in the high-impedance state.
P10 to P17 D8 to D15	8 (TTL)	I/O I/O	Port 1: I/O port Data: 8 to15 for data bus If TMP94C251A is external ROM type and is start with 16-bit data bus, these pins are initialized to this function. When TMP94C251A doesn't access external memories, these pins are put in the high-impedance state.
P40 to P47 A0 to A7	8	I/O Output	Port 4: I/O port Address: 0 to 7 for address bus TMP94C251A is external ROM type, these pins are initialized to this function. When TMP94C251A doesn't access external memories, these pins don't change.
P50 to P57 A8 to A15	8	I/O Output	Port 5: I/O port Address: 8 to 15 for address bus TMP94C251A is external ROM type, these pins are initialized to this function. When TMP94C251A doesn't access external memories, these pins don't change.
P60 to P67 A16 to A23	8	I/O Output	Port 6: I/O port Address: 16 to 23 for address bus TMP94C251A is external ROM type, these pins are initialized to this function. When TMP94C251A doesn't access external memories, these pins don't change.
P70 $\overline{\text{RD}}$	1	Output Output	Port 70: Output port (output "high" when initialized) Read: Strobe signal for reading external memory When TMP94C251A doesn't access external memory, doesn't output strobe. TMP94C251A is external ROM type, these pins are initialized to this function.
P71 $\overline{\text{WRL}}$	1	Output Output	Port 71: Output port (output "high" when initialized) Write LL: Strobe signal for writing data on pins D0 to D7 When TMP94C251A doesn't access external memory, doesn't output strobe.
P72 $\overline{\text{WRH}}$	1	Output Output	Port 72: Output port (output "high" when initialized) Write LH: Strobe signal for writing data on pins D8 to D15 When TMP94C251A doesn't access external memory, doesn't output strobe.
P73	1	Output	Port 73: Output port (output "high" when initialized)
P74	1	Output	Port 74: Output port (output "high" when initialized)

Table 2.2.2 Pin Names and Functions (2/6)

Pin Name	Number of Pins	I/O	Functions
P75 $\overline{\text{BUSRQ}}$	1	I/O Input	Port 75: I/O port Bus request: Signal used to request high impedance for memory interface signals. If these signals are used as port, there are not change. The memory interface signals are follows: A0 to A23, D0 to D15, $\overline{\text{RD}}$, $\overline{\text{WRLL}}$, $\overline{\text{WRLH}}$ The output signals of memory controller.
P76 $\overline{\text{BUSAK}}$	1	Output Output	Port 76: Output port (output "high" when initialized) Bus acknowledge: Signal indicating that request of $\overline{\text{BUSRQ}}$ signal is accepted.
P80 $\overline{\text{CS0}}$	1	Output Output	Port 80: Output port (output "high" when initialized) Chip select 0: Outputs "low" if address is within specified address area.
P81 $\overline{\text{CS1}}$ $\overline{\text{RAS0}}$	1	Output Output Output	Port 81: Output port (output "high" when initialized) Chip select 1: Outputs "low" if address is within specified address area. Row address strobe 0: Outputs $\overline{\text{RAS}}$ strobe for DRAM if address is within specified address area.
P82 $\overline{\text{CS2}}$	1	Output Output	Port 82: Output port (output "high" when initialized) Chip select 2: Outputs "low" if address is within specified address area.
P83 $\overline{\text{CS3}}$ $\overline{\text{RAS1}}$	1	Output Output Output	Port 83: Output port (output "high" when initialized) Chip select 3: Outputs "low" if address is within specified address area. Row address strobe 1: Outputs $\overline{\text{RAS}}$ strobe for DRAM if address is within specified address area.
P84 $\overline{\text{CS4}}$	1	Output Output	Port 84: Output port (output "high" when initialized) Chip select 4: Outputs "low" if address is within specified address area.
P85 $\overline{\text{CS5}}$	1	Output Output	Port 85: Output port (output "high" when initialized) Chip select 5: Outputs "low" if address is within specified address area.
P86 $\overline{\text{WAIT}}$	1	I/O Input	Port 86: I/O port Wait: Signal used to request CPU bus wait.

Table 2.2.3 Pin Names and Functions (3/6)

Pin Name	Number of Pins	I/O	Functions
PA0 $\overline{CAS0}$ $\overline{LCAS0}$	1	Output Output Output	Port A0: Output port (output “high” when initialized) Column address strobe 0: Outputs $\overline{CAS}$ strobe for DRAM if address is within specified address area. Lower column address strobe 0: Outputs lower $\overline{CAS}$ strobe for DRAM if address is within specified address area.
PA1 $\overline{UCAS0}$	1	Output Output	Port A1: Output port (output “high” when initialized) Upper Column address strobe 0: Outputs upper $\overline{CAS}$ strobe for DRAM if address is within specified address area.
PA2 $\overline{OE0}$	1	Output Output	Port A2: Output port (output “high” when initialized) Output enable 0: Outputs read enable signal for DRAM.
PA3 $\overline{OE1}$	1	Output Output	Port A3: Output port (output “high” when initialized) Output enable 1: Outputs read enable signal for DRAM.
PA4 $\overline{WE0}$	1	Output Output	Port A4: Output port (output “high” when initialized) Write enable 0: Outputs write enable signal for DRAM.
PB0 $\overline{CAS1}$ $\overline{LCAS1}$	1	Output Output Output	Port B0: Output port (output “high” when initialized) Column address strobe 1: Outputs $\overline{CAS}$ strobe for DRAM if address is within specified address area. Lower column address strobe 1: Outputs lower $\overline{CAS}$ strobe for DRAM if address is within specified address area.
PB1 $\overline{UCAS1}$	1	Output Output	Port B1: Output port (output “high” when initialized) Upper Column address strobe 1: Outputs upper $\overline{CAS}$ strobe for DRAM if address is within specified address area.
PB2	1	Output	Port B2: Output port (output “high” when initialized)
PB3	1	Output	Port B3: Output port (output “high” when initialized)
PB4 $\overline{WE1}$	1	Output Output	Port B4: Output port (output “high” when initialized) Write enable 1: Outputs write enable signal for DRAM.

Table 2.2.4 Pin Names and Functions (4/6)

Pin Name	Number of Pins	I/O	Functions
PC0 TO1 TO7	1	I/O Output Output	Port C0: I/O port Timer output 1: 8-bit timer 0 or 1 output Timer output 7: 16-bit timer 7 output
PC1 TO3 TOB	1	I/O Output Output	Port C1: I/O port Timer output 3: 8-bit timer 2 or 3 output Timer output B: 16-bit timer B output
PD0 TO4	1	I/O Output	Port D0: I/O port Timer output 4: 16-bit timer 4 output
PD1 TI4 INT4	1	I/O Input Input	Port D1: I/O port Timer input 4: 16-bit timer 4 input Interrupt request pin 4: Interrupt request pin with programmable rising/falling edge
PD2 TI5 INT5	1	I/O Input Input	Port D2: I/O port Timer input 5: 16-bit timer 4 input Interrupt request pin 5: Interrupt request pin with rising edge
PD4 TO6	1	I/O Output	Port D4: I/O port Timer output 6: 16-bit timer 6 output
PD5 TI6 INT6	1	I/O Input Input	Port D5: I/O port Timer input 6: 16-bit timer 6 input Interrupt request pin 6: Interrupt request pin with programmable rising/falling edge
PD6 TI7 INT7	1	I/O Input Input	Port D6: I/O port Timer input 7: 16-bit timer 6 input Interrupt request pin 7: Interrupt request pin with rising edge
PE0 TO8	1	I/O Output	Port E0: I/O port Timer output 8: 16-bit timer 8 output
PE1 TI8 INT8	1	I/O Input Input	Port E1: I/O port Timer input 8: 16-bit timer 8 input Interrupt request pin 8: Interrupt request pin with programmable rising/falling edge
PE2 TI9 INT9	1	I/O Input Input	Port E2: I/O port Timer input 9: 16-bit timer 8 input Interrupt request pin 9: Interrupt request pin with rising edge
PE4 TOA	1	I/O Output	Port E4: I/O port Timer output A: 16-bit timer A output
PE5 TIA INTA	1	I/O Input Input	Port E5: I/O port Timer input A: 16-bit timer A input Interrupt request pin A: Interrupt request pin with programmable rising/falling edge
PE6 TIB INTB	1	I/O Input Input	Port E6: I/O port Timer input B: 16-bit timer A input Interrupt request pin B: Interrupt request pin with rising edge

Table 2.2.5 Pin Names and Functions (5/6)

Pin Name	Number of Pins	I/O	Functions
PF0 TXD0	1	I/O Output	Port F0: I/O port Serial send data 0
PF1 RXD0	1	I/O Input	Port F1: I/O port Serial receive data 0
PF2 $\overline{\text{CTS0}}$ SCLK0	1	I/O Input I/O	Port F2: I/O port Serial data receive enable 0 Serial clock I/O 0
PF4 TXD1	1	I/O Output	Port F4: I/O port Serial send data 1
PF5 RXD1	1	I/O Input	Port F5: I/O port Serial receive data 1
PF6 $\overline{\text{CTS1}}$ SCLK1	1	I/O Input I/O	Port F6: I/O port Serial data receive enable 1 Serial clock I/O 1
PG0 to PG7 AN0 to AN7	8	Input Input	Port G: Input port Analog input: Input to 10-bit AD converter
DAOUT0	1	Output	DA output 0: Output from 8-bit DA converter 0
DAOUT1	1	Output	DA output 1: Output from 8-bit DA converter 1
PH0 TC0	1	I/O Output	Port H0: I/O port Terminal count 0: Outputs “high” strobe when counter value of micro-DMA channel 0 is “0”.
PH1 TC1	1	I/O Output	Port H1: I/O port Terminal count 1: Outputs “high” strobe when counter value of micro-DMA channel 1 is “0”.
PH2 TC2	1	I/O Output	Port H2: I/O port Terminal count 2: Outputs “high” strobe when counter value of micro-DMA channel 2 is “0”.
PH3 TC3	1	I/O Output	Port H3: I/O port Terminal count 3: Outputs “high” strobe when counter value of micro-DMA channel 3 is “0”.
PH4 INT0	1	I/O Input	Port H4: I/O port (schmitt input) Interrupt request pin 0: Interrupt request pin with programmable level/rising edge. (schmitt input)
PZ0 to PZ7	8	I/O	Port Z: I/O port
$\overline{\text{NMI}}$	1	Input	Non-maskable interrupt request pin: Interrupt request pin with falling edge. Can also be operated at rising edge by program. (schmitt input)
$\overline{\text{WDTOUT}}$	1	Output	Watchdog timer output pin

Table 2.2.6 Pin Names and Functions (6/6)

Pin Name	Number of Pins	I/O	Functions
AM0, 1	2	Input	Address mode: Selects external data bus width. AM1 = "low" AM0 = "low": Start with 8-bit external data bus AM1 = "low" AM0 = "high": Start with 16-bit external data bus AM1 = "high" AM0 = "low": Don't use this setting AM1 = "high" AM0 = "high": Don't use this setting
TEST0, 1	2	Input	Test: Input "low" when using
CLK	1	Output	Clock output: Outputs system clock
X1/X2	2	I/O	Oscillator connecting pin
$\overline{\text{RESET}}$	1	Input	Reset: Initializes LSI (with pull-up resistor) (schmitt input)
VREFH	1	Input	Pin for reference voltage input to AD converter ("high" level)
VREFL	1	Input	Pin for reference voltage input to AD converter ("low" level)
DAREFH	1	Input	Pin for reference voltage input to DA converter ("high" level)
DAREFL	1	Input	Pin for reference voltage input to DA converter ("low" level)
ADVCC	1	–	Power supply pin for 10-bit AD converter
ADVSS	1	–	GND pin for 10-bit AD converter (0 V)
DAVCC	1	–	Power supply pin for 8-bit DA converter
DAVSS	1	–	GND pin for 8-bit DA converter (0 V)
CLVCC	1	–	Power supply pin for clock doubler
CLVSS	1	–	GND pin for clock doubler
DVCC	8	–	Power supply pin (+5 V) (Connect all DVCC pins to +5V)
DVSS	8	–	GND pin (0 V) (Connect all DVSS pins to GND(0V).)

3. Operation

The following is a block-by-block description of the functions and basic operation of TMP94C251A.

3.1 CPU

TMP94C251A contains an advanced, high-speed 32-bit CPU (900/H2 CPU).

3.1.1 CPU Outline

900/H2 CPU is high-speed and high-performance CPU based on 900/H CPU. 900/H2 CPU has expanded 32-bit internal data bus to process instructions more quickly.

Functional differences between 900/H2 CPU and 900/H CPU are as follows:

	900/H2 CPU
Width of CPU Address Bus	24-bit
Width of CPU Data Bus	32-bit
Internal Operating Frequency	20 MHz
Minimum Bus Cycle	1-clock access (50 ns @ 20 MHz)
Bus Sizing Function	8/16-bit
Internal RAM	32-bit 1-clock access
Internal I/O	8/16/32-bit 2-clock access
External Device	8/16-bit 2-clock access (can insert some waits)
Minimum Instruction Execution Cycle	1-clock (50 ns @20 MHz)
Conditional Jump	2-clock (100 ns @20 MHz)
Instruction Queue Buffer	12-byte
Instruction Set	No MIN instruction No LDX instruction
CPU mode	No MIN (minimum) mode
Micro DMA	8-channel

3.1.2 Reset Operation

When resetting the TMP94C251A microcontroller, ensure that the power supply voltage is within the operating voltage range, and that the internal high-frequency oscillator has stabilized. Then set the $\overline{\text{RESET}}$ input to low level at least for 10 system clocks (2 μs at 10 MHz).

When the reset is accepted, the CPU:

- Set the program counter (PC) to the reset vector stored at addresses FFFF00H to FFFF02H.

  PC (7:0) ← Value at address FFFF00H
  PC (15:8) ← Value at address FFFF01H
  PC (23:16) ← Value at address FFFF02H
- Sets the stack pointer (XSP) to 00000000H
- Sets bits IFF2 to IFF0 of the status register (SR) to 111 (This sets the interrupt level mask register to level 7).
- Clears bits RFP1 to RFP0 of the status register (SR) to 00 (This sets the register banks to 0).

After reset is released, the CPU begins execution from the instruction at the location specified in the PC. Other than the changes described above, reset does not alter any internal CPU registers.

When reset is accepted, processing of the internal I/O, port, and other pins are as follows:

- Initializes the internal I/O registers as table of “Special Function Register” in section 5.
- Set ports pins to general-purpose input port mode.
- Sets the $\overline{\text{WDTOUT}}$ pin to “Low”. (However, when reset is released, sets to “High”.)

When external reset is released, built-in clock doubler begins operation and after the stable time (1.6384 ms at 20 MHz) elapse of the circuit, internal reset is released.

The operation of memory controller and DRAM controller cannot be insured until power supply becomes stable after power-on reset. The external RAM data provided before turning on the TMP94C251A may be spoiled because the control signals are unstable until power supply becomes stable after power on reset.

3.1.3 Data Bus Size after Reset Release

The start data bus size is determined depending on the state of a AM1/AM0 pins just after reset release. Then the external memory is accessed as follows.

AM1	AM0	Start mode
"0"	"0"	8 bit data bus (1wait)
"0"	"1"	16 bit data bus (1wait)
"1"	"0"	Don't use this setting
"1"	"1"	Don't use this setting

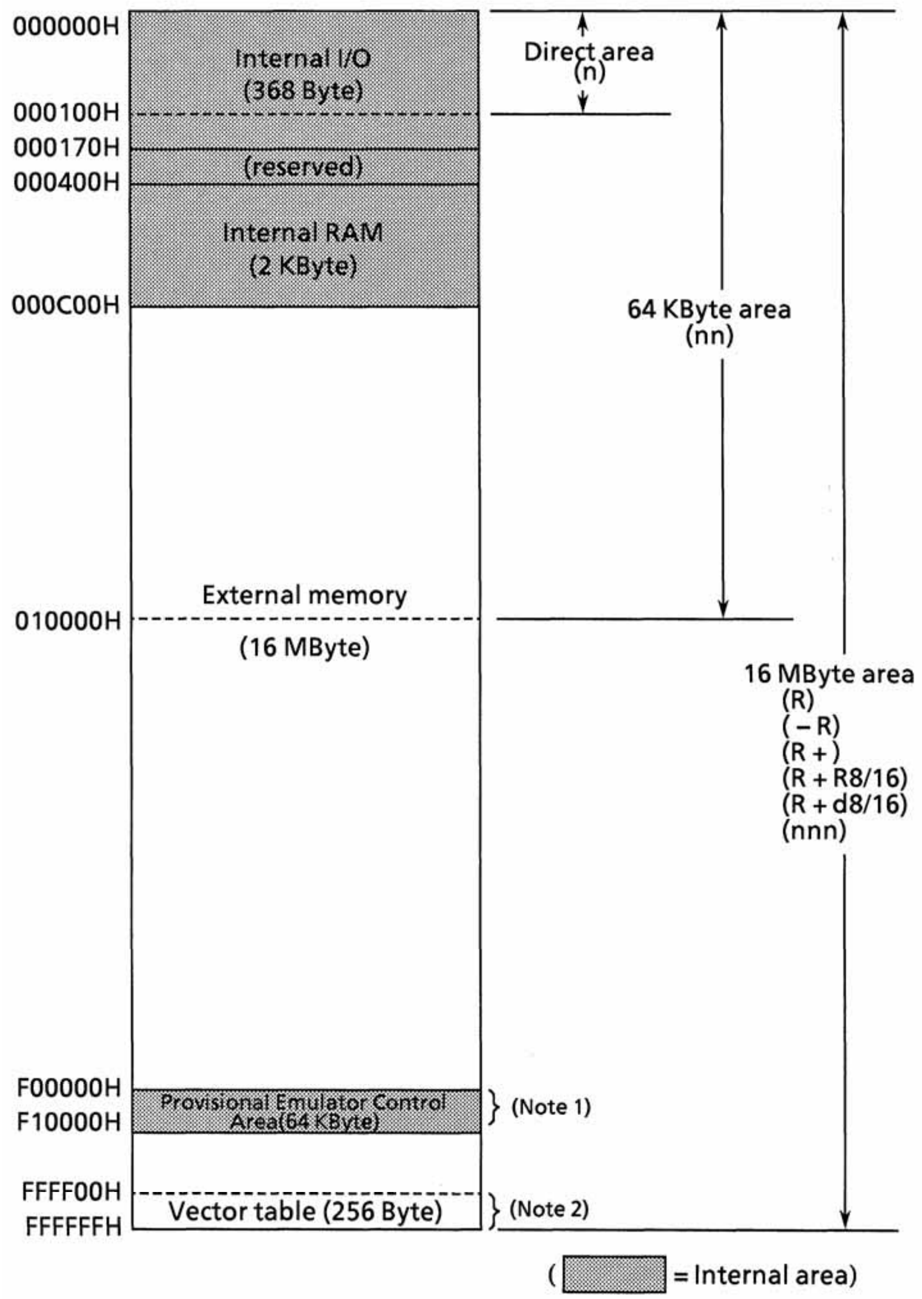
For the details, refer to chapter 3.6 "Memory Controller".

3.1.4 Setting of TEST0, TEST1

Connect TEST0, TEST1 pin to "GND" to use.

3.2 Memory Map

Figure 3.2.1 is a memory map of the TMP94C251A.



Note 1: Emulator control area is for emulator, it is mapped F00000H to F10000H address. Don't use this area. This area is reserved. On emulator $\overline{WR}$ signal and $\overline{RD}$ signal are asserted, when this area is accessed. Be careful to use external memory.

Note 2: Don't use the last 16-byte area (FFFFF0H to FFFFFFH). This area is reserved.

Figure 3.2.1 Memory Map

3.3 Interrupts

TLCS-900/H2 interrupts are controlled by the CPU interrupt mask flip-flops <IFF2:0> and the internal interrupt controller. Interrupts can come from a total of 38 sources:

- Interrupts from CPU itself: 2 (Software interrupt and illegal instructions)
- Interrupts from external pins ($\overline{\text{NMI}}$, INT0, INT4 to INTB): 10
- Interrupts from internal I/O: 18
- Interrupts from micro DMA: 8

Individual interrupt vector numbers (Fixed) are allocated to each interrupt source. Six levels of priority (Variable) can be allocated to maskable interrupts. The priority of non-maskable interrupts is fixed at “7” (The highest priority).

When an interrupt is generated, the interrupt controller sends the priority value of that interrupt to the CPU. If more than one interrupt is generated simultaneously, the interrupt with the highest priority (7 non-maskable interrupts is the highest) is sent to the CPU.

The CPU compares the priority value with the value of the CPU interrupt mask register <IFF2:0>, and accepts the interrupt if the priority is higher or equal to the value in the CPU interrupt mask register. The value of the interrupt mask register <IFF2:0> can be modified using the EI instruction (EI num sets <IFF2:0> to num). For example, executing “EI3” enables acceptance of non-maskable interrupts and maskable interrupts with a priority of 3 or higher set in the interrupt controller.

The DI instruction (Sets <IFF2:0> to “7”) is operationally the same as specifying “EI7”. As maskable interrupts have priorities in the range of 0 to 6, the DI instruction disables acceptance of maskable interrupts. The EI instruction is valid immediately after its execution.

In addition to the general-purpose interrupt processing mode described above, there is also a micro DMA processing mode. The micro DMA is a mode used by the CPU to automatically transfer 1/2/4 byte. It enables the CPU to transfer to the internal or external memories and the built-in I/O at high speed.

Furthermore, TMP94C251A has a software start function to request by software except that micro DMA is requested by interrupt sources.

Figure 3.3.1 is a flowchart showing overall interrupt processing.

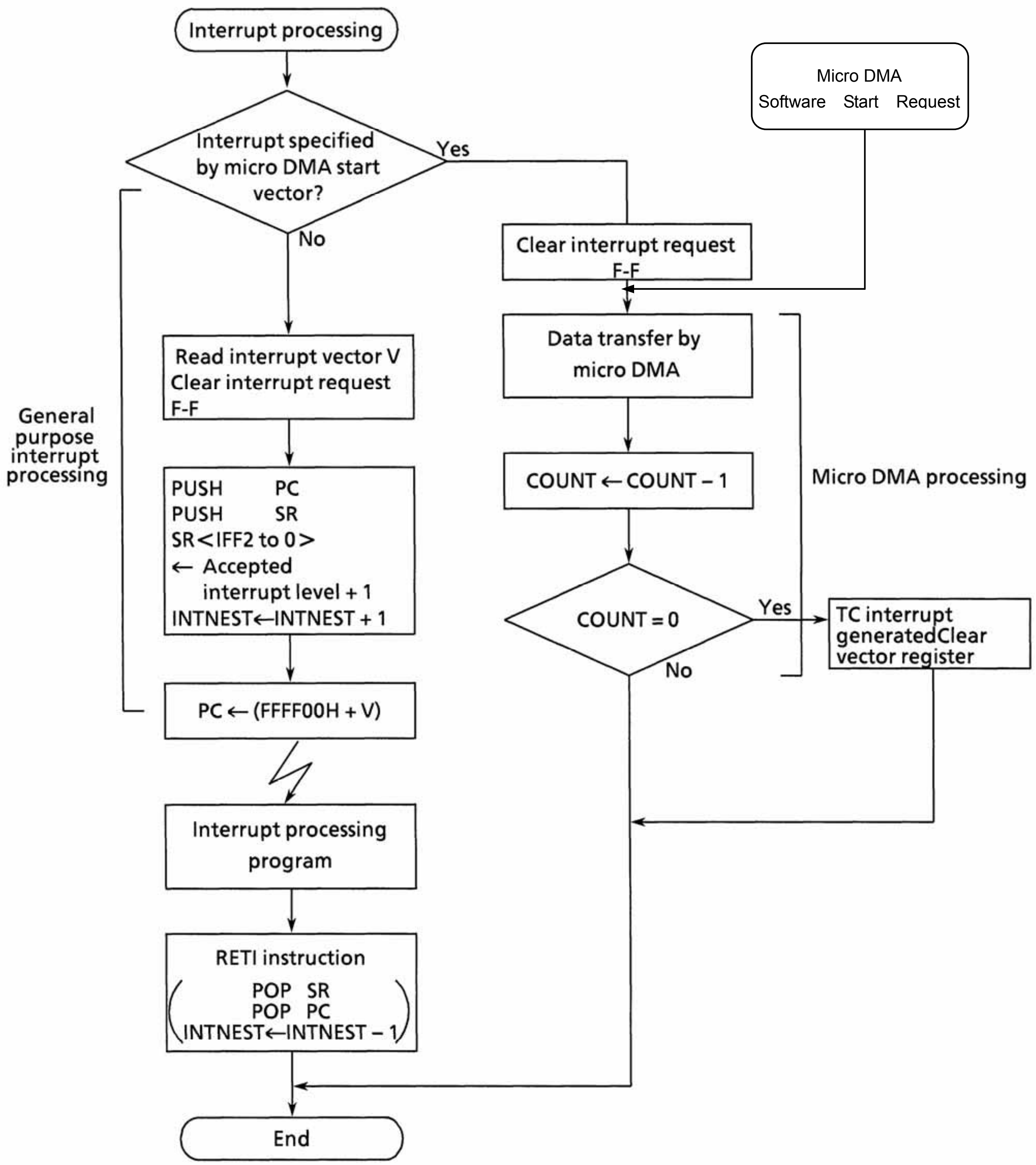


Figure 3.3.1 Interrupt Processing Flowchart

3.3.1 General-purpose Interrupt Processing

When accepting an interrupt the CPU operates as follows, which is the same as it is in TLCS-900/L and TLCS-900/H.

(1) The CPU reads the interrupt vector from the interrupt controller. When more than one interrupt with the same level is generated simultaneously, the interrupt vectors in accordance with the default priority (which is fixed as follows: the smaller the vector value, the higher the priority), then clears the interrupt request.

(2) The CPU pushes the program counter (PC) and the status register (SR) to the system stack area (Area indicated by the XSP).

(3) The CPU sets a value in the CPU interrupt mask register <IFF2:0> that is higher by 1 than the value of the accepted interrupt level. However, if the value is 7, 7 is set without an increment.

(4) The CPU sets the interrupt nesting counter (INTNEST) to +1.

(5) The CPU jumps to address FFFF00H + interrupt vector, then starts the interrupt processing routine.

All the above processing is completed in 10 states (Internal operation with 500 ns at 20 MHz) in the most approximate processing (The external memory is 32-bit data bus 0 wait, the stack area is the built-in RAM and the stack pointer value is an integer multiple of 4).

To return to the main routine after completion of the interrupt processing, the RETI instruction is usually used. Executing this instruction restores the contents of the program counter and the status registers, and decrements the interrupt nesting counter (INTNEST).

Though acceptance of non-maskable interrupts cannot be disabled by program, acceptance of maskable interrupts can. A priority can be set for each source of maskable interrupts. The CPU accepts an interrupt request with a priority higher than the value in the CPU mask register <IFF2:0>. The CPU mask register <IFF2:0> is set to a value higher by 1 than the priority of the accepted interrupt. Thus, if an interrupt with a level higher than the interrupt being processed is generated, the CPU accepts the interrupt with the higher level, causing interrupt processing to nest.

If an interrupt generated while the CPU is performing processes (1) to (5) for an earlier interrupt, the new interrupt is sampled immediately after the start instruction of the interrupt processing routine is executed. Setting DI as the start instruction disables maskable interrupt nesting.

Resetting initializes the CPU mask register <IFF2:0> to 7; therefore, maskable interrupts are disabled.

The addresses FFFF00H to FFFFFFH (256 bytes) of TMP94C251A are assigned for interrupt vector area. The interrupt vector area is depended on the derivative products.

Table 3.3.1 TMP94C251A Interrupt Table

Default priority	Type	Interrupt source	Vector	Address refer to vector	DMA start vector
1	Non-maskable	"SWI 0" instruction or RESET	0000H	FFFF00H	–
2		"SWI 1" instruction or default vector	0004H	FFFF04H	–
3		"SWI 2" instruction or "INT-UNDEF"	0008H	FFFF08H	–
4		"SWI 3" instruction	000CH	FFFF0CH	–
5		"SWI 4" instruction	0010H	FFFF10H	–
6		"SWI 5" instruction	0014H	FFFF14H	–
7		"SWI 6" instruction	0018H	FFFF18H	–
8		"SWI 7" instruction	001CH	FFFF1CH	–
9		$\overline{\text{NMI}}$ Pin	0020H	FFFF20H	–
10		INTWD: Watch-dog timer	0024H	FFFF24H	–
–	Maskable	(Micro-DMA)	–	–	–
11		INT0 Pin	0028H	FFFF28H	0AH (Note1)
12		INT4 Pin	002CH	FFFF2CH	0BH
13		INT5 Pin	0030H	FFFF30H	0CH
14		INT6 Pin	0034H	FFFF34H	0DH
15		INT7 Pin	0038H	FFFF38H	0EH
–		(Reserved)	003CH	FFFF3CH	–
16		INT8 Pin	0040H	FFFF40H	10H
17		INT9 Pin	0044H	FFFF44H	11H
18		INTA Pin	0048H	FFFF48H	12H
19		INTB Pin	004CH	FFFF4CH	13H
20		INTT0: 8-bit timer (Timer 0)	0050H	FFFF50H	14H
21		INTT1: 8-bit timer (Timer 1)	0054H	FFFF54H	15H
22		INTT2: 8-bit timer (Timer 2)	0058H	FFFF58H	16H
23		INTT3: 8-bit timer (Timer 3)	005CH	FFFF5CH	17H
24		INTTR4: 16-bit timer (Treg 4)	0060H	FFFF60H	18H
25		INTTR5: 16-bit timer (Treg 5)	0064H	FFFF64H	19H
26		INTTR6: 16-bit timer (Treg 6)	0068H	FFFF68H	1AH
27		INTTR7: 16-bit timer (Treg 7)	006CH	FFFF6CH	1BH
28		INTTR8: 16-bit timer (Treg 8)	0070H	FFFF70H	1CH
29		INTTR9: 16-bit timer (Treg 9)	0074H	FFFF74H	1DH
30		INTTRA: 16-bit timer (Treg A)	0078H	FFFF78H	1EH
31		INTTRB: 16-bit timer (Treg B)	007CH	FFFF7CH	1FH
32		INTRX0: Serial receive 0	0080H	FFFF80H	20H (Note2)
33		INTTX0: Serial send 0	0084H	FFFF84H	21H
34		INTRX1: Serial receive 1	0088H	FFFF88H	22H (Note2)
35		INTTX1: Serial send 1	008CH	FFFF8CH	23H
36		INTAD: AD conversion completion	0090H	FFFF90H	24H
37		INTTC0: micro-DMA completion Ch.0	0094H	FFFF94H	25H
38		INTTC1: micro-DMA completion Ch.1	0098H	FFFF98H	26H
39		INTTC2: micro-DMA completion Ch.2	009CH	FFFF9CH	27H
40		INTTC3: micro-DMA completion Ch.3	00A0H	FFFFA0H	28H
41		INTTC4: micro-DMA completion Ch.4	00A4H	FFFFA4H	29H
42		INTTC5: micro-DMA completion Ch.5	00A8H	FFFFA8H	2AH
43		INTTC6: micro-DMA completion Ch.6	00ACH	FFFFACH	2BH
44		INTTC7: micro-DMA completion Ch.7	00B0H	FFFFB0H	2CH
		(Reserved) : (Reserved)	00B4H : 00FCH	FFFFB4H : FFFFFCH	– : –

Note 1: When starting-up micro DMA, set at Edge detect mode.

Note 2: Micro DMA processing cannot be applied.

3.3.2 Micro DMA

TMP94C251A supports the micro DMA function. For interrupt requests set for micro DMA, micro DMA processing is performed at the highest priority for maskable interrupts, regardless of the set interrupt level.

Since the micro DMA has eight channels, it can transfer continuously by the burst specification which is described later.

(1) Micro DMA operation

When an interrupt request occurs for an interrupt specified by the micro DMA start vector register, micro DMA sends data to the CPU with the highest priority for maskable interrupts, regardless of the interrupt level set for the interrupt. If IFF = 7, micro DMA request is not accept.

The micro DMA function has eight channels. This allows micro DMA to be set for up to eight interrupts at the same time.

When micro DMA is accepted, the interrupt request F/F for the micro DMA channel is cleared, data are transferred (1/2/4 byte) once from the transfer source address to the transfer destination address (The addresses are set in the control register), and the transfer counter is decremented. If the decremented result is 0, the CPU informs a micro DMA transfer end to the interrupt controller. The interrupt controller generates a micro DMA transfer end interrupt (INTTCn). The CPU clears the micro DMA start vector register (DMAnV) 0, disables the next micro DMA startup, and terminates the micro DMA processing. If the decremented result is other than zero, micro DMA processing is terminated without the burst specification which is described later. In this case, the transfer end interrupt (INTTCN) is not generated.

When the interrupt source is used only to start micro DMA, the interrupt level must be set to “0”.

When the interrupt request generates until the interrupt sources are set to the micro DMA start vector, the CPU performs the general-purpose interrupt processing at the interrupt level of 1 to 6.

When simultaneously using the same interrupt resource for both the micro DMA and general-purpose interrupts as described above, set the level of the interrupt source used to start micro DMA lower than the levels of all other interrupt sources.

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

If multiple-channel micro DMA requests occur at the same time, the priority is determined by the channel numbers, not the interrupt levels. The lower the channel number, the higher the priority. (CH0 (High) to CH2 (Low))

The transfer source and transfer destination addresses are set in 32-bit control registers. However, as only 24-bit addresses are output, the address space available to micro DMA is 16 Mbytes.

Three transfer modes are supported: 1-byte transfer, 2-byte transfer and 4-byte transfer. For each transfer mode, it is possible to specify whether to increment, decrement, or fix source and destination addresses after transfer.

These modes facilitate data transfer from I/O to memory, from memory to I/O, and from I/O to I/O. For transfer mode details, see “Transfer Mode Register Details” later in this manual.

As a 16-bit transfer counter is used, micro DMA can perform a maximum of 65536 transfers (Initializing the counter to 0000H specifies the maximum number of transfers) and the software start (Total 35 interrupt sources) can be used to start micro DMA processing.

Figure 3.3.2 shows the micro DMA cycle for transfer destination address INC mode (the same apart from counter mode). (Condition: 0-wait built-in RAM in the transfer address area)

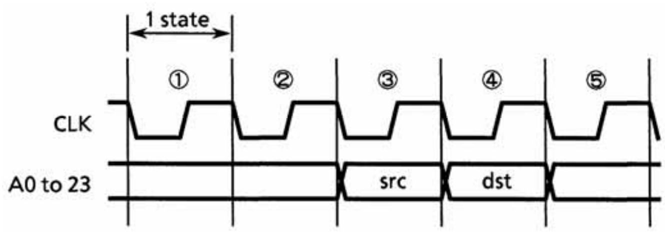


Figure 3.3.2 Micro DMA Cycle Timing

States 1, 2: Instruction fetch cycle (Prefetches the next instruction code)
State 3: Micro DMA read cycle
State 4: Micro DMA write cycle
State 5: (The same as in state 1, 2)

(2) Software start function

In addition to starting the micro DMA function by interrupts, TMP94C251A includes a micro DMA software start function that starts micro DMA on the generation of the write cycle to the DMAR register.

Writing “1” to each bit of DMAR register causes micro DMA once. At the end of transfer, the bits of the DMAR register which support the end channel are automatically cleared to “0”.

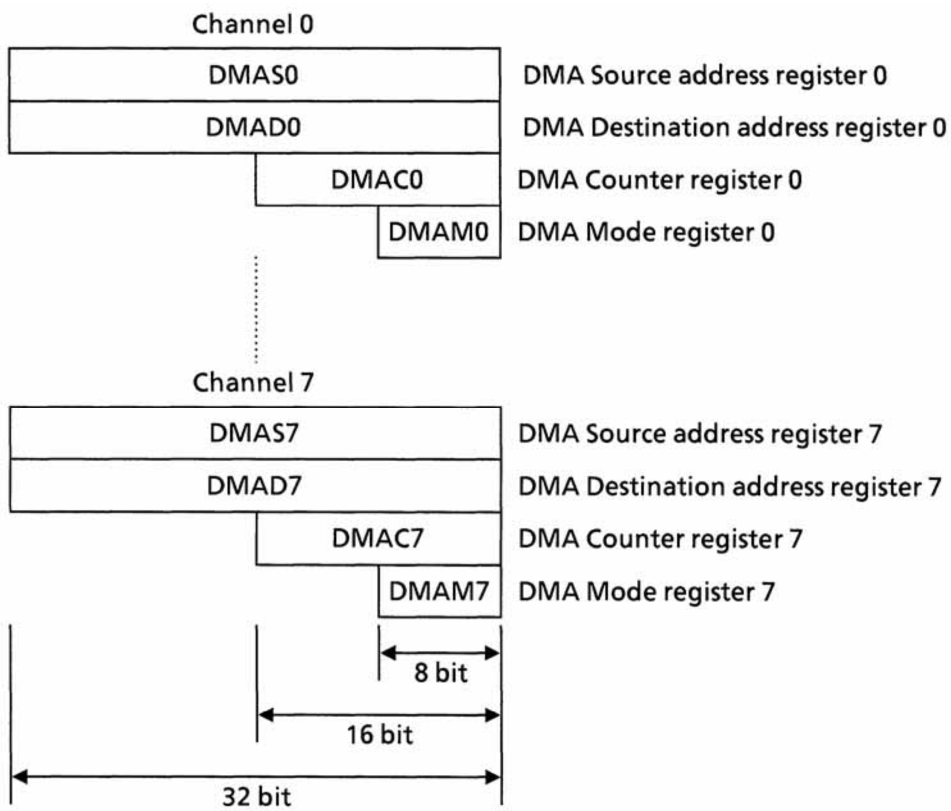
Writing again to the DMAB register triggers another software start, provided the micro DMA trance counter is set to other than “0”.

When the burst is specified by DMAB register, data is continuously transferred until the value in the micro DMA transfer counter is “0” after startup of the micro DMA.

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMAR	DMA Request	109h	DMA Request							
			DREQ7	DREQ6	DREQ5	DREQ4	DREQ3	DREQ2	DREQ1	DREQ0
			R/W							
		(no RMW)	0	0	0	0	0	0	0	0

(3) Transfer control register

The transfer source address and the transfer destination address are set by the following registers. These registers set data using “LDC cr,r” instruction.



(4) DMA mode register details

0 0 0	mode	DMAM0 to 7

DMAM [4:0]	Operation	Execution time
000zz	Destination INC mode (DMADn +) ← (DMASn) DMACn ← DMACn – 1 if DMACn = 0 then INT.	5 States
001zz	Destination DEC mode (DMADn –) ← (DMASn) DMACn ← DMACn – 1 if DMACn = 0 then INT.	5 States
010zz	Source INC mode (DMADn) ← (DMASn +) DMACn ← DMACn – 1 if DMACn = 0 then INT.	5 States
011zz	Source DEC mode (DMADn) ← (DMASn –) DMACn ← DMACn – 1 if DMACn = 0 then INT.	5 States
100zz	Destination and Source INC mode (DMADn +) ← (DMASn +) DMACn ← DMACn – 1 if DMACn = 0 then INT.	6 States
101zz	Destination and Source DEC mode (DMADn –) ← (DMASn –) DMACn ← DMACn – 1 if DMACn = 0 then INT.	6 States
110zz	Destination and Source fixed mode (DMADn) ← (DMASn) DMACn ← DMACn – 1 if DMACn = 0 then INT.	5 States
11100	Counter mode DMASn ← DMASn + 1 DMACn ← DMACn – 1 if DMACn = 0 then INT.	5 States

ZZ: 00 = 1 byte transfer
01 = 2 byte transfer
10 = 4 byte transfer
11 = (reserved)

Note: The execution time is measured at 1 state = 50 ns (operation at internal 20 MHz).

3.3.3 Interrupt Controller Operation

Figure 3.3.3 is a block diagram of the interrupt circuit. The left-hand side of the diagram shows the interrupt controller circuit. The right-hand side shows the CPU interrupt request signal circuit and the halt release circuit.

For each interrupt channel (36 channels in total), an interrupt request flag (Flip-flop), an interrupt priority setting register, and a micro DMA start vector register. The interrupt request flag latches interrupt request from the peripherals. The flag is cleared to zero in the following cases: When reset occurs, when the CPU reads the channel vector of an interrupt it has received, when the CPU receives a micro DMA request (when micro DMA is set), when the micro DMA burst transfer is terminated, and when an instruction that clears the interrupt for that channel is executed (by writing "0" to the clear bit in the interrupt priority setting register).

The interrupt priority can be set independently for each interrupt source by writing the priority to the interrupt priority setting register (e.g., INTE0AD, INTE12). Six interrupt priorities from 1 to 6 are provided. Setting "0" (or "7") disables the interrupt request. The priority of non-maskable interrupts (NMI pin, watchdog timer) is fixed at 7. If interrupt requests with the same level are generated at the same time, the default priority (The interrupt with the lowest priority or, in other words, the interrupt with the lowest vector value) is used to determine which interrupt request to accept first.

Reading the 3rd bit and the 7th bit in the interrupt priority setting register sees the state of the interrupt request flag and whether there are the interrupt request of each channel.

The interrupt controller sends the interrupt request with the highest priority among the simultaneous interrupts and its vector address to the CPU. The CPU compares the priority value <IFF2:0> set in the status register by the interrupt request signal with the priority value set; if the latter is higher, the interrupt is accepted. Then the CPU sets a value higher than the priority value by 1 in the CPU SR <IFF2:0>. Interrupt request where the priority value equals or is higher than the set value are accepted simultaneously during the previous interrupt routine.

When interrupt processing is completed (after execution of the RETI instruction), the CPU restores the priority value saved in the stack before the interrupt was generated to the CPU SR<IFF2:0>.

The interrupt controller also has eight registers used to store the micro DMA start vector. Writing the start vector of the interrupt source for the micro DMA processing (See Table 3.3.1), enables the corresponding interrupt to be processed by micro DMA processing. The values must be set in the micro DMA parameter register (e.g., DMAS and DMAD) prior to the micro DMA processing.

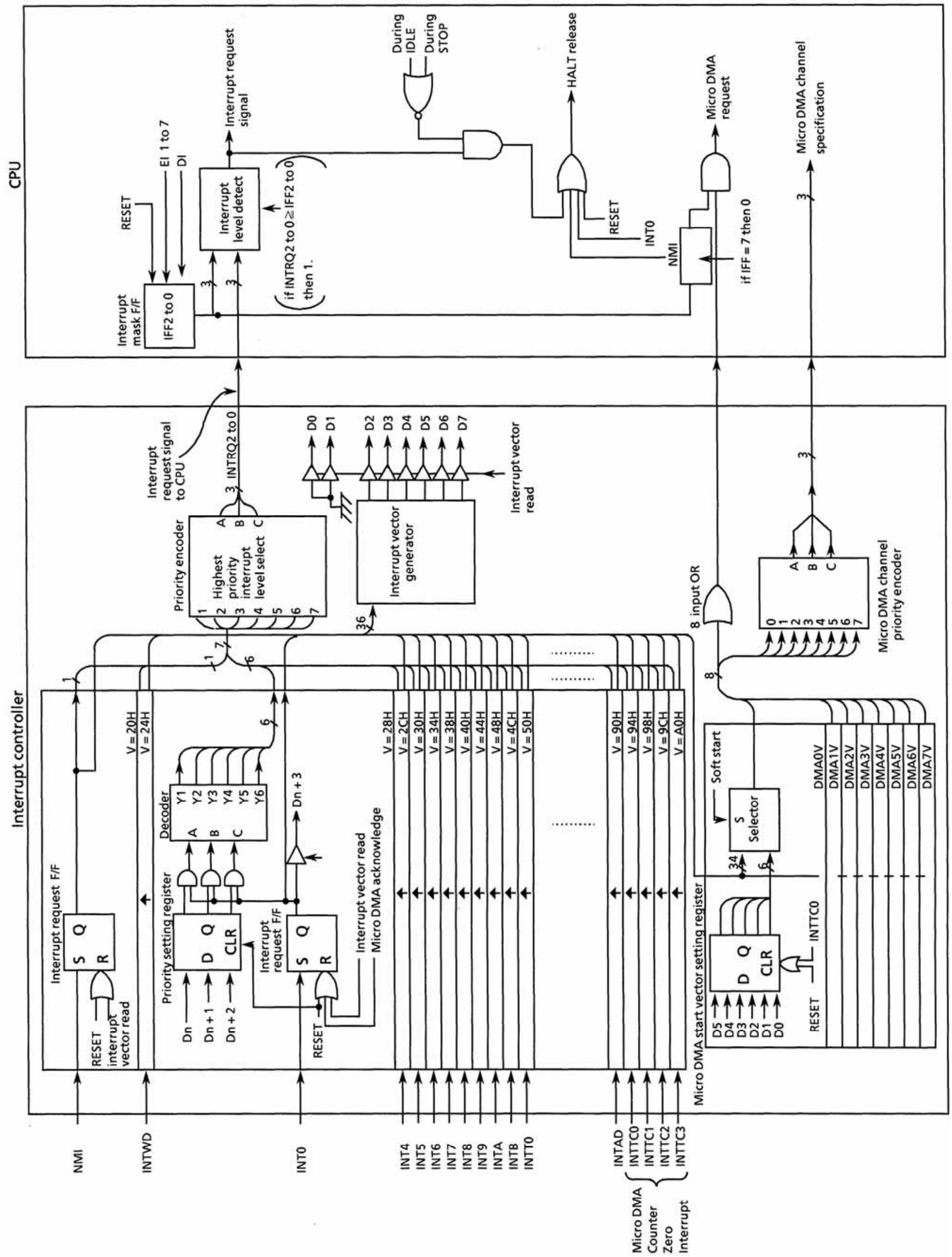


Figure 3.3.3 Block Diagram of Interrupt Controller

(1) Interrupt priority setting register

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE0AD	INT0 & INTAD Enable	F0h	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE45	INT4 & INT5 Enable	E0h	INT5				INT4			
			I5C	I5M2	I5M1	I5M0	I4C	I4M2	I4M1	I4M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE67	INT6 & INT7 Enable	E1h	INT7				INT6			
			I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE89	INT8 & INT9 Enable	E2h	INT9				INT8			
			I9C	I9M2	I9M1	I9M0	I8C	I8M2	I8M1	I8M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTEAB	INTA & INTB Enable	E3h	INTB				INTA			
			IBC	IBM2	IBM1	IBM0	IAC	IAM2	IAM1	IAM0
			B	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTET01	INTT0 & INTT1 Enable	E4h	INTT1 (Timer1)				INTT0(Timer0)			
			IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTET23	INTT2 & INTT3 Enable	E5h	INTT3 (Timer3)				INTT2 (Timer2)			
			IT3C	IT3M2	IT3M1	IT3M0	IT2C	IT2M2	IT2M1	IT2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTET45	INTTR4 & INTTR5 Enable	E6h	INTTR5 (TREG5)				INTTR4 (TREG4)			
			IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTET67	INTTR6 & INTTR7 Enable	E7h	INTTR7 (TREG7)				INTTR6 (TREG6)			
			IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTET89	INTTR8 & INTTR9 Enable	E8h	INTTR9 (TREG9)				INTTR8 (TREG8)			
			IT9C	IT9M2	IT9M1	IT9M0	IT8C	IT8M2	IT8M1	IT8M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETAB	INTTRA & INTTRB Enable	E9h	INTTRB (TREGB)				INTTRA (TREGA)			
			ITBC	ITBM2	ITBM1	ITBM0	ITAC	ITAM2	ITAM1	ITAM0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTES0	INTRX0 & INTTX0 Enable	EAh	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTES1	INTRX1 & INTTX1 Enable	EBh	INTTX1				INTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC01	INTTC0 & INTTC1 Enable	ECh	INTTC1				INTTC0			
			ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC23	INTTC2 & INTTC3 Enable	EDh	INTTC3				INTTC2			
			ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC45	INTTC4 & INTTC5 Enable	EEh	INTTC5				INTTC4			
			ITC5C	ITC5M2	ITC5M1	ITC5M0	ITC4C	ITC4M2	ITC4M1	ITC4M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC67	INTTC6 & INTTC7 Enable	EFh	INTTC7				INTTC6			
			ITC7C	ITC7M2	ITC7M1	ITC7M0	ITC6C	ITC6M2	ITC6M1	ITC6M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTNMWDT	NMI & INTWD Enable	F7h	NMI				INTWD			
			ITCNM	—	—	—	ITCWD	—	—	—
			R				R			
			0	—	—	—	0	—	—	—

"1" indicates interrupt request

IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Prohibits 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	Prohibit interrupt request.

Note: Changing of the interrupt priority setting register should be carried out after execution of DI instruction.

(2) External interrupt control

Symbol	Name	Address	7	6	5	4	3	2	1	0
IIMC	Interrupt Input Mode Control	F6h (no RMW)	—	—	—	—	—	—	I0LE	NMIREE
									R/W	
			—	—	—	—	—	—	0	0
									0: INT0 edge mode 1: INT0 level mode	1: Operate even at $\overline{NMI}$ rise edge

Note:

*INT0 level Enable

0	Rising edge detect INT
1	"H"level INT

*$\overline{NMI}$ rising edge Enable

0	INT request generation at falling edge
1	INT request generation at rising/falling edge

Note 1: Disable INT0 request before changing INT0 pin mode from level-sense to edge-sense.
Setting example:

```
DI
LD (IIMC), xxxxxx0xB   ;   Switches from level to edge.
LD (INTCLR), 0AH       ;   Clears interrupt request flag.
EI
```

Note 2: See electrical characteristics in section 4 for external interrupt input pulse width.

Setting of External Interrupt Pin Function

Interrupt	Pin name	Mode	Setting method
$\overline{NMI}$	—	Falling edge	IIMC<NMIREE> = 0
		Falling and rising edges	IIMC<NMIREE> = 1
INT0	PH4	Rising edge	IIMC<I0LE> = 0, PHFC<PH4F> = 1
		Level	IIMC<I0LE> = 1, PHFC<PH4F> = 1
INT4	PD1	Rising edge	T4MOD<CAP45M1:0> = 0, 0 or 0, 1 or 1, 1
		Falling edge	T4MOD<CAP45M1:0> = 1, 0
INT5	PD2	Rising edge	—
INT6	PD5	Rising edge	T6MOD<CAP67M1:0> = 0, 0 or 0, 1 or 1, 1
		Falling edge	T6MOD<CAP67M1:0> = 1, 0
INT7	PD6	Rising edge	—
INT8	PE1	Rising edge	T8MOD<CAP89M1:0> = 0, 0 or 0, 1 or 1, 1
		Falling edge	T8MOD<CAP89M1:0> = 1, 0
INT9	PE2	Rising edge	—
INTA	PE5	Rising edge	TAMOD<CAPABM1:0> = 0, 0 or 0, 1 or 1, 1
		Falling edge	TAMOD<CAPABM1:0> = 1, 0
INTB	PE6	Rising edge	—

(3) Interrupt request flag clear register

The interrupt request flag is cleared by writing the micro DMA start vector, which is listed in Table 3.3.1, to the INTCLR register.

For example, to clear the INT0 interrupt flag, operate the following register after execution of DI instruction.
Clears INT0 interrupt request flag

INTCLR←0AH

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTCLR	Interrupt Clear Control	F8h (no RMW)	–	–	–	–	–	–	–	–
			W							
			0	0	0	0	0	0	0	0
			Interrupt Vector							

(4) Micro DMA start vector register

This register assigns micro DMA processing to an interrupt source. The interrupt source with a micro DMA start vector that matches the vector set in this register is assigned as the micro DMA start source.

When the micro DMA transfer counter value reaches 0, the micro DMA transfer end interrupt corresponding to the channel is set to the interrupt controller, the micro DMA start vector register is cleared, and the micro DMA start source of the channel is cleared. Therefore, to continue micro DMA processing, set the micro DMA start vector register again during the processing of the micro DMA transfer end interrupt.

If the same vector is set in the micro DMA start vector registers of more than one channel, the channel with the lowest number has a higher priority.

Accordingly, if the same vector is set in the micro DMA start vector registers of two channels, the interrupt generated in the channel with the lower number is executed until the micro DMA transfer is complete. If the micro DMA start vector of this channel is not set again, the next micro DMA is started for the channel with the higher number (Micro DMA chaining).

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMA0V	DMA0 Start Vector	100h			DMA0 Start Vector					
			–	–	DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
					R/W					
			–	–	0	0	0	0	0	0
DMA1V	DMA1 Start Vector	101h			DMA1 Start Vector					
			–	–	DMA1V5	DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
					R/W					
			–	–	0	0	0	0	0	0
DMA2V	DMA2 Start Vector	102h			DMA2 Start Vector					
			–	–	DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
					R/W					
			–	–	0	0	0	0	0	0
DMA3V	DMA3 Start Vector	103h			DMA3 Start Vector					
			–	–	DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
					R/W					
			–	–	0	0	0	0	0	0
DMA4V	DMA4 Start Vector	104h			DMA4 Start Vector					
			–	–	DMA4V5	DMA4V4	DMA4V3	DMA4V2	DMA4V1	DMA4V0
					R/W					
			–	–	0	0	0	0	0	0
DMA5V	DMA5 Start Vector	105h			DMA5 Start Vector					
			–	–	DMA5V5	DMA5V4	DMA5V3	DMA5V2	DMA5V1	DMA5V0
					R/W					
			–	–	0	0	0	0	0	0
DMA6V	DMA6 Start Vector	106h			DMA6 Start Vector					
			–	–	DMA6V5	DMA6V4	DMA6V3	DMA6V2	DMA6V1	DMA6V0
					R/W					
			–	–	0	0	0	0	0	0
DMA7V	DMA7 Start Vector	107h			DMA7 Start Vector					
			–	–	DMA7V5	DMA7V4	DMA7V3	DMA7V2	DMA7V1	DMA7V0
					R/W					
			–	–	0	0	0	0	0	0

(5) Micro DMA burst specification

Specifying the micro DMA burst continues the micro DMA transfer until the transfer counter register reaches 0 after micro DMA start. Setting a bit which corresponds to the micro DMA channel of the DMAB registers mentioned below to “1” specifies a burst.

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMAB	DMA Burst	108h	DMA Burst							
			DBST7	DBST6	DBST5	DBST4	DBST3	DBST2	DBST1	DBST0
			R/W							
			0	0	0	0	0	0	0	0

(6) Notes

The instruction execution unit and the bus interface unit of this CPU operate independently. Therefore, immediately before an interrupt is generated, if the CPU fetches an instruction that clears the corresponding interrupt request flag, the CPU may execute the instruction that clears the interrupt request flag between accepting and reading the interrupt vector. In this case, the CPU reads the default vector 0004H and reads the interrupt vector at address FFFF04H.

To avoid the above problem, place instructions that clear interrupt request flags after a DI instruction.

And in the case of setting an interrupt enable again by EI instruction after the execution of clearing instruction, execute EI instruction after clearing and more than 3-instructions (e.g., "NOP" * 3times). If placed EI instruction without waiting NOP instruction after execution of clearing instruction, interrupt will be enable before request flag is 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.

In addition, take care as the following 2 circuits are exceptional and demand special attention.

INT0 level mode	INT0 in level mode is not an edge-detect interrupt, so the interrupt request flip-flop function is canceled. The peripheral interrupt request bypasses the S input of the flip-flop, and acts as the Q output. Changing modes from edge to level automatically clears the interrupt request flag.

	If the CPU enters the interrupt response sequence as a result of setting INT0 from 0 to 1, INT0 must be held at 1 until the interrupt response sequence is completed. If the INT0 level mode is used to release a halt, INT0 must be held at 1 from the time INT0 changes from 0 to 1, to the time when the halt is released. (Ensure that INT0 does not go back 0 due to noise before the halt is released.)
	When switching modes from level to edge, any interrupt request flag set in level mode is not cleared. Accordingly, clear the interrupt request flag using the following sequence.
	DI
	LD (IIMC), 00H; Switches from level to edge.
	LD (INTCLR), 0AH; Clears interrupt request flag.
	NOP; Wait EI execution
	NOP
	NOP
	EI

INTRX	The interrupt request flip-flop can only be cleared by reset or by reading the serial channel receive buffer, not by an instruction.

Note: The following instructions or pin changes are equivalent to instructions that clear the interrupt request flag.

INT0: Instructions that switch to level mode after an interrupt request is generated in edge mode.
The pin input changes from high to low after an interrupt request is generated in level mode. ("H" → "L")

INTRX: Instructions that read the receive buffer.

3.4 Standby Function

[1] HALT mode

Executing the HALT instruction sets either RUN, IDLE, or STOP mode depending on the content of WDMOD<HALTM1:0>.

(1) RUN: Halts the CPU only. Power dissipation remains almost unchanged.

(2) IDLE: Operates only the internal oscillator, while halts all other circuits.

(3) STOP: Halts all internal circuits, including the internal oscillator.

[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 = ($\overline{\text{NMI}}$ and 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 IDLE or STOP mode (RUN 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.

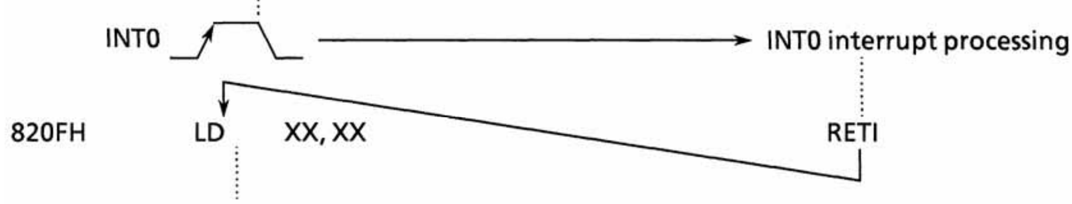
- Release by reset

  All HALT modes can be released by a reset. However, when releasing STOP mode, allow sufficient reset time (at least 2 μs) 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.

On execution of the HALT instruction, the device enters standby state in RUN mode. Release halt using INT0.

```
Address
8200H      LD     (PHCR), 00H        ; Sets to input mode for PH4/INT0
8203H      LD     (PHFC), 10H        ; Sets to INT0 for PH4/INT0
8206H      LD     (INTE0AD), 06H     ; Sets interrupt level to "6" for INT0
8209H      EI     5                  ; Sets interrupt level to "5" for CPU
820BH      LD     (WDMOD), 00H       ; Sets HALT mode to "RUN"
820EH      HALT                      ; Halts CPU
```



(1) RUN mode

Figure 3.4.1 is the timing chart for releasing a halt in RUN mode using an interrupt.

In RUN mode, the MCU internal system clock does not stop after the HALT instruction is executed. Only CPU instruction execution stops. Therefore, the CPU performs repeated dummy cycles until the halt state is released.

In the halt state, interrupt requests are sample on the cycle of the CLK signal.

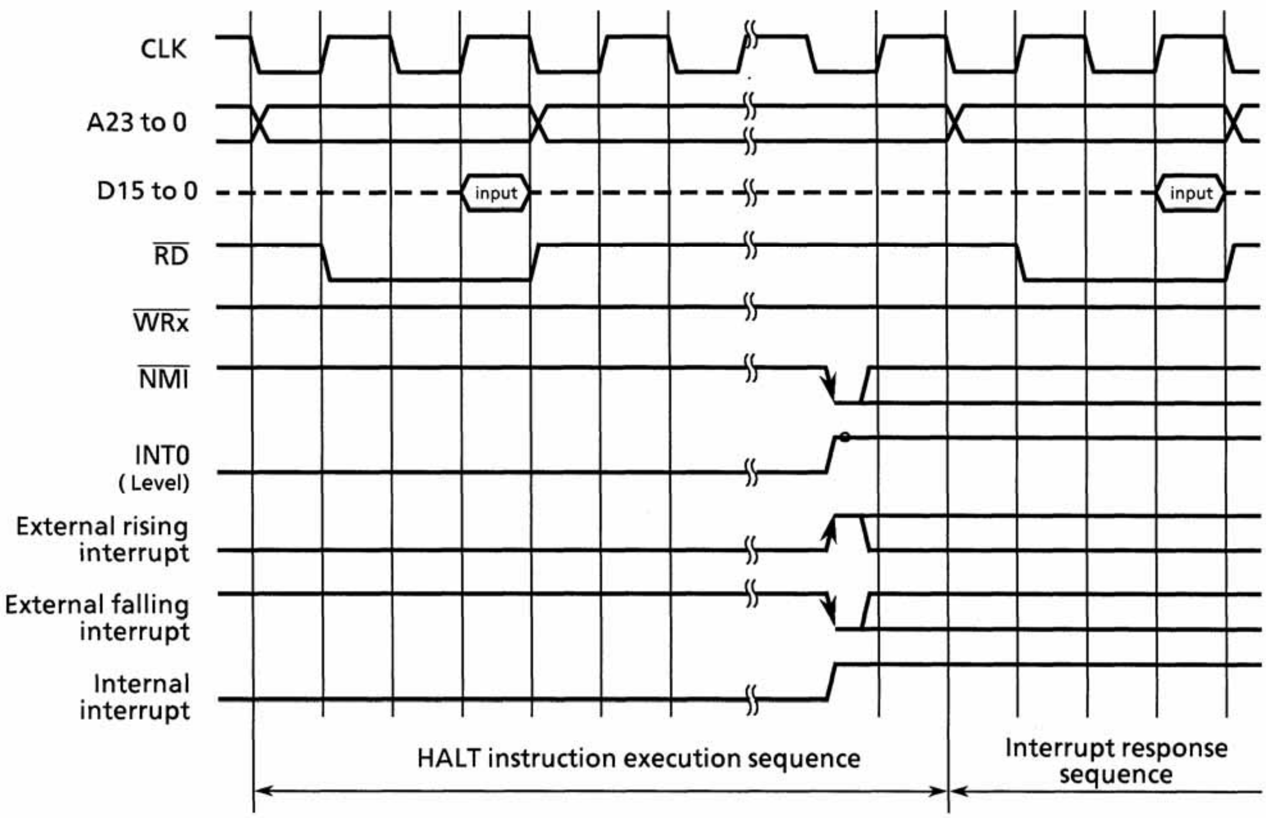


Figure 3.4.1 Timing Chart for Releasing Halt in RUN Mode Using Interrupt

(2) IDLE mode

Figure 3.4.2 is the timing chart for releasing a halt in IDLE mode using an interrupt.

In IDLE mode, the MCU internal system clock stops. Only the internal oscillator functions.

In the halt state, interrupt requests are sampled synchronously to the system clock. The release from the halt state (Operation restart), however, is synchronized with the clock.

In IDLE mode, interrupt requests other than external interrupts (NMI, INT0) are disabled.

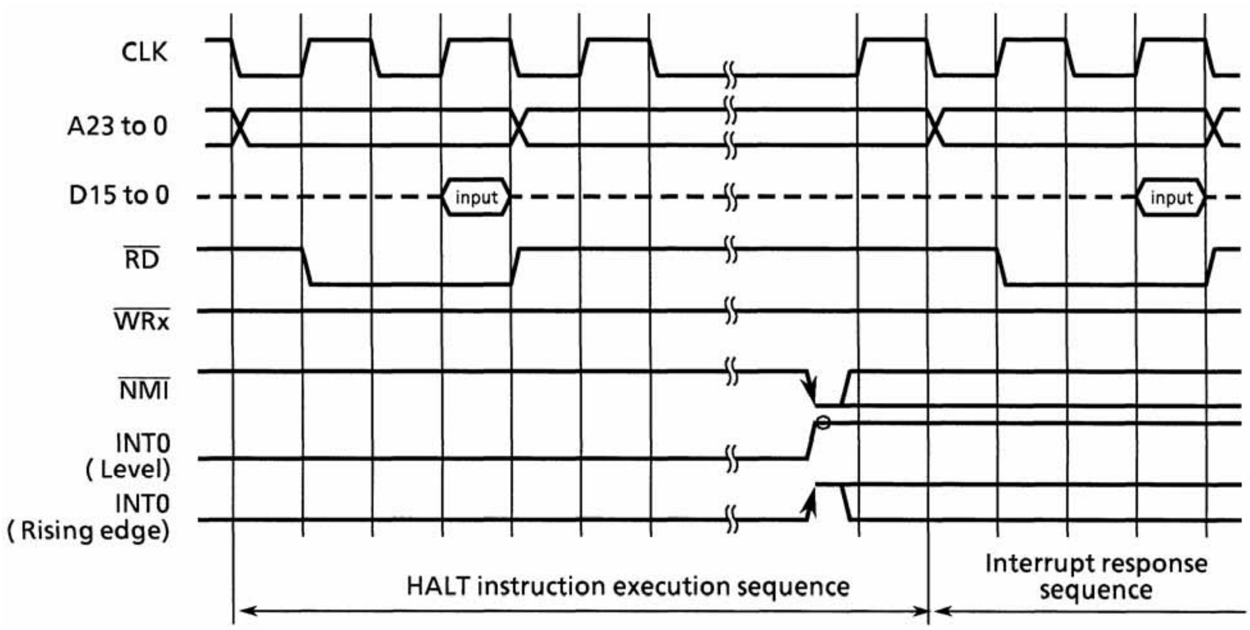


Figure 3.4.2 Timing Chart for Releasing Halt in IDLE Mode Using Interrupt

(3) STOP mode

Figure 3.4.3 is the timing chart for releasing a halt in STOP mode using an interrupt.

In STOP mode, all internal circuits stop, including the internal oscillator. Also, in STOP mode, all pins, apart from a few exceptions, are set to high impedance and are disconnected from the internal circuit of the MCU.

However, setting WDMOD<DRVE> in the internal I/O register to “1” specifies that pins maintain the states prior to the halt. Reset clears the register to “0”.

When the CPU receives an interrupt request, the internal oscillation restarts. Then, after the time set by the warm-up counter for the internal oscillation to stabilize, the system clock starts its output. The CLKMOD<WARM> bit sets the warm-up time. Setting this bit to 0 specifies a warm-up time of 2^{15} clock cycles; setting the bit to 1 specifies a warm-up time of 2^{17} clock cycles. Reset clears CLKMOD<WARM> to “0”.

STOP mode can only be released by an NMI pin or INT0 pin interrupt, or by reset.

When STOP mode is released by other than reset, the system clock starts its output after the time set by the warm-up counter for the internal oscillation to stabilize. When using reset to release stop mode, input reset signals long enough for stable oscillation.

In systems with an external oscillator, the warm-up counter also operates when STOP mode is released. Therefore, such systems also require a warm-up time between input of release signal and system clock output.

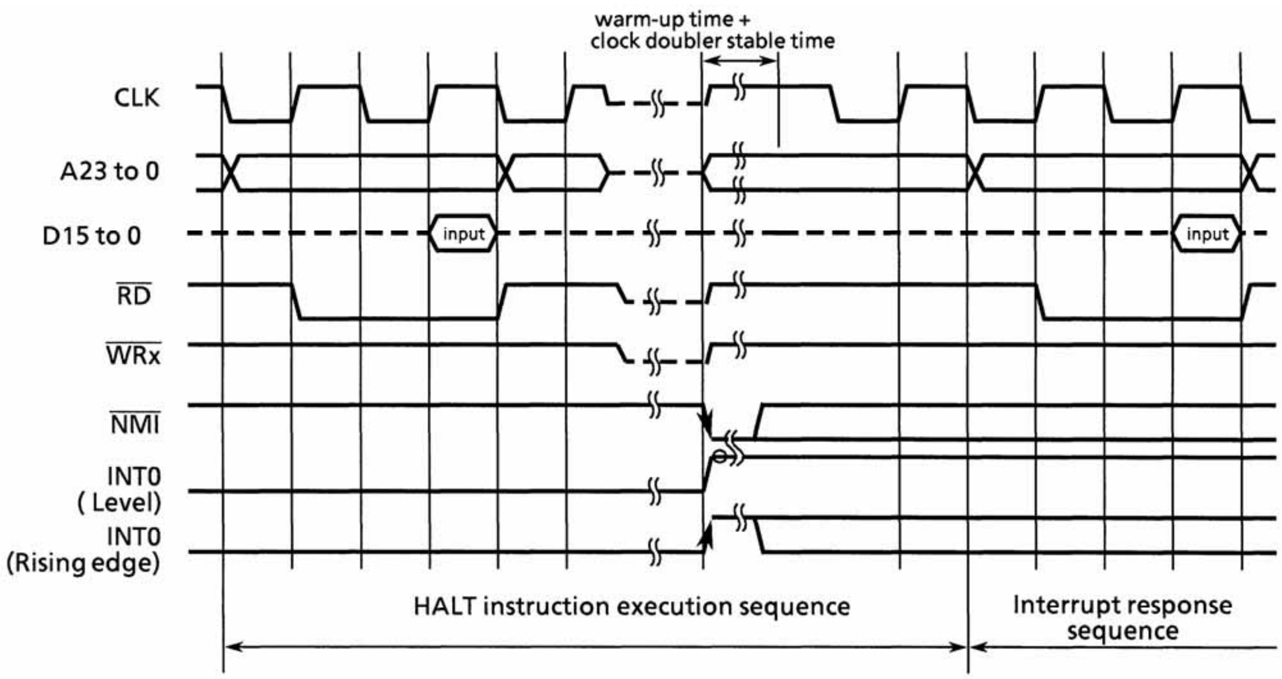


Figure 3.4.3 Timing Chart for Releasing Halt in STOP Mode Using Interrupt

Table 3.4.1 Pin states in STOP mode

Pin Name	Mode	DRVE = 0	DRVE = 1
P00 to P37/D0 to D15	D0 to D15 (input / output) P00 to P17 (input) P00 to P17 (output)	High-Z Disable output	High-Z Disable output
P40 to P67/A0 to A23	input output	Disable High-Z	Disable output
P70/$\overline{RD}$	output	High-Z	output
P71, P72/$\overline{WRL}$ to $\overline{WRH}$	output	High-Z	output
P73, P74	output	High-Z	output
P75/$\overline{BUSRQ}$	input output	Disable High-Z	Disable output
P76/$\overline{BUSAK}$	output	High-Z	output
P80 to P85/$\overline{CS0}$ to $\overline{CS5}$	output	High-Z	output
P86/$\overline{WAIT}$	input output	Disable High-Z	Disable output
PA0 to PA4/$\overline{CAS0}$ to $\overline{WE0}$	output	High-Z	output
PB0 to PB4/$\overline{CAS1}$ to $\overline{WE1}$	output	High-Z	output
PC0, PC1/TO1 to TO3	input output	Disable High-Z	Disable output
PD0 to PD6/TO4 to TI7	input output	Disable High-Z	Disable output
PE0 to PE6/TO8 to TIB	input output	Disable High-Z	Disable output
PF0 to PF6/TXD0 to SCLK1	input output	Disable High-Z	Disable output
PG0 to PG7/AN0 to AN7	input	Disable	Disable
DAOUT0, DAOUT1	output	High-Z	output
PH0 to PH3/TC0 to TC3	input output	Disable High-Z	Disable output
PH4/INT0	input output	Enable High-Z	Enable output
PZ0 to PZ7	input output	Disable output	Disable output
$\overline{NMI}$	input	Enable	Enable
$\overline{WDTOUT}$	output	output	output
AM0, AM1	input	Enable	Enable
TEST0, TEST1	input	Enable	Enable
CLK	output	output	output
X1 X2	input output	Disable "High"	Disable "High"
$\overline{RESET}$	input	Enable	Enable

Output: Maintains output states prior to a halt.
Enables: Input is valid. When the input pin is set to middle electric potential, through current.
Disabled: Input is invalid. As the input gate is disabled, no through current.
High-Z: The output is set to high impedance.
"High": The output is set to high electric potential.

Table 3.4.2 I/O Operation During Halt and Release

Halt Mode		RUN	IDLE	STOP
WDMOD 〈HALTM1, 0〉		00	10	01
Operation Block	CPU	Halt		
	I/O Port	Operation	Operation	See Table 3.4 (1)
	8-bit timer	Operation	Halts	
	16-bit timer	Operation	Halts	
	Serial interface	Operation	Halts	
	AD converter	Operation	Halts	
	DA converter	Operation	Halts	
	Watchdog timer	Operation	Halts	
	DRAM controller	Operation	Halts	
	Interrupt controller	Operation	Halts	

Interrupt Mask and Request Level Settings			Interrupt Request Level ≥ Interrupt Mask <IFF2:0>			Interrupt Request Level *2 < Interrupt Mask <IFF2:0>		
HALT Mode			RUN	IDLE	STOP	RUN	IDLE	STOP
HALT Release Source	Interrupt	$\overline{\text{NMI}}$	♦	♦	♦ *1	–	–	–
		INTWD	♦	×	×	–	–	–
		INT0	♦	♦	♦ *1	○	○	○ *1
		INT4 to 9, A, B	♦	×	×	×	×	×
		INTT0 to 3	♦	×	×	×	×	×
		INTTR4 to 9, A, B	♦	×	×	×	×	×
		INTRXD0, 1	♦	×	×	×	×	×
		INTTXD0, 1	♦	×	×	×	×	×
		INTAD	♦	×	×	×	×	×
	$\overline{\text{RESET}}$		♦	♦	♦	♦	♦	♦

♦ : After a halt is released, interrupt processing begins. (Reset initializes the LSI.)

○ : After a halt is released, processing begins from the next address following the HALT instruction.

× : Cannot be used to release a halt.

*1 : Halt is released after the warm-up time has elapsed.

*2 : Same as a DI instruction.

3.5 Functions of Ports

TMP94C251A has I/O port pins which are shown in Table 3.5.1. In addition to functioning as general-purpose I/O ports, these pins are also used by internal CPU and I/O functions.

Table 3.5.1 Port Functions (1/2)

Port Name	Pin Name	Number of Pins	I/O	I/O Setting	Pin Name for built-in function
Port 0	P00 to P07	8	I/O	Bit	D0 to D7
Port 1	P10 to P17	8	I/O	Bit	D8 to D15
Port 4	P40 to P47	8	I/O	Bit	A0 to A7
Port 5	P50 to P57	8	I/O	Bit	A8 to A15
Port 6	P60 to P67	8	I/O	Bit	A16 to A23
Port 7	P70	1	Output	(Fixed)	$\overline{\text{RD}}$
	P71	1	Output	(Fixed)	$\overline{\text{WRL}}$
	P72	1	Output	(Fixed)	$\overline{\text{WRH}}$
	P73	1	Output	(Fixed)	
	P74	1	Output	(Fixed)	
	P75	1	I/O	Bit	$\overline{\text{BUSRQ}}$
	P76	1	Output	(Fixed)	$\overline{\text{BUSAK}}$
Port 8	P80	1	Output	(Fixed)	$\overline{\text{CS0}}$
	P81	1	Output	(Fixed)	$\overline{\text{CS1}}$/$\overline{\text{RAS0}}$
	P82	1	Output	(Fixed)	$\overline{\text{CS2}}$
	P83	1	Output	(Fixed)	$\overline{\text{CS3}}$/$\overline{\text{RAS1}}$
	P84	1	Output	(Fixed)	$\overline{\text{CS4}}$
	P85	1	Output	(Fixed)	$\overline{\text{CS5}}$
	P86	1	I/O	Bit	$\overline{\text{WAIT}}$
Port A	PA0	1	Output	(Fixed)	$\overline{\text{CAS0}}$/$\overline{\text{LCAS0}}$
	PA1	1	Output	(Fixed)	$\overline{\text{UCAS0}}$
	PA2	1	Output	(Fixed)	$\overline{\text{OE0}}$
	PA3	1	Output	(Fixed)	$\overline{\text{OE1}}$
	PA4	1	Output	(Fixed)	$\overline{\text{WE0}}$
Port B	PB0	1	Output	(Fixed)	$\overline{\text{CAS1}}$/$\overline{\text{LCAS1}}$
	PB1	1	Output	(Fixed)	$\overline{\text{UCAS1}}$
	PB2	1	Output	(Fixed)	
	PB3	1	Output	(Fixed)	
	PB4	1	Output	(Fixed)	$\overline{\text{WE1}}$
Port C	PC0	1	I/O	Bit	TO1/TO7
	PC1	1	I/O	Bit	TO3/TOB
Port D	PD0	1	I/O	Bit	TO4
	PD1	1	I/O	Bit	TI4/INT4
	PD2	1	I/O	Bit	TI5/INT5
	PD4	1	I/O	Bit	TO6
	PD5	1	I/O	Bit	TI6/INT6
	PD6	1	I/O	Bit	TI7/INT7

Table 3.5.2 Port Functions (2/2)

Port Name	Pin Name	Number of Pins	I/O	I/O Setting	Pin Name for built-in function
Port E	PE0	1	I/O	Bit	TO8
	PE1	1	I/O	Bit	TI8/INT8
	PE2	1	I/O	Bit	TI9/INT9
	PE4	1	I/O	Bit	TOA
	PE5	1	I/O	Bit	TIA/INTA
	PE6	1	I/O	Bit	TIB/INTB
Port F	PF0	1	I/O	Bit	TXD0
	PF1	1	I/O	Bit	RXD0
	PF2	1	I/O	Bit	$\overline{\text{CTS0}}$ /SCLK0
	PF4	1	I/O	Bit	TXD1
	PF5	1	I/O	Bit	RXD1
	PF6	1	I/O	Bit	$\overline{\text{CTS1}}$ /SCLK1
Port G	PG0 to PG7	8	Input	(Fixed)	AN0 to AN7
Port H	PH0	1	I/O	Bit	TC0
	PH1	1	I/O	Bit	TC1
	PH2	1	I/O	Bit	TC2
	PH3	1	I/O	Bit	TC3
	PH4	1	I/O	Bit	INT0
Port Z	PZ0 to PZ7	8	I/O	(Fixed)	–

3.5.1 Port 0 (P00 to P07/D0 to D7)

Port 0 is an 8-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register P0CR and function register P0FC.

In addition to functioning as a general-purpose I/O port, port 0 can also function as data bus (D0 to D7).

Setting the AM1 and AM0 pins as shown below and resetting the device initialize port 0 to the following function pins:

AM1	AM0	Function Setting after reset is released.
0	0	Data bus (D0 to D7)
0	1	Data bus (D0 to D7)
1	0	Don't use this setting
1	1	Don't use this setting

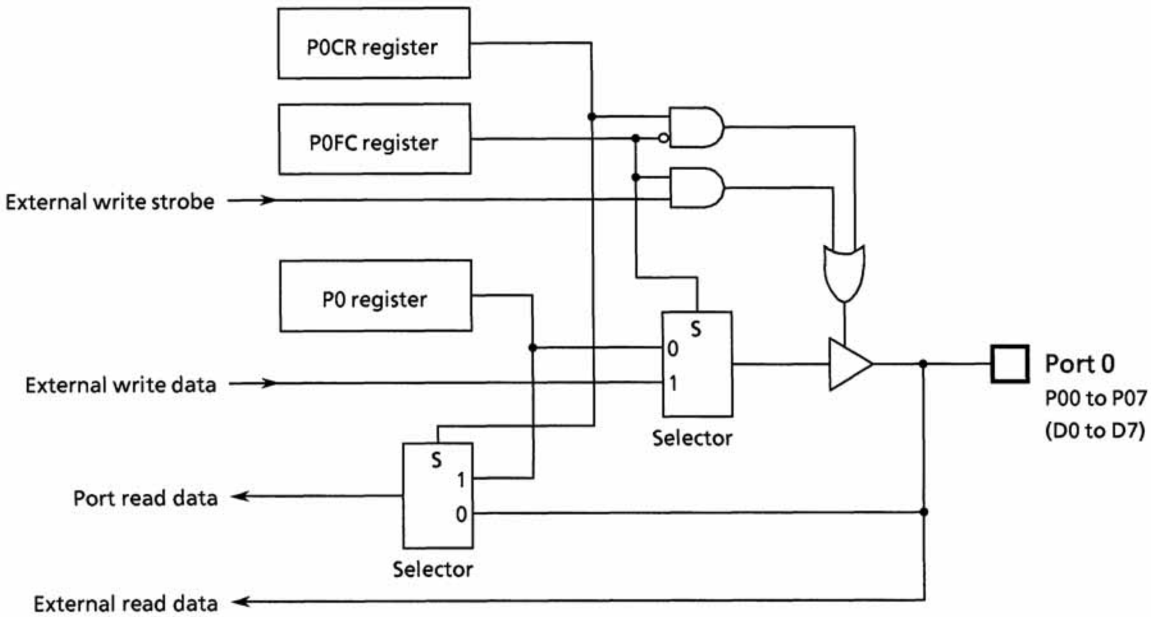


Figure 3.5.1 Port 0

Table 3.5.3 Port 0 Registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
P0	PORT0	00h	P07	P06	P05	P04	P03	P02	P01	P00
			R/W							
			0	0	0	0	0	0	0	0
			Input / Output							
P0CR	PORT0 Control Register	02h	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P0FC	PORT0 Function Register	03h	–	–	–	–	–	–	–	P0F
			W							
			–	–	–	–	–	–	–	0/1
			0: PORT 1: Data Bus (D7 to D0)							

Note)Read-modify-write is prohibited for P0CR and P0FC.

3.5.2 Port 1 (P10 to P17/D8 to D15)

Port 1 is an 8-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register P1CR and function register P1FC.

In addition to functioning as a general-purpose I/O port, port 1 can also function as data bus (D8 to D15).

Setting the AM1 and AM0 pins as shown below and resetting the device initialize port 1 to the following function pins:

AM1	AM0	Function Setting after reset is released.
0	0	Input port
0	1	Data bus (D8 to D15)
1	0	Don't use this setting
1	1	Don't use this setting

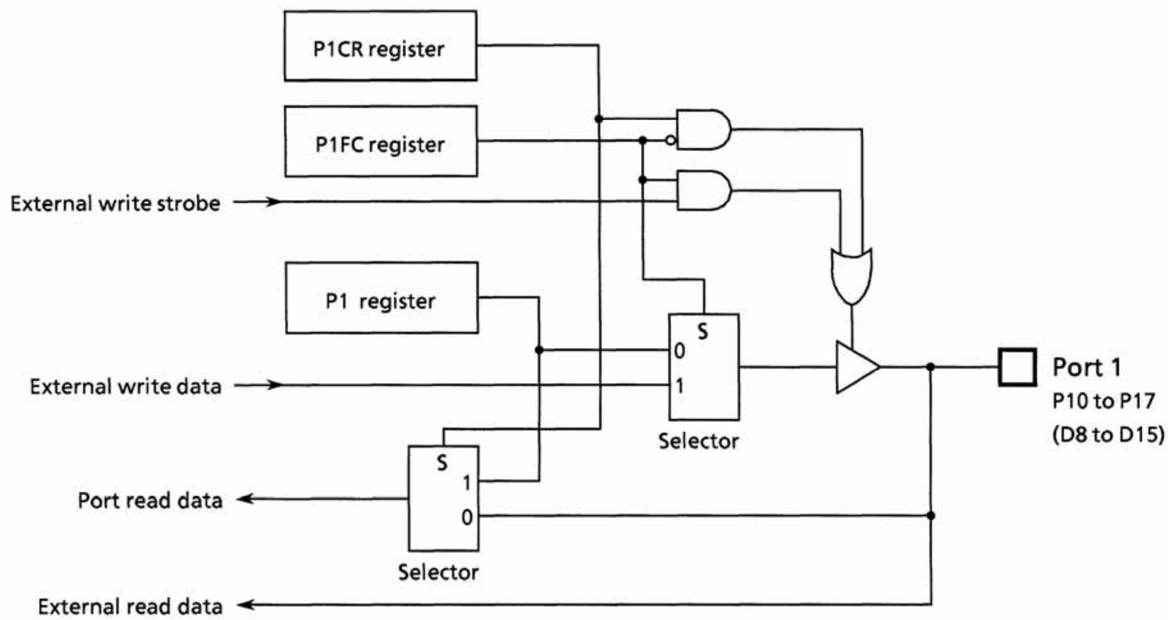


Figure 3.5.2 Port 1

Table 3.5.4 Port 1 Registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
P1	PORT1	04h	P17	P16	P15	P14	P13	P12	P11	P10
			R/W							
			0	0	0	0	0	0	0	0
			Input / Output							
P1CR	PORT1 Control Register	06h	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P1FC	PORT1 Function Register	07h	–	–	–	–	–	–	–	P1F
			W							
			–	–	–	–	–	–	–	0/1
			0: PORT 1: Data Bus (D15 to D8)							

Note)Read-modify-write is prohibited for P1CR and P1FC.

3.5.3 Port 4 (P40 to P47/A0 to A7)

Port 4 is an 8-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register P4CR and function register P4FC.

In addition to functioning as a general-purpose I/O port, port 4 can also function as data bus (A0 to A7). When accessing internal memory and internal I/O, these pins retain the addresses of the previous bus cycle.

Setting the AM1 and AM0 pins as shown below and resetting the device initialize port 4 to the following function pins:

AM1	AM0	Function Setting after reset is released.
0	0	Address bus (A0 to A7)
0	1	Address bus (A0 to A7)
1	0	Don't use this setting
1	1	Don't use this setting

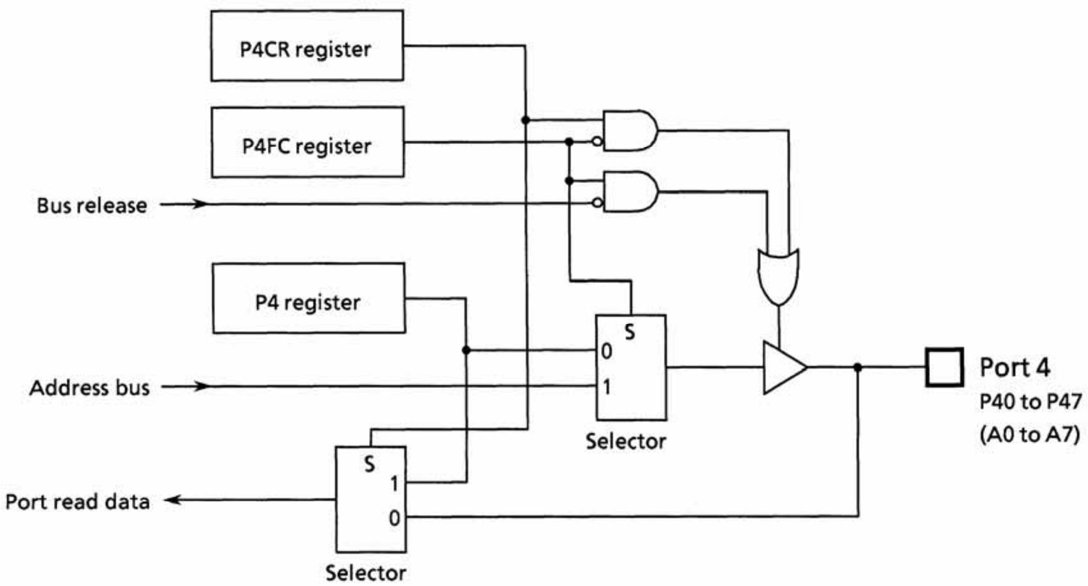


Figure 3.5.3 Port 4

Table 3.5.5 Port 4 Registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
P4	PORT4	10h	P47	P46	P45	P44	P43	P42	P41	P40
			R/W							
			0	0	0	0	0	0	0	0
			Input / Output							
P4CR	PORT4 Control Register	12h	P47C	P46C	P45C	P44C	P43C	P42C	P41C	P40C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P4FC	PORT4 Function Register	13h	P47F	P46F	P45F	P44F	P43F	P42F	P41F	P40F
			W							
			1	1	1	1	1	1	1	1
			0: PORT 1: Adddress Bus (A7 to A0)							

Note)Read-modify-write is prohibited for P4CR and P4FC.

3.5.4 Port 5 (P50 to P57/A8 to A15)

Port 5 is an 8-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register P5CR and function register P5FC.

In addition to functioning as a general-purpose I/O port, port 5 can also function as data bus (A8 to A15). When accessing internal memory and internal I/O, these pins retain the addresses of the previous bus cycle.

Setting the AM1 and AM0 pins as shown below and resetting the device initialize port 5 to the following function pins:

AM1	AM0	Function Setting after reset is released.
0	0	Address bus (A8 to A15)
0	1	Address bus (A8 to A15)
1	0	Don't use this setting
1	1	Don't use this setting

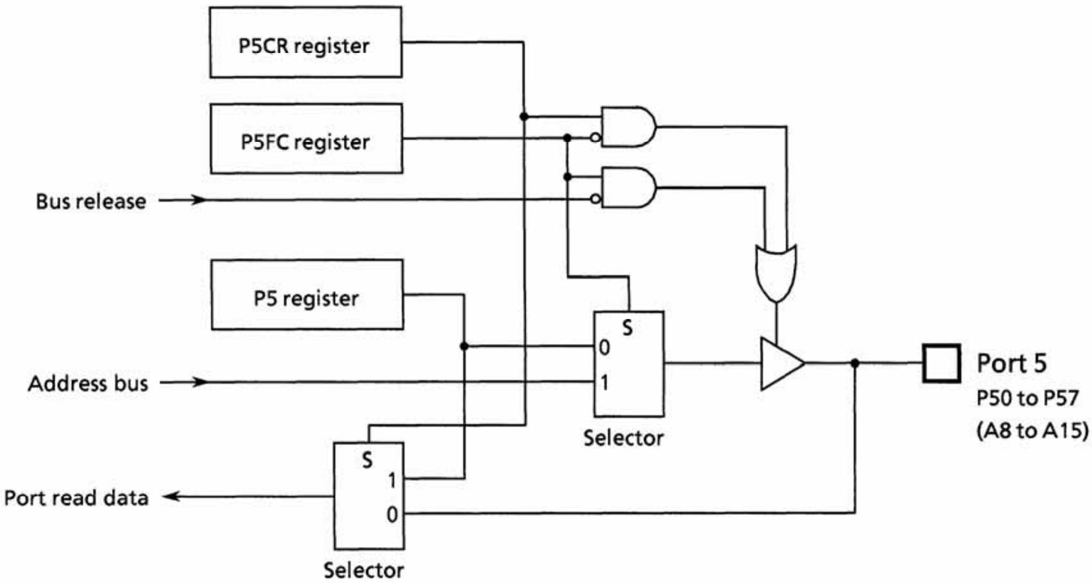


Figure 3.5.4 Port 5

Table 3.5.6 Port 5 Registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
P5	PORT5	14h	P57	P56	P55	P54	P53	P52	P51	P50
			R/W							
			0	0	0	0	0	0	0	0
			Input / Output							
P5CR	PORT5 Control Register	16h	P57C	P56C	P55C	P54C	P53C	P52C	P51C	P50C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P5FC	PORT5 Function Register	17h	P57F	P56F	P55F	P54F	P53F	P52F	P51F	P50F
			W							
			1	1	1	1	1	1	1	1
			0: PORT 1: Adddress Bus (A15 to A8)							

Note)Read-modify-write is prohibited for P5CR and P5FC.

3.5.5 Port 6 (P60 to P67/A16 to A23)

Port 6 is an 8-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register P6CR and function register P6FC.

In addition to functioning as a general-purpose I/O port, port 6 can also function as data bus (A16 to A23). When accessing internal memory and internal I/O, these pins retain the addresses of the previous bus cycle.

Setting the AM1 and AM0 pins as shown below and resetting the device initialize port 6 to the following function pins:

AM1	AM0	Function Setting after reset is released.
0	0	Address bus (A16 to A23)
0	1	Address bus (A16 to A23)
1	0	Don't use this setting
1	1	Don't use this setting

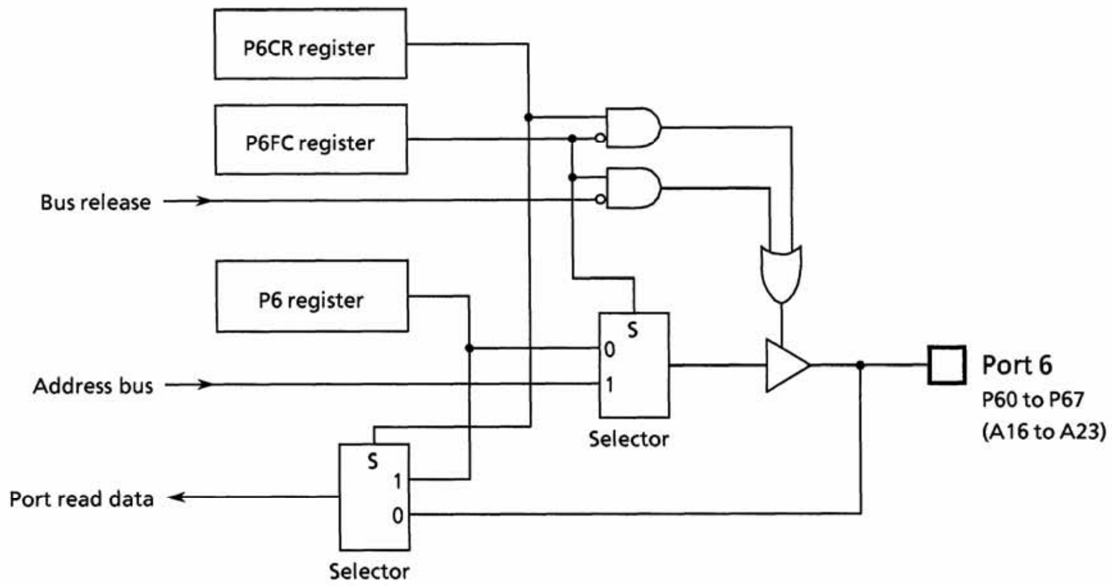


Figure 3.5.5 Port 6

Table 3.5.7 Port 6 Registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
P6	PORT6	18h	P67	P66	P65	P64	P63	P62	P61	P60
			R/W							
			0	0	0	0	0	0	0	0
			Input / Output							
P6CR	PORT6 Control Register	1Ah	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P6FC	PORT6 Function Register	1Bh	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
			W							
			1	1	1	1	1	1	1	1
			0: PORT 1: Adddress Bus (A23 to A16)							

Note)Read-modify-write is prohibited for P6CR and P6FC.

3.5.6 Port 7 (P70 to P76)

Port 7 is an 8-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register P7CR and function register P7FC.

In addition to functioning as a general-purpose I/O port, port 7 can also function as read/write strobe signals to connect with an external memory and control signals to release bus.

A reset initializes P71 to P74 and P76 pins to output port mode, and P75 pin to input port mode.

Setting the AM0 and AM1 pins as shown below and resetting the device initialize port 70 to the following function pins:

AM1	AM0	Function Setting after reset is released
0	0	$\overline{RD}$
0	1	$\overline{RD}$
1	0	Don't use this setting
1	1	Don't use this setting

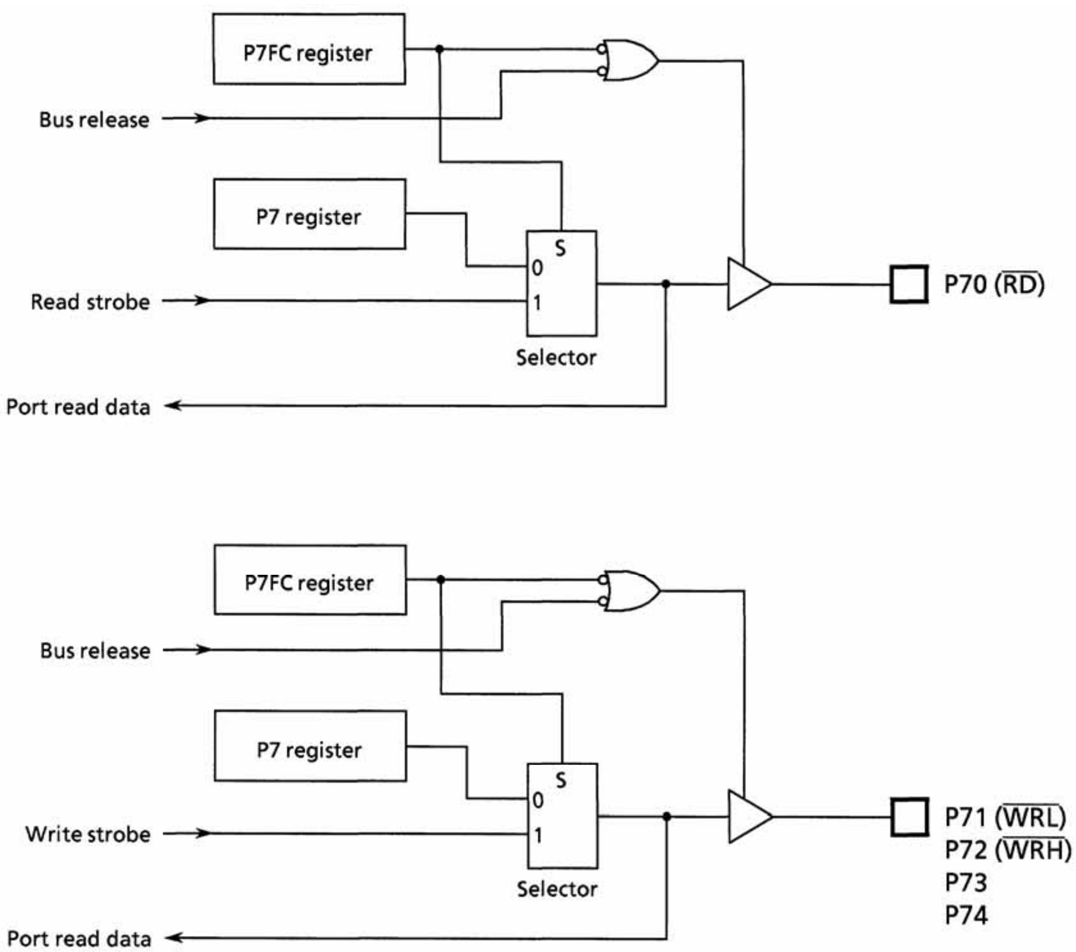


Figure 3.5.6 Port 7 (P70 to P74)

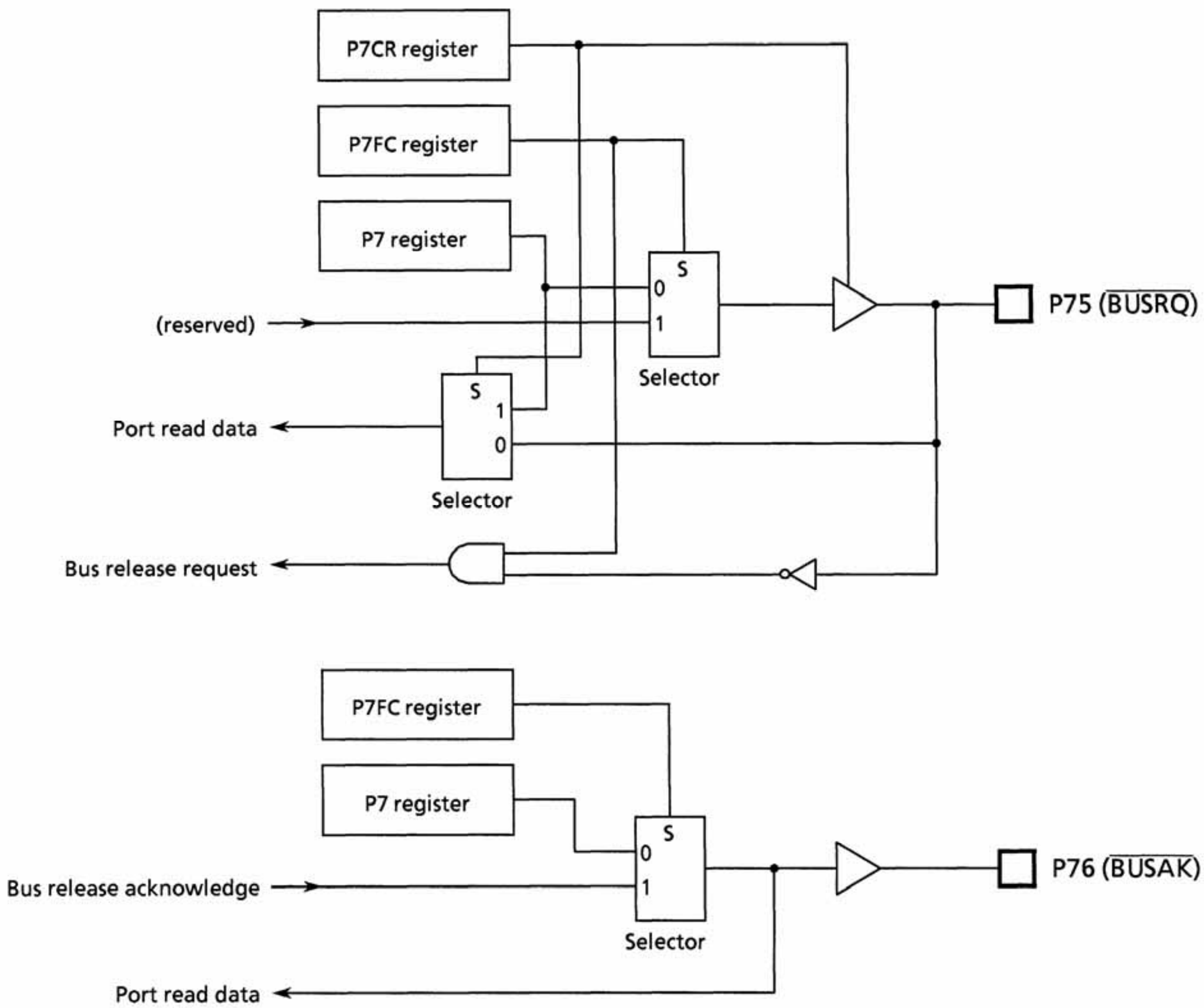


Figure 3.5.7 Port 7 (P75, P76)

Table 3.5.8 Port 7 Registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
P7	PORT7	1Ch	–	P76	P75	P74	P73	P72	P71	P70
				R/W						
			–	1	1	1	1	1	1	1
				Output	In/Out	Output				
P7CR	PORT7 Control Register	1Eh	–	–	P75C	–	–	–	–	–
					W					
			–	–	0	–	–	–	–	–
			0: Input 1: Output							
P7FC	PORT7 Function Register	1Fh	–	P76F	P75F	P74F	P73F	P72F	P71F	P70F
				W						
			–	0	0	0	0	0	0	1
				0: PORT 1: $\overline{BUSAK}$	0: PORT 1: $\overline{BUSRQ}$	0: PORT 1: reserved	0: PORT 1: reserved	0: PORT 1: $\overline{WRH}$	0: PORT 1: $\overline{WRL}$	0: PORT 1: $\overline{RD}$

Note)Read-modify-write is prohibited for P7CR and P7FC.

3.5.7 Port 8 (P80 to P86)

Port 8 is a 7-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register P8CR and function register P8FC.

In addition to functioning as a general-purpose I/O port, port 8 can also function as chip selection to connect with an external memory and wait input.

A reset initializes P80 to P85 pins to output port mode, and P86 pin to input port mode.

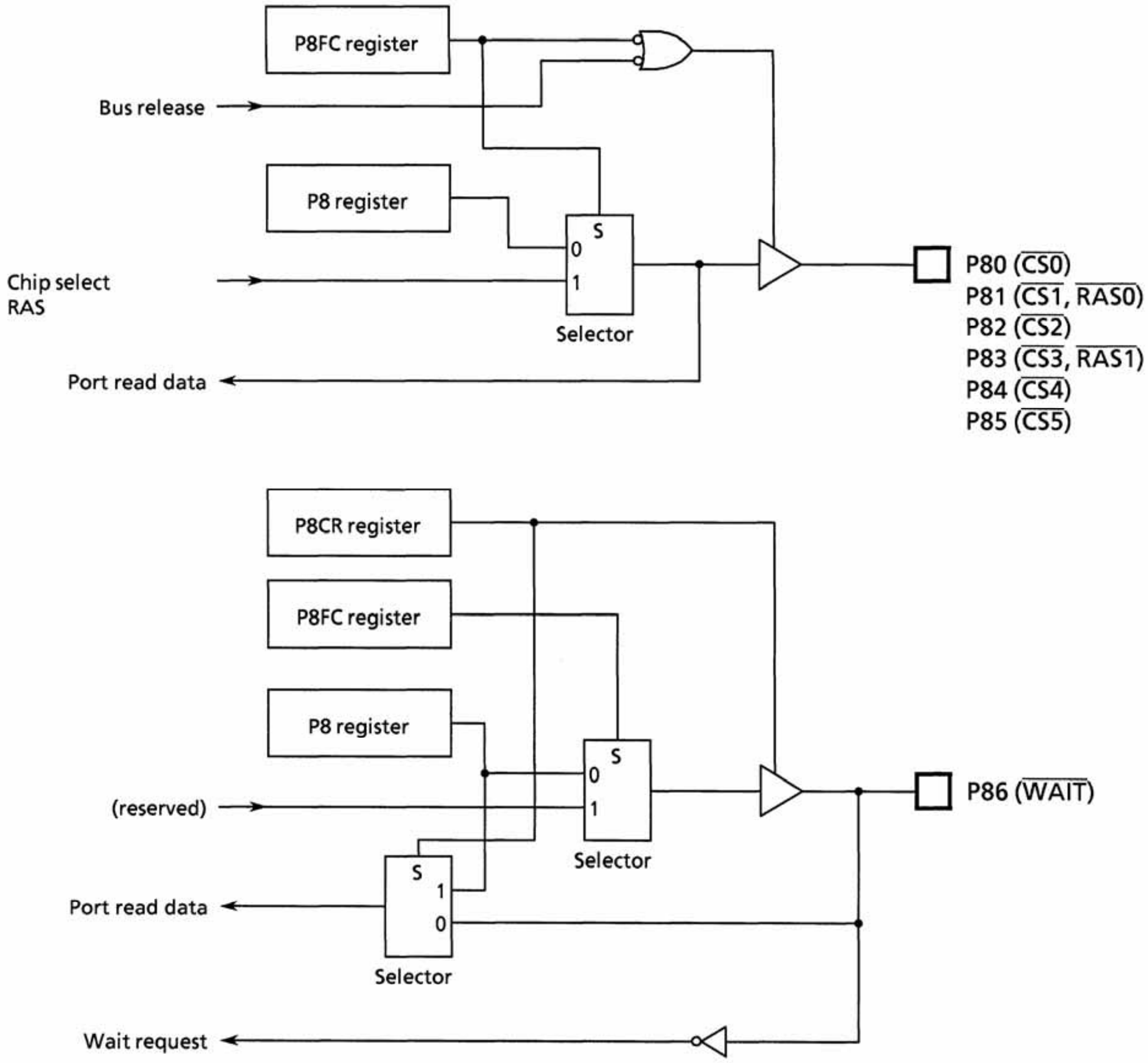


Figure 3.5.8 Port 8

Table 3.5.9 Port 8 Registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
P8	PORT8	20h	–	P86	P85	P84	P83	P82	P81	P80
				R/W						
			–	0	1	1	1	0	1	1
				In/Out	Output					
P8CR	PORT8 Control Register	22h	–	P86C	–	–	–	–	–	–
				W						
			–	0	–	–	–	–	–	–
				0: Input 1: Output						
P8FC	PORT8 Function Register	23h	–	P86F	P85F	P84F	P83F	P82F	P81F	P80F
				W						
			–	0	0	0	0	0	0	0
				0: PORT 1: $\overline{\text{WAIT}}$	0: PORT 1: $\overline{\text{CS5}}$	0: PORT 1: $\overline{\text{CS4}}$	0: PORT 1: $\overline{\text{CS3}}$ $\overline{\text{RAS1}}$	0: PORT 1: $\overline{\text{CS2}}$	0: PORT 1: $\overline{\text{CS1}}$ $\overline{\text{RAS0}}$	0: PORT 1: $\overline{\text{CS0}}$

Note)Read-modify-write is prohibited for P8CR and P8FC.

3.5.8 Port A (PA0 to PA4)

Port A is a 5-bit general-purpose I/O port.

In addition to functioning as a general-purpose I/O port, port A can also function as external DRAM (Channel 0) connection.

A reset initializes port A to output port mode.

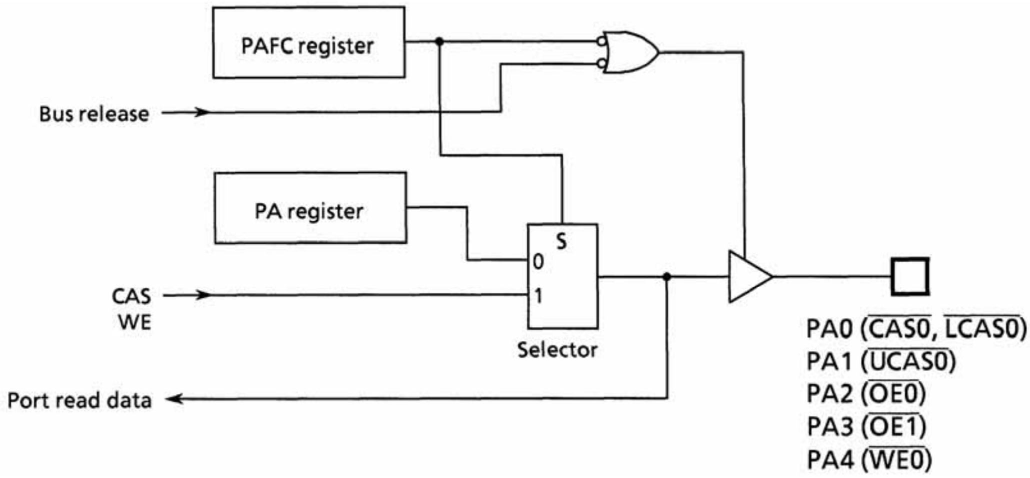


Figure 3.5.9 Port A

Table 3.5.10 Port A Registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
PA	PORTA	28h	–	–	–	PA4	PA3	PA2	PA1	PA0
						R/W				
			–	–	–	1	1	1	1	1
						Output				
PAFC	PORTA Function Register	2Bh	–	–	–	PA4F	PA3F	PA2F	PA1F	PA0F
						W				
			–	–	–	0	0	0	0	0
						0: PORT 1: WE0	0: PORT 1: OE1	0: PORT 1: OE0	0: PORT 1: UCAS0	0: PORT 1: CAS0 LCAS0

Note)Read-modify-write is prohibited for PAFC.

3.5.9 Port B (PB0 to PB4)

Port B is a 5-bit general-purpose I/O port.

In addition to functioning as a general-purpose I/O port, port A can also function as external DRAM (Channel 1) connection.

A reset initializes port A to output port mode.

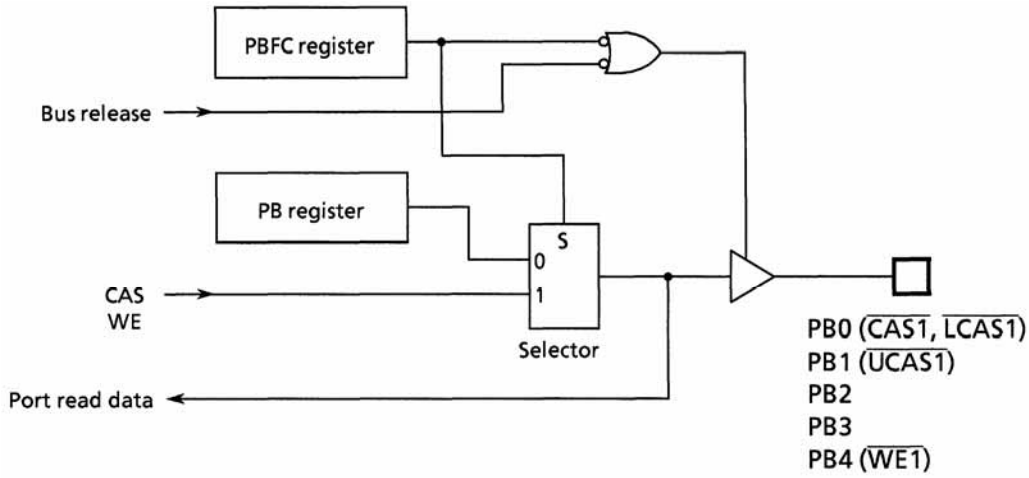


Figure 3.5.10 Port B

Table 3.5.11 Port B Registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
PB	PORTB	2Ch	–	–	–	PB4	PB3	PB2	PB1	PB0
						R/W				
			–	–	–	1	1	1	1	1
						Output				
PBFC	PORTB Function Register	2Fh	–	–	–	PB4F	PB3F	PB2F	PB1F	PB0F
						W				
			–	–	–	0	0	0	0	0
						0: PORT 1: $\overline{\text{WE1}}$	0: PORT 1: reserved	0: PORT 1: reserved	0: PORT 1: $\overline{\text{UCAS1}}$	0: PORT 1: $\overline{\text{CAS1}}$ $\overline{\text{LCAS1}}$

Note)Read-modify-write is prohibited for PBFC.

3.5.10 Port C (PC0 to PC4)

Port C is a 2-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register PCCR and function register PCFC.

In addition to functioning as a general-purpose I/O port, port C can also function as 8-bit timer or 16-bit timer output.

A reset initializes port C to output port mode.

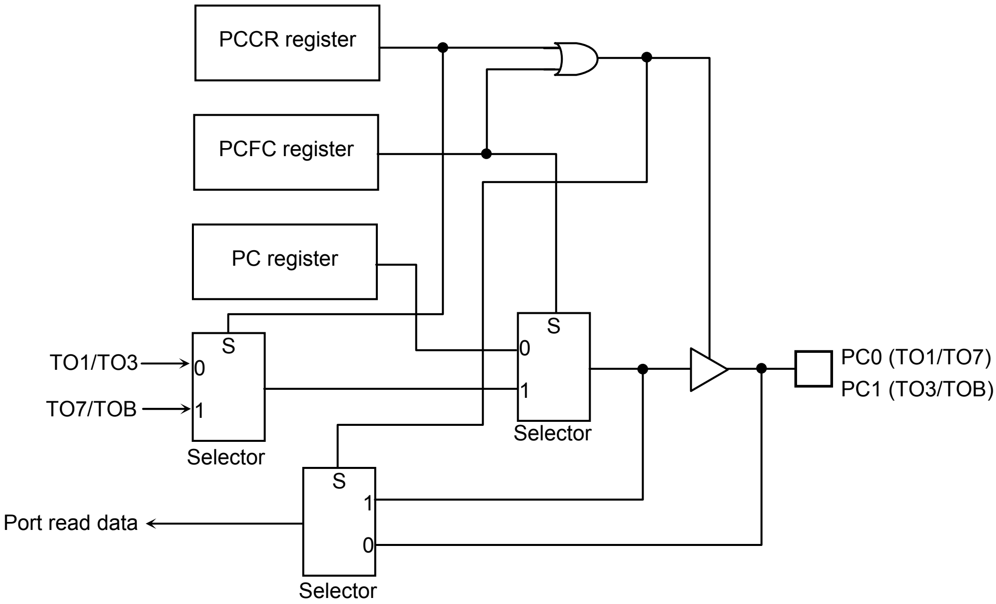


Figure 3.5.11 Port C

Table 3.5.12 Port C Registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
PC	PORTC	30h	–	–	–	–	–	–	PC1	PC0
									R/W	
			–	–	–	–	–	–	0	0
									Input / Output	
PCCR	PORTC Control Register	32h	–	–	–	–	–	–	PC1C	PC0C
									W	
			–	–	–	–	–	–	0	0
									(See below)	
PCFC	PORTC Function Register	33h	–	–	–	–	–	–	PC1F	PC0F
									W	
			–	–	–	–	–	–	0	0
									(See below)	

PCFC	PCCR	function PC1	function PC0
0	0	Input Port	
0	1	Output Port	
1	0	TO3	TO1
1	1	TOB	TO7

Note)Read-modify-write is prohibited for PCCR and PCFC.

3.5.11 Port D (PD0 to PD6)

Port D is a 6-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register PDCR and function register PDFC.

In addition to functioning as a general-purpose I/O port, port D can also function as 16-bit timer I/O and interrupt input.

A reset initializes port D to output port mode.

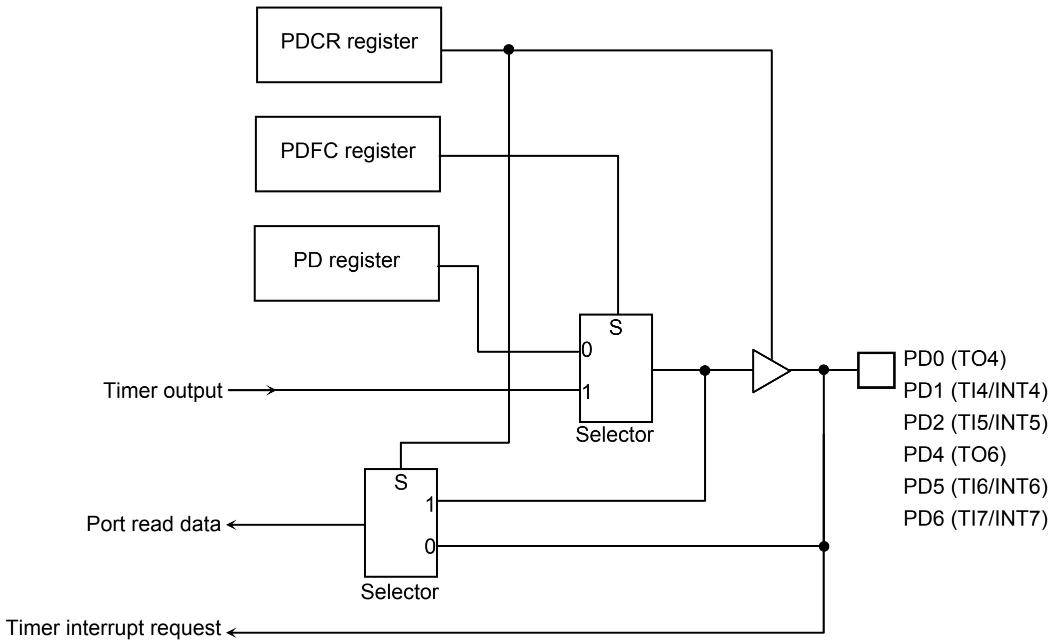


Figure 3.5.12 Port D

Table 3.5.13 Port D Registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
PD	PORTD	34h	–	PD6	PD5	PD4	–	PD2	PD1	PD0
				R/W				R/W		
			–	0	0	0	–	0	0	0
				Input/Output				Input/Output		
PDCR	PORTD Control Register	36h	–	PD6C	PD5C	PD4C	–	PD2C	PD1C	PD0C
				W				W		
			–	0	0	0	–	0	0	0
				0: Input 1: Output				0: Input 1: Output		
PDFC	PORTD Function Register	37h	–	PD6F	PD5F	PD4F	–	PD2F	PD1F	PD0F
				W				W		
			–	0	0	0	–	0	0	0
				0: PORT 1: TI7 INT7	0: PORT 1: TI6 INT6	0: PORT 1: TO6		0: PORT 1: TI5 INT5	0: PORT 1: TI4 INT4	0: PORT 1: TO4

Note)Read-modify-write is prohibited for PDCR and PDFC.

3.5.12 Port E (PE0 to PE6)

Port E is a 6-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register PECR and function register PEFC.

In addition to functioning as a general-purpose I/O port, port E can also function as 8-bit timer or 16-bit timer output and interrupt input.

A reset initializes port E to input port mode.

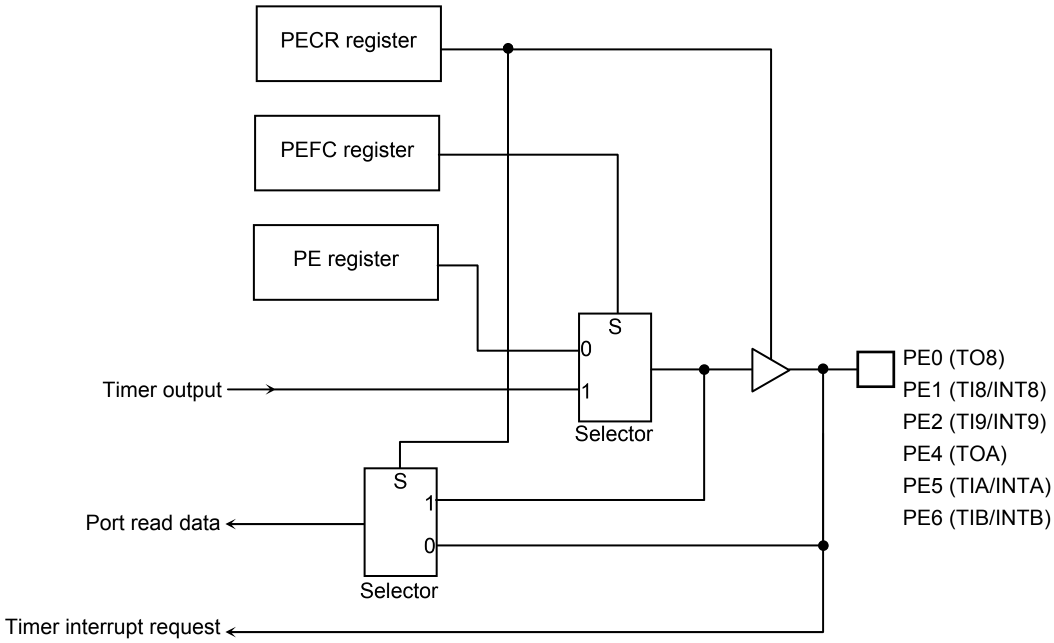


Figure 3.5.13 Port E

Table 3.5.14 Port E Registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
PE	PORTE	38h	–	PE6	PE5	PE4	–	PE2	PE1	PE0
				R/W				R/W		
			–	0	0	0	–	0	0	0
				Input/Output				Input/Output		
PECR	PORTE Control Register	3Ah	–	PE6C	PE5C	PE4C	–	PE2C	PE1C	PE0C
				W				W		
			–	0	0	0	–	0	0	0
				0: Input 1: Output				0: Input 1: Output		
PEFC	PORTE Function Register	3Bh	–	PE6F	PE5F	PE4F	–	PE2F	PE1F	PE0F
				W				W		
			–	0	0	0	–	0	0	0
				0: PORT 1: TIB INTB	0: PORT 1: TIA INTA	0: PORT 1: TOA		0: PORT 1: TI9 INT9	0: PORT 1: TI8 INT8	0: PORT 1: TO8

Note)Read-modify-write is prohibited for PECR and PEFC.

3.5.13 Port F (PF0 to PF6)

Port F is a 6-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register PFCR and function register PFFC.

In addition to functioning as a general-purpose I/O port, port F can also function as I/O functions of serial interface.

A reset initializes port F to input port mode.

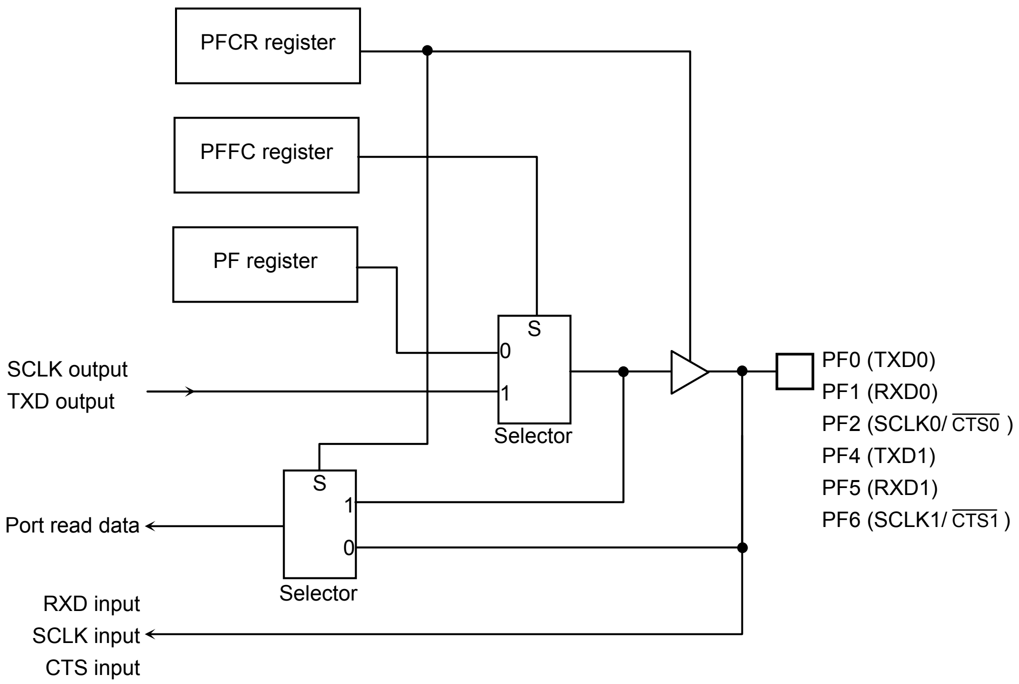


Figure 3.5.14 Port F

Table 3.5.15 Port F Registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
PF	PORTF	3Ch	–	PF6	PF5	PF4	–	PF2	PF1	PF0
					R/W				R/W	
			–	0	0	0	–	0	0	0
					Input/Output				Input/Output	
PFCR	PORTF Control Register	3Eh	–	PF6C	PF5C	PF4C	–	PF2C	PF1C	PF0C
					W				W	
			–	0	0	0	–	0	0	0
					0: Input 1: Output				0: Input 1: Output	
PFFC	PORTF Function Register	3Fh	–	PF6F	PF5F	PF4F	–	PF2F	PF1F	PF0F
					W				W	
			–	0	0	0	–	0	0	0
				0: PORT 1: $\overline{CTS1}$ SCLK1	0: PORT 1: RxD1	0: PORT 1: TxD1		0: PORT 1: $\overline{CTS0}$ SCLK0	0: PORT 1: RxD0	0: PORT 1: TxD0

Note)Read-modify-write is prohibited for PFCR and PFFC.

3.5.14 Port G (PG0 to PG7)

Port G is an 8-bit general-purpose input-only port.

In addition to functioning as a general-purpose I/O port, port G can also function as I/O functions of AD converter.

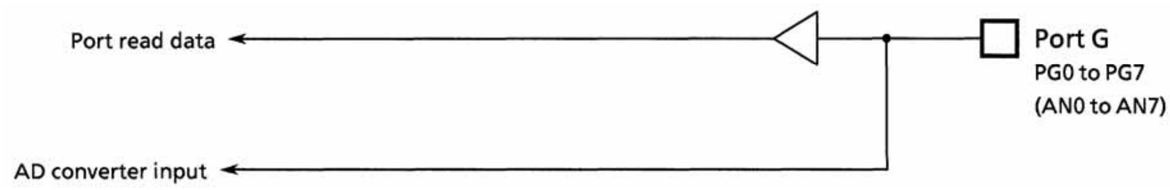


Figure 3.5.15 Port G

Table 3.5.16 Port G Register

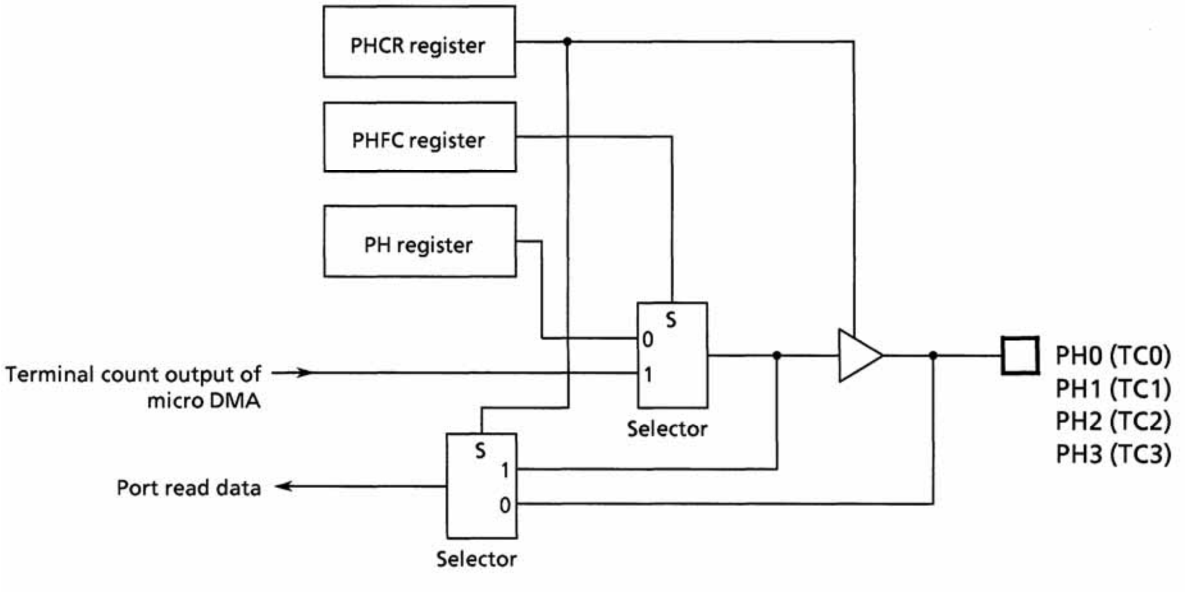
Symbol	Name	Address	7	6	5	4	3	2	1	0
PG	PORTG	40h	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
			R							
			Input							

3.5.15 Port H (PH0 to PH4)

Port H is a 5-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register PHCR and function register PHFC.

In addition to functioning as a general-purpose I/O port, port H can also function as terminal count output function of micro DMA and interrupt input function.

A reset initializes port H to input port mode.



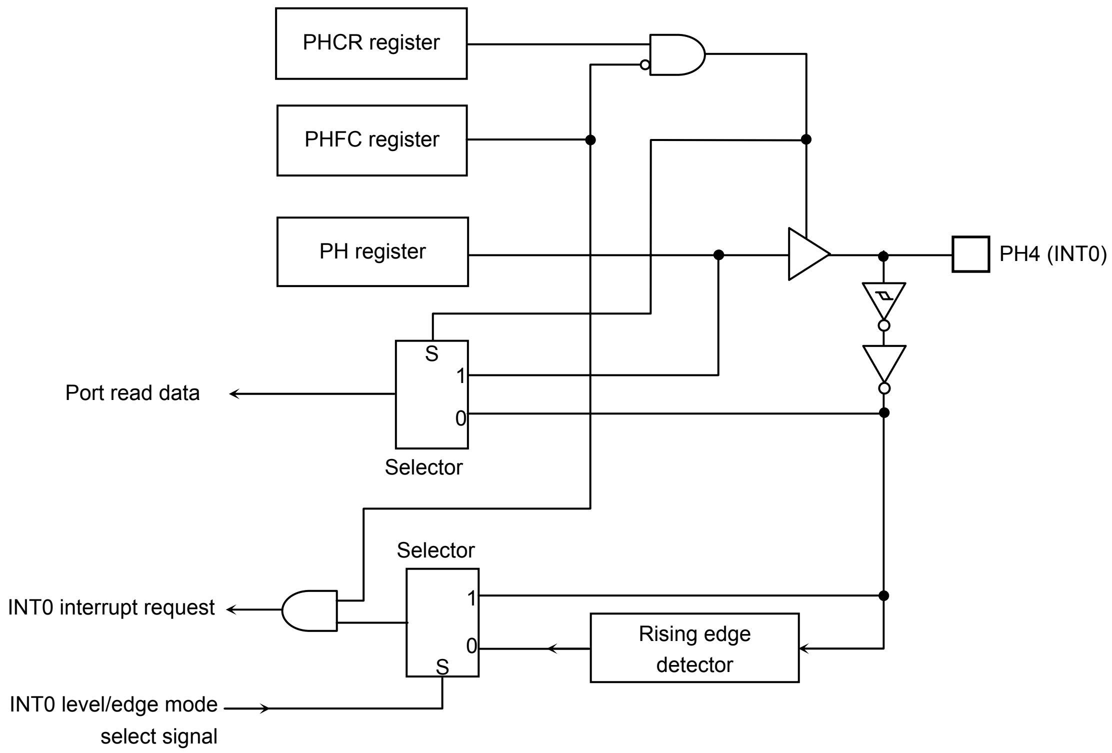


Figure 3.5.16 Port H

Table 3.5.17 Port H Registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
PH	PORTH	44h	–	–	–	PH4	PH3	PH2	PH1	PH0
						R/W				
			–	–	–	0	0	0	0	0
						Input/Output				
PHCR	PORTH Control Register	46h	–	–	–	PH4C	PH3C	PH2C	PH1C	PH0C
						W				
			–	–	–	0	0	0	0	0
						0: Input 1: Output				
PHFC	PORTH Function Register	47h	–	–	–	PH4F	PH3F	PH2F	PH1F	PH0F
						W				
			–	–	–	0	0	0	0	0
						0: PORT 1: INT0	0: PORT 1: TC3	0: PORT 1: TC2	0: PORT 1: TC1	0: PORT 1: TC0

Note)Read-modify-write is prohibited for PHCR and PHFC.

3.5.16 Port Z (PZ0 to PZ7)

Port Z is an 8-bit general-purpose I/O port. Bits can be individually set as either inputs or outputs by control register PZCR.

A reset initializes port Z to input port mode.

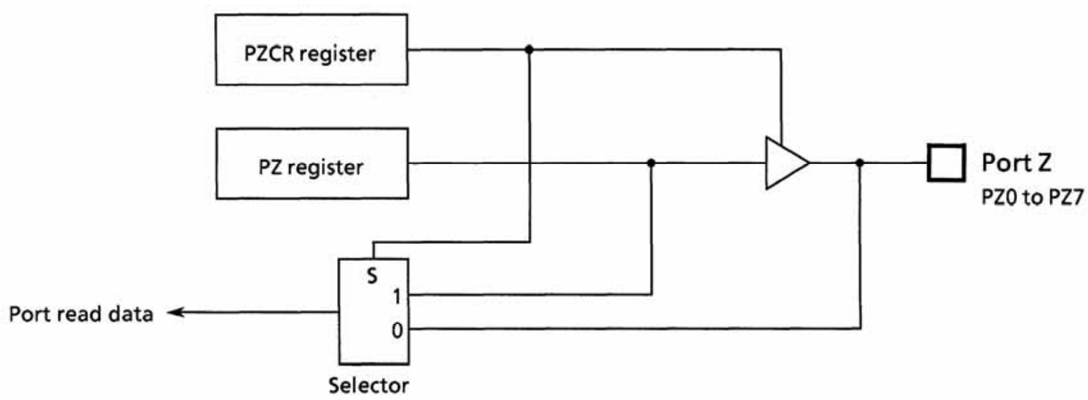


Figure 3.5.17 Port Z

Table 3.5.18 Port Z Registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
PZ	PORTZ	68h	PZ7	PZ6	PZ5	PZ4	PZ3	PZ2	PZ1	PZ0
			R/W							
			0	0	0	0	0	0	0	0
			Input / Output							
PZCR	PORTZ Control Register	6Ah	PZ7C	PZ6C	PZ5C	PZ4C	PZ3C	PZ2C	PZ1C	PZ0C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							

Note)Read-modify-write is prohibited for PZCR.

3.6 Memory Controller

3.6.1 Functions

TMP94C251A has a memory controller with a variable 6-block address area that controls as follows.

(1) 6-block address area support

Specifies a start address and a block size respectively for 6-block address area (block 0 to 5).

(2) Connecting memory specification

Specifies SRAM, ROM and DRAM as memories to connect with the respective block address areas. DRAM is specified only in block 1 and block 3.

When SRAM or ROM is specified, a usual bus cycle is executed. When DRAM is specified, DRAM is effectively accessed with built-in DRAM controller. The page access of ROM is also supported in block 2. For details, see 3.6.4 “ROM Control (Page mode)”.

(3) Data bus size selection

Whether 8 bits, 16 bits is selected as the data bus size of the respective block address areas.

(4) Wait control

Wait specification bit in the control register and WAIT input pin control the number of waits in the external bus cycle. Read cycle and write cycle can specify the number of waits individually. The number of waits is controlled in three mode mentioned below.

0 waits, 1 wait, 2 waits, 3 waits, N waits (controls with $\overline{\text{WAIT}}$ pin)

(5) DRAM control

TMP94C251A has DRAM controller to control refresh and DRAM accessing.

This document describes in order of the operation after reset release, basic functions and ROM page mode.

Each section explains the operation and the register setting method and the signal timing. The register setting method is mentioned as the lists of registers in the final.

3.6.2 Control Register and Operation after Reset Release

This section describes the registers to control the memory controller, the state after reset release and necessary settings.

(1) Control register

The control registers of the memory controller are as follows.

- Control register: BnCSH/BnCSL (n = 0 to 5)
  Sets the basic functions of the memory controller, that is the connecting memory type, the data bus size, the number of waits to be read and written.
- Memory start address register: MSARn (n = 0 to 5)
  Sets a start address in the respective block address areas.
- Memory address mask register: MAMRn (n = 0 to 5)
  Sets a block size in the respective block address areas.

In addition to setting of the above-mentioned registers, it is necessary to set the following registers to control ROM page mode access and DRAM.

- Page ROM control register: PMEMCR
  Sets to executed ROM page mode accessing.
- DRAM control register: DRAMnCRL/DRAMnCRH (n = 0 to 1)
  Sets DRAM access.
- DRAM refresh control register: DRAMnREF (n = 0 to 1)
  Sets DRAM refresh operation.

(2) Operation After Reset Release

The start data bus size is determined depending on the state of AM1/AM0 pins just after reset release. Then, the external memory is accessed as follows.

AM1	AM0	Start mode
0	0	Start with 8-bit data bus
0	1	Start with 16-bit data bus
1	0	Don't use this setting
1	1	Don't use this setting

AM1/AM0 pins are valid only just after reset release. In the other cases, the data bus width is set to the value set to BnBUS bit of the control register.

After reset, only control register (B2CSH/B2CSL) of the block address area 2 is automatically valid. The data bus width which is specified by AM1/AM0 pins is loaded to the bit to specify the bus width of the control register in the block address area 2. The block address area 2 is set to addresses 000000H to FFFFFFH after reset.

After reset release, the block address areas are specified by the memory start address register (MSARn) and the memory address mask register (MAMRn). Then the control register (BnCS) is set.

Set the enable bit (BnE) of the control register to "1" to enable the setting. Set relevant registers to access ROM page mode and DRAM.

3.6.3 Basic Functions and Register Setting

In this section, setting of the block address area, the data bus width, the connecting memory and the number of waits out of the memory controller's functions are described.

(1) Block address area specification

The block address area is specified by two registers.

The memory start address register (MSARn) sets the start address of the block address areas. The memory controller compares between the register value and the address every bus cycles. The address bit which is masked by the memory address mask register (MAMRn) is not compared by the memory controller. The block address area size is determined by setting the memory address mask register. The set value in the register is compared with the block address area on the bus. If the compared result is a match, the memory controller sets the chip select signal (CSn) to "Low".

(i) Setting memory start address register

The MnS23 to MnS16 bits of the memory start address register respectively correspond with addresses A23 to A16. The lower start address A15 to A0 are always set to address 0000H. Therefore, the start address of the block address area are set to addresses 000000H to FF0000H every 64 Kbytes.

(ii) Setting memory address mask registers

The memory address mask register sets whether an address bit is compared or not. Set the register to “0” to compare, or to “1” not to compare.

The address bit to be set is depended on the block address area.

Block address area 0: A20 to A8

Block address area 1: A21 to A8

Block address area 2 to 5: A22 to A15

The above-mentioned bits are always compared. The block address area size is determined by the compared result.

The size to be set depending on the block address areas is as follows.

	256	512	32K	64K	128K	256K	512K	1M	2M	4M	8M
CS0	○	○	○	○	○	○	○	○	○		
CS1	○	○		○	○	○	○	○	○	○	
CS2 to 5			○	○	○	○	○	○	○	○	○

[Unit: Byte]

Note: After reset release, only the control register of the block address area 2 is valid. The control register of the block address area 2 has B2M bit. Setting B2M bit to “0” sets the block address area 2 to addresses 000000H to FFFFFFH. Setting B2M bit to “1” specifies the start address and the address area size as it is in the other block address area.

(iii) Example of register setting

To set the block address area 1 to 512 bytes from address 110000H, set the register as follows.

MSAR1 Register

bit	7	6	5	4	3	2	1	0
bit Symbol	M1S23	M1S22	M1S21	M1S20	M1S19	M1S18	M1S17	M1S16
Specified value	0	0	0	1	0	0	0	1

M1S23 to M1S16 bits of the memory start address register MSAR1 correspond with addresses A23 to A16.

A15 to A0 are set to “0”. Therefore, setting MSAR1 to the above-mentioned value specifies the start address of the block address area 1 to address 110000H.

The start address is set as it is in the other block address areas.

MAMR1 Register

bit	7	6	5	4	3	2	1	0
bit Symbol	M1V21	M1V20	M1V19	M1V18	M1V17	M1V16	M1V15 to 9	M1V8
Specified value	0	0	0	0	0	0	0	1

M1V21 to M1V16 and M1V8 bits of the memory address mask register MAMR1 set whether addresses A21 to A16 and A8 are compared or not. Set the register to "0" to compare, or to "1" not to compare. M1V15 to M1V9 bits set whether addresses A15 to A9 are compared or not with 1 bit. A23 and A22 are always compared.

Setting the above-mentioned compares A23 to A9 with the values set as the start addresses. Therefore, 512 bytes of addresses 110000H to 1101FFH are set as the block address area 1, and compared with the addresses on the bus. If the compared result is a match, the chip select signal CS1 is set to "Low".

The other block address area sizes are specified like this.

A23 and A22 are always compared in the block address area 0. Whether A20 to A8 are compared or not is set to the register.

Similarly, A23 is always compared in the block address areas 2 to 5. Whether A22 to A15 are compared or not is set to the register.

Note: When the set block address area overlaps with the built-in memory area, or both two address areas overlap, the block address areas are processed according to priority as follows.

Built-in I/O > Built-in memory > Block address area 0 > 1 > 2 > 3 > 4 > 5

Note also that any accessed areas outside the address spaces set by $\overline{\text{CS0}}$ to $\overline{\text{CS5}}$ are processed as the $\overline{\text{CS2}}$ space. Therefore, settings of $\overline{\text{CS2}}$ apply for the control of wait cycles, data bus width, etc., and the $\overline{\text{CS2}}$ signal is output.

(2) Connection memory specification

Setting the BnOM1 to BnOM0 bit of the control register (BnCSH) specifies the memory type to be connected with the block address areas. The interface signal is output according to the set memory as follows.

BnOM1, BnOM0 Bit (BnCSH register)

BnOM1	BnOM0	Function
0	0	SRAM/ROM (Default)
0	1	(Reserved)
1	0	DRAM
1	1	(Reserved)

DRAM is set only in the block address are 1 and 3.

When ROM is selected, the page mode is accessed. It is possible to specify only in the block address area 2.

(3) Data bus width specification

The data bus width is set for every block address areas. The bus size is set by the BnBUS1 and BnBUS0 bits of the control register (BnCSH) as follows.

BnBUS Bit (BnCSH register)

BnBUS1	BnBUS0	Function
0	0	8-bit bus mode (Default)
0	1	16-bit bus mode
1	0	(Reserved)
1	1	(Reserved)

This way of changing the data bus size depending on the address being accessed is called "dynamic bus sizing". The part where the data is output to is depended on the data size, the bus width and the start address.

Note: Since there is a possibility of abnormal writing/reading of the data if two memories with different bus width are put in consecutive address, do not execute a access both memories with one command.

Operand data size (bit)	Operand start address	Memory data size (bit)	CPUaddress	CPU data D15 to D8	CPU data D7 to D0
8	4n + 0	8/16	4n + 0	xxxxx	b7 to b0
	4n + 1	8	4n + 1	xxxxx	b7 to b0
		16	4n + 1	b7 to b0	xxxxx
	4n + 2	8/16	4n + 2	xxxxx	b7 to b0
		8	4n + 3	xxxxx	b7 to b0
	4n + 3	16	4n + 3	b7 to b0	xxxxx
16	4n + 0	8	(1) 4n + 0 (2) 4n + 1	xxxxx xxxxx	b7 to b0 b15 to b8
		16	4n + 0	b15 to b8	b7 to b0
	4n + 1	8	(1) 4n + 1 (2) 4n + 2	xxxxx xxxxx	b7 to b0 b15 to b8
		16	(1) 4n + 1 (2) 4n + 2	b7 to b0 xxxxx	xxxxx b15 to b8
	4n + 2	8	(1) 4n + 2 (2) 4n + 1	xxxxx xxxxx	b7 to b0 b15 to b8
		16	4n + 2	b15 to b8	b7 to b0
	4n + 3	8	(1) 4n + 3 (2) 4n + 4	xxxxx xxxxx	b7 to b0 b15 to b8
		16	(1) 4n + 3 (2) 4n + 4	b7 to b0 xxxxx	xxxxx b15 to b8
32	4n + 0	8	(1) 4n + 0 (2) 4n + 1 (3) 4n + 2 (4) 4n + 3	xxxxx xxxxx xxxxx xxxxx	b7 to b0 b15 to b8 b23 to b16 b31 to b24
		16	(1) 4n + 0 (2) 4n + 2	b15 to b8 b31 to b24	b7 to b0 b23 to b16
	4n + 1	8	(1) 4n + 1 (2) 4n + 2 (3) 4n + 3 (4) 4n + 4	xxxxx xxxxx xxxxx xxxxx	b7 to b0 b15 to b8 b23 to b16 b31 to b24
		16	(1) 4n + 1 (2) 4n + 2 (3) 4n + 4	b7 to b0 b23 to b16 xxxxx	xxxxx b15 to b8 b31 to b24
	4n + 2	8	(1) 4n + 2 (2) 4n + 3 (3) 4n + 4 (4) 4n + 5	xxxxx xxxxx xxxxx xxxxx	b7 to b0 b15 to b8 b23 to b16 b31 to b24
		16	(1) 4n + 2 (2) 4n + 4	b15 to b8 b31 to b24	b7 to b0 b23 to b16
	4n + 3	8	(1) 4n + 3 (2) 4n + 4 (3) 4n + 5 (4) 4n + 6	xxxxx xxxxx xxxxx xxxxx	b7 to b0 b15 to b8 b23 to b16 b31 to b24
		16	(1) 4n + 3 (2) 4n + 4 (3) 4n + 6	b7 to b0 b23 to b16 xxxxx	xxxxx b15 to b8 b31 to b24

xxxxx: During a read, data input to the bus is ignored. At write, the bus is at high impedance and the write strobe signal remains non to active.

(4) Wait control

The external bus cycle completes a wait of two states at least (100ns at 20 MHz). Setting the BnWW2 to BnWW0 and BnWR2 to BnWR0 of the control register (BnCSL) specifies the number of waits in the read cycle and the write cycle. BnWW is set with the same method as BnWR.

BnWW/BnWR Bit (BnCSL register)

BnWW2 BnWR2	BnWW1 BnWR1	BnWW0 BnWR0	Function
0	0	1	2 states (0 wait) access fixed mode
0	1	0	3 states (1 wait) access fixed mode (Default)
1	0	1	4 states (2 wait) access fixed mode
1	1	0	5 states (3 wait) access fixed mode
0	1	1	WAIT pin input mode
others			(Reserved)

Note: When DRAM is specified as a connecting memory, setting should be 3 states (1 wait) or more. In the case of DRAM access, the WAIT pin input mode cannot be used.

(i) Waits number fixed mode

The bus cycle is completed with the set states. The number of states is selected from 2 states (0 waits) to 5 states (3 waits).

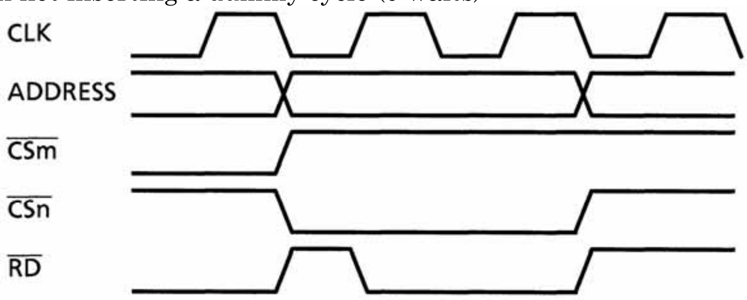
(ii) WAIT pin input mode

This mode samples the WAIT input pins. It continuously samples the WAIT pin state and inserts a wait if the pin is active. The bus cycle is minimum 2 states. The bus cycle is completed when the wait signal is non-active ("High" level) at 2 states. The bus cycle extends if the wait signal is active at 2 states and more.

BnREC Bit (BnCSH register)

0	No dummy cycle is inserted (default).
1	Dummy cycle is inserted.

When not inserting a dummy cycle (0 waits)



When inserting a dummy cycle (0 waits)

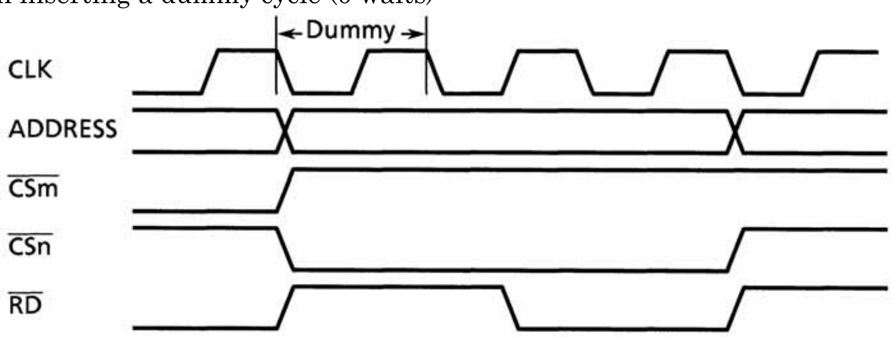
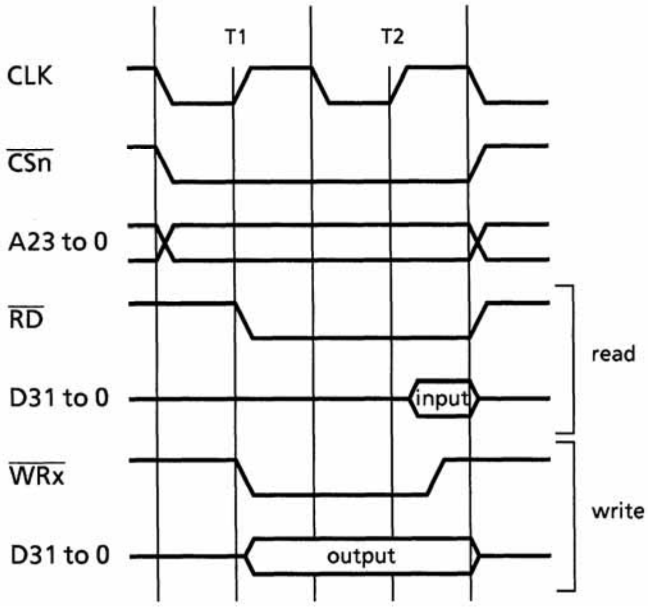


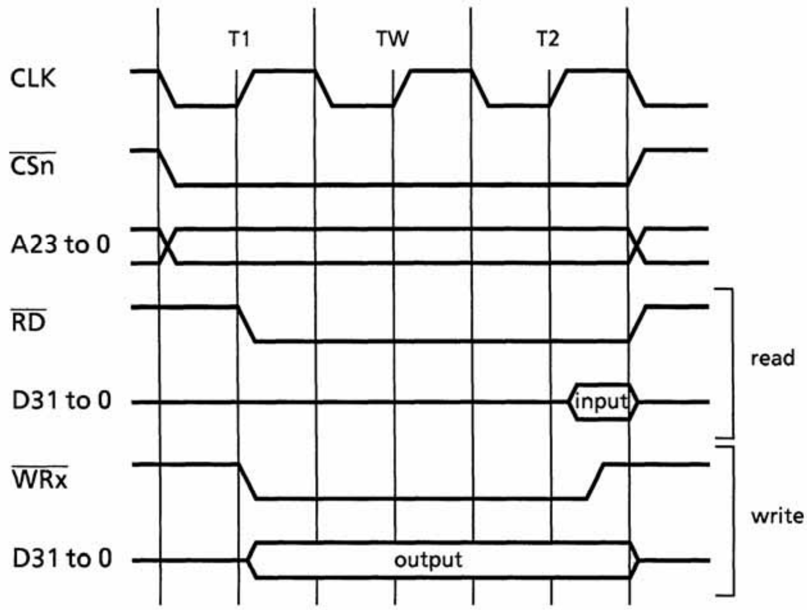
Figure 3.6.1 Read Cycle when Dummy Cycle is Inserted

(5) Basic bus timing

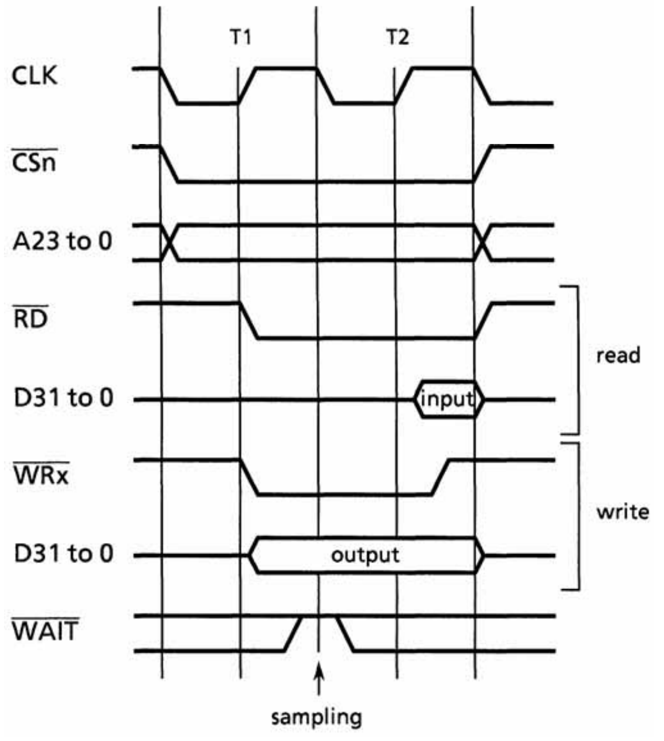
External read/write bus cycle (0 waits)



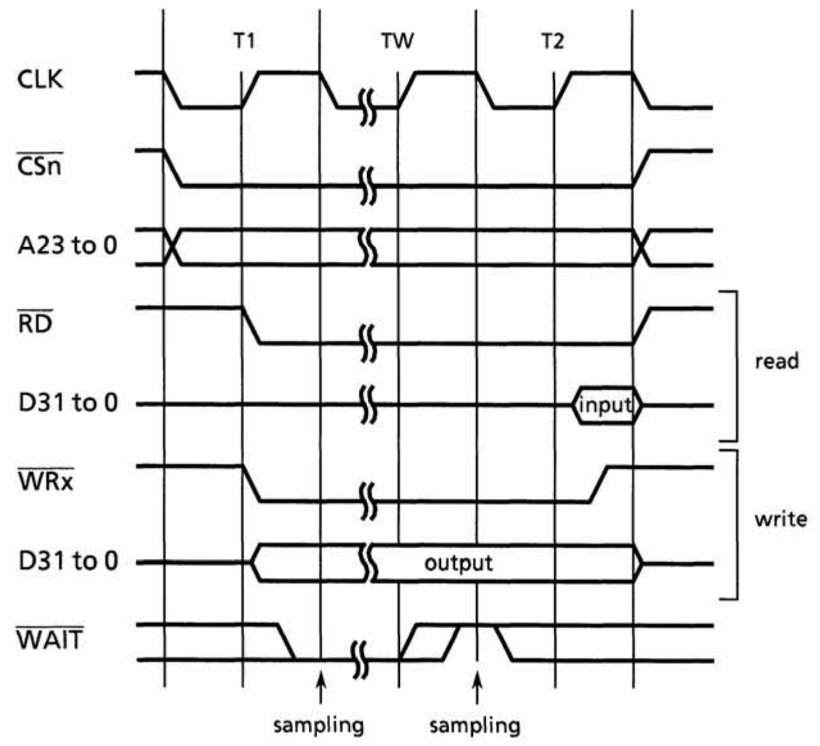
External read/write bus cycle (1 wait)



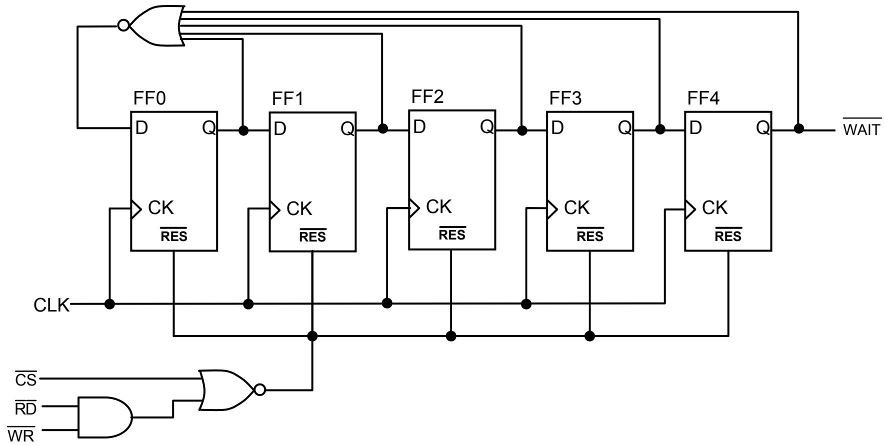
External read/write bus cycle (0 waits at WAIT pin input mode)

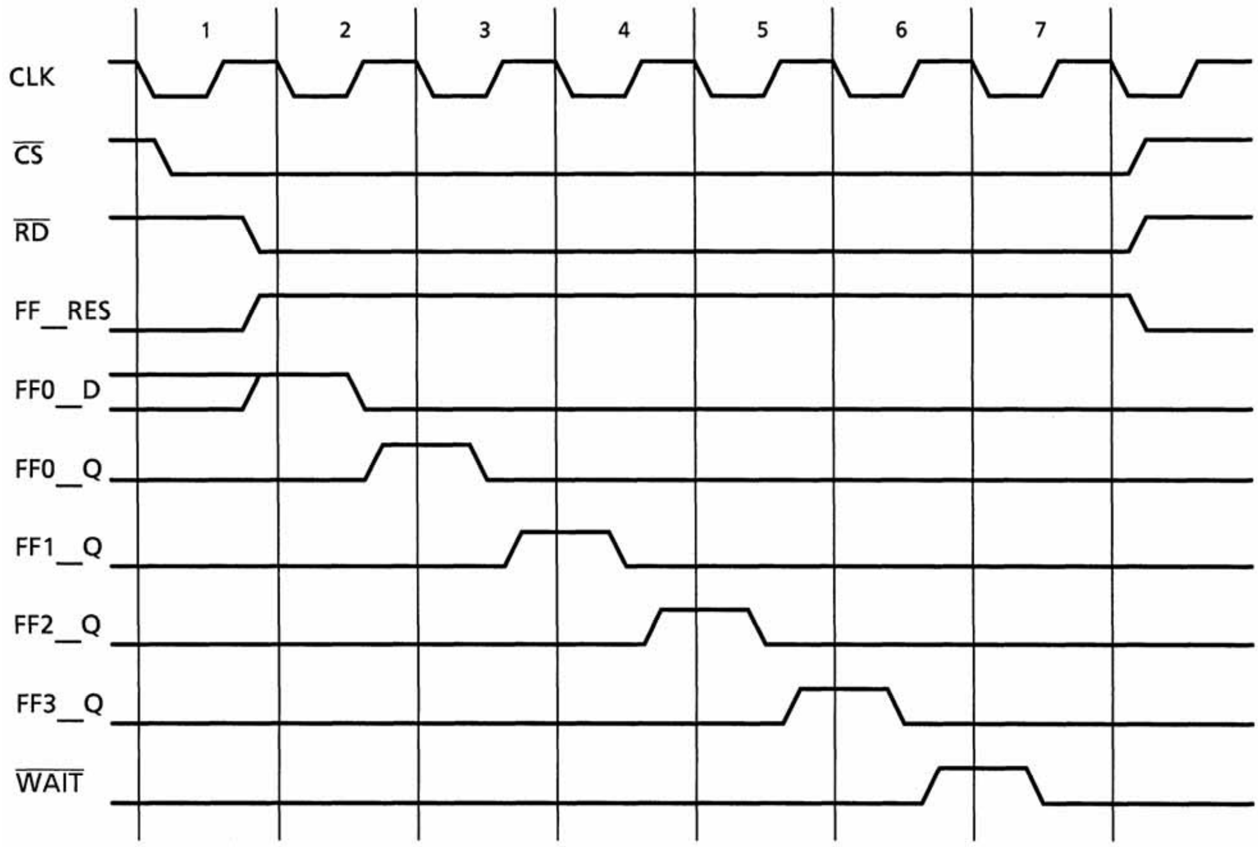


External read/write bus cycle (n waits at WAIT pin input mode)



Example of WAIT input cycle (5 waits)





3.6.4 ROM Control (Page mode)

This section describes ROM page mode accessing and how to set the registers. ROM page mode is set by the page ROM control register.

(1) Operation and how to set the registers

TMP94C251A supports ROM access of the page mode. The ROM access of the page mode is specified only in the block address area 2.

ROM page mode is set by the page ROM control register (PMEMCR). Setting OPGE bit of the PMEMCR register to "1" sets the memory access of the block address area 2 to ROM page mode access.

The number of read cycles is set by the OPWR1, OPWR0 bit of the PMEMCR register.

OPWR1/OPWR0 Bit (PMEMCR register)

OPWR1	OPWR0	Number of cycle in a page
0	0	1 state (n-1-1-1 mode) (n ≧ 2)
0	1	2 states (n-2-2-2 mode) (n ≧ 3)
1	0	3 states (n-3-3-3 mode) (n ≧ 4)
1	1	(Reserved)

Note: Set the number of waits "n" to the control register (BnCSL) in each block address area.

The page size (The number of bytes) of ROM in the CPU side is set to the PR1 and 0 bits of the PMEMCR register. When data is read out until a border of the set page, the controller completes the page reading operation. The start data of the next page is read in the normal cycle. The following data is set to page read again.

PR1/PR0 Bit (PMEMCR register)

RR1	RR0	ROM Page Size
0	0	64 Bytes
0	1	32 Bytes
1	0	16 Bytes (Default)
1	1	8 Bytes

(2) Signal timing pulse

For the signal timing pulse, see ROM read cycle in section 4.3.2.

3.6.5 List of Registers

The memory control registers and the settings are described as follows. For the addresses of the registers, see "List of Special Function Registers" in section 5.

(1) Control register

The control register is a pair of BnCSL and BnCSH. (n is a number of the block address area.) BnCSL has the same configuration regardless of the block address areas. In BnCSH, only B2CSH which is corresponded to the block address area 2 has a different configuration from the others.

BnCSL (n = 0 to 5)

	7	6	5	4	3	2	1	0
bit Symbol	–	BnWW2	BnWW1	BnWW0	–	BnWR	BnWR1	BnWR0
Read/Write		W				W		
After reset	–	0	1	0	–	0	1	0

Note)Read-modify-write is prohibited.

BnWW[2:0] Specifies the number of write waits.

001 = 2 states (0 waits) access 010 = 3 states (1 wait) access
101 = 4 states (2 waits) access 110 = 5 states (3 waits) access
011 = WAIT pin input mode Others = (Reserved)

BnWR[2:0] Specifies the number of read waits.

001 = 2 states (0 waits) access 010 = 3 states (1 wait) access
101 = 4 states (2 waits) access 110 = 5 states (3 waits) access
011 = WAIT pin input mode Others = (Reserved)

Note: When DRAM is specified as a connecting memory, setting should be 3 states (1 wait) or more. In the case of DRAM access, the WAIT pin input mode cannot be used.

B2CSH

	7	6	5	4	3	2	1	0
bit Symbol	B2E	B2M	–	B2REC	B2OM1	B2OM0	B2BUS1	B2BUS0
Read/Write	W	W		W	W		W	
After reset	1	0	0	0	0	0	1	0/1

Note)Read-modify-write is prohibited.

B2E Enable bit

0 = No chip select signal output
1 = Chip select signal output (Default)

Note: After reset release, only the enable bit B2E of B2CS register is valid ("1").

B2M Block address area specification

0 = Sets the block address area of CS2 to addresses 000000H to FFFFFFH. (Default)

1 = Sets the block address area of CS2 to programmable.

Note: After reset release, the block address area 2 is set to addresses 000000H to FFFFFFH.

B2REC Sets the dummy cycle for data output recovery time.

0 = Not insert a dummy cycle (Default)

1 = Insert a dummy cycle

B2OM [1:0]

00 = SRAM or ROM (Default)

Others = (Reserved)

B2BUS [1:0] Sets the data bus width

00 = 8 bits (Default)

01 = 16 bits

10 = (Reserved)

11 = (Reserved)

Note: The value of B2BUS bit is set according to the state of AM[1:0] pin after reset release.

BnCSH (n = 0,1,3,4,5)

	7	6	5	4	3	2	1	0
bit Symbol	BnE	–	–	BnREC	BnOM1	BnOM0	BnBUS1	BnBUS0
Read/Write	W			W	W		W	
After reset	0	–	0	0	0	0	0	0

Note)Read-modify-write is prohibited.

BnE Enable bit

0 = No chip select signal output (Default)

1 = Chip select signal

BnREC Sets the dummy cycle for data output recovery time.

0 = Not insert a dummy cycle (Default)

1 = Insert a dummy cycle

BnOM [1:0] Sets the connecting device.

00 = SRAM or ROM (Default)

01 = (Reserved)

10 = DRAM

11 = (Reserved)

Note: DRAM is set only by B1CS and B3CS.

BnBUS [1:0] Sets data bus width.

00 = 8 bits (Default)

01 = 16 bits

10 = (Reserved)

11 = (Reserved)

(2) Block address register

A start address and an address area of the block address area are specified by the memory start address register (MSARn) and the memory address mask register (MAMRn). The memory start address register sets all start addresses similarly regardless of the block address areas. The bit to be set by the memory address mask register is depended on the block address area.

MSARn (n = 0 to 5)

	7	6	5	4	3	2	1	0
bit Symbol	MnS23	MnS22	MnS21	MnS20	MnS19	MnS18	MnS17	MnS16
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

MnS [23:16] Sets a start address.
Sets the start address of the block address areas. The bit are corresponding to the addresses A23 to A16.

MAMR0

	7	6	5	4	3	2	1	0
bit Symbol	M0V20	M0V19	M0V18	M0V17	M0V16	M0V15	M0V14-9	M0V8
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

M0V [20:8]
Enables or masks comparison of the addresses. M0V20 to M0V8 are corresponding to addresses A20 to A8. The bit of M0V14 to M0V9 are corresponding to addresses A14 to A9 by 1 bit. If “0” is set, the comparison between the value of the address bus and the start address is enabled. If “1” is set, the comparison is masked.

MAMR1

	7	6	5	4	3	2	1	0
bit Symbol	M1V21	M1V20	M1V19	M1V18	M1V17	M1V16	M1V15-9	M1V8
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

M1V [21:8]
Enables or masks comparison of the addresses. M1V20 to M1V8 are corresponding to addresses A21 to A8. The bit of M1V15 to M1V9 are corresponding to addresses A15 to A9 by 1 bit. If “0” is set, the comparison between the value of the address bus and the start address is enabled. If “1” is set, the comparison is masked.

MAMRn (n = 2 to 5)

	7	6	5	4	3	2	1	0
bit Symbol	MnV22	MnV21	MnV20	MnV19	MnV18	MnV17	MnV16	MnV15
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

MnV [22:15]

Enables or masks comparison of the addresses. MnV22 to MnV15 are corresponding to addresses A22 to A15. If "0" is set, the comparison between the value of the address bus and the start address is enabled. If "1" is set, the comparison is masked.

(3) Page ROM control register (PMEMCR)

The page ROM control register sets page ROM accessing. ROM page accessing is executed only in the block address area 2.

	7	6	5	4	3	2	1	0
bit Symbol	–	–	–	OPGE	OPWR1	OPWR0	PR1	PR0
Read/Write				R/W	R/W		R/W	
After reset	–	–	–	0	0	0	1	0

OPGE enable bit

0 = No ROM page mode accessing (Default)

1 = ROM page mode accessing

OPWR [1:0] Specifies the number of waits.

00 = 1 state (n-1-1-1 mode) ($n \geq 2$) (Default)

01 = 2 states (n-2-2-2 mode) ($n \geq 3$)

10 = 3 states (n-3-3-3 mode) ($n \geq 4$)

01 = (Reserved)

Note: Set the number of waits "n" to the control register (BnCSL) in each block address area.

PR [1:0] ROM page size

00 = 64 bytes

01 = 32 bytes

10 = 16 bytes (Default)

11 = 8 bytes

Table 3.6.1 Control Register (1/2)

Symbol	Address	7	6	5	4	3	2	1	0
B0CSL	140H	–	B0WW2	B0WW1	B0WW0	–	B0WR2	B0WR1	B0WR0
			W				W		
		–	0	1	0	–	0	1	0
B0CSH	141H	B0E	–	–	B0REC	B0OM1	B0OM0	B0BUS1	B0BUS0
		W			W	W			
		0	–	0	0	0	0	0	0
MAMR0	142H	M0V20	M0V19	M0V18	M0V17	M0V16	M0V15	M0V14-V9	M0V8
		R/W							
		1	1	1	1	1	1	1	1
MSAR0	143H	M0S23	M0S22	M0S21	M0S20	M0S19	M0S18	M0S17	M0S16
		R/W							
		1	1	1	1	1	1	1	1
B1CSL	144H	–	B1WW2	B1WW1	B1WW0	–	B1WR2	B1WR1	B1WR0
			W				W		
		–	0	1	0	–	0	1	0
B1CSH	145H	B1E	–	–	B1REC	B1OM1	B1OM0	B1BUS1	B1BUS0
		W			W	W			
		0	–	0	0	0	0	0	0
MAMR1	146H	M1V21	M1V20	M1V19	M1V18	M1V17	M1V16	M1V15-V9	M1V8
		R/W							
		1	1	1	1	1	1	1	1
MSAR1	147H	M1S23	M1S22	M1S21	M1S20	M1S19	M1S18	M1S17	M1S16
		R/W							
		1	1	1	1	1	1	1	1
B2CSL	148H	–	B2WW2	B2WW1	B2WW0	–	B2WR2	B2WR1	B2WR0
			W				W		
		–	0	1	0	–	0	1	0
B2CSH	149H	B2E	B2M	–	B2REC	B2OM1	B2OM0	B2BUS1	B2BUS0
		W			W	W			
		1	0	0	0	0	0	0	0
MAMR2	14AH	M2V22	M2V21	M2V20	M2V19	M2V18	M2V17	M2V16	M2V15
		R/W							
		1	1	1	1	1	1	1	1
MSAR2	14BH	M2S23	M2S22	M2S21	M2S20	M2S19	M2S18	M2S17	M2S16
		R/W							
		1	1	1	1	1	1	1	1
B3CSL	14CH	–	B3WW2	B3WW1	B3WW0	–	B3WR2	B3WR1	B0WR0
			W				W		
		–	0	1	0	–	0	1	0
B3CSH	14DH	B3E	–	–	B3REC	B3OM1	B3OM0	B3BUS1	B3BUS0
		W			W	W			
		0	–	0	0	0	0	0	0
MAMR3	14EH	M3V22	M3V21	M3V20	M3V19	M3V18	M3V17	M3V16	M3V15
		R/W							
		1	1	1	1	1	1	1	1
MSAR3	14FH	M3S23	M3S22	M3S21	M3S20	M3S19	M3S18	M3S17	M3S16
		R/W							
		1	1	1	1	1	1	1	1
B4CSL	150H	–	B4WW2	B4WW1	B4WW0	–	B4WR2	B4WR1	B4WR0
			W				W		
		–	0	1	0	–	0	1	0
B4CSH	151H	B4E	–	–	B4REC	B4OM1	B4OM0	B4BUS1	B4BUS0
		W			W	W			
		0	–	0	0	0	0	0	0

Note)Read-modify-write is prohibited for B0CSL,B0CSH, B1CSL,B1CSH, B2CSL,B2CSH, B3CSL,B3CSH, B4CSL and B4CSH.

Table 3.6.2 Control Register (2/2)

Symbol	Address	7	6	5	4	3	2	1	0
MAMR4	152H	M4V22	M4V21	M4V20	M4V19	M4V18	M4V17	M4V16	M4V15
		R/W							
		1	1	1	1	1	1	1	1
MSAR4	153H	M4S23	M4S22	M4S21	M4S20	M4S19	M4S18	M4S17	M4S16
		R/W							
		1	1	1	1	1	1	1	1
B5CSL	154H	–	B5WW2	B5WW1	B5WW0	–	B5WR2	B5WR1	B5WR0
			W				W		
		–	0	1	0	–	0	1	0
B5CSH	155H	B5E	–	–	B5REC	B5OM1	B5OM0	B5BUS1	B5BUS0
		W			W	W			
		0	–	0	0	0	0	0	0
MAMR5	156H	M5V22	M5V21	M5V20	M5V19	M5V18	M5V17	M5V16	M5V15
		R/W							
		1	1	1	1	1	1	1	1
MSAR5	157H	M5S23	M5S22	M5S21	M5S20	M5S19	M5S18	M5S17	M5S16
		R/W							
		1	1	1	1	1	1	1	1
PMEMCR	166H	–	–	–	OPGE	OPWR1	OPWR0	PR1	PR0
					R/W				
		–	–	–	0	0	0	1	0

Note)Read-modify-write is prohibited for B5CSL and B5CSH.

3.6.6 Cautions

If the parasitic capacitance of the read signal (Output enable signal) is greater than that of the Chip Select signal, it is possible that an unintended read cycle occurs due to a delay in the read signal. Such an unintended read cycle may cause a trouble as in the case of (a) in Figure 3.6.2.

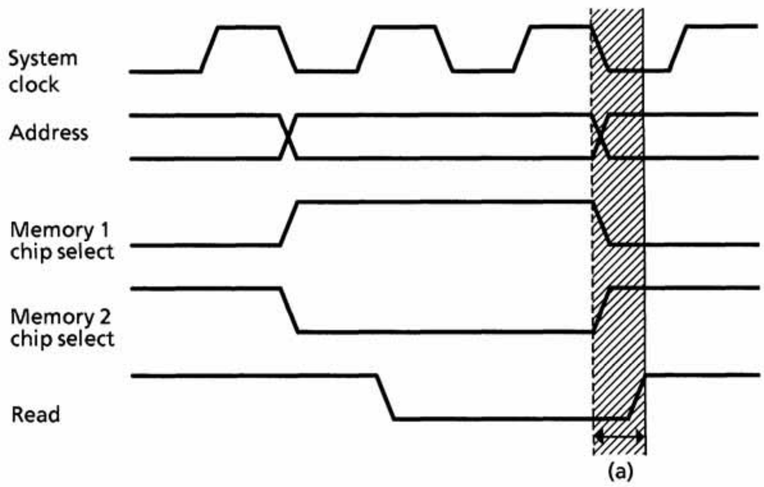


Figure 3.6.2 Read Signal Delay Read cycle

Example: When using an externally connected flash E²PROM which uses JEDEC standard commands, note that the toggle bit may not be read out correctly. If the Read signal in the cycle immediately preceding the access to the flash E²PROM does not go High in time, as shown in Figure 3.6.3, an unintended Read cycle like the one shown in (b) may occur.

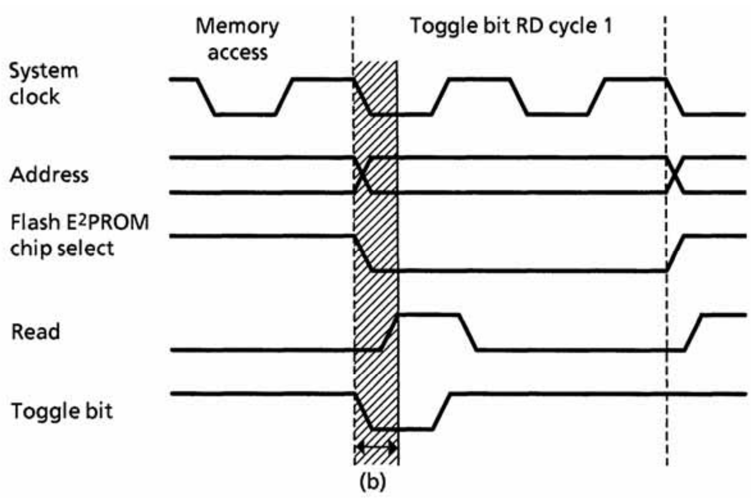


Figure 3.6.3 Flash E²PROM Toggle Bit Read cycle

When the toggle bit reverses with this unexpected read cycle, TMP94C251A always reads same value of the toggle bit, and cannot read the toggle bit correctly. To avoid this phenomena, the data polling control is recommended.

3.7 DRAM Controller

TMP94C251A has a two-channel DRAM controller. In addition, it controls DRAM access, address multiplexed, refresh, etc. as followings.

- Mapping area
  Block address area 1 … 256 to 4 Mbytes
  Block address area 3 … 32K to 8 Mbytes
- Memory access mode
  2CAS (16-bit bus), 1CAS (8-bit bus)
  Supports the page mode.
  Note) The 32-bit bus mode is supported only in the channel 1.
- Memory access address length
  Selects of 8 to 11 bits.
- Refresh mode
  CAS-before-RAS refresh mode
- Refresh interval
  Programmable (78 to 384 states)
- Refresh cycle width
  Programmable (2 to 9 states)
- Self-refresh
  Sets the self-refresh mode.
- Arbitration between refresh and access
  Refresh is prior to access. Wait is automatically inserted to the access cycle.
- Operation during bus release
  While the bus is released, there is a mode to support only DRAM refresh operation.

The data bus width and the number of waits to access DRAM are set according to the set value to the control register (B1CSH, B3CSH) in the block address area 1 and 3. This wait setting should be 3 states (1 wait) or more. In the case of DRAM access, the WAIT pin input mode cannot be used. The DRAM control register (DRAM0CRL/H, DRAM1CRL/H) and the DRAM refresh control register (DRAM0REF, DRAM1REF) set the other values.

DRAM accessing and refresh are explained with the setting method of the registers.

(1) DRAM access pin

The DRAM accessing is performed by the following pins. The functions of the pins are depended on the connected data bus width. The data bus width is set to the control register (B1CSH, B3CSH) in the block address area 1 and 3.

Pin Name	Bus Size	
	8 bit	16 bit
$\overline{\text{CS1}}$/$\overline{\text{RAS0}}$ (P81)	$\overline{\text{RAS0}}$	$\overline{\text{RAS0}}$
$\overline{\text{WE0}}$ (PA4)	$\overline{\text{WE0}}$	$\overline{\text{WE0}}$
$\overline{\text{OE0}}$ (PA2)	$\overline{\text{OE0}}$	$\overline{\text{OE0}}$
$\overline{\text{UCAS0}}$ (PA1)	–	$\overline{\text{UCAS0}}$
$\overline{\text{CAS0}}$/$\overline{\text{LCAS0}}$ (PA0)	$\overline{\text{CAS0}}$	$\overline{\text{LCAS0}}$
$\overline{\text{CS3}}$/$\overline{\text{RAS1}}$ (P83)	$\overline{\text{RAS1}}$	$\overline{\text{RAS1}}$
$\overline{\text{WE1}}$ (PB4)	$\overline{\text{WE1}}$	$\overline{\text{WE1}}$
$\overline{\text{OE1}}$ (PA3)	$\overline{\text{OE1}}$	$\overline{\text{OE1}}$
$\overline{\text{UCAS1}}$ (PB1)	–	$\overline{\text{UCAS1}}$
$\overline{\text{CAS1}}$/$\overline{\text{LCAS1}}$ (PB0)	$\overline{\text{CAS1}}$	$\overline{\text{LCAS1}}$

(2) DRAM access control

The DRAM control register (DRAM0CRL/H, DRAM1CRL/H) sets the DRAM access mode. The following explains the operations of the modes and the setting of the register.

(i) Address multiplexing

In TMP94C251A, the internal address multiplexer outputs the row/column address. The multiplexed address lines depend on the bus size: 8-bit or 11-bit.

- Setting method

  The MUXWn1 and MUXWn0 bits of the DRAM control register specify the multiplexed address width. The value set as follows is valid by setting MUXEn bit to “1”.

MUXWn

MUXWn1	MUXWn0	Multiplexed address length
0	0	8 bit (default)
0	1	9 bit
1	0	10 bit
1	1	11 bit

The multiplexed access bus size is depended on the data bus width after the multiplexed address width is set.

Column Address (A0 to A12 Pins)	Row Address							
	8 bit multiplexed address length		9 bit multiplexed address length		10 bit multiplexed address length		11 bit multiplexed address length	
	8 bit	16 bit	8 bit	16 bit	8 bit	16 bit	8 bit	16 bit
A0	A8	A0	A9	A0	A10	A0	A11	A0
A1	A9	A9	A10	A10	A11	A11	A12	A12
A2	A10	A10	A11	A11	A12	A12	A13	A13
A3	A11	A11	A12	A12	A13	A13	A14	A14
A4	A12	A12	A13	A13	A14	A14	A15	A15
A5	A13	A13	A14	A14	A15	A15	A16	A16
A6	A14	A14	A15	A15	A16	A16	A17	A17
A7	A15	A15	A16	A16	A17	A17	A18	A18
A8	A8	A16	A17	A17	A18	A18	A19	A19
A9	A9	A9	A9	A18	A19	A19	A20	A20
A10	A10	A10	A10	A10	A10	A20	A21	A21
A11	A11	A11	A11	A11	A11	A11	A11	A22
A12	A12	A12	A12	A12	A12	A12	A12	A12

In addition, the following multiplexed method is provided to connect 16 M bit DRAM and 64 M bit DRAM.

Column Address (A0 to A12 Pins)	Row Address	
	16 M	64 M
	16 bit	16 bit
A0	A0	A0
A1	A13	A14
A2	A14	A15
A3	A15	A16
A4	A16	A17
A5	A17	A18
A6	A18	A19
A7	A19	A20
A8	A20	A21
A9	A9	A22
A10	A10	A10
A11	A11	A11
A12	A12	A12
A13	–	A13

The following procedure can be used to set the multiplexed method.
Select a multiplexed method(16M/64M) to set "DRAMSELn" bit of DRAM control register.
"DRAMSELn" setting can be used by setting "ChnEN" bit of DRMEXT register to "1".

DRAMSEL n Bit

DRAMSEL n	DRAM TYPE
0	16 Mbit
1	64 Mbit

Note: This mode does not support development tools.

(ii) Page mode access

The DRAM page mode is accessed by setting the PGEn bit of the DRAM control register to “1”.

In the usual DRAM accessing, the number of waits is set to the control register (BnCSL) in the block address areas.

However in the page mode accessing, it is set to the DRAM control register.

- Setting method

  The number of waits in writing is set to the PnWW1, PnWW0 bits. The number of waits in reading is set to the PnWR1, PnWR0 bits. The setting method is the same as follows.

PnWW/PnWR Bit

PnWW1 PnWR1	PnWW0 PnWR0	Function
0	0	(reserved)
0	1	1 wait (n-2-2-2 mode) (n ≧ 3)
1	0	2 wait (n-3-3-3 mode) (n ≧ 4)
1	1	(reserved)

Note: Set the number of waits “n” to the control register (BnCSL) in each block address area.

(iii) DRAM access signal timing

For details of the signal timing pulse, see DRAM bus cycle in section 4.3.3.

(3) DRAM refresh controller

TMP94C251A supports three refresh controls as followings,

- CAS-before-RAS interval refresh
- CAS-before-RAS self-refresh
- Dummy refresh

The DRAM refresh control register (DRAM0REF, DRAM1REF) and the SRFC bit of the DRAM control register control the DRAM refresh operation. The followings explain the setting method and the operations.

(i) CAS-before-RAS interval refresh

In the CAS-before-RAS interval refresh mode, the RAS and CAS signals which are necessary for DRAM refresh are created according to the refresh interval and the refresh cycle width.

- Execution procedure

  Setting the RCn bit of the DRAM refresh control register (DRAM0REF, DRAM1REF) to "1" inserts the refresh cycle. The refresh cycle width is set to RWn2 to RWn0 bit, and the refresh cycle insertion interval is set to RSn2 to RSn0 bit. When using DRAM, set to at least three cycles.

  The RWn bit is set as follows.

RWn

RW02 RW12	RW01 RW11	RW00 RW10	Refresh cycle width
0	0	0	2 cycle (default)
0	0	1	3 cycle
0	1	0	4 cycle
0	1	1	5 cycle
1	0	0	6 cycle
1	0	1	7 cycle
1	1	0	8 cycle
1	1	1	9 cycle

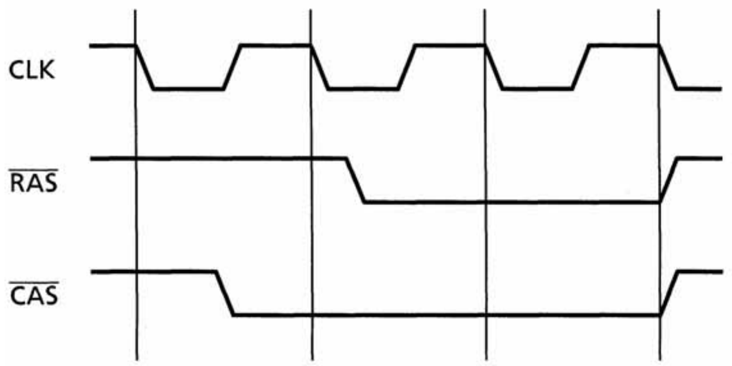
The refresh insertion interval is set in accordance with the setting of the RSn bit. The refresh cycle insertion interval is set in accordance with the frequency of the system clock as follows.

RSn

RS02 RS12	RS01 RS11	RS00 RS10	Insertion Interval (cycle)	Clock Frequency (MHz)		
				16.00	19.66	20.00
0	0	0	78	4.88	3.97	3.90
0	0	1	154	9.63	7.83	7.70
0	1	0	188	11.75	9.56	9.40
0	1	1	226	14.13	11.50	11.30
1	0	0	246	15.38	12.51	12.30
1	0	1	302	18.88	15.36	15.10
1	1	0	308	19.25	15.67	15.40
1	1	1	384	24.00	19.53	19.20

(Unit: μs)

- Refresh cycle timing



(ii) CAS-before-RAS self-refresh

The CAS-before-RAS self-refresh (Hereinafter refered to as self-refresh mode) used when the clock supplied is stopped by a HALT instruction while refreshing using the CAS-before-RAS interval refresh mode (Hereinafter refereed to as interval mode). (To stop clock supplied by a HALT instruction, the standby function is set in the IDLE mode or the STOP mode.)

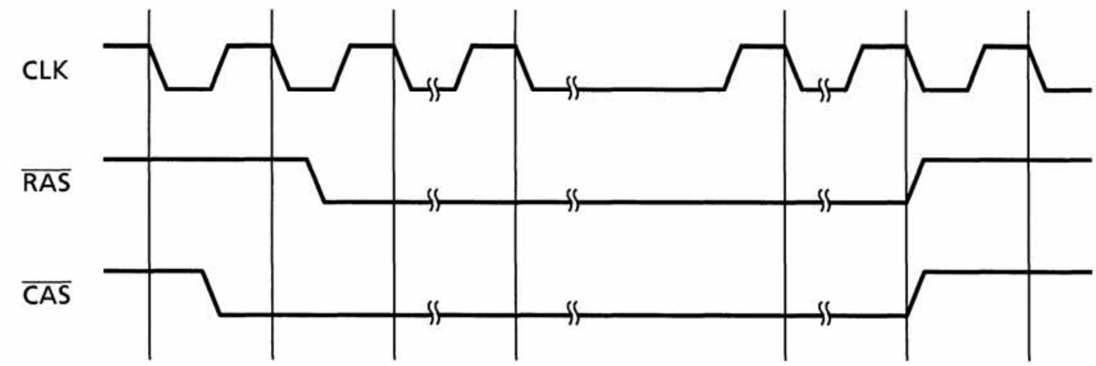
- Execution procedure

Setting the SFRCn bit of the DRAM control register to “0” during refresh in usual interval mode executes self-refresh.

In the self-refresh mode, the $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ signals maintain their low levels after turning to “Low”, as it is in the interval mode.

When HALT is released and the clock is supplied, “1” is set to the SFRCn bit by HALT release detector. The self-refresh mode is automatically released. But “1” isn’t set in “RUN mode”. After the self-refresh mode is released, $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ signals turn to “High”. The usual refresh is executed to return to the interval refresh mode.

- Self-refresh cycle timing



(iii) Dummy refresh

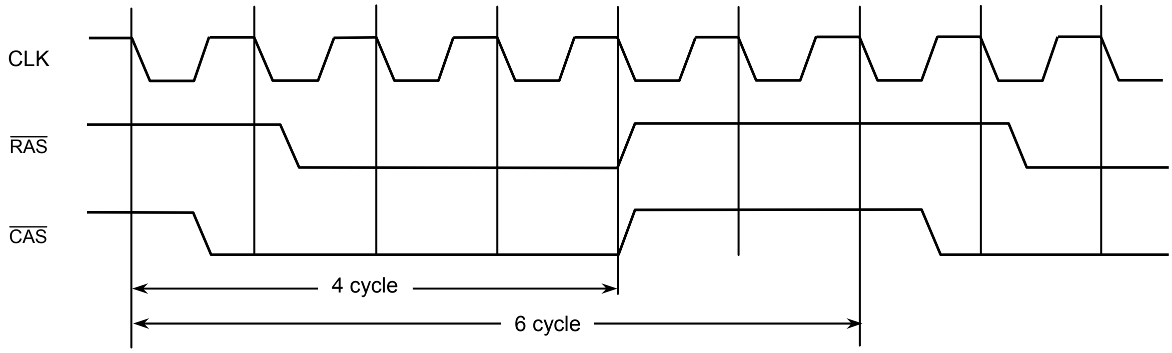
The dummy refresh executes CAS-before-RAS interval refresh successively.

The refresh cycle width is fixed to 4 states; the interval, to 6 states.

- Execution procedure

  Setting the DMn bit of the DRAM refresh control register (DRAM0REF, DRAM1REF) to "1" generates the dummy refresh. Dummy refresh is released by writing "0" to the DMn bit, by enabling DAM access control, or setting the RCn bit of the DRAM refresh control register to "1" and setting to the interval refresh mode. When dummy refresh mode is released by enabling DRAM access control or by setting the RCn bit of the DRAM refresh control register, the DMn bit is not cleared to zero.

- Cycle timing



(4) Priorities

As the DRAM refresh cycle is asynchronous to the CPU operating cycle, the refresh cycle may overlap with DRAM read and write cycles. If an overlap occurs, the DRAM controller gives priority to the cycle that started first. If the refresh cycle and DRAM access request are generated at the same time, the refresh cycle is given priority. In this case, the DRAM controller automatically inserts wait states in the memory access cycle until the refresh cycle completes.

(5) Refresh in the bus release mode

TMP94C251A has a bus release function. DRAM accessing pins ($\overline{RAS}$, $\overline{CAS}$) include two modes; either to release mode (Set to high impedance) in the same way as other pins, or to non-release mode (Output refresh signals). The BRMn bit of DRAM control register sets these modes.

BRMn

BRM0 BRM1	Bus release mode
0	Releases bus release-only pin (default)
1	Supports only refresh operation

- DRAM accessing pin release mode

  When "0" is input to the BRMn bit and the bus release request pin ($\overline{BUSRQ}$) is set to active, TMP94C251A acknowledges a bus release request. When the current bus cycle completes, TMP94C251A first set the DRAM accessing pins ($\overline{RAS}$, $\overline{CAS}$) to high, then turns the output buffer off to set the pins to high impedance. As the refresh cycle is asynchronous to the access cycle, when a refresh request is generated and has to wait, the refresh cycle is generated and the bus is released.

  Only one refresh request generated during the bus release is held. The refresh cycle is generated immediately upon return of the bus mastership to TMP94C251A at bus release completion.

- DRAM accessing pin non-release mode

  When "1" is input to the BRMn bit, DRAM accessing pins do not release the bus when a bus release request occurs.

  The pins continue to operate but support refresh cycles only. In DRAM accessing pin non-release mode, the bus release timing is not affected by refresh requests.

(6) List of registers

The registers to control DRAM controller and the settings are described as follows. For the addresses of the registers, see List of Special Function Registers in section 5.

DRAM can be set to the connecting memory only in the block address area 1 and 3. DRAMEXT, DRAM0CRL, DRAM0CRH and DRAM0REF control DRAM (Channel 0) in the block address area 1. DRAMEXT, DRAM1CRL, DRAM1CRH and DRAM1REF control DRAM (Channel 1) in the block address area 3.

DRAM0CRL

	7	6	5	4	3	2	1	0
bit Symbol	SFRC0	–	BRM0	DRAMSEL0	MUXE0	MUXW01	MUXW00	MAC0
Read/Write	R/W		R/W	R/W	R/W	R/W		R/W
After reset	1	–	0	0	0	0	0	0

DRAM1CRL

	7	6	5	4	3	2	1	0
bit Symbol	SFRC1	–	BRM1	DRAMSEL1	MUXE1	MUXW11	MUXW10	MAC1
Read/Write	R/W		R/W	R/W	R/W	R/W		R/W
After reset	1	–	0	0	0	0	0	0

- SFRC0/1 self-refresh control

  0 = Self-refresh
  1 = No self-refresh

- BRM0/1 bus release mode control

  0 = Also releases DRAM pin in bus release mode
  1 = Does not release DRAM pin in bus release mode. Supports only refresh.

- DRAMSEL0/1 multiplex system

  0 = 16 Mbits
  1 = 64 Mbits

- MUXE0/1 address multiplex

  0 = Disable
  1 = Enable (Make this setting when using DRAM.)

- MUXW0/1[1:0] multiplex address length control

  00 = 8 bits
  01 = 9 bits
  10 = 10 bits
  11 = 11 bits

- MAC0/1 enable bit

  0 = No DRAM access control
  1 = DRAM access control

DRAM0CRH

	7	6	5	4	3	2	1	0
bit Symbol	P0WW1	P0WW0	P0WR1	P0WR0	PGE0	–	–	–
Read/Write	R/W		R/W		R/W			
After reset	1	0	1	0	0	–	–	–

DRAM1CRH

	7	6	5	4	3	2	1	0
bit Symbol	P1WW1	P1WW0	P1WR1	P1WR0	PGE1	–	–	–
Read/Write	R/W		R/W		R/W			
After reset	1	0	1	0	0	–	–	–

- P0/1WW[1:0] specifies the number of DRAM page mode write waits.

  00 = (Reserved)
  01 = 2 states (n-2-2-2 mode) (n ≥ 3)
  10 = 3 states (n-3-3-3 mode) (n ≥ 4)
  11 = (Reserved)

  Note: Set the number of waits “n” in the corresponding control register (BnCSL).

- P0/1WR[1:0] specifies the number of DRAM page mode read waits.

  00 = (Reserved)
  01 = 2 states (n-2-2-2 mode) (n ≥ 3)
  10 = 3 states (n-3-3-3 mode) (n ≥ 4)
  11 = (Reserved)

  Note: Set the number of waits “n” to the control register (BnCSL) in each block address area.

- PGE0/1 page mode access enable

  0 = No page mode access

  1 = Page mode access

  Note: Please set the same value to PGE0 and PGE1.
  Setting the different value may occur malfunction.

DRAM0REF

	7	6	5	4	3	2	1	0
bit Symbol	DM0	RS02	RS01	RS00	RW02	RW01	RW00	RC0
Read/Write	R/W	R/W			R/W			R/W
After reset	0	0	0	0	0	0	0	0

DRAM1REF

	7	6	5	4	3	2	1	0
bit Symbol	DM1	RS12	RS11	RS10	RW12	RW11	RW10	RC1
Read/Write	R/W	R/W			R/W			R/W
After reset	0	0	0	0	0	0	0	0

- DM0/1 dummy refresh cycle control

  0 = No dummy refresh cycle
  1 = Dummy refresh cycle

- RS0/1[2:0] refresh cycle insertion interval

  000 = 78 cycles
  001 = 154 cycles
  010 = 188 cycles
  011 = 226 cycles
  100 = 246 cycles
  101 = 302 cycles
  110 = 308 cycles
  111 = 384 cycles

- RW0/1[2:0] refresh cycle width

  000 = 2 cycles
  001 = 3 cycles
  010 = 4 cycles
  011 = 5 cycles
  100 = 6 cycles
  101 = 7 cycles
  110 = 8 cycles
  111 = 9 cycles

- RC0/1 enable bit

  0 = No refresh cycle
  1 = Refresh cycle

DRAMEXT

	7	6	5	4	3	2	1	0
bit Symbol	CH0EN	—	—	—	CH1EN	—	—	—
Read/Write	R/W				R/W			
After reset	0	—	—	—	0	—	—	—

- CH0EN/CH1EN DRAMSELn bit enable

  0 = "DRAMSELn" setting is invalid.
  1 = "DRAMSELn" setting is valid.

(7) Register setting examples

The following shows an example of setting block address space 1 (CS1) with addresses 100000H to 1FFFFFH (1-Mbyte space), 8-bit data bus width, write 3 states, read 3 states, no dummy cycle for data bus recovery, and 8-bit address multiplex DRAM mode.

MSAR1 = 10H

MAMR1 = 3FH

B1CSL = 22H

B1CSH = 88H

DRAM0CRL = 8DH

The following shows an example of setting block address space 3 (CS3) with addresses 300000H to 3FFFFFH (1-Mbyte space), 16-bit data bus width, write/read 1 wait, page access, and 10-bit address multiplex DRAM mode.

MSAR3 = 30H

MAMR3 = 1FH

B3CSH = 89H

DRAM1CRL = 8DH

DRAM1CRH = 58H

Table 3.7.1 List of Registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
DRAMEXT	DRAM Select Enable	15Ch	CH0EN	–	–	–	CH1EN	–	–	–
			R/W							
			0	–	–	–	0	–	–	–
			DRAMSEL0 0: Disable 1: Enable				DRAMSEL1 0: Disable 1: Enable			
DRAM0CRL	DRAM 0 Control Register L	160h	SFRC0	–	BRM0	DRAMSEL0	MUXE0	MUXW01	MUXW00	MAC0
			1	–	0	0	0	0	0	0
			Self-refresh 0: Exec. 1: Rele.		Bus release mode control 0: Rele. 1: Not release	DRAM select 0: 16 Mbit 1: 64 Mbit	address multiplex 0: disable 1: Enable	Multiplexed length address 00: 8 bit 01: 9 bit 10: 10 bit 11: 11 bit		memory access control 0: Disable 1: Enable
DRAM0CRH	DRAM 0 Control Register H	161h	P0WW1	P0WW0	P0WR1	P0WR0	PGE0	–	–	–
			R/W							
			1	0	1	0	0	–	–	–
			00: (Reserved) 01: 1 wait (n-2-2-2 mode) 10: 2 wait (n-3-3-3 mode) 11: (Reserved)		00: (Reserved) 01: 1 wait (n-2-2-2 mode) 10: 2 wait (n-3-3-3 mode) 11: (Reserved)		DRAM page access 1:Enable			
DRAM1CRL	DRAM 1 Control Register L	162h	SFRC1	–	BRM1	DRAMSEL1	MUXE1	MUXW11	MUXW10	MAC1
			R/W							
			1	–	0	0	0	0	0	0
			Self-refresh 0: Exec. 1: Rele.		Bus release mode control 0: Rele. 1: Not release	DRAM select 0: 16 Mbit 1: 64 Mbit	address multiplex 0: disable 1: Enable	Multiplexed length address 00: 8 bit 01: 9 bit 10: 10 bit 11: 11 bit		memory access control 0: Disable 1: Enable
DRAM1CRH	DRAM 1 Control Register H	163h	P1WW1	P1WW0	P1WR1	P1WR0	PGE1	–	–	–
			R/W							
			1	0	1	0	0	–	–	–
			00: (Reserved) 01: 1 wait (n-2-2-2 mode) 10: 2 wait (n-3-3-3 mode) 11: (Reserved)		00: (Reserved) 01: 1 wait (n-2-2-2 mode) 10: 2 wait (n-3-3-3 mode) 11: (Reserved)		DRAM page access 1:Enable			
DRAM0REF	DRAM 0 Refresh Control	164h	DM0	RS02	RS01	RS00	RW02	RW01	RW00	RC0
			R/W							
			0	0	0	0	0	0	0	0
			Dummy cycle 0: Prohi bit 1: Execute	Refresh cycle insertion at 000: 78 100: 246 001: 154 101: 302 010: 188 110: 308 011: 226 111: 384			Refresh cycle width 000: 2 100: 6 001: 3 101: 7 010: 4 110: 8 011: 5 111: 9			Refresh cycle 0: Not insert 1: insert
DRAM1REF	DRAM 1 Refresh Control	165h	DM1	RS12	RS11	RS10	RW12	RW11	RW10	RC1
			R/W							
			0	0	0	0	0	0	0	0
			Dummy cycle 0: Prohi bit 1: Execute	Refresh cycle insertion at 000: 78 100: 246 001: 154 101: 302 010: 188 110: 308 011: 226 111: 384			Refresh cycle width 000: 2 100: 6 001: 3 101: 7 010: 4 110: 8 011: 5 111: 9			Refresh cycle 0: Not insert 1: insert

3.8 8-Bit Timers

TMP94C251A incorporates four 8-bit timers (Timers 0 to 3). Each timer can operate independently or be cascaded to form two 16-bit timers. The 8-bit timers have the following four operating modes.

- 8-bit interval timer mode (4 channels)
- 16-bit interval timer mode (2 channels)

The above two modes can be combined (for example, two 8-bit timers and one 16-bit timers)

- 8-bit programmable square wave (PPG: Variable cycle, variable duty) output mode (2 channels)
- 8-bit PWM (Pulse width modulation: Variable duty at fixed cycle) output mode (2 channels)

Figure 3.8.1 is a block diagram for 8-bit timers (Timers 0, 1).

Timers 2 and 3, have the same circuit configuration as timers 0 and 1.

Each interval 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) is provided for the timer pairs: timers 0 and 1, timers 2 and 3.

Of the input clock sources for interval timers, the ϕT1, ϕT4, ϕT16, and ϕT256 internal clocks are obtained from the 9-bit prescaler shown in Figure 3.8.2.

The 8-bit timer operating mode and the timer flip-flops are controlled by six control registers (T01MOD, T23MOD, TFFCR, T8RUN, T16RUN, and TRDC)

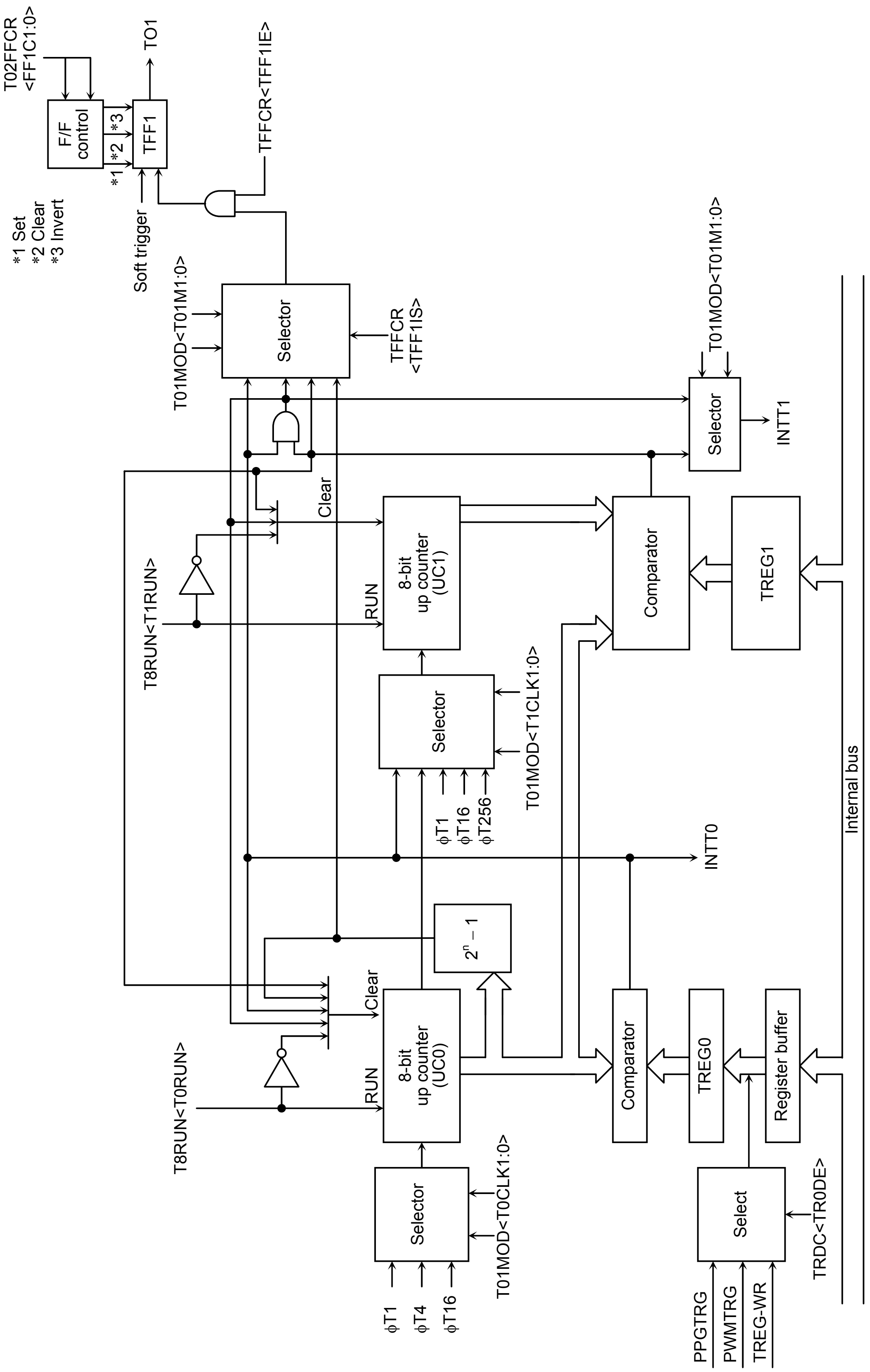


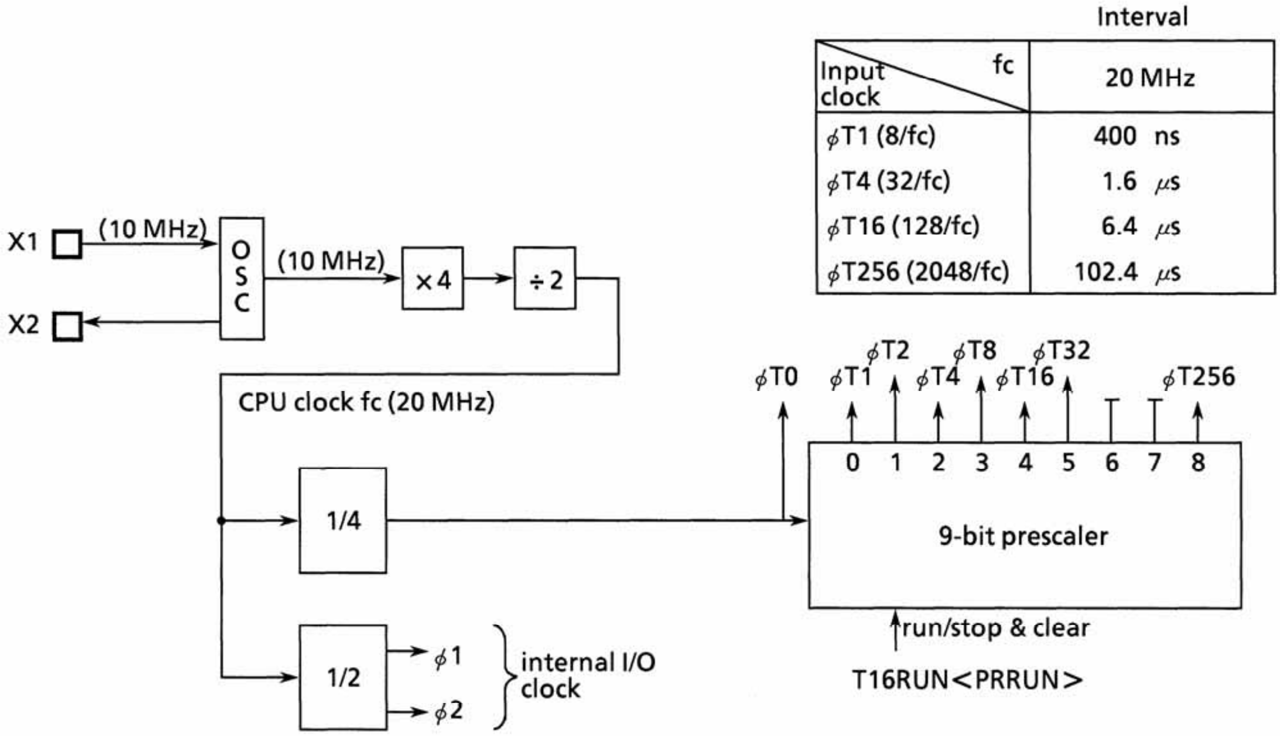
Figure 3.8.1 8-Bit Timer Block Diagram (Timers 0, 1)

[1] Prescaler

The input to the 9-bit prescaler is the CPU fundamental clock (fc) divided by four (fc/4). The prescaler generates an input clock for the 8-bit timers, the 16-bit timer/event counters, and baud rate generator, for example.

The 8-bit timers can use the following four clock signals: ϕT1, ϕT4, ϕT16, and ϕT256.

To set the prescaler to count or stop, use timer control register T16RUN<PRRUN>. Setting T16RUN<PRRUN> to "1" starts the count. Clearing <PRRUN> to "0" clears and stops the prescaler. Resetting clears <PRRUN> to "0", and clears and stops the prescaler.



Interval

Input clock \ fc	20 MHz
ϕT1 (8/fc)	400 ns
ϕT4 (32/fc)	1.6 μs
ϕT16 (128/fc)	6.4 μs
ϕT256 (2048/fc)	102.4 μs

Note: The number in the parenthesis indicates the frequency when TMP94C251A operates is the maximum frequency.

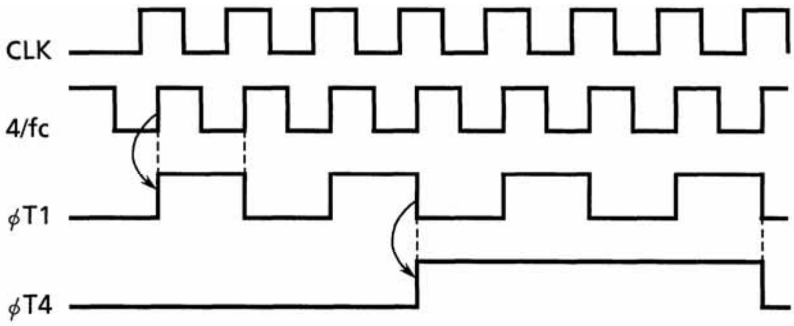


Figure 3.8.2 Prescaler

[2] Up counter

The up counter is an 8-bit binary counter that counts up using the input clock specified by timer 0 and 1 mode registers T01MOD, T23MOD.

The timer 0, 2 input clocks are selected from internal clocks ϕT1, ϕT4, and ϕT16 in accordance with the T01MOD, T23MOD settings.

The timer 1, 3 input clocks vary according to the operating mode. When the up-counter is set to 16-bit timer mode, timer 0, 2 overflow output is used as an input clock.

When the up-counter is set to other than 16-bit timer mode, two further settings are available: internal clocks ϕT1, ϕT16, or ϕT256 based on the T01MOD, T23MOD settings, and timer 0, 2 comparator output (Match detect).

Example: If T01MOD<T01M1:0> is set to "01", the timer 0 overflow output is used as the timer 1 input clock (16-bit timer mode).
T01MOD<T01M1:0> is "00" and <T1CLK1:0> is "01", ϕT1 is used as the timer 1 input clock (8-bit timer mode).

The T01MOD, T23MOD registers also set the operating mode. A reset sets the up-counter to 8-bit timer mode.

To control the count, stop, and clear functions of each up-counter interval timer, use timer control register T8RUN. A reset clears all up-counters and stops the timers.

[3] Timer registers

The timer registers are 8-bit registers for setting interval times. When the setting of timer registers TREG0 to TREG3 matches the up-counter value, the comparator match detect signal becomes active. If "00H" is set, the match detect signal is activated when the up-counter overflows.

Timer registers TREG0, TREG2 have a double-buffer configuration and are paired with a register buffer.

TREG0, TREG2 enable or disable the double-buffer using timer register double-buffer control register TRDC<TR0/2DE>. Setting <TR0/2DE> to "0" disables the double-buffer; setting <TR0/2DE> to "1" enables the double-buffer.

With the double-buffer 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 an interval comparison match in programmable pulse generation (PPG) mode.

A reset initializes <TR0/2DE> to "0", disabling the double-buffer. When using the double-buffer, first write data to the timer register and set <TR0/2DE> to "1", then write the following data to the register buffer.

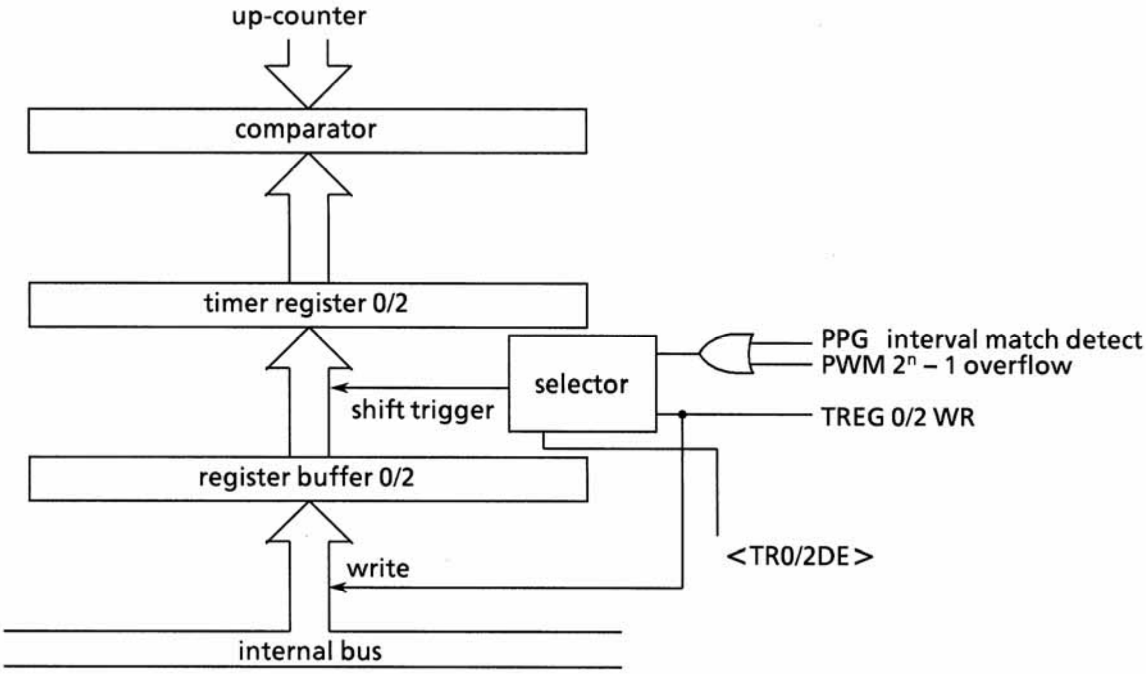


Figure 3.8.3 Timer Register 0/2/4/6 Configuration

Note: The timer register and register buffer are allocated to the same address in memory. When <TR0/2DE> is set to "0", the same value is written to both the register buffer and the timer register. When <TR0/2DE> is set to "1", the value is written to the register buffer only.

The timer registers are allocated in memory as follows.

The timer register TREG0, TREG1, TREG2 and TREG3 are write-only; cannot read data from them. As the initial values are undefined, when using an 8-bit timer, be sure to write values.

T01MOD (0084H)

	7	6	5	4	3	2	1	0
bit Symbol	T01M1	T01M0	PWM01	PWM00	T1CLK1	T1CLK0	T0CLK1	T0CLK0
Read/Write	R/W		R/W		R/W		R/W	
After reset	0	0	0	0	0	0	0	0
Function	00: 8 bit Timer 01: 16 bit Timer 10: 8 bit PPG 11: 8 bit PWM		00: – 01: $2^6 - 1$ PWM 10: $2^7 - 1$ interval 11: $2^8 - 1$		00: TO0TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: Reserved 01: ϕT1 10: ϕT4 11: ϕT16	

Timer 0 input clock

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

Timer 1 input clock

	<T01M1,0> ≠ "01"	<T01M1,0> = "01"
00	Timer 0 comparator output	Timer 0 overflow output (16-bit timer mode)
01	Internal clock ϕT1	
10	Internal clock ϕT16	
11	Internal clock ϕT256	

PWM0 interval selection
(Except for PWM mode, don't care)

00	——
01	$2^6 - 1$
10	$2^7 - 1$
11	$2^8 - 1$

Timer 0 and 1 operating mode selection

00	8-bit timer x 2 (timers 0,1)
01	16-bit timer
10	8-bit programmable square wave output
11	8-bit PWM output (timer 0) + 8-bit timer (timer 1)

Figure 3.8.4 Timer 0/1 Mode Register (T01MOD)

		7	6	5	4	3	2	1	0
T23MOD (0085H)	bit Symbol	T23M1	T23M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
	Read/Write	R/W		R/W		R/W		R/W	
	After reset	0	0	0	0	0	0	0	0
	Function	00: 8 bit Timer 01: 16 bit Timer 10: 8 bit PPG 11: 8 bit PWM		00: – 01: $2^6 - 1$ PWM 10: $2^7 - 1$ interval 11: $2^8 - 1$		00: TO2TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: Reserved 01: ϕT1 10: ϕT4 11: ϕT16	

Timer 2 input clock

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

Timer 3 input clock

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

PWM2 interval selection
(Except for PWM mode, don't care)

00	——
01	$2^6 - 1$
10	$2^7 - 1$
11	$2^8 - 1$

Timer 2 and 3 operating mode selection

00	8-bit timer x 2 (timers 2, 3)
01	16-bit timer
10	8-bit programmable square wave output
11	8-bit PWM output (timer 2) + 8-bit timer (timer 3)

Figure 3.8.5 Timer 2/3 Mode Register (T23MOD)

TFFCR (0082H)

	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	–		0	0	–		0	0
Function	00: Invert TFF3 01: Set TFF3 10: Clear TFF3 11: Don't care		1: TFF3 Invert Enable	0: Inverts by 8-bit timer 2. 1: Inverts by timer 3.	00: Invert TFF1 01: Set TFF1 10: Clear TFF1 11: Don't care		1: TFF1 Invert Enable	0: Inverts by 8-bit timer 0. 1: Inverts by timer 1.

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

0	Inverts by timer 0.
1	Inverts by timer 1.

Timer flip-flop TFF1 inversion

0	Disable
1	Enable

Timer flip-flop TFF1 control

00	Inverts TFF1 value (software inversion).
01	Sets TFF1 to "1".
10	Clears TFF1 to "0".
11	Don't care

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

0	Inverts by timer 2.
1	Inverts by timer 3.

Timer flip-flop TFF3 inversion

0	Disable
1	Enable

Timer flip-flop TFF3 control

00	Inverts TFF3 value (software inversion).
01	Sets TFF3 to "1".
10	Clear TFF3 to "0".
11	Don't care

Note)Read-modify-write is prohibited.

Figure 3.8.6 8-Bit Timer Flip-Flop Control Register (TFFCR)

8-bit Timers (0, 1, 2, 3)

T8RUN (0080H)		7	6	5	4	3	2	1	0
	bit Symbol	–	–	–	–	T3RUN	T2RUN	T1RUN	T0RUN
	Read/Write					R/W			
	After reset	–	–	–	–	0	0	0	0
	Function					Timer Run / Stop Control 0: Stop & Clear 1: Run (Count up)			

Timer 0 operation

0	Stop and clear
1	Count up

Timer 1 operation

0	Stop and clear
1	Count up

Timer 2 operation

0	Stop and clear
1	Count up

Timer 3 operation

0	Stop and clear
1	Count up

Figure 3.8.7 8-Bit Timer Operation Control Register (T8RUN)

TRDC (0081H)		7	6	5	4	3	2	1	0
	bit Symbol	–	–	–	–	–	–	TR2DE	TR0DE
	Read/Write							R/W	
	After reset	–	–	–	–	–	–	0	0
	Function							0: Double Buffer Disable 1: Double Buffer Enable	

Timer register 0 double-buffer control

0	Disable
1	Enable

Timer register 2 double-buffer control

0	Disable
1	Enable

Figure 3.8.8 Timer Register Double-Buffer Control Register (TRDC)

T16RUN (009EH)		7	6	5	4	3	2	1	0
	bit Symbol	PRRUN	–	–	–	TARUN	T8RUN	T6RUN	T4RUN
	Read/Write	R/W				R/W			
	After reset	0	–	–	–	0	0	0	0
	Function	Prescaler 0: Stop 1: Run				16 Bit Timer Run / Stop Control 0: Stop & Clear 1: Run (Count up)			

16-bit Timer (timer 4) operation

0	Stop and clear
1	Count Up

16-bit Timer (timer 6) operation

0	Stop and clear
1	Count Up

16-bit Timer (timer 8) operation

0	Stop and clear
1	Count Up

16-bit Timer (timer A) operation

0	Stop and clear
1	Count Up

Prescaler operation

0	Stop and clear
1	Count Up

Figure 3.8.9 16-Bit Timer Operation Control Register (T16RUN)

Symbol	Address	7	6	5	4	3	2	1	0
TREG0	88h	–							
		W							
		Undefined							
TREG1	89h	–							
		W							
		Undefined							
TREG2	8Ah	–							
		W							
		Undefined							
TREG3	8Bh	–							
		W							
		Undefined							

Note)Read-modify-write is prohibited.

Figure 3.8.10 Timer Register

[4] Comparator

The comparator compares the up-counter value with the timer register value. If the values match, the comparator clears the up counter to 0 and generates an interrupt (INTT0 to INTT3). If the timer flip-flop invert is enabled at this time, the comparator inverts the timer flip-flop value.

[5] Timer flip-flops (Timer F/F)

Each interval timer match detect signal (Comparator output) inverts the timer flip-flops and outputs the values to timer output pins TO1 (also used as PC0), TO3 (also used as PC1).

One timer flip-flop is provided for a timer pair: TFF1 for timer pair 0, 1; TFF3 for pair 2, 3. TFF1 is output to pin TO1, TFF3 to pin TO3.

The following explains the operation of the 8-bit timers.

(1) 8-bit timer mode

Four interval timers 0 to 3 can be used independently as 8-bit interval timers. As all the timers operate the same, the following describes timer 1 only.

[1] Generating a fixed-interval interrupt

When using timer 1 to generate a timer 1 interrupt (INTT1) for each fixed interval, first halt timer 1, then set the operating mode, input clock, and interval in T01MOD and TREG1. Next, enable INTT1, and start timer 1 counting.

Example: If a timer 1 interrupt is required every 40 μs at fc = 20 MHz, set the registers in the following order:

		MSB 7 6 5 4 3 2 1 0 LSB	
T8RUN	←	X X X X – – 0 –	Stops timer 1 and clears it to "0".
T01MOD	←	0 0 X X 0 1 – –	Sets 8-bit timer mode and sets the input clock to ϕT1 (0.4 μs at fc = 20 MHz).
TREG1	←	0 1 1 0 0 1 0 0	Sets 40μs ÷ ϕT1 = 100 (64H) in the timer register.
INTET01	←	X 1 0 1 – – – –	Sets INTT1 to level 5.
T16RUN	←	1 X X X – – – –	Starts Prescaler
T8RUN	←	X X X X – – 1 –	Starts timer 1 counting.

Note: X: Don't care –: No change

For input clock selection, see the following table.

Table 3.8.1 Selecting Interrupt Interval and the Input Clock Using 8-Bit Timer

Input Clock	Interrupt Interval (at fc = 20 MHz)	Resolution
ϕT1 (8/fc)	0.4 μs to 102.4 μs	0.4 μs
ϕT4 (32/fc)	1.6 μs to 409.6 μs	1.6 μs
ϕT16 (128/fc)	6.4 μs to 1.639 ms	6.4 μs
ϕT256 (2048/fc)	102.4 μs to 26.22 ms	102.4 μs

[2] Generating a square wave with a 50% duty cycle

Invert the timer flip-flop at fixed intervals and output the timer flip-flop values to the timer output pin (TO1).

Example: To output a square wave from pin TO1 with an interval of 2.4 μs at fc = 20 MHz, set the registers in the following order. Use either timer 0 or 1. The example shows the register settings for timer 1.

		MSB 7 6 5 4 3 2 1 0 LSB	
T8RUN	←	X X X X – – 0 –	Stops timer 1 and clears it to 0.
T01MOD	←	0 0 X X 0 1 – –	Sets 8-bit timer mode and sets the input clock to ϕT1.
TREG1	←	0 0 0 0 0 0 1 1	Sets 2.4 μs ÷ ϕT1 ÷ 2 = 3 in the timer register.
TFFCR	←	– – – – 1 0 1 1	Clears TFF1 to 0 and sets it to invert on a match detect signal from timer 1.
PCCR	←	X X X X X X – 0	Sets PC0 to TO1.
PCFC	←	X X X X – – – 1	
T16RUN	←	1 X X X – – – –	Starts PRESCALER
T8RUN	←	X X X X – – 1 –	Starts timer 1 counting.

Note: X: Don't care –: No change

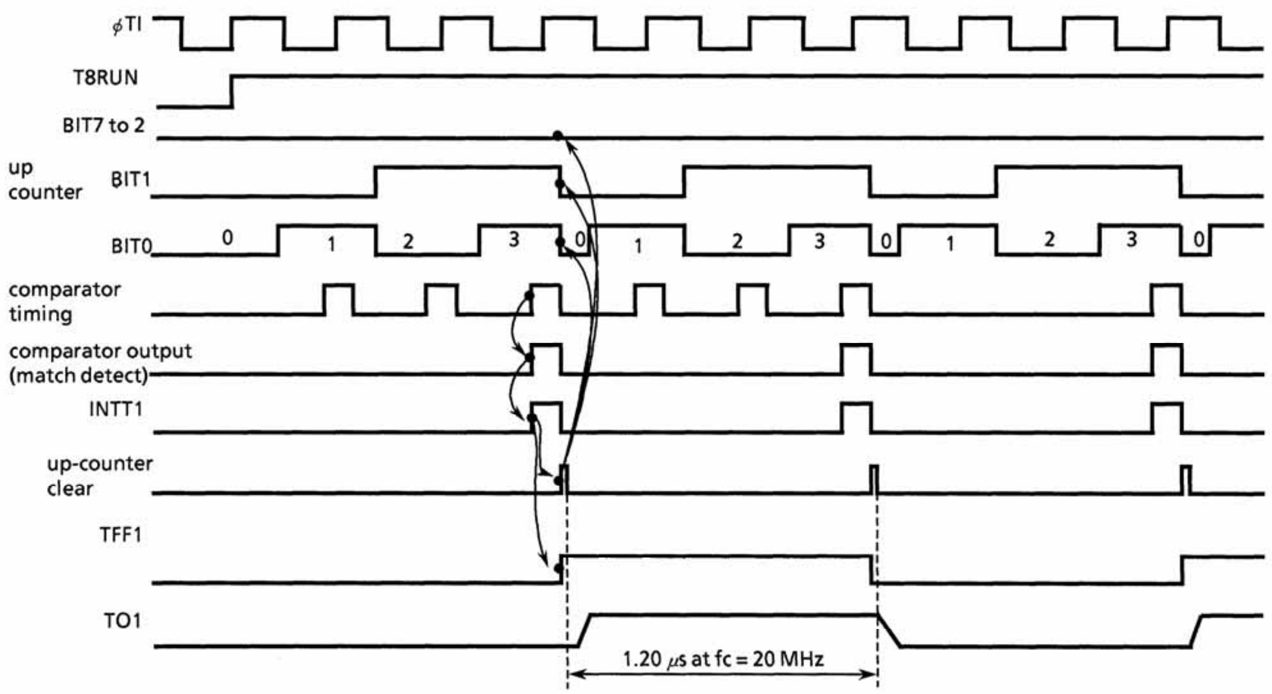


Figure 3.8.11 Square Wave (50% Duty) Output Timing Chart

[3] Setting timer 1 to count up at timer 0 match output

Set 8-bit timer mode and set the timer 1 input clock to timer 0 comparator output.

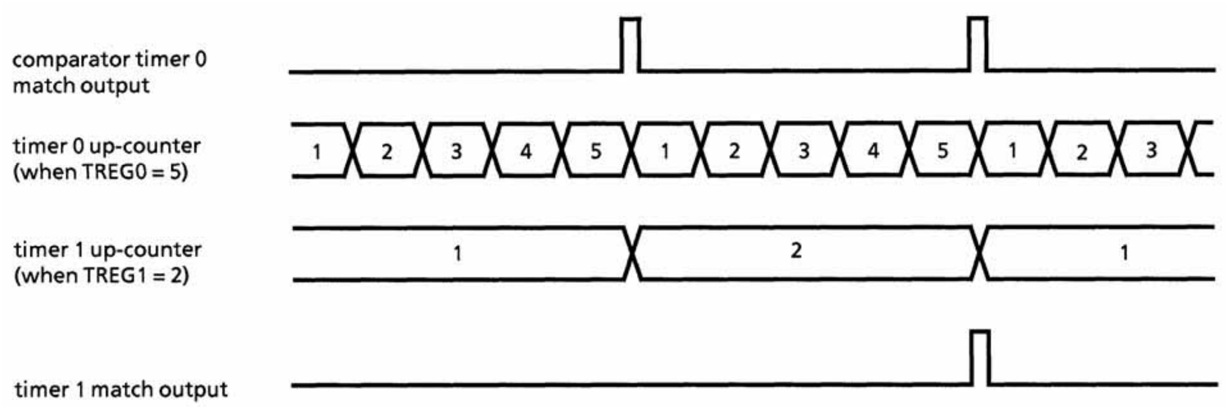


Figure 3.8.12 Timer 1 Count-up due to Matching Output of Timer 0

[4] Output invert by software

The timer flip-flop (Timer F/F) value can be inverted independently of timer operation.

For example, writing “00” to TFFCR<FF1C1:0> inverts the TFF1 value; writing “00” to TFFCR<FF3C1:0> inverts the TFF3 value.

[5] Timer flip-flop (Timer F/F) initialization

The timer flip-flop value can be initialized to “0” or “1” independently of timer operation.

For example, to set TFF1 to 0, write “10” to TFFCR<FF1C1:0>. To set TFF1 to 1, write “01” to TFFCR<FF3C1:0>.

Note: The timer flip-flop or timer register value cannot be read.

(2) 16-bit timer mode

Timers 0 and 1, 2 and 3 can be paired to configure 16-bit interval timers.

As timers 0 and 1, 2 and 3 operate the same, the following describes timers 0 and 1 only.

To cascade-connect timers 0 and 1 and configure a 16-bit interval timer, set mode register T01MOD<T01M1:0> to “01”.

When setting 16-bit timer mode, the input clock for timer 1 is provided by the overflow output of timer 0, irrespective of the clock control register TCLK setting.

Table 3.8.2 Selection of 16-Bit Timer (Interrupt) Interval and Input Clock

Input Clock	Interrupt Interval (fc = 25 MHz)	Resolution
ϕT1 (8/fc)	0.4 μs to 26.214 ms	0.4 μs
ϕT4 (32/fc)	1.6 μs to 104.858 ms	1.6 μs
ϕT16 (128/fc)	6.4 μs to 419.430 ms	6.4 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 the compare, while entering data in TREG1 starts the compare).

Setting example: To generate interrupt INTT1 every 0.4 s at fc = 20 MHz, set the following values in timer registers TREG0 and TREG1:
Using ϕT16 (= 6.4 μs at 20 MHz) as a timer input clock,
0.4 s ÷ 6.4 μs = 62500 = F424H
Therefore, set TREG1 to F4H, and TREG0 to 24H.

A match between up-counter UC0 and TREG0 triggers the timer 0 comparator to generate a match detect signal, but does not clear up-counter UC0. No interrupt INTT0 is generated.

A match between up-counter UC1 and TREG1 at comparator timing triggers the timer 1 comparator to generate a match detect signal. When comparator match detect signals for both timer 0 and timer 1 are generated, up-counter 0 and up-counter 1 are cleared to 0 and interrupt INTT1 only is generated. When invert is enabled, the value of timer flip-flop TFF1 is inverted.

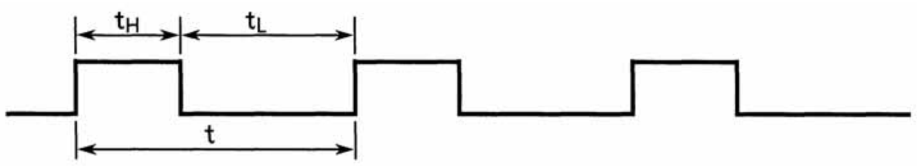
	Timer 0			Timer 1		
	INT T0	TO1	Match Value	INT T1	TO1	Match Value
16-bit timer mode (timer 1 counts up on timer 0 overflow)	no interrupt generated	output disabled	TREG0 (timer 1 continues counting up at match)	interrupt generated	output enabled	TREG1*2^8 + TREG0 (full 16 bits)
8-bit timer mode (timer 1 counts up on timer 0 match)	interrupt generated	output enabled (timer 0 or timer 1)	TREG0 clear at match	interrupt generated	output enabled (timer 0 or timer 1)	TREG1* TREG0 (product)

(3) 8-bit programmable pulse generation output mode

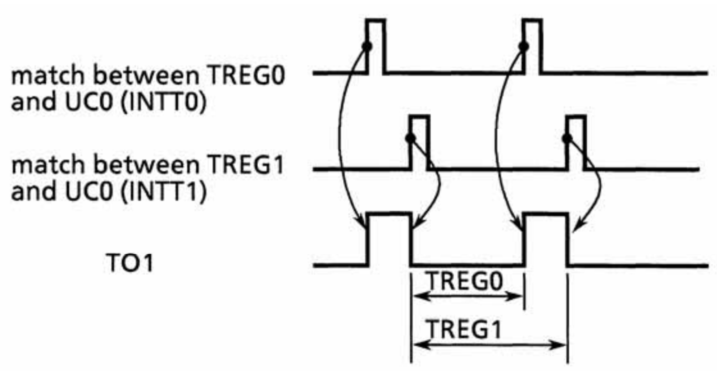
Timers 0, 2 can output variable frequencies and square waves (Pulses) with variable duty. The output pulse can be set to either active low or active high.

Timers 1, 3 cannot be used in this mode.

Timer 0 outputs from pin TO1 (also used as PC0), timer 2 outputs from pin TO3 (also used as PC1).



As timers 0, 2 operate the same, the following describes timer 0 only.



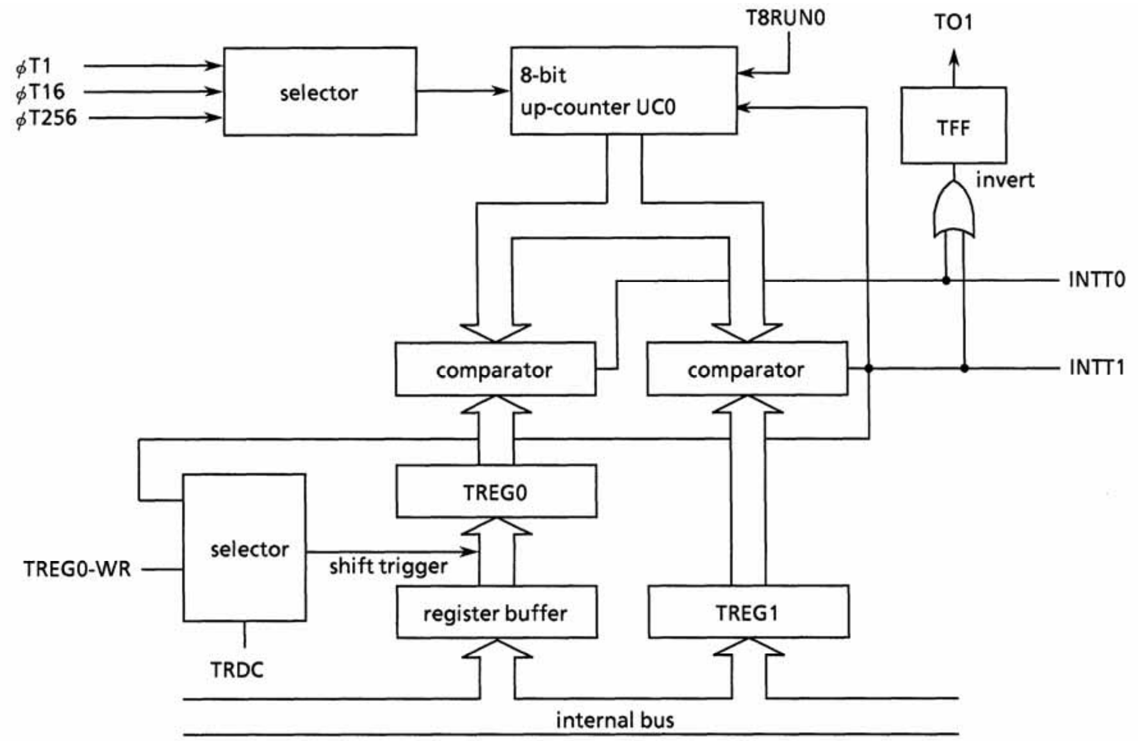
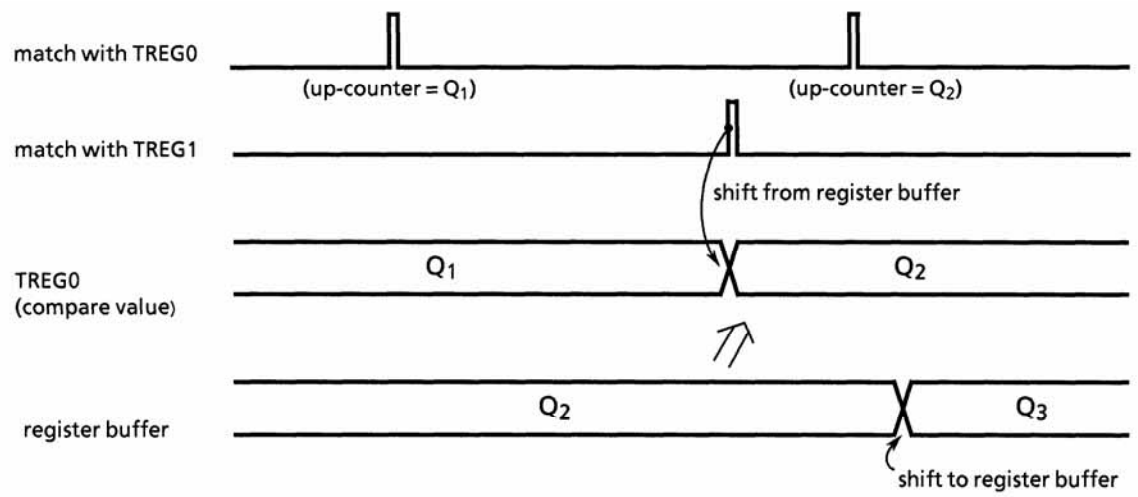


Figure 3.8.13 8-Bit PPG Output Mode Block Diagram

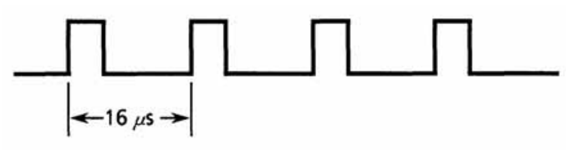
Enabling the TREG0 double-buffer in this mode shifts the register buffer value to TREG0 when TREG1 matches UC0.

Using the double-buffer facilitates output of waveforms with a low duty ratio (when changing the duty).



Register Buffer Operation

Example: Output a 1/4-duty 62.5 kHz pulse (at fc = 20 MHz)



- Determine the set value in the timer register.

  Setting the frequency to 62.5 kHz generates a square wave with a cycle of t = 1/62.5 kHz = 16 μs.

  Using ϕT1 = 0.4 μs (at fc = 20 MHz) results in:

  16 μs ÷ 0.4 μs = 40

  Accordingly, set timer register 1 (TREG1) to TREG1 = 40 = 28H.

  Next, set the duty to 1/4 as follows: t × 1/4 = 16 μs × 1/4 = 4 μs

  Accordingly, set timer register 0 (TREG0) to TREG0 = 10 = 0AH.

		MSB							LSB	
		7	6	5	4	3	2	1	0	
T8RUN	←	X	X	X	X	-	-	0	0	Stops timers 0 and 1, and clear them to 0.
T01MOD	←	1	0	X	X	0	1	0	1	Sets 8-bit PPG mode and sets the input clock to ϕT1.
TFFCR	←	-	-	-	-	0	1	1	x	Sets TFF1 to "1" and enables invert.
										Setting to "10" obtains a negative logic output wave.
TREG0	←	0	0	0	0	1	0	1	0	Writes 0AH.
TREG1	←	0	0	1	0	1	0	0	0	Writes 28H.
PCCR	←	X	X	X	X	-	-	1	-	Sets P91 to TO1.
PCFC	←	X	X	X	X	-	-	1	X	
T8RUN	←	X	X	X	X	-	-	1	1	Starts timers 0 and 1 counting.
T16RUN	←	1	X	X	X	-	-	-	-	Starts prescaler

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

(4) 8-bit pulse width modulation (PWM) output mode

Only timers 0, 2 support this mode, which allows up to two pulse width modulation outputs with 8-bit resolution.

For timer 0, PWM is output to pin TO1 (also used as PC0). For timer 2 PWM is output to pin TO3 (also used as PC1).

Timers 1, 3 can be used as 8-bit timers.

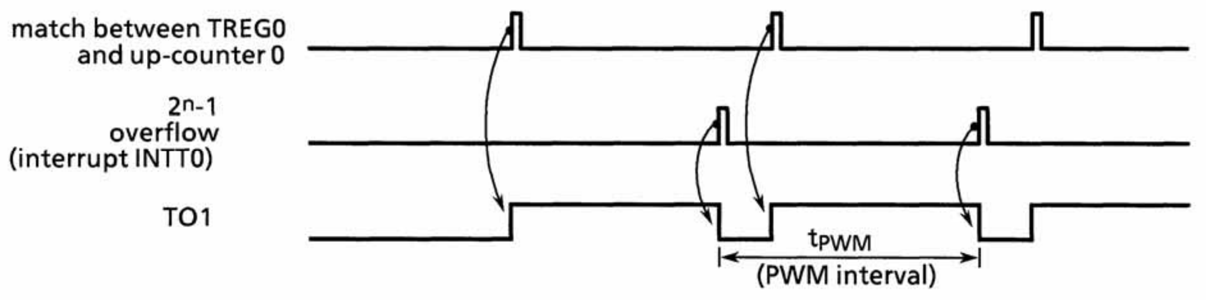
As timers 0, 2 operate the same, the following describes timer 0 only.

Timer output is inverted when the up-counter UC0 setting and the timer register TREG setting match, or when 2^n-1 (T01MOD specifies one of n = 6, n = 7, or n = 8) counter overflow occurs. The up-counter UC0 is cleared by the 2^n-1 counter overflow.

In 8-bit PWM output mode, the following conditions must be satisfied:

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

(Timer register setting) $\neq 0$



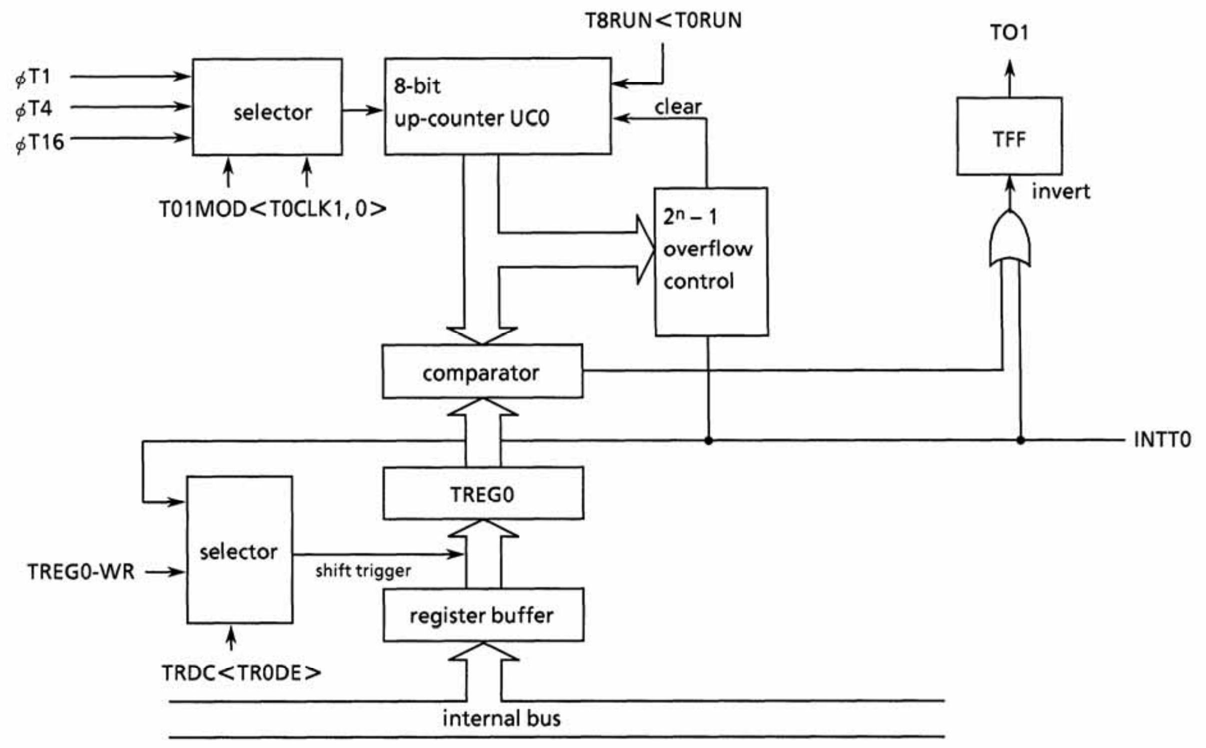
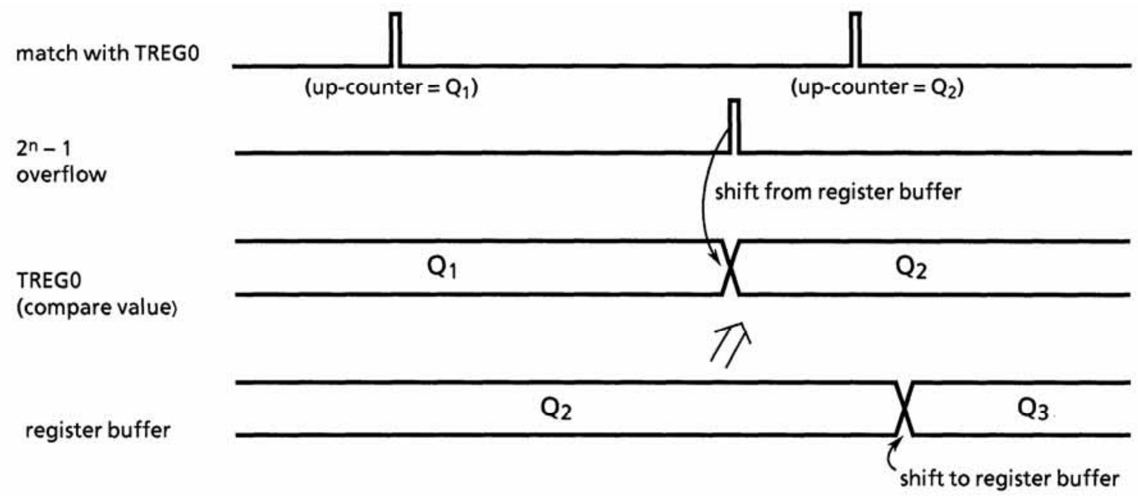


Figure 3.8.14 8-Bit PWM Output Mode Block Diagram

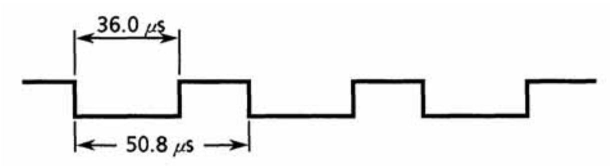
Enabling the TREG0 double-buffer in this mode shifts the register buffer value to TREG0 when $2^n - 1$ overflow is detected.

Using the double-buffer facilitates output of waveforms with a low duty ratio (when changing the duty).



Register Buffer Operation

Example: Output the following PWM waveform to pin TO1 using timer 0 for fc = 20 MHz:



To realize a PWM interval of 50.8 µs using ϕT1 = 0.4 µs (at fc = 20 MHz)

$50.8\ \mu s \div 0.4\ \mu s = 127 = 2^n - 1$

Accordingly, set n = 7.

As the low-level interval is 36.0 µs, at ϕT1 = 0.4 µs,

set $36.0\ \mu s \div 0.4\ \mu s = 90$ = 5AH in TREG0

		MSB 7	6	5	4	3	2	1	LSB 0	
T8RUN	←	X	X	X	X	–	–	–	0	Stops timer 0 and clears it to 0.
T01MOD	←	1	1	1	0	–	–	0	1	Sets to 8-bit PWM mode (interval = $2^7 - 1$) and sets the input clock to ϕT1.
TFFCR	←	–	–	–	–	1	0	1	X	Clears TFF1 and sets to invert enable.
TREG0	←	0	1	0	1	1	0	1	0	Writes 5AH.
PCCR	←	X	X	X	X	X	X	–	1	Sets PC0 to TO1.
PCFC	←	X	X	X	X	X	X	–	1	
T16RUN	←	1	X	X	X	–	–	–	–	Starts prescaler
T8RUN	←	X	X	X	X	–	–	–	1	Starts timer 0 counting.

Note: X: Don't care –: No change

Table 3.8.3 Setting PWM Interval and 2^n-1 Counter

	PWM Interval (at fc = 20 MHz)		
	ϕT1	ϕT4	ϕT16
$2^6 - 1$	25.2 µs (39.7 kHz)	100.8 µs (9.92 kHz)	403.2 µs (2.48 kHz)
$2^7 - 1$	50.8 µs (19.6 kHz)	203.2 µs (4.92 kHz)	810 µs (1.23 kHz)
$2^8 - 1$	102 µs (9.80 kHz)	408 µs (2.45 kHz)	1.63 ms (0.61 kHz)

(5) Table 3.8.4 shows the settings for all 8-bit timer modes.

Table 3.8.4 Setting Register for All Timer Modes

Timer Mode (for 8-bit timer x 2 channels)	Mode T01M (T23M)	PWM0 (PWM2)	Upper Timer Input Clock T1CLK (T3CLK)	Lower Timer Input Clock T0CLK (T2CLK)	Invert Select FF1IS (FF3IS)
16-bit timer (full 16 bits) × 1ch	01	–	–	(ϕT1, 4, 16)	–
8-bit timer (8-bit × 8-bit mode × 1ch) (inputs lower timer comparator output to upper timer)	00	–	00	(ϕT1, 4, 16)	0: lower timer 1: upper timer
8-bit timer × 2ch	00	–	(ϕT1, 16, 256)	(ϕT1, 4, 16)	0: lower timer 1: upper timer
8-bit PPG × 1ch	10	–	–	(ϕT1, 4, 16)	–
8-bit PWM × 1ch (lower) 8-bit timer × 1ch (upper)	11	PWM interval	(ϕT1, 16, 256)	(ϕT1, 4, 16)	–

3.9 16-Bit Timers

TMP94C251A incorporates four multi-function 16-bit timer/event counters (Timers 4, 6, 8, and A).

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable pulse generation (PPG) output mode
- Frequency measurement mode
- Pulse width modulation (PWM) mode
- Time differential measurement mode

The timer/event counters have a 16-bit up counter, two 16-bit timer registers (One with a double-buffer configuration), two 16-bit capture registers, two comparators, capture input control, and timer flip-flops and accompanying F/F control circuit.

The timer/event counter is controlled by four control registers: T4MOD/T6MOD/T8MOD/TAMOD, T4FFCR/T6FFCR/T8FFCR/TAFFCR, T16RUN and T16CR.

Figure 3.9.1, Figure 3.9.2, Figure 3.9.3, and Figure 3.9.4 are block diagrams of a 16-bit timer/event counter (Timer 4, 6, 8, A).

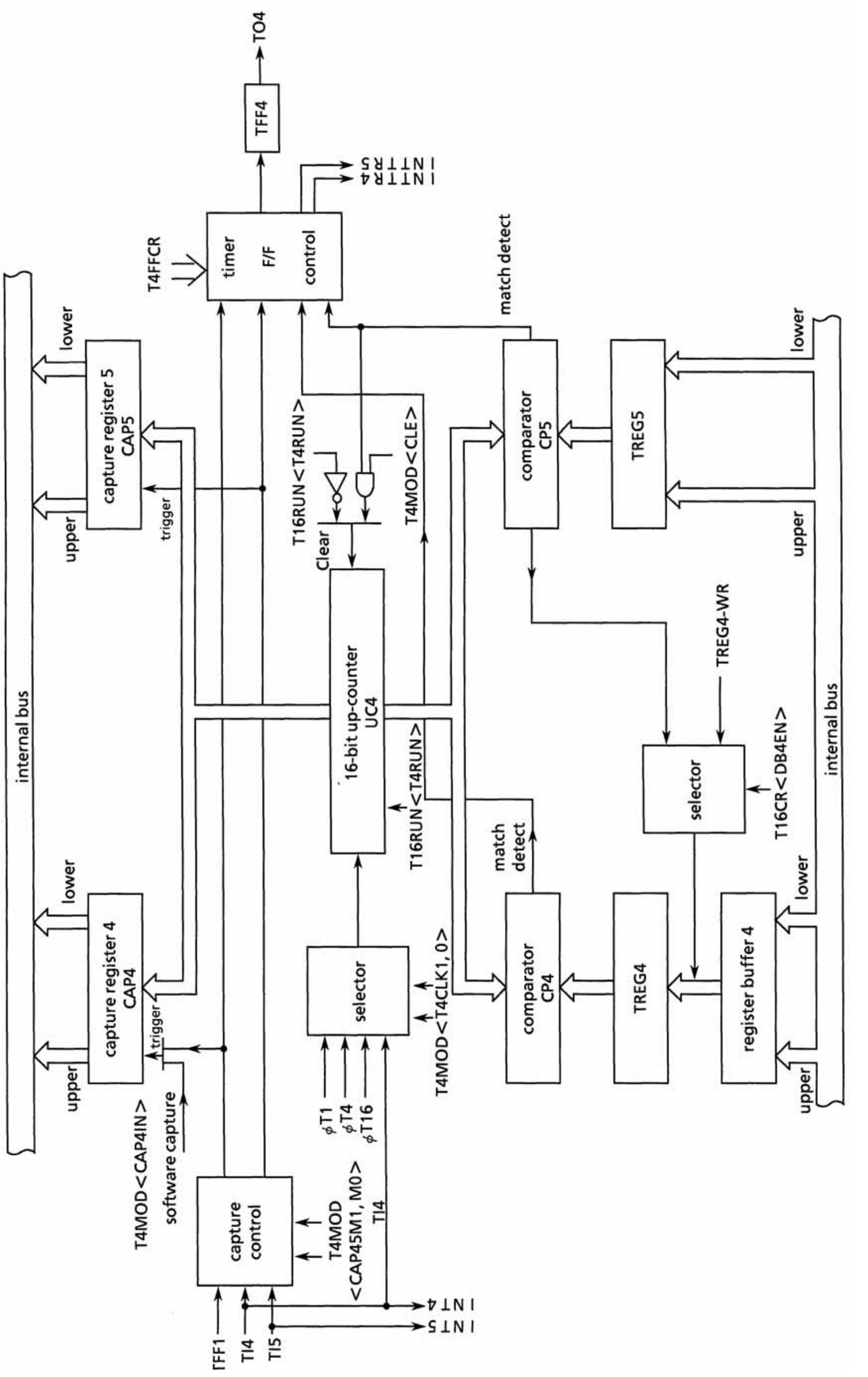


Figure 3.9.1 16-Bit Timer Block Diagram (Timer 4)

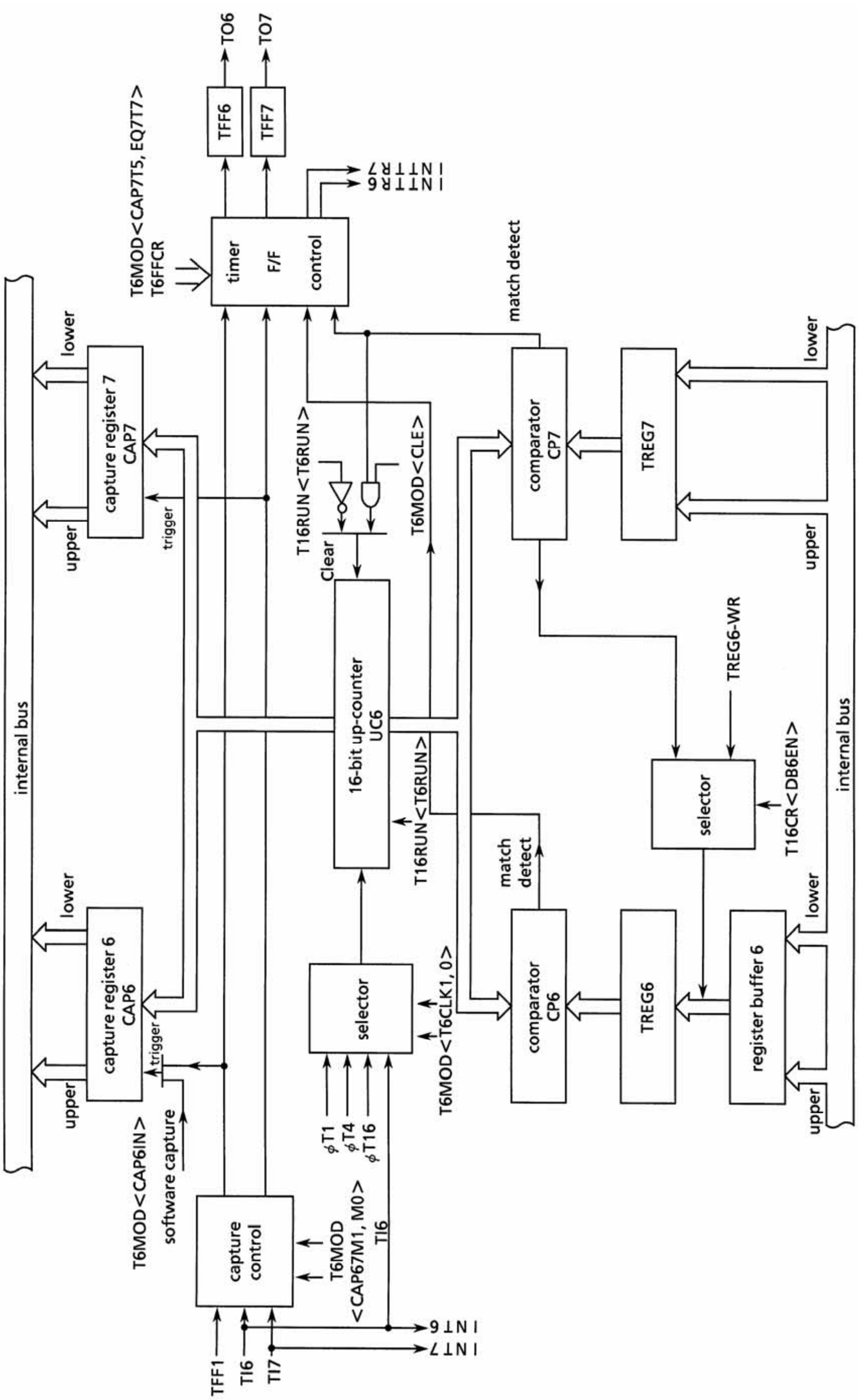


Figure 3.9.2 16-Bit Timer Block Diagram (Timer 6)

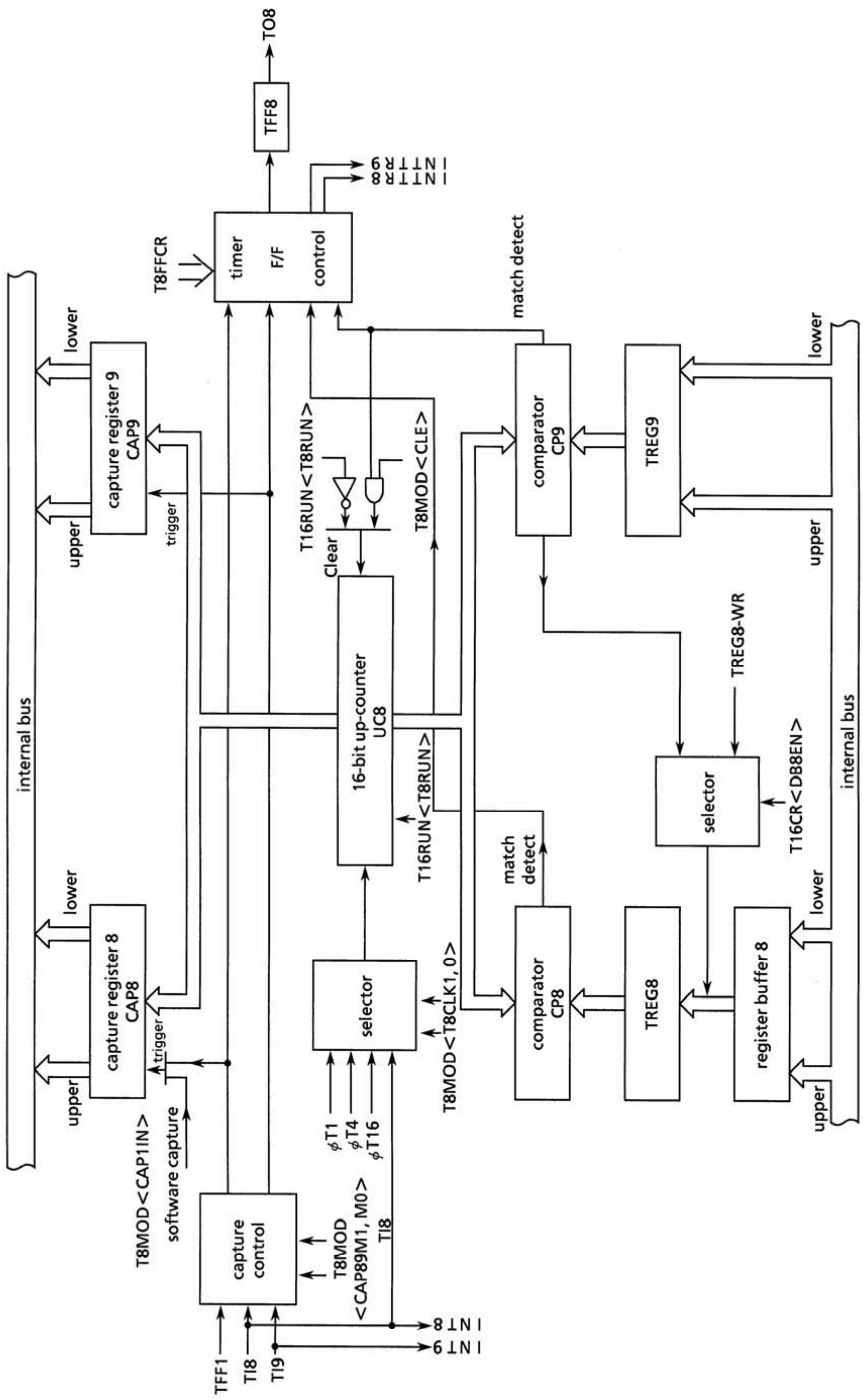


Figure 3.9.3 16-Bit Timer Block Diagram (Timer 8)

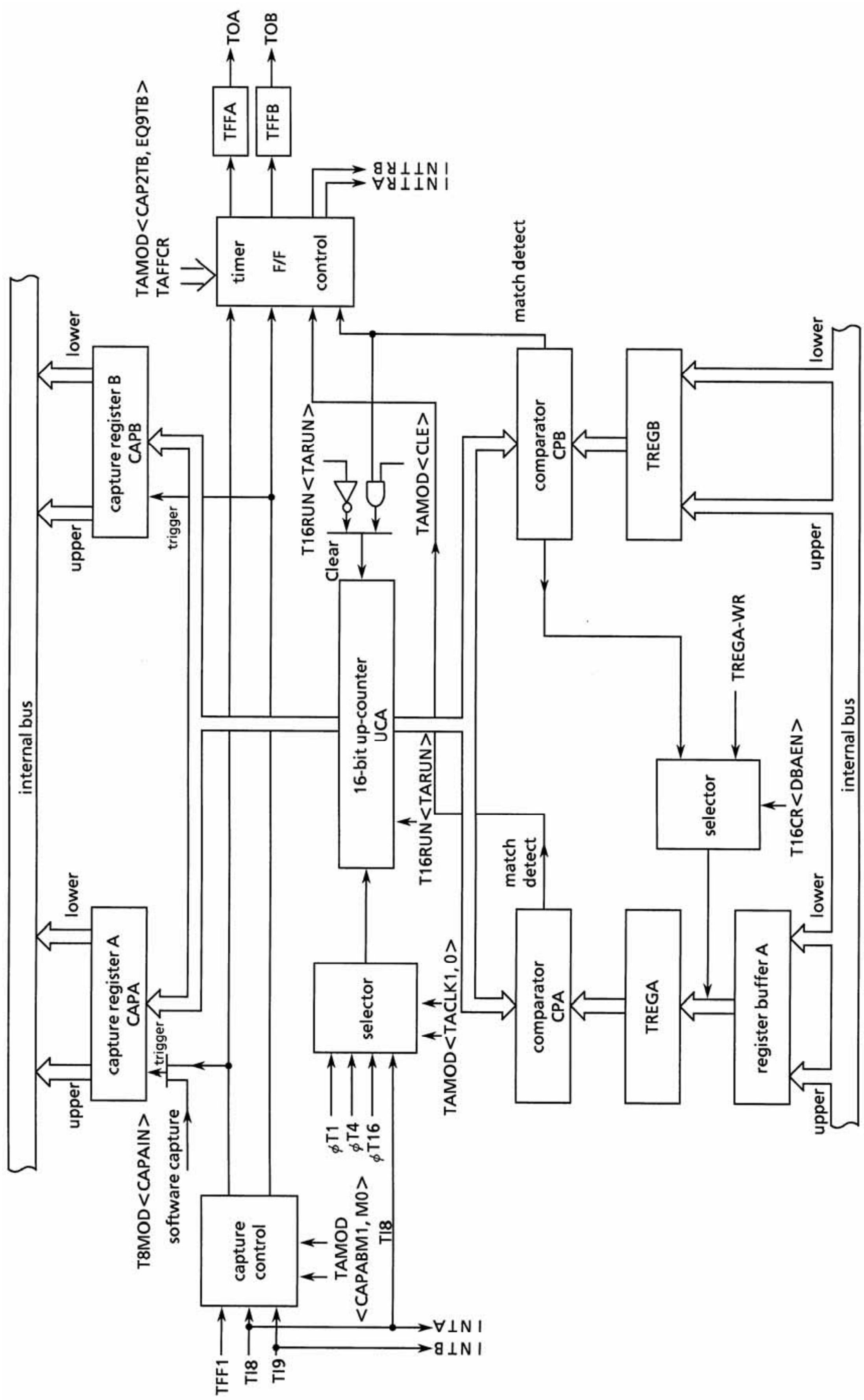


Figure 3.9.4 16-Bit Timer Block Diagram (Timer A)

T4MOD (0098H)

	7	6	5	4	3	2	1	0
bit Symbol	–	–	CAP4IN	CAP45M1	CAP45M0	CLE	T4CLK1	T4CLK0
Read/Write			W	R/W		R/W	R/W	
After reset	–	–	–	0	0	0	0	0
Function			0: Soft-Capture 1: Don't care	Capture timing 00: Disable 01: TI4↑ TI5↑ 10: TI4↑ TI4↓ 11: TFF1↑ TFF1↓		1: UC4 Clear Enable	Source clock 00: TI4 01: ϕT1 10: ϕT4 11: ϕT16	

Timer 4 input clock

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

Up-counter UC4 clear

0	Disables up-counter clear.
1	Clears by match with TREG5.

Note)Read-modify-write is prohibited.

Figure 3.9.5 16-Bit Timer Mode Control Register (T4MOD) (1/2)

T4MOD (0098H)

	7	6	5	4	3	2	1	0
bit Symbol	–	–	CAP4IN	CAP45M1	CAP45M0	CLE	T4CLK1	T4CLK0
Read/Write			W	R/W		R/W	R/W	
After reset	–	–	–	0	0	0	0	0
Function			0: Soft-Capture 1: Don't care	Capture timing 00: Disable 01: TI4↑ TI5↑ 10: TI4↑ TI4↓ 11: TFF1↑ TFF1↓		1: UC4 Clear Enable	Source clock 00: TI4 01: ϕT1 10: ϕT4 11: ϕT16	

Timer 4 capture timing

	Capture control	INT4 control
00	Capture disable	
01	CAP4 on TI4 rising CAP5 on TI5 rising	INT4 generated on TI4 rising
10	CAP4 on TI4 rising CAP5 on TI4 falling	INT4 generated on TI4 falling
11	CAP4 on TFF1 rising CAP5 on TFF1 falling	INT4 generated on TI4 rising

Software capture

0	Loads up-counter 4 value to CAP4.
1	Disables software capture to CAP4.

Note)Read-modify-write is prohibited.

Figure 3.9.6 16-Bit Timer Control Register (T4MOD) (2/2)

T4FFCR
(0099H)

	7	6	5	4	3	2	1	0
bit Symbol	–	–	CAP5T4	CAP4T4	EQ5T4	EQ4T4	TFF4C1	TFF4C0
Read/Write			R/W	R/W	R/W	R/W	W	
After reset	–	–	0	0	0	0	–	
Function			TFF4 invert trigger 0: Disable trigger 1: Enable trigger				00: Invert TFF4 01: Set TFF4 10: Clear TFF4 11: Don't care	
			At loading of up-counter value to CAP5	At loading of up-counter value to CAP4	At match between up-counter and TREG5	At match between up-counter and TREG4		

Timer flip-flop 4 (TFF4) control

00	Inverts TFF4 value (software invert).
01	Sets TFF4 to "1".
10	Clears TFF4 to "0".
11	Don't care

Timer flip-flop 4 (TFF4) invert trigger

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

Note)Read-modify-write is prohibited.

Figure 3.9.7 16-Bit Timer 4 F/F Control (T4FFCR)

T6MOD
(00A8H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP7T7	EQ7T7	CAP6IN	CAP67M1	CAP67M0	CLE	T6CLK1	T6CLK0
Read/Write	R/W		W	R/W		R/W	R/W	
After reset	0	0	–	0	0	0	0	0
Function	TFF7 invert trigger 0: Disable trigger 1: Enable trigger At loading of up-counter value to CAP7	At match between up-counter and TREG7	0: Soft-Capture 1: Don't care	Capture timing 00: Disable 01: TI6↑ TI7↑ 10: TI6↑ TI6↓ 11: TFF1↑ TFF1↓		1: UC6 Clear Enable	Source clock 00: TI6 01: ϕT1 10: ϕT4 11: ϕT16	

→ Timer 6 input clock

00	External input clock (TI6)
01	Internal clock ϕT1 (8/fc)
10	Internal clock ϕT4 (32/fc)
11	Internal clock ϕT16 (128/fc)

→ Up-counter UC6 clear

0	Disables up-counter clear.
1	Clears by match with TREG7.

Note)Read-modify-write is prohibited.

Figure 3.9.8 16-Bit Timer Mode Control Register (T6MOD) (1/2)

T6MOD (00A8H)

	7	6	5	4	3	2	1	0
bit Symbol	CAP7T7	EQ7T7	CAP6IN	CAP67M1	CAP67M0	CLE	T6CLK1	T6CLK0
Read/Write	R/W		W	R/W		R/W	R/W	
After reset	0	0	–	0	0	0	0	0
Function	TFF7 invert trigger 0: Disable trigger 1: Enable trigger At loading of up-counter value to CAP7	At match between up-counter and TREG7	0: Soft-Capture 1: Don't care	Capture timing 00: Disable 01: TI6↑ TI7↑ 10: TI6↑ TI6↓ 11: TFF1↑ TFF1↓		1: UC6 Clear Enable	Source clock 00: TI6 01: ϕT1 10: ϕT4 11: ϕT16	

→ Timer 6 capture timing

	Capture control	INT6 control
00	Capture disable	
01	CAP6 on TI6 rising CAP7 on TI7 rising	INT6 generated onTI6 rising
10	CAP6 on TI6 rising CAP7 on TI6 falling	INT6 generated on TI6 falling
11	CAP6 on TFF1 rising CAP7 on TFF1 falling	INT6 generated on TI6 rising

→ Software capture

0	At loading of up-counter 6 value to CAP6
1	Disables software capture to CAP6

→ Invert trigger of timer flip-flop 7 (TFF7)

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

Note)Read-modify-write is prohibited.

Figure 3.9.9 16-Bit Timer Control Register (T6MOD) (2/2)

T6FFCR
(00A9H)

	7	6	5	4	3	2	1	0
bit Symbol	TFF7C1	TFF7C0	CAP7T6	CAP6T6	EQ7T6	EQ6T6	TFF6C1	TFF6C0
Read/Write	W		R/W	R/W	R/W	R/W	W	
After reset	0	0	0	0	0	0	–	
Function	00: Invert TFF7 01: Set TFF7 10: Clear TFF7 11: Don't care		TFF6 invert trigger 0: Disable trigger 1: Enable trigger				00: Invert TFF6 01: Set TFF6 10: Clear TFF6 11: Don't care	
			At loading of up-counter value to CAP7	At loading of up-counter value to CAP6	At match between up-counter and TREG7	At match between up-counter and TREG6		

Timer flip-flop 6 (TFF6) control

00	Inverts TFF6 value (software invert).
01	Sets TFF6 to "1".
10	Clears TFF6 to "0".
11	Don't care

Timer flip-flop 6 (TFF6) invert trigger

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

Timer flip-flop 7 (TFF7) control

00	Inverts TFF7 value (software invert).
01	Sets TFF7 to 1.
10	Clears TFF7 to 0.
11	Don't care

Note)Read-modify-write is prohibited.

Figure 3.9.10 16-Bit Timer 6 F/F Control (T6FFCR)

T8MOD (00B8H)		7	6	5	4	3	2	1	0
	bit Symbol	–	–	CAP8IN	CAP89M1	CAP89M0	CLE	T8CLK1	T8CLK0
	Read/Write			W	R/W		R/W	R/W	
	After reset	–	–	–	0	0	0	0	0
	Function			0: Soft-Capture 1: Don't care	Capture timing 00: Disable 01: TI8↑ TI9↑ 10: TI8↑ TI8↓ 11: TFF1↑ TFF1↓		1: UC8 Clear Enable	Source clock 00: TI8 01: ϕT1 10: ϕT4 11: ϕT16	

Timer 8 input clock

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

Up-counter UC8 clear

0	Disables up-counter clear.
1	Clears by match with TREG9.

Note)Read-modify-write is prohibited.

Figure 3.9.11 16-Bit Timer Mode Control Register (T8MOD) (1/2)

T8MOD (00B8H)

	7	6	5	4	3	2	1	0
bit Symbol	–	–	CAP8IN	CAP89M1	CAP89M0	CLE	T8CLK1	T8CLK0
Read/Write			W	R/W		R/W	R/W	
After reset	–	–	–	0	0	0	0	0
Function			0: Soft-Capture 1: Don't care	Capture timing 00: Disable 01: TI8↑ TI9↑ 10: TI8↑ TI8↓ 11: TFF1↑ TFF1↓		1: UC8 Clear Enable	Source clock 00: TI8 01: ϕT1 10: ϕT4 11: ϕT16	

Timer 8 capture timing

	Capture control	INTA control
00	Capture disable	
01	CAP8 on TI8 rising CAP9 on TI9 rising	INT8 generated onTI8 rising
10	CAP8 on TI8 rising CAP9 on TI8 falling	INT8 generated on TI8 falling
11	CAP8 on TFF1 rising CAP9 on TFF1 falling	INT8 generated on TI8 rising

Software capture

0	At loading of up-counter 8 value to CAP8
1	Disables software capture to CAP8

Note)Read-modify-write is prohibited.

Figure 3.9.12 16-Bit Timer Control Register (T8MOD) (2/2)

T8FFCR (00B9H)

	7	6	5	4	3	2	1	0
bit Symbol	–	–	CAP9T8	CAP8T8	EQ9T8	EQ8T8	TFF8C1	TFF8C0
Read/Write			R/W	R/W	R/W	R/W	W	
After reset	–	–	0	0	0	0	–	
Function			TFF8 invert trigger 0: Disable trigger 1: Enable trigger				00: Invert TFF8 01: Set TFF8 10: Clear TFF8 11: Don't care	
			At loading of up-counter value to CAP9	At loading of up-counter value to CAP8	At match between up-counter and TREG9	At match between up-counter and TREG8		

Timer flip-flop 8 (TFF8) control

00	Inverts TFF8 value (software invert).
01	Sets TFF8 to "1".
10	Clears TFF8 to "0".
11	Don't care

Timer flip-flop 8 (TFF8) invert trigger

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

Note)Read-modify-write is prohibited.

Figure 3.9.13 16-Bit Timer 8 F/F Control (T8FFCR)

		7	6	5	4	3	2	1	0
TAMOD (00C8H)	bit Symbol	CAPBTB	EQBTB	CAPAIN	CAPABM1	CAPABM0	CLE	TACLK1	TACLK0
	Read/Write	R/W		W	R/W		R/W	R/W	
	After reset	0	0	1	0	0	0	0	0
	Function	TFFB invert trigger 0: Disable trigger 1: Enable trigger		0: Soft-Capture 1: Don't care	Capture timing 00: Disable 01: TIA↑ TIB↑ 10: TIA↑ TIA↓ 11: TFF1↑ TFF1↓		1: UCA Clear Enable	Source clock 00: TI6 01: ϕT1 10: ϕT4 11: ϕT16	
		At loading of up-counter value to CAPB	At match between up-counter and TREGB						

Timer A input clock

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

Up-counter UCA clear

0	Disables up-counter clear.
1	Clears by match with TREGB.

Note)Read-modify-write is prohibited.

Figure 3.9.14 16-Bit Timer Mode Control Register (TAMOD) (1/2)

TAMOD (00C8H)

	7	6	5	4	3	2	1	0
bit Symbol	CAPBTB	EQBTB	CAPAIN	CAPABM1	CAPABM0	CLE	TACLK1	TACLK0
Read/Write	R/W		W	R/W		R/W	R/W	
After reset	0	0	–	0	0	0	0	0
Function	TFFB invert trigger 0: Disable trigger 1: Enable trigger At loading of up-counter value to CAPB	At match between up-counter and TREGB	0: Soft-Capture 1: Don't care	Capture timing 00: Disable 01: TIA↑ TIB↑ 10: TIA↑ TIA↓ 11: TFF1↑ TFF1↓		1: UCA Clear Enable	Source clock 00: TIA 01: ϕT1 10: ϕT4 11: ϕT16	

Timer A capture timing

	Capture control	INTA control
00	Capture disable	
01	CAPA on TIA rising CAPB on TIB rising	INTA generated onTIA rising
10	CAPA on TIA rising CAPB on TIA falling	INTA generated on TIA falling
11	CAPA on TFF1 rising CAPB on TFF1 falling	INTA generated on TIA rising

Software capture

0	At loading of up-counter A value to CAPA
1	Disables software capture to CAPA

Invert trigger of timer flip-flop B (TFFB)

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

CAPBTB: At loading of up-counter value to CAPB
EQBTB: At match between up-counter and TREGB

Note)Read-modify-write is prohibited.

Figure 3.9.15 16-Bit Timer Control Register (TAMOD) (2/2)

TAFFCR (00C9H)

	7	6	5	4	3	2	1	0
bit Symbol	TFFBC1	TFFBC0	CAPBTA	CAPATA	EQBTA	EQATA	TFFAC1	TFFAC0
Read/Write	W		R/W	R/W	R/W	R/W	W	
After reset	0	0	0	0	0	0	–	
Function	00: Invert TFFB 01: Set TFFB 10: Clear TFFB 11: Don't care		TFFA invert trigger 0: Disable trigger 1: Enable trigger At loading of up-counter value to CAPB	At loading of up-counter value to CAPA	At match between up-counter and TREGB	At match between up-counter and TREGA	00: Invert TFFA 01: Set TFFA 10: Clear TFFA 11: Don't care	

Timer flip-flop A (TFFA) control

00	Inverts TFFA value (software invert).
01	Sets TFFA to "1".
10	Clears TFFA to "0".
11	Don't care

Timer flip-flop A (TFFA) invert trigger

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

Timer flip-flop B (TFFB) control

00	Inverts TFFB value (software invert).
01	Sets TFFB to 1.
10	Clears TFFB to 0.
11	Don't care

Note)Read-modify-write is prohibited.

Figure 3.9.16 16-Bit Timer A F/F Control (TAFFCR)

T16RUN (009EH)

	7	6	5	4	3	2	1	0
bit Symbol	PRRUN	–	–	–	TARUN	T8RUN	T6RUN	T4RUN
Read/Write	R/W				R/W			
After reset	0	–	–	–	0	0	0	0
Function	Prescaler 0 : STOP 1 : RUN				16-Bit Timer RUN /STOP Control 0 : STOP & Clear 1 : RUN (Count up)			

16-bit timer (timer 4) operation

0	Stop and clear
1	Count up

16-bit timer (timer 6) operation

0	Stop and clear
1	Count up

16-bit timer (timer 8) operation

0	Stop and clear
1	Count up

16-bit timer (timer A) operation

0	Stop and clear
1	Count up

Figure 3.9.17 16-Bit Timer Operation Control Register (T16RUN)

T16CR (009FH)

	7	6	5	4	3	2	1	0
bit Symbol	–	–	–	–	DBAEN	DB8EN	DB6EN	DB4EN
Read/Write					R/W			
After reset	–	–	–	–	0	0	0	0
Function					1: Double buffer Enable			

Double-buffer control

0	Disable
1	Enable

Figure 3.9.18 16-Bit Timer (4, 6, 8, A) Control Register (T16CR)

Symbol	Address	7	6	5	4	3	2	1	0
TREG4L	90h	–							
		W							
		Undefined							
TREG4H	91h	–							
		W							
		Undefined							
TREG5L	92h	–							
		W							
		Undefined							
TREG5H	93h	–							
		W							
		Undefined							
TREG6L	A0h	–							
		W							
		Undefined							
TREG6H	A1h	–							
		W							
		Undefined							
TREG7L	A2h	–							
		W							
		Undefined							
TREG7H	A3h	–							
		W							
		Undefined							
TREG8L	B0h	–							
		W							
		Undefined							
TREG8H	B1h	–							
		W							
		Undefined							
TREG9L	B2h	–							
		W							
		Undefined							
TREG9H	B3h	–							
		W							
		Undefined							
TREGAL	C0h	–							
		W							
		Undefined							
TREGAH	C1h	–							
		W							
		Undefined							
TREGBL	C2h	–							
		W							
		Undefined							
TREGBH	C3h	–							
		W							
		Undefined							

Note)Read-modify-write is prohibited.

Figure 3.9.19 Timer Register

Symbol	Address	7	6	5	4	3	2	1	0
CAP4L	94h	–							
		R							
		Undefined							
CAP4H	95h	–							
		R							
		Undefined							
CAP5L	96h	–							
		R							
		Undefined							
CAP5H	97h	–							
		R							
		Undefined							
CAP6L	A4h	–							
		R							
		Undefined							
CAP6H	A5h	–							
		R							
		Undefined							
CAP7L	A6h	–							
		R							
		Undefined							
CAP7H	A7h	–							
		R							
		Undefined							
CAP8L	B4h	–							
		R							
		Undefined							
CAP8H	B5h	–							
		R							
		Undefined							
CAP9L	B6h	–							
		R							
		Undefined							
CAP9H	B7h	–							
		R							
		Undefined							
CAPAL	C4h	–							
		R							
		Undefined							
CAPAH	C5h	–							
		R							
		Undefined							
CAPBL	C6h	–							
		R							
		Undefined							
CAPBH	C7h	–							
		R							
		Undefined							

Figure 3.9.20 Capture Register

[1] Up counter

The up counter is a 16-bit binary counter that counts up using the input clock specified by 16-bit timer mode control registers T4MOD<T4CLK1:0>, T6MOD<T6CLK1:0>, T8MOD<T8CLK1:0>, and TAMOD<TACLK1:0>.

The input clock is selected from internal clocks ϕT1, ϕT4, and ϕ16 output from the 9-bit prescaler (shared with the 8-bit timers), or the external clocks output from pin TI4 (also used as PD1/INT4), pin TI6 (also used as PD5/INT6), pin TI8 (also used as PE1/INT8), and pin TIA (also used as PE5/INTA). A reset initializes <T4CLK1:0> /<T8CLK1:0>/<T9CLK1:0>/<TACLK1:0> to “00”, selecting an external input clock on pin TI4/TI6/TI8/TIA as the input clock.

To control the count, stop, and clear functions for the counter, use timer control register T16RUN<T4RUN, T6RUN, T8RUN, TARUN>.

If up counter clearing is enabled, up counter UC4/UC6/UC8/UCA are cleared to 0 when up counter UC4/UC6/UC8/UCA matches timer register TREG6/TREG7/TREG9/ TREGB. The clear enable/disable is set with T4MOD<CLE>, T6MOD<CLE>, T8MOD <CLE>, and TAMOD<CLE>.

When clear disable is set, the counter operates as a free-running counter.

[2] Timer registers

Each timer has two internal 16-bit registers for setting counter values. When the value set in the timer register matches the value of the up counter UC4/UC6/UC8/UCA, the comparator match detect signal is activated.

Setting data for both H and L registers (TREG4L/H, TREG5L/H, TREG6L/H, TREG7L/H, TREG8L/H, TREG9L/H, TREGAL/H, TREGBL/H) are always needed. For example, either using the 2-byte data load instruction or the 1-byte data load instruction twice; first to write data to the lower 8 bits, then to write data to the upper 8 bits.

Timer registers TREG4, TREG6, TREG8, and TREGA have a double-buffer configuration and are paired with a register buffer. Timer registers TREG4/TREG6/ TREG8/TREGA enable/disable the double-buffer function using timer control register T16CR<DB4EN, DB6EN, DB8EN, DBAEN>. Setting <DB4EN, DB6EN, DB8EN, DBAEN> to 0 disables the double buffer; setting <DB4EN, DB6EN, 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 UC4/UC6/UC8/UCA and timer register TREG5/TREG7/TREG9/TREGB.

A reset initializes T16CR<DB4EN, DB6EN, DB8EN, DBAEN> to “0”, disabling the double buffer. When using the double buffer, write data to the timer register and set <DB4EN, DB6EN, DB8EN, DBAEN> to “1”, then write the next data to the register buffer.

TREG4/TREG6/TREG8/TREGA and the register buffer are allocated to the same addresses in memory (000090H, 000091H/0000A0H, 0000A1H/0000B0H, 0000B1H/0000C0H, 0000C1H).

When <DB4EN, DB6EN, DB8EN, DBAEN>is set to “0”, the same value is written to TREG4/TREG6/TREG8/TREGA and to their respective register buffers. When <DB4EN, DB6EN, DB8EN, DBAEN> is set to “1”, the value is written to the register buffers only. Therefore, disable the register buffers before writing the initial values to the timer registers.

As the timer registers are undefined after a reset, be sure to write data to the upper and lower registers before using the timers.

[3] Capture register

The capture register is a 16-bit register for latching the up counter value.
Data in the capture registers should be read all 16 bits(H and L).
When reading the capture register, use the 2-byte data load instruction, or the 1-byte data load instruction twice; first to read data from the lower 8 bits, then to read data from the upper 8 bits.

[4] Capture input control

The capture input control circuit controls the timing to latch the up counter UC4/UC6/UC8/UCA value to capture registers CAP4, CAP5/CAP6, CAP7/CAP8, CAP9/CAPA, CAPB.
Set the capture register latch timing using T4MOD<CAP45M1:0>/T6MOD <CAP67M1:0>/T8MOD<CAP89M1:0>/TAMOD<CAPABM1:0>.

- When T4MOD<CPA45M1:0>/T6MOD<CAP67M1:0>/T8MOD<CAP89M1:0>/ DAMOD<CAPABM1:0> = "00",
  the capture function is disabled. Resetting disables the capture function.

- When T4MOD<CPA45M1:0>/T6MOD<CAP67M1:0>/T8MOD<CAP89M1:0>/ DAMOD<CAPABM1:0> = "01",
  On the TI4 (also used as PD1/INT4)/TI6 (also used as PD5/INT6)/TI8 (also used as PE1/INT8)/TIA (also used as PE5/INTA) input rising edge, the up counter value is loaded to capture register CAP4/CAP6/CAP8/CAPA. On the TI5 (also used as PD2/INT5)/TI7 (also used as PD6/INT7)/TI9 (also used as PE2/INT9)/TIB (also used as PE6/INTB) input rising edge, the up counter value is loaded to capture register CAP5/CAP7/CAP9/CAPB (Time differential measurement).

- When T4MOD<CPA45M1:0>/T6MOD<CAP67M1:0>/T8MOD<CAP89M1:0>/ DAMOD<CAPABM1:0> = "10",
  On the TI4/TI6/TI8/TIA input rising edge, the up counter value is loaded to capture register CAP4/CAP6/CAP8/CAPA. On the input falling edge, the up counter value is loaded to capture register CAP5/CAP7/CAP9/CAPB. In this mode only, interrupt INT4/INT6 is generated on a falling edge (Pulse width measurement).

- When T4MOD<CPA45M1:0>/T6MOD<CAP67M1:0>/T8MOD<CAP89M1:0>/ DAMOD<CAPABM1:0> = "11",
  On the timer flip-flop TFF1 rising edge, the up counter value is loaded to capture register CAP4/CAP6/CAP8/CAPA. On the falling edge, the up counter value is loaded to capture register CAP5/CAP7/CAP9/CAPB.
  The up counter value can also be loaded to a capture register on a software request. When "0" is written to T4MOD<CAP4IN>/T6MOD<CAP6IN>/T8MOD <CAP8IN>/TAMOD <CAPAIN>, the up counter value at that time is loaded to capture register CAP4/CAP6/CAP8 /CAPA. The prescaler must be set to RUN (Set T16RUN<PRRUN> to "1").

[5] Comparator

A 16-bit comparator compares the up counter UC4/UC6/UC8/UCA value with the value set in the timer register (TREG4, TREG5/TREG6, TREG7/TREG8, TREG9/TREGA, TREGB) to detect a match.

On detection of a match, the comparator generates interrupt INTTR4/INTTR5, INTTR6/INTTR7, INTTR8/INTTR9, INTTRA/INTTRB. Only a match with TREG5/TREG7/TREG9/TREGB clears up counter UC4/UC6/UC8/UCA. (Setting T4MOD <CLE>/T6MOD<CLE>/T8MOD<CLE>/TAMOD<CLE> to “0” disables UC4/UC6/ UC8/UCA clearing.)

[6] Timer flip-flop TFF4/TFF6/TFF8/TFFA

This flip-flop is inverted by a match detect signal from the comparator and a latch signal to the capture register.

Enable or disable the invert for each interrupt source using T4FFCR<CAP5T4, CAP4T4, EQ5T4, EQ4T4>/T6FFCR<CAP7T6, CAP6T6, EQ7T6, EQ6T6>/T8FFCR <CAP9T8, CAP8T8, EQ9T8, EQ8T8>/TAFFCR<CAPBTA, CAPATA, EQBTA, EQATA>.

To invert TFF4/TFF6/TFF8/TFFA write “00” to T4FFCR<TFF4C1:0>/T6FFCR <TFF6C1:0>/T8FFCR<TFF8C1:0>/TAFFCR<TFFAC1:0>. Writing “01” sets TFF4/TFF6/TFF8/TFFA to 1; “10” clears TFF4/TFF6/TFF8/TFFA to 0.

The TFF4/TFF6/TFF8/TFFA value can be output to timer output pin TO4 (also used as PD0)/TO6 (also used as PD4)/TO8 (also used as PE0)/TOA (also used as PE4).

[7] Timer flip-flop TFF7/TFFB

This flip-flop is inverted by a match detect signal between up counter UC6/A and timer register TREG7/B, and a latch signal to capture register CAP7/B.

Enable or disable the invert for each interrupt source using T6MOD<CAP7T7, EQ7T7>/TAMOD<CAPBTB, EQBTB>.

To invert TFF7/B, write “00” to T6FFCR<TFF7C1:0>/TAFFCR<TFFBC1:0>. Writing “01” sets TFF7/B to 1; “10” clears TFF7/B to 0.

The TFF7/B value can be output to timer output pin TO7 (also used as PC3)/TOB (also used as PC1).

Note: Only timer 6 and timer A contain this flip-flop (TFF7/TFFB).

(1) 16-bit timer mode

Timers 4, 6, 8 and A operate independently. As both timers operate the same, the following describes timer 4 only.

Example: Generate fixed-interval interrupts

Set an interval time in timer register TREG5 and generate interrupt INTTR5.

		7 6 5 4 3 2 1 0	
T16RUN	←	- X X X - - - 0	Stop timer 4.
INTET45	←	X 1 0 0 X 0 0 0	Enables INTTR5 (set to level 4) and disables INTTR4.
T8FFCR	←	1 1 0 0 0 0 1 1	Disables trigger.
T8MOD	←	X X 1 0 0 1 * *	Sets input clock to an internal clock, and disables
		(** = 01, 10, 11)	capture function.
TREG5	←	* * * * * * * *	Sets interval time.
		* * * * * * * *	(16 bits)
T16RUN	←	1 X X X - - - 1	Starts timer 4.

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

(2) 16-bit event counter mode

Setting external clock TI4/TI6/TI8/TIA as an input clock in 16-bit timer mode results in an event counter. To obtain a counter value, load the counter value into a capture register using "software capture" and read the captured value from the capture register.

The counter counts up at the TI4/TI6/TI8/TIA input rising edge.

The TI4/TI6/TI8/TIA pin is also used as PD1/INT4, PD5/INT6, PE1/INT8, PE5/INTA.

As timers 4, 6, 8 and A operate the same, the following describes timer 4 only.

		7 6 5 4 3 2 1 0	
T16RUN	←	- X X X - - - 0	Stop s timer 4.
PDCR	←	- - - - - - - -	Sets PD1 to input mode.
INTET45	←	X 1 0 0 X 0 0 0	Enables INTTR5 (level 4) and disables INTTR4.
T4FFCR	←	1 1 0 0 0 0 1 1	Disables trigger.
T4MOD	←	X X 1 0 0 1 0 0	Sets input clock to TI4.
TREG5	←	* * * * * * * *	Sets the count (16 bits).
		* * * * * * * *	
T16RUN	←	1 X X X - - - 1	Starts timer 4.

Note: Set the prescaler to RUN when using a 16-bit counter as an event counter.

(3) 16-bit programmable pulse generation (PPG) output mode

As timers 4, 6, 8 and A operate the same, the following describes timer 4 only.

To enter PPG mode, set the device to invert timer flip-flop TFF4 and output the TFF4 value from the TO4 pin (also used as PD0) at a match between up counter UC4 and the TREG4/TREG5 register value.

The following condition must be satisfied: (TREG4 setting) < (TREG5 setting).

		7	6	5	4	3	2	1	0	
T16RUN	←	-	X	X	X	-	-	-	0	Stops timer 4.
TREG4	←	*	*	*	*	*	*	*	*	Sets the duty. (16 bits)
		*	*	*	*	*	*	*	*	
TREG5	←	*	*	*	*	*	*	*	*	Sets the interval. (16 bits)
		*	*	*	*	*	*	*	*	
T16CR	←	X	X	X	X	-	-	-	1	Enables TREG4 double-buffer. (Duty/interval modified by interrupt INTTR5)
T4FFCR	←	1	1	0	0	1	1	1	0	Sets TFF4 to invert at detection of a match with TREG4 or TREG5. Sets TFF4 initial value to "0".
T4MOD	←	X	X	1	0	0	1	*	*	Sets the input clock to the internal clock, and disable the capture function.
		(** = 01, 10, 11)								
PDCR	←	-	-	-	-	-	-	-	1	Allocates PD0 to TO4.
PDFC	←	X	-	-	-	X	-	-	1	
T16RUN	←	1	X	X	X	-	-	-	1	Starts timer 4.

Note: X: Don't care –: No change

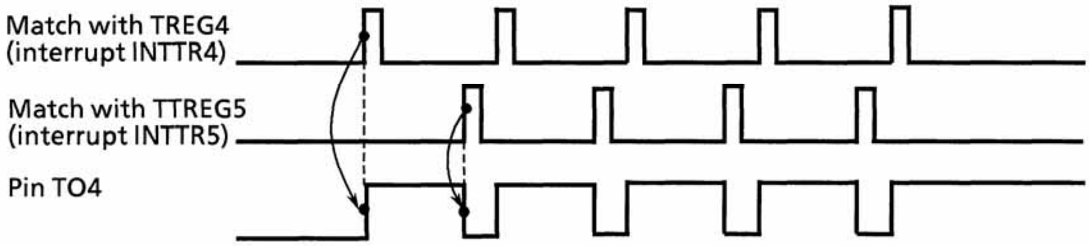


Figure 3.9.21 Programmable Pulse Generation (PPG) Output Waveform

Enabling the TREG4 double-buffer in this mode shifts the value of register buffer 4 to TREG4 when TREG5 matches UC4. Using the double-buffer facilitates output of waveforms with a low duty ratio.

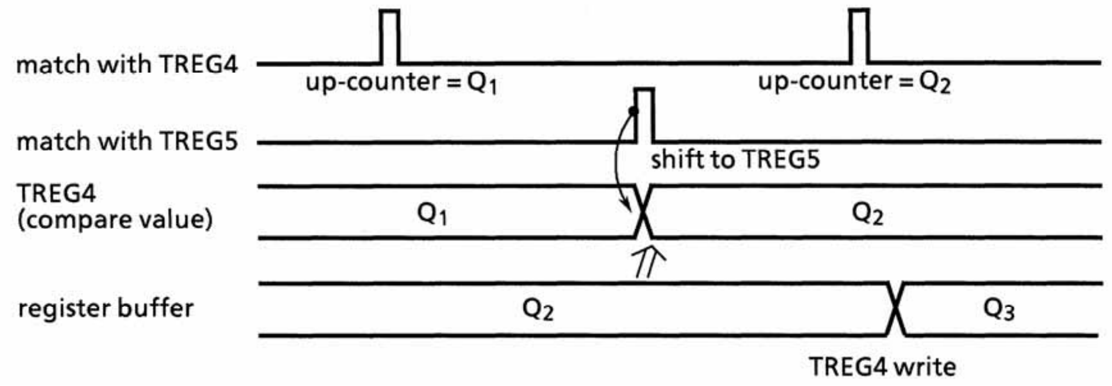


Figure 3.9.22 Register Buffer Operation

The following is a block diagram of this mode.

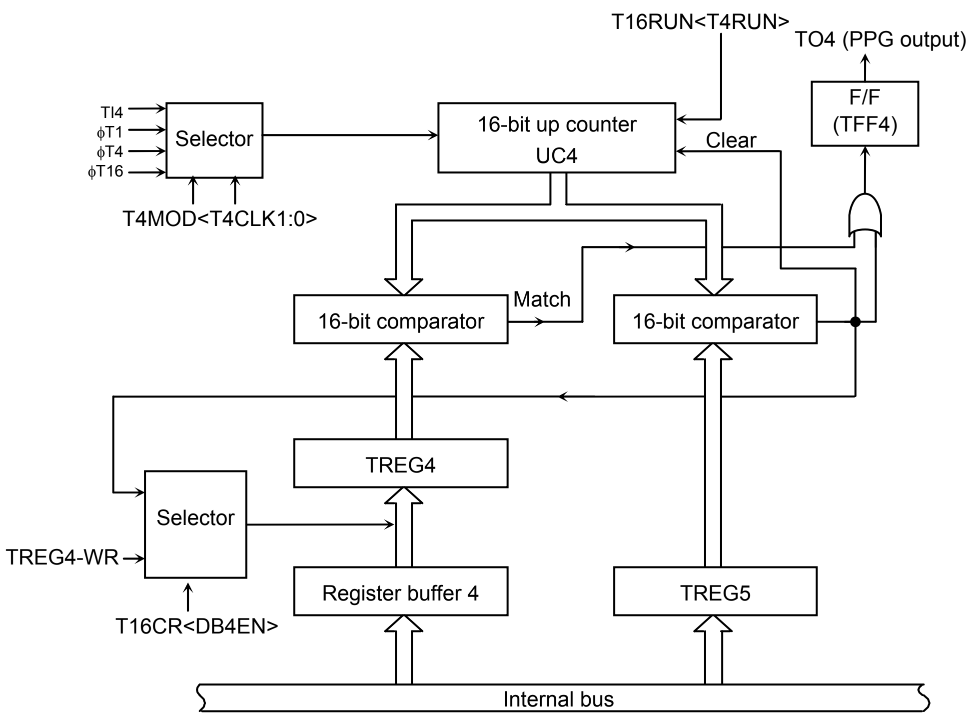


Figure 3.9.23 16-Bit PPG Mode Block Diagram

(4) Capture function application example

As timers 4, 6, 8 and A operate the same, the following describes timer 4 only.

The following features of the 16-bit timer can be enabled or disabled as required: loading of up counter UC4 value to capture registers CAP4 and CAP5, inversion of timer flip/flop TFF4 on a match detect signal from comparators CP4 and CP5, and outputting of TFF4 to pin TO4. Many functions can be obtained by combining these features with interrupts. For example:

[1] One-shot pulse output from the external trigger pulse

[2] Frequency measurement

[3] Pulse width measurement

[4] Time differential measurement

[1] One-shot pulse output from external trigger pulse

Set up counter UC4 to free-running using internal clock input. Input the external trigger pulse from pin TI4, and load the up counter value to capture register CAP4 on the TI4 input rising edge (Set T4MOD<CAP45M1:0> to "01").

On the TI4 input rising edge, add the value of capture register CAP4 at interrupt INT4 (c) to the delay time (d), and set timer register TREG4 to the sum of these values (c + d). Add the pulse width of the one-shot pulse (p) to TREG4, and set TREG5 to the result (c + d + p). On interrupt INT4, set register T4FFCR <EQ5T4, EQ4T4> to "enable the inversion of timer flip-flop TFF4 only when the up counter matches with TREG4 or TREG5". On interrupt INTTR5, disable the inversion of timer flip-flop TFF8.

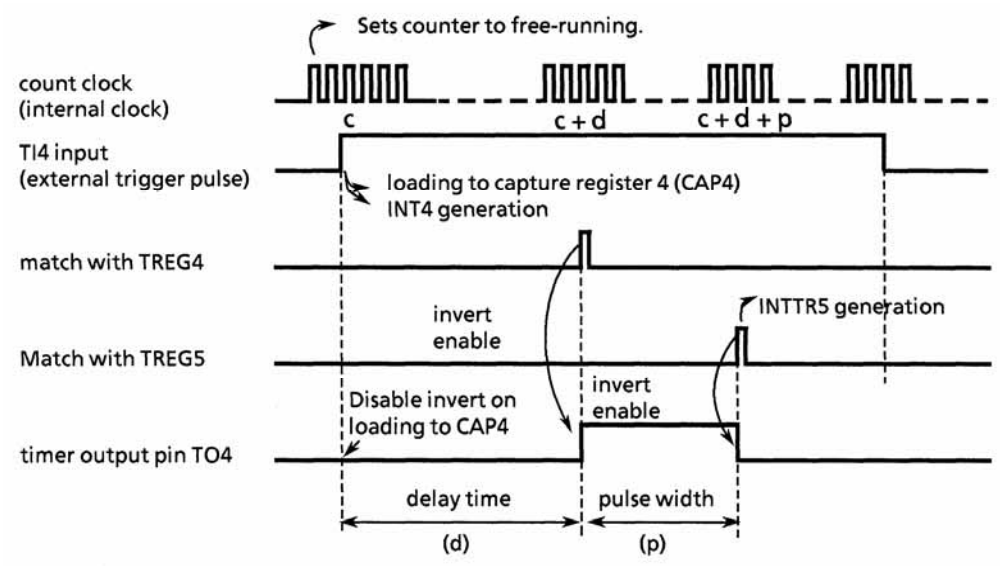


Figure 3.9.24 One-Shot Pulse Output (With delay)

Setting Example: On pin TI4, output a 2ms one-shot pulse with a 3ms-delay after an external trigger pulse.

Main setting

```
T4MOD   ←  X X 1 0 1 0 0 1      Sets to free-running.
                                Sets to count at φT1.
                                Loads to CAP4 at TI4 input rising edge.
T4FFCR  ←  1 1 0 0 0 0 1 0      Clears TFF4 to 0.
                                Disables TFF4 invert.
PDCR    ←  X - - - X - - 1
PDFC    ←  X - - - X - - 1      Allocates PD0 to TO4.
INTE45  ←  - - - - X 1 0 0      Enables INT4 and disables INTTR4 and INTTR5.
INTET45 ←  X 0 0 0 X 0 0 0
T16RUN  ←  1 X X X - - - 1      Starts timer 4.
```

Settings at INT4

```
TREG4   ←  CAP4+3ms/φT1
TREG5   ←  TREG4+2ms/φT1
T4FFCR  ←  - - - - 1 1 - -      Enables TFF4 invert on match with TREG4 or TREG5.
INTET45 ←  X 1 0 0 - - - -      Enables INTTR5.
```

Settings at INTTR5

```
T4FFCR  ←  - - - - 0 0 - -      Disables TFF4 invert on match with TREG4 or TREG5.
INTET45 ←  X 0 0 0 - - - -      Disables INTTR5.
```

Note: X: Don't care –: No change

If delay time is not required, invert timer flip-flop TFF4 by loading to capture register 4 (CAP4). Set timer register TREG5 to the sum of the one-shot pulse width (p) and the value of CAP4 at interrupt INT4 (c) (c + p). Enable TFF4 invert on match between TREG5 and up counter UC4. On interrupt INTTR5, disable the timer flip-flop TFF4 invert.

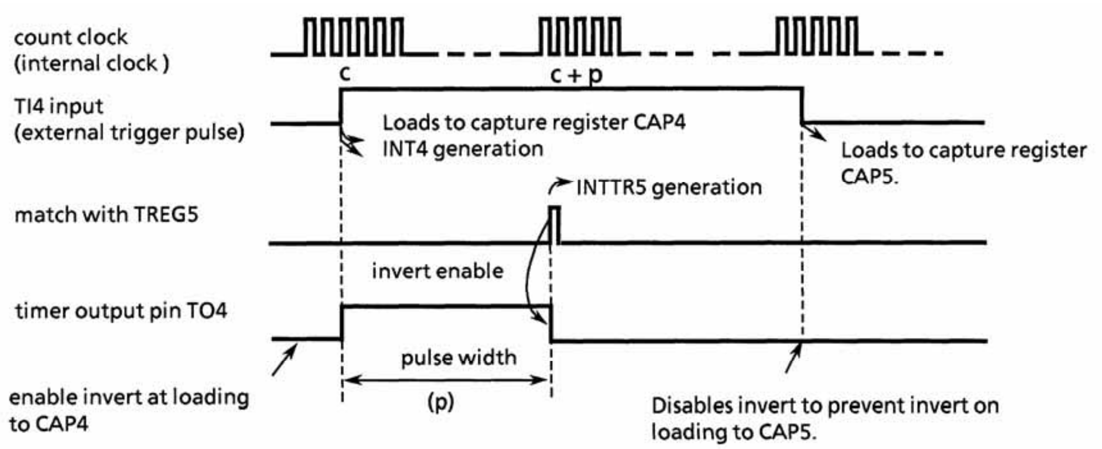


Figure 3.9.25 One-Shot Pulse Output (No Delay)

[2] Frequency measurement

This mode is used to measure the frequency of the external clock. Input the external clock on pin TI4 and measure its frequency with the 8-bit timers (Timers 0, 1) and the 16-bit timer/event counter (Timer 4).

Set the TI4 input as the timer 4 input clock, and load the value of up counter UC4 to capture register CAP4 when timer flip/flop TFF4 of the 8-bit timer (Timer 0, 1) rises, and to capture register CAP5 when timer flip/flop TFF4 falls.

The frequency is determined from the difference between capture registers CAP4 and CAP5 at the 8-bit timer interrupts (INTT0 or INTT1).

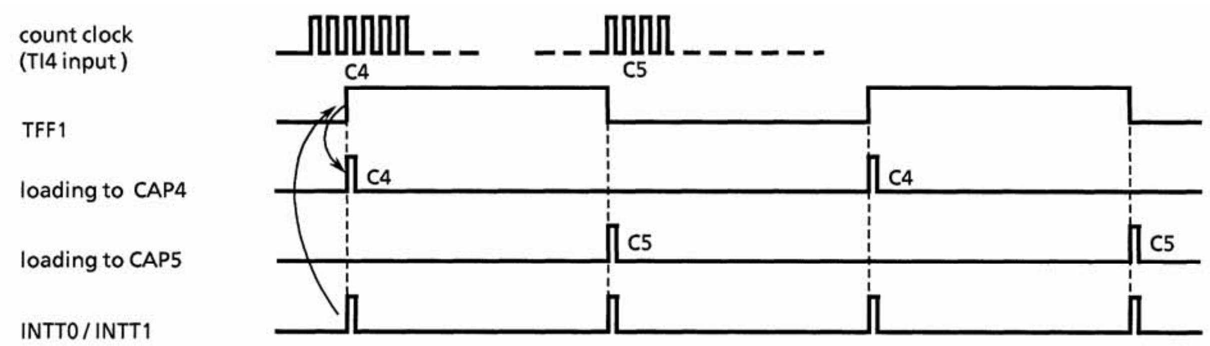


Figure 3.9.26 Frequency Measurement

For example, if TFF1 is set to “1” for 0.5 s by the 8-bit timers, and the difference between CAP4 and CAP5 is 100, the frequency is 100 ÷ 0.5 [s] = 200 [Hz].

[3] Pulse width measurement

This mode is used for measuring the “high” level width of an external pulse. Input the external pulse through pin TI4 and set the 16-bit timer/event counter to free-running count-up using an internal clock. Load the up counter UC4 value into capture register CAP4 and CAP5 on the rising and falling edge respectively of the external pulse. Interrupt INT4 is generated on the falling edge of pin TI4.

The pulse width can now be determined according to the difference between CAP4 and CAP5, and the internal clock interval.

For example, if the difference between CAP4 and CAP5 is 100 and the internal clock interval is 0.8 μs, the pulse width is 100 × 0.8 μs = 80 μs.

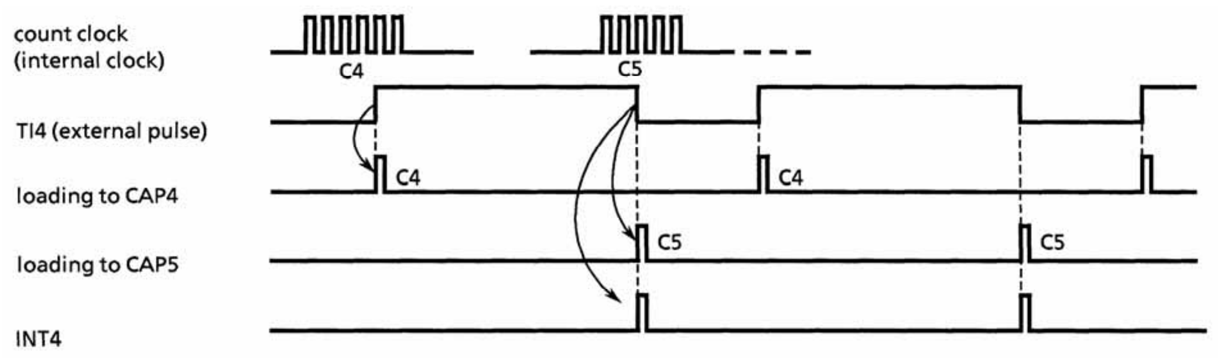


Figure 3.9.27 Pulse Width Measurement

Note: Only in pulse width measurement mode where T4MOD<CAP45M1, 0> = “10”, external interrupt INT4 is generated at the falling edge of pin TI4. In other modes, external interrupt INT4 is generated at the rising edge.

Determine the “low” level width at the second INT4 using the difference between the value of C5 at the first interrupt and the value of C4 at the second interrupt.

[4] Time differential measurement

This mode measures the time difference between the rising edge of the external pulses input to pins TI4 and TI5.

Set the 16-bit timer/event counter (Timer 4) to free-running count-up using an internal clock. When a rising edge is detected in the pulse on pin TI4, the up counter UC4 value is loaded into capture register CAP4 and interrupt INT4 is generated.

Similarly, when a rising edge is detected in the pulse on pin TI5, the up counter UC4 value is loaded into capture register CAP5 and interrupt INT5 is generated.

When the up counter values are loaded to CAP4 and CAP5, the time difference can be determined from the difference between CAP4 and CAP5.

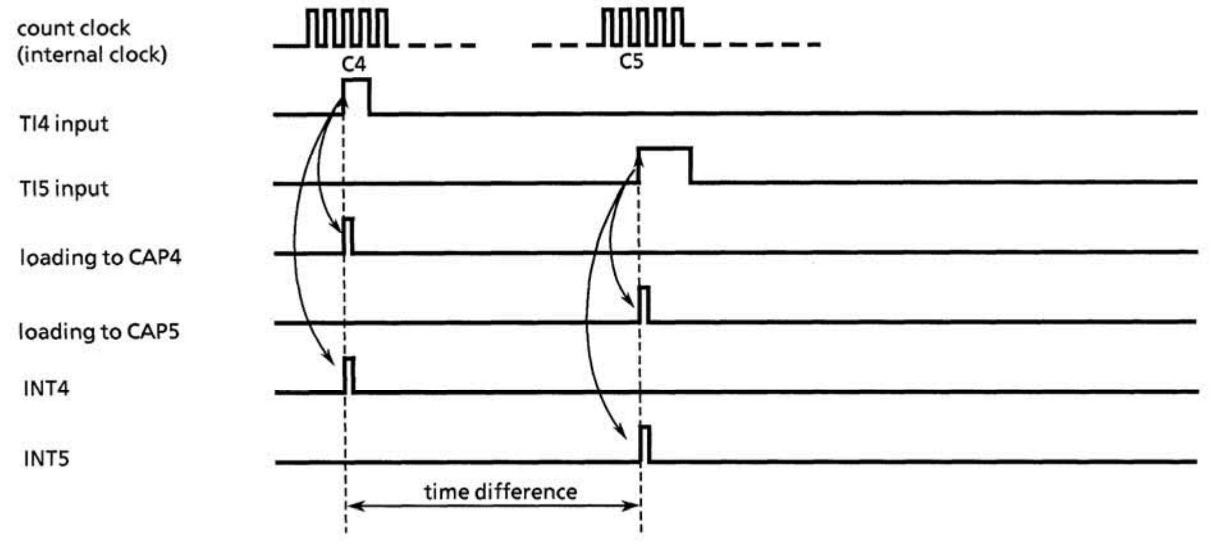


Figure 3.9.28 Time Differential Measurement

(5) Phase output mode

Set the up counter UC4/6/8/A to free-running and output a signal with any phase differential. As timers 4, 6, 8 and A operate the same, the following describes timer 6 only.

A match between up counter UC6 and TREG6 or TREG7 inverts TFF6 or TFF7 respectively, and outputs the invert values to TO6 and TO7 respectively.

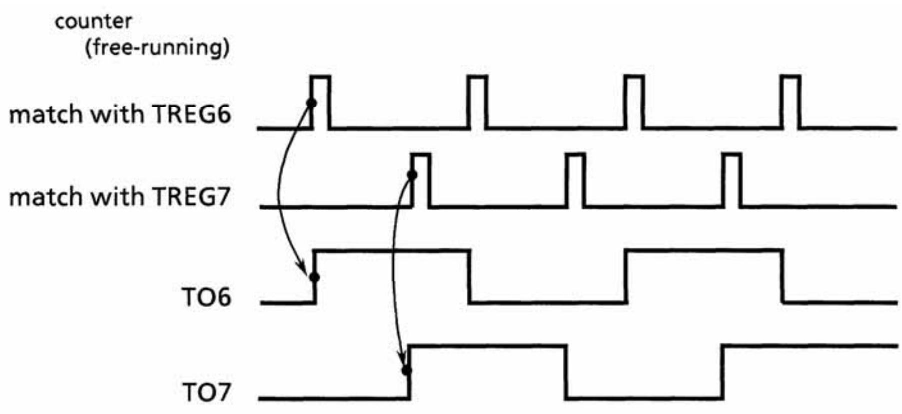


Figure 3.9.29 Phase Output

The following table shows the interval (Counter overflow time) of the above waveform output.

	16 MHz	20 MHz
ϕT1	32.77 ms	26.214 ms
ϕT4	131.07 ms	104.856 ms
ϕT16	524.29 ms	419.424 ms

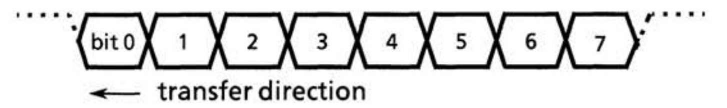
3.10 Serial Channel

TMP94C251A features two built-in serial input/output channels.
The serial channel operating modes are as follows:

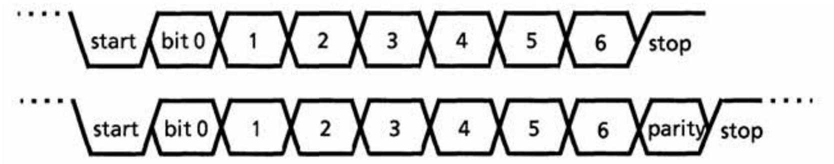
- I/O interface mode — Mode 0: For receiving and transmitting I/O data for I/O extension, and for receiving and transmitting synchronous I/O data signals (SCLK).
- Universal asynchronous receiver transmitter (UART) mode
  - Mode 1 : 7-bit transmit/receive data
  - Mode 2: 8-bit transmit/receive data
  - Mode 3: 9-bit transmit/receive data

Parity bits can be added in modes 1 and 2.
Figure 3.10.1 shows the data formats (for one frame) in each mode.

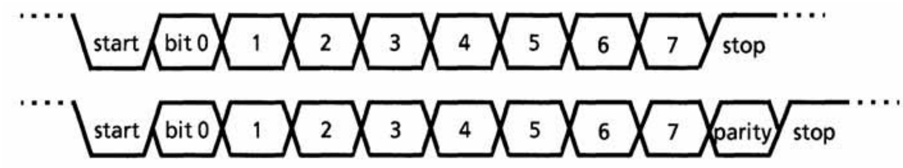
- Mode 0 (I/O interface mode)



- Mode 1 (7-bit UART mode)



- Mode 2 (8-bit UART mode)



- Mode 3 (9-bit UART mode)

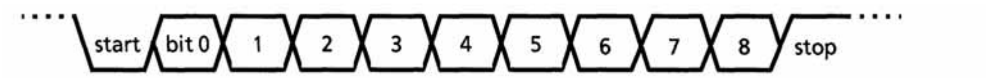


Figure 3.10.1 Data Formats

Serial channel buffer registers temporarily hold data to be transmitted or received (full-duplex), allowing independent transmission and reception.

Note that in I/O interface mode, the serial clock (SCLK) is shared between reception and transmission (half-duplex).

The buffer register for reception features a double-buffer configuration to prevent overrun error; an extra frame holds data until the data are read by the CPU. That is, a receive buffer holds the data already received, while the buffer register receives the next frame of data.

By using $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ (as no $\overline{\text{RTS}}$ pin is provided, a pin in any port must be controlled by software), it is possible to halt data transmission until the CPU reads the data received after each frame (handshake function).

In UART mode, a check function prevents data receive operations from starting due to erroneous start bits being generated by noise or other interference on the line. The channel starts receiving data only when the start bit is detected as normal in at least two of three samplings.

When the transmit buffer is empty, an INTTX interrupt is generated to request the CPU to supply the next data to transmit. When the receive buffer has data to be read by the CPU, an INTRX interrupt is generated.

When an overrun error, parity error, or framing error is detected at data reception, the corresponding flag <OERR, PERR, FERR> is set in the control register (SC0CR/SC1CR) of the relevant serial channel.

Serial channels 0 and 1 have a dedicated baud rate generator, which can set any baud rate by dividing the frequency of internal input clocks (ϕT0, ϕT2, ϕT8, and ϕT32) from the 9-bit prescaler (shared with 8/16 bit timers) by a value between 1 and 16.

In addition to the clock from the internal baud rate generator, an arbitrary baud rate can be obtained from the external clock input (SCLK0/1). Moreover, in I/O interface mode, a sync signal (SCLK0/1) can be input and data transfer performed using this external clock.

3.10.1 Control Registers

Each serial channel is controlled by three control registers (SC0CR, SC0MOD, and BR0CR for channel 0). Transmit/receive data are stored in a register in each channel (SC0BUF for channel 0).

SC0MOD
(00D2H)

	7	6	5	4	3	2	1	0
bit Symbol	TB8	CTSE	RXE	–	SM1	SM0	SC1	SC0
Read/Write	R/W			R/W	R/W			
After reset	Undefined	0	0	0	0	0	0	0
Function	Transmit data bit 8	Handshake function control 0: CTS disable 1: CTS enable	Receive control 0: receive disable 1: receive enable	Fixed to "0"	Serial transfer mode 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial transfer clock (UART) 00: TO2 trigger 01: baud rate generator 10: internal clock ϕ1 11: external clock (SCLK0)	

Serial transfer clock (UART)

00	Timer 2 match detect signal
01	Baud rate generator
10	Internal clock ϕ1
11	External clock (SCLK0)

Serial transfer mode

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

Receive control

0	Receive disable
1	Receive enable

Handshake function ($\overline{CTS}$ pin)

0	Disable (transmission always enabled)
1	Enable

Transmit data bit 8

Note: SC1MOD (D6H) is provided for channel 1.

Figure 3.10.2 Serial Mode Control Register (SC0MOD, Channel 0)

SC0CR (00D1H)

	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	Receive data bit 8	Parity 0: Odd 1: Even	Parity addition 0: disable 1: enable	1: Error Overrun	Parity	Framing	0: SCLK0 1: SCLK0	0: Baud rate generator 1: SCLK0 pin input

I/O interface input clock select

0	Baud rate generator
1	SCLK0 pin input

Edge select in SCLK0 input operations

0	Data transmit/receive ( ) at SCLK0 rising edge
1	Data transmit/receive ( ) at SCLK0 falling edge

Framing error flag
Parity error flag
Overrun error flag

Cleared to "0" when read.

Parity addition enable

0	Disable
1	Enable

Even parity addition/check

0	Odd parity
1	Even parity

Receive data bit 8

Notes: SC1CR (D5H) is provided for channel 1.
As the error flags are all cleared after reading, when testing with a bit test instruction, test more than just a single bit.

Figure 3.10.3 Serial Control Register (SC0CR, Channel 0)

BR0CR (00D3H)

	7	6	5	4	3	2	1	0
bit Symbol	–	–	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write			R/W					
After reset	0	–	0	0	0	0	0	0
Function	Fixed to "0".		00: ϕT0 (4/fc) 01: ϕT2 (16/fc) 10: ϕT8 (64/fc) 11: ϕT32 (256/fc)		Sets divisor.			

Sets baud rate generator divisor.

0000	Divide by 16
0001	Divide by 1 (no division)
0010 ∫ 1111	Divide by 2 to 15

Baud rate generator input clock select

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

Notes: BR1CR (D7H) is provided for channel 1.
To use the baud rate generator, set T16RUN<PRRUN> to "1" and run the prescaler.
The baud rate generator frequency can be divided by 1 in UART mode only. Do not use this setting in I/O interface mode.

Figure 3.10.4 Baud Rate Generator Control Register (BR0CR, Channel 0)

SC0BUF (D0H)

7	6	5	4	3	2	1	0	
TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(transmit)

7	6	5	4	3	2	1	0	
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(receive)

Note)Read-modify-write is prohibited.

Figure 3.10.5 Serial Transmit/Receive Register (SC0BUF, Channel 0)

SC1MOD (00D6H)

	7	6	5	4	3	2	1	0
bit Symbol	TB8	CTSE	RXE	–	SM1	SM0	SC1	SC0
Read/Write	R/W			R/W	R/W			
After reset	Undefined	0	0	0	0	0	0	0
Function	Transmit data bit 8	Handshake function control 0: CTS disable 1: CTS enable	Receive control 0: receive disable 1: receive enable	Fixed to “0”	Serial transfer mode 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial transfer clock (UART) 00: TO2 trigger 01: baud rate generator 10: internal clock φ1 11: external clock SCLK1	

Serial transfer clock (UART)

00	Timer 2 match detect signal
01	Baud rate generator
10	Internal clock φ1
11	External clock SCLK1

Serial transfer mode

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

Receive control

0	Receive disable
1	Receive enable

Handshake function ($\overline{\text{CTS}}$ pin)

0	Disable (transmission always enabled)
1	Enable

Transmit data bit 8

Figure 3.10.6 Serial Mode Control Register (SC1MOD, Channel 1)

SC1CR (00D5H)

	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	Receive data bit 8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: Error Overrun	Parity	Framing	0: SCLK1 ( ) 1: SCLK1 ( )	0: Baud rate generator 1: SCLK1 pin input

I/O interface input clock select

0	Baud rate generator
1	SCLK1 pin input

Edge select in SCLK1 input operations

0	Data transmit/receive ( ) at SCLK1 rising edge
1	Data transmit/receive ( ) at SCLK1 falling edge

Framing error flag
Parity error flag
Overrun error flag
} Cleared to "0" when read

Parity addition enable

0	Disable
1	Enable

Even parity addition/check

0	Odd parity
1	Even parity

Receive data bit 8

Note: As the error flags are all cleared after reading, when testing with a bit test instruction, test more than just a single bit.

Figure 3.10.7 Serial Control Register (SC1CR, Channel 1)

BR1CR (00D7H)

	7	6	5	4	3	2	1	0
bit Symbol	–	–	BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
Read/Write			R/W					
After reset	0	–	0	0	0	0	0	0
Function	Fixed to "0".		00: ϕT0 (4/fc) 01: ϕT2 (16/fc) 10: ϕT8 (64/fc) 11: ϕT32 (256/fc)		Sets divisor.			

Baud rate generator divisor setting

0000	Divide by 16
0001	Divide by 1 (no division)
0010 ～ 1111	Divide by 2 to 15

Baud rate generator 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)

Notes: To use the baud rate generator, set T16RUN<PRRUN> to "1" and run the prescaler. The baud rate generator frequency can be divided by 1 in UART mode only. Do not use this setting in I/O interface mode.

Figure 3.10.8 Baud Rate Generator Control Register (BR1CR, Channel 1)

SC1BUF (00D4H)

7	6	5	4	3	2	1	0	
TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(transmit)

7	6	5	4	3	2	1	0	
RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(receive)

Note)Read-modify-write is prohibited.

Figure 3.10.9 Serial Transmit/Receive Buffer Register (SC1BUF, Channel 1)

PFFC (003FH)

	7	6	5	4	3	2	1	0
bit Symbol	–	PF6F	PF5F	PF4F	–	PF2F	PF1F	PF0F
Read/Write		W				W		
After reset	–	0	0	0	–	0	0	0
Function		0: PORT 1: $\overline{CTS1}$ SCLK1	0: PORT 1: RxD1	0: PORT 1: TxD1		0: PORT 1: $\overline{CTS0}$ SCLK0	0: PORT 1: RxD0	0: PORT 1: TxD0

Sets PF0 to TxD0 output

0	Port input / output
1	TxD0 Output

Sets PF1 to RxD0 input

0	Port input / output
1	RxD0 Input

Sets PF2 to $\overline{CTS0}$ input or SCLK0 input / output

0	Port input / output
1	$\overline{CTS0}$, SCLK0 output

Sets PF4 to TxD1 output

0	Port input / output
1	TxD1 Output

Sets PF5 to RxD1 input

0	Port input / output
1	RxD1 Input

Sets PF6 to $\overline{CTS1}$ input or SCLK1 input / output

0	Port input / output
1	$\overline{CTS1}$, SCLK1 output

Figure 3.10.10 Port F Function Register (PFFC)

3.10.2 Configuration

Figure 3.10.11 is a block diagram of serial channel 0. Serial channel 1 has the same circuit configuration.

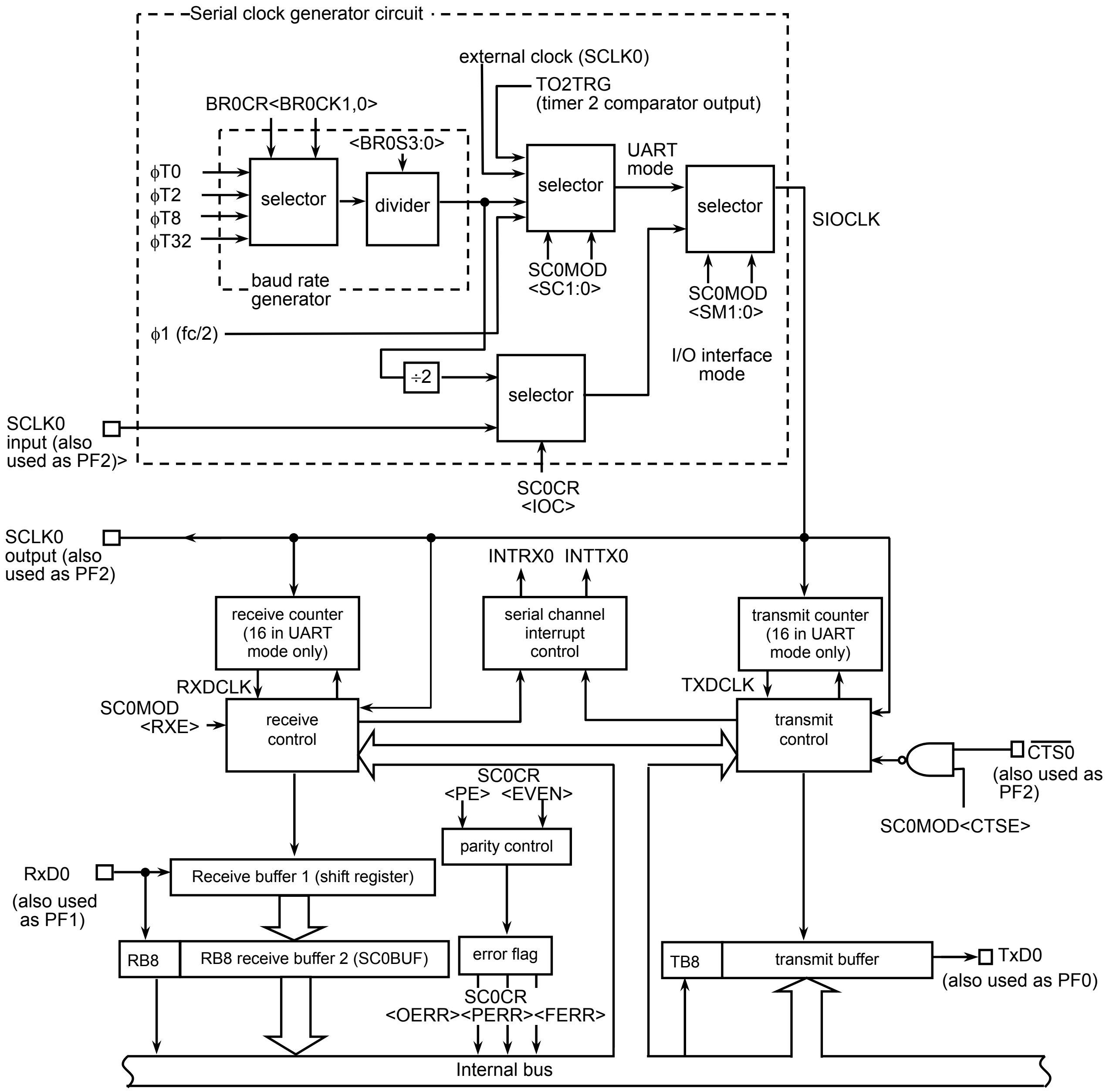


Figure 3.10.11 Serial Channel 0 Block Diagram

[1] Baud rate generator

The baud rate generator is a circuit to generate the transmission clock signals that control the serial channel transmission rate.

The baud rate generator input clock is one of $\phi T0$ (4/fc), $\phi T2$ (16/fc), $\phi T8$ (64/fc), or $\phi T32$ (256/fc) from the 9-bit prescaler that the baud rate generator shares with the timers.

Bits 5 and 4 <BR0CK1:0>/<BR1CK1:0> of the baud rate generator control register (BR0CR/BR1CR) select the input clock.

The baud rate generator features a built-in 4-bit divider. Set the transmission rate by dividing the frequency by 1 to 16 using the divider.

Baud rates using the baud rate generator are determined as follows:

- UART mode

$$\text{Baud Rate} = \frac{\text{Baud rate generator input clock}}{\text{Baud rate generator divisor}} \div 16$$

- I/O interface mode

$$\text{Baud Rate} = \frac{\text{Baud rate generator input clock}}{\text{Baud rate generator divisor}} \div 2$$

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

$\phi T0$ = 4/fc

$\phi T2$ = 16/fc

$\phi T8$ = 64/fc

$\phi T32$ = 256/fc

Accordingly, with the source clock set to 19.6608 MHz, when $\phi T2$ (16/fc) is selected as input clock and the divisor is 8, the baud rate in UART mode is:

$$\text{Baud Rate} = \frac{\text{fc}/16}{8} \div 16$$
$$= 19.6608 \times 10^6 \div 16 \div 8 \div 16 = 9600 \text{ (bps)}$$

Table 3.10.1 shows examples of the baud rates in UART mode.

In UART mode, the serial channels use 8-bit timer 2 to obtain the baud rate. Table 3.10.2 shows examples of baud rates using timer 2.

Moreover, the external clock input can also be used as the serial clock. The baud rate in this case is determined as follows.

Baud rate = External clock input ÷ 16

Table 3.10.1 UART Mode Baud Rate Selection (1) (Using baud rate generator)

Unit: Kbps

fc [MHz]	Input Clock / Divisor	ϕT0 (4/fc)	ϕT2 (16/fc)	ϕT8 (64/fc)	ϕT32 (256/fc)
18.432000	15	19.2000	4.800	1.200	0.300
19.660800	8	38.400	9.600	2.400	0.600
↑	16	19.200	4.800	1.200	0.300

Note: In I/O interface mode, the transmission rate is eight times the values shown in this table.

Table 3.10.2 UART Mode Baud Rate Selection (2) (Using timer 2 input clock ϕT1)

Unit: Kbps

TREG2 / fc	20 MHz	19.6608 MHz	16 MHz
01H			
02H		76.8	62.5
03H			
04H		38.4	31.25
05H	31.25		
06H			
08H		19.2	
0CH			
10H		9.6	

Baud rate calculation (using timer 2):

$$\text{Transmission rate:} = \frac{\text{fc}}{\text{TREG2} \times \underline{8} \times 16}$$

(Where timer 2 input clock is ϕT1)

Input clocks for timer 0

ϕT1 = 8/fc

ϕT4 = 32/fc

ϕT16 = 128/fc

Note: In I/O interface mode, the timer 2 match signal cannot be used as a transmission clock.

[2] Serial clock generator circuit

This circuit generates the transmit/receive basic clock.

- In I/O interface mode

  In SCLK output mode where SC0CR/SC1CR<IOC> is set to "0", the basic clock (SIOCLK) is generated by dividing the output of the baud rate generator by 2.

  In SCLK input mode where SC0CR/SC1CR<IOC> is set to "1" the basic clock is derived from the rising or falling edge of the SCLK input, as determined by the setting of the SC0CR/SC1CR<SCLKS> register.

- In universal asynchronous receiver transmitter (UART) mode

  Basic clock SIOCLK is selected from one of the following depending on the setting of the <SC1:0> bits of the SC0MOD or SC1MOD register: the clock from the baud rate generator, internal clock $\phi 1$ (500K bps at fc = 16 MHz), a match detect signal from timer 2, or an external clock.

[3] Receive counter

The receive counter is a 4-bit binary counter that counts by the SIOCLK clock and is used in universal asynchronous receiver transmitter (UART) mode. Sixteen cycles of SIOCLK are used to receive one bit of data. The data are sampled three times: at the 7th, 8th, and 9th clock cycles.

The data received are checked by the majority rule applied to the three samples.

For example, if the sampled data bits are 1, 0, 1 at the 7th, 8th, and 9th clock cycles respectively, the data are determined as "1". If the samplings are 0, 0, 1, the data received are determined as "0".

[4] Receive control section

- In I/O interface mode

  In SCLK output mode where SC0CR/SC1CR<IOC> is set to "0", the RxD0/1 pin is sampled at the rising edge of the shift clock output on the SCLK0/1 pin.

  In SCLK input mode where SC0CR/SC1CR<IOC> is set to "1", the RxD0/1 pin is sampled at the rising or falling edge of SCLK input as determined by the setting of the SC0CR/SC1CR<SCLKS> register.

- In universal asynchronous receiver transmitter (UART) mode

  The receive control section has a circuit for detecting the start bit by the majority rule. If two or more 0s are detected among three samples, the circuit recognizes the bit as a start bit and begins receiving. Data being received are also checked by the majority rule.

[5] Receive buffer

The receive buffer has a double-buffer configuration to prevent overrun error. Receive buffer 1 (a shift register buffer) stores the data received bit by bit. When the receive buffer contains seven or eight bits of data, the data are transferred to receive buffer 2 (SC0BUF/SC1BUF), generating interrupt INTRX0/INTRX1.

The CPU reads only receive buffer 2 (SC0BUF/SC1BUF). Data can be stored in receive buffer 1 even before the CPU reads receive buffer 2.

However, receive buffer 2 must be read before all bits of the next data unit are received by buffer 1. Otherwise, an overrun error occurs and the contents of receive buffer 1 are lost, although the contents of receive buffer 2 and SC0CR<RB8>/SC1CR<RB8> are preserved. Reading receive buffer 2 (SC0BUF/SC1BUF) clears interrupt request flags INTRX0<IRX0C> and INTRX1<IRX1C>.

In 8-bit UART mode with parity added, the parity bit is stored in SC0CR<RB8>/SC1CR <RB8>. In 9-bit UART mode, the MSB is stored in SC0CR<RB8>/SC1CR<RB8>.

[6] Transmit counter

The transmit counter is a 4-bit binary counter for use in universal asynchronous receiver transmitter (UART) mode. Like the receive counter, the transmit counter counts by the SIOCLK clock, generating transmission clock TxDCLK every 16 clock cycles.

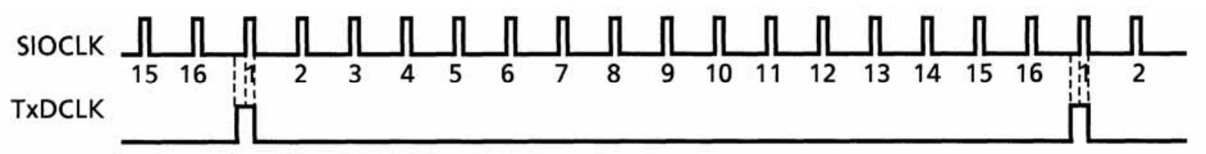


Figure 3.10.12 Transmission Clock Generation

[7] Transmit control section

- In I/O interface mode

In SCLK output mode where SC0CR/SC1CR<IOC> is set to “0”, the data in the transmit buffer is output bit by bit to the TxD0/1 pin at the rising edge of the shift clock output on the SCLK0/1 pin.

In SCLK input mode where SC0CR/SC1CR<IOC> is set to “1”, the data in the transmit buffer is output bit by bit to the TxD0/1 pin at the rising or falling edge of SCLK input as determined by the setting of the SC0CR/SC1CR<SCLKS> register.

- In universal asynchronous receiver transmitter (UART) mode

  When the CPU writes data in the transmit buffer, transmission begins from the next rising edge of the TxDCLK, generating transmission shift clock TxDSFT.

Handshake Function

The serial channels use the $\overline{\text{CTS}}$ pin to transmit data in units of frames, thus preventing an overrun error. Use SC0MOD/SC1MOD<CTSE> to enable or disable the handshake function.

When $\overline{\text{CTS}}$ goes high, data transmission is halted after the completion of the current transmission and is not restarted until $\overline{\text{CTS}}$ returns to low. An INTTX interrupt is generated to request the CPU for the next data to transmit. When the CPU writes the data to the transmit buffer, processing enters standby mode.

An $\overline{\text{RTS}}$ pin is not provided, but a handshake function can easily be configured if the receiver sets any port assigned to the RTS function to high (in the receive interrupt routine) after data receive, and requests the transmitter to temporarily halt transmission.

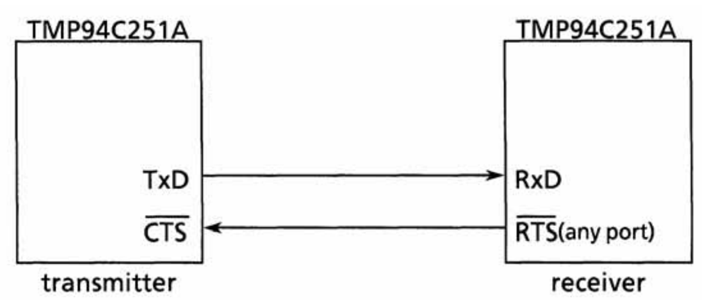
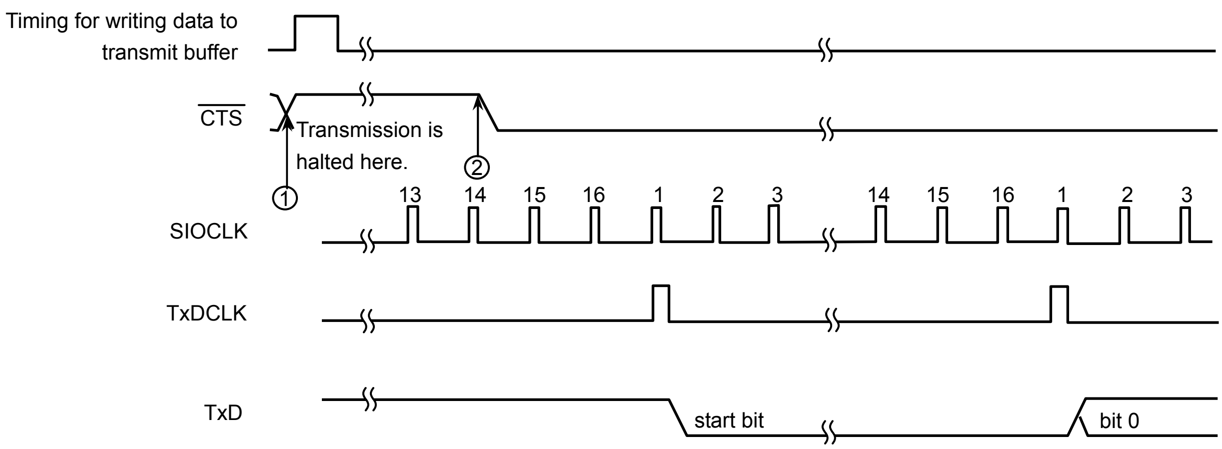


Figure 3.10.13 Handshake Function



Note 1: When the $\overline{\text{CTS}}$ signal rises during transmission, transmission of the next data frame halts after transmission of the current data frame is complete.

Note 2: Transmission begins at the first TxDCLK clock falling edge after the $\overline{\text{CTS}}$ signal falls.

Figure 3.10.14 $\overline{\text{CTS}}$ (Clear to send) Signal Timing

[8] Transmit buffer

Transmit buffer (SC0BUF/SC1BUF) shifts out and transmits the transmit data written by the CPU, beginning with the least significant bit, using the transmission shift clock (TxDSFT) generated by the transmission control section. When all bits are shifted out, the empty transmit buffer generates interrupt INTTX0/INTTX1.

[9] Parity control circuit

When serial channel control register SC0CR<PE>/SC1CR<PE> is set to “1”, data are transmitted and received with parity. However, parity can be added only in 7-bit or 8-bit UART mode. The SC0CR<EVEN>/SC1CR<EVEN> register selects even/odd parity.

At transmission, the parity control circuit automatically generates parity according to the data written in the transmit buffer (SC0BUF/SC1BUF). In 7-bit UART mode, the parity bit is stored in SC0BUF<TB7>/SC1BUF<TB7> prior to transmission. In 8-bit UART mode, parity is stored in SC0MOD<TB8>/ SC1MOD<TB8> prior to transmission. Set both <PE> and <EVEN> before writing the transmit data in the transmit buffer.

At receiving, data are first shifted into receive buffer 1. The parity control circuit automatically generates parity according to the data transferred to receive buffer 2 (SC0BUF/SC1BUF). In 7-bit UART mode, the generated parity is compared with the received parity in SC0BUF<RB7>/SC1BUF<RB7>. In 8-bit UART mode, the generated parity is compared with the received parity in SC0CR<RB8>/SC1CR<RB8>. If the parities differ, a parity error occurs and the SC0CR<PERR>/SC1CR<PERR> flag is set.

[10] Error flags

Three error flags improve the reliability of data reception.

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/SC1BUF), an overrun error occurs.

2. Parity error <PERR>

   The parity generated according to the data shifted into receive buffer 2 (SC0BUF/SC1BUF) is compared with the parity bit received from the RxD pin. If the parities are not equal, a parity error occurs.

3. Framing error <FERR>

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

[11] Signal generation timing

1) In UART mode

Receive

Mode	9 Bit	8 Bit + Parity	8 Bit, 7 Bit + Parity, 7 Bit
Interrupt generation timing	Center of last bit (bit 8)	Center of last bit (parity bit)	Center of stop bit
Framing error generation timing	Center of stop bit	Center of stop bit	Center of stop bit
Parity error generation timing	————	Center of last bit (parity bit)	←
Overrun error generation timing	Center of last bit (bit 8)	Center of last bit (parity bit)	Center of stop bit

Note: In 9-bit and 8-bit + Parity mode, interrupts coincide with the ninth bit pulse. Thus, when servicing the interrupt, it is necessary to wait for a 1-bit period (to allow the stop bit to be transferred) to allow checking for a framing error.

Transmit

Mode	9 Bit	8 Bit + Parity	8 Bit, 7 Bit + Parity, 7 Bit
Interrupt generation timing	Immediately before stop bit is sent	←	←

2) In I/O interface mode

Transmission interrupt generation timing	SCLK output mode	Immediately after rise of last SCLK signal (See Figure 3.10 (17))
	SCLK input mode	Immediately after rise of last SCLK signal (rising mode), immediately after fall in falling mode (See Figure 3.10 (18))
Receive interrupt generation timing	SCLK output mode	When received data are transferred to receive buffer 2 (SC0BUF/SC1BUF) (immediately after final SCLK) (See Figure 3.10 (19))
	SCLK input mode	When received data are transferred to receive buffer 2 (SC0BUF/SC1BUF) (immediately after final SCLK) (See Figure 3.10 (20))

3.10.3 Operation

(1) Mode 0 (I/O interface mode)

This mode is used to increase the number of I/O pins for transmitting or receiving data to an external shift register or other external destinations.

This mode consists of SCLK output mode for outputting a synchronous clock (SCLK), and SCLK input mode for inputting a synchronous clock (SCLK) from an external source.

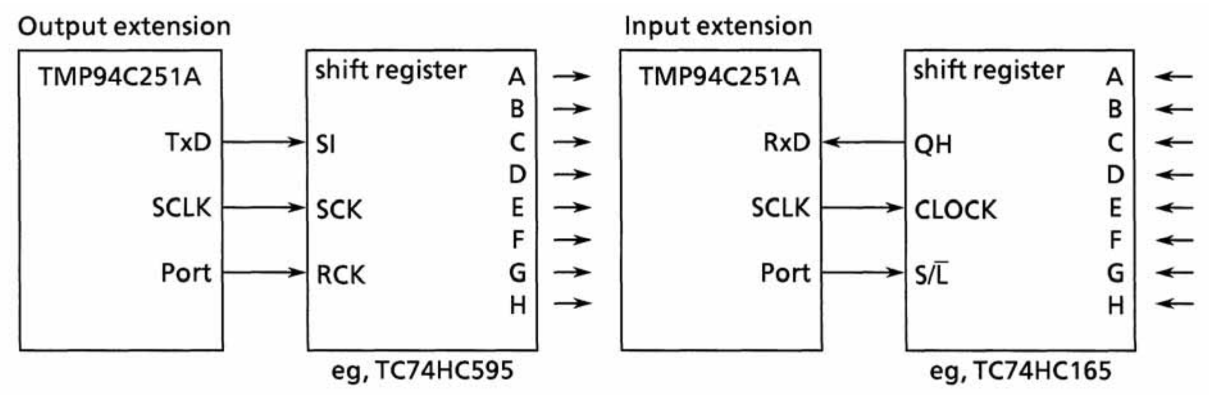


Figure 3.10.15 Example of SCLK Output Mode Connection

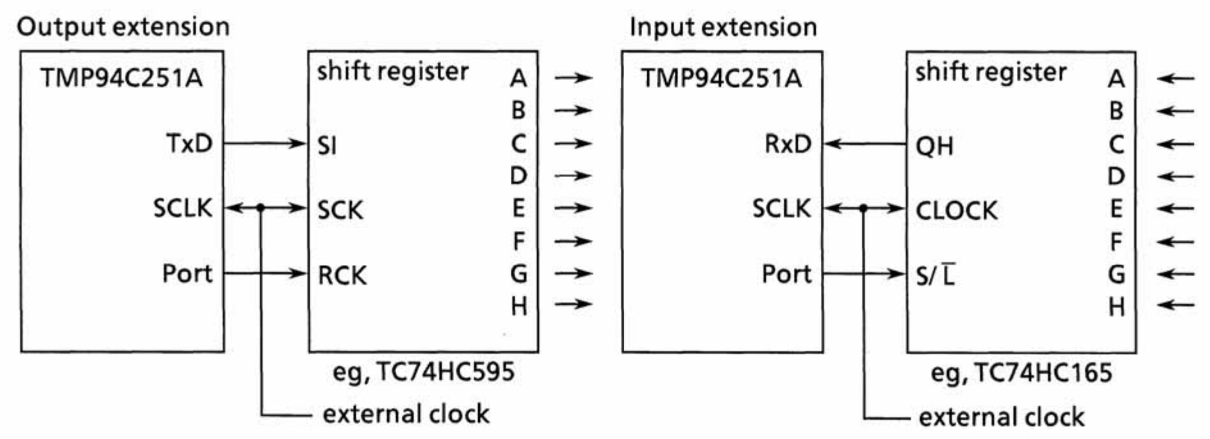


Figure 3.10.16 Example of SCLK Input Mode Connection

[1] Transmission (Example: Ch1)

In SCLK output mode, each time the CPU transmits data to the transmit buffer, eight data bits are output from the TxD0/1 pin, and a synchronous clock signal is output from the SCLK0/1 pin. When all data are output, INTES0<ITX0C>/INTES1<ITX1C> is set, generating interrupt INTTX0/1.

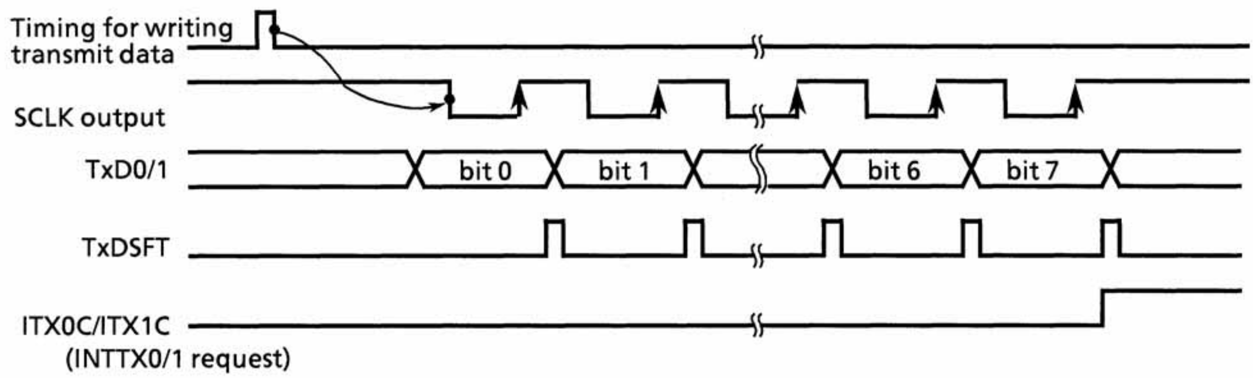


Figure 3.10.17 Data Transmission in I/O Interface Mode (SCLK output mode)

In SCLK input mode, 8-bit data are output from TxD0/1 pin when SCLK input becomes active while data are written in the transmission buffer by CPU.

When all data are output, INTES0<ITX0C>/INTES1<ITX1C> is set, generating interrupt INTTX0/1.

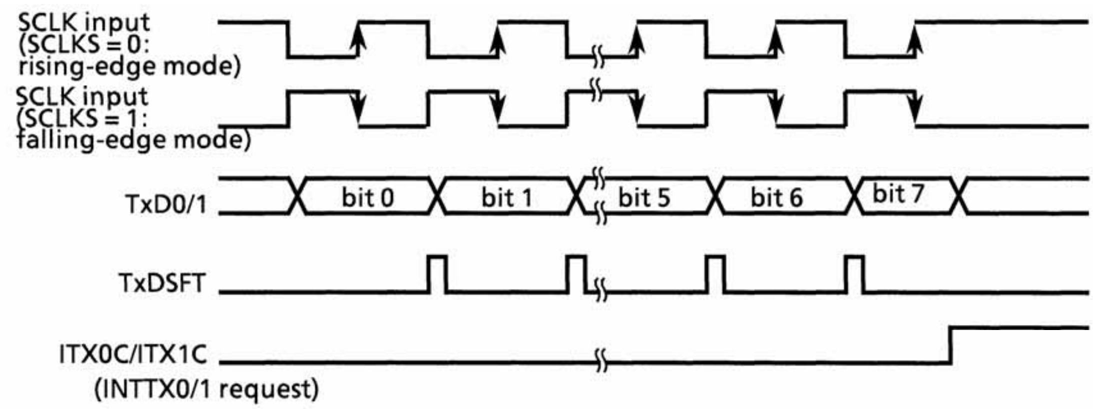


Figure 3.10.18 Data Transmission in I/O Interface Mode (SCLK input mode)

[2] Receiving (Example: Ch1)

In SCLK output mode, whenever the CPU reads the received data and clears the receive interrupt flag INTES0<IRX0C>/INTES1<IRX1C>, a synchronous clock is output from the SCLK0/1 pin and the next data frame is shifted to receive buffer 1. When an 8-bit data frame has been received, it is transferred to receive buffer 2 (SC0BUF/SC1BUF), and INTES0<IRX0C>/INTES1<IRX1C> is set again, generating interrupt INTRX0/1.

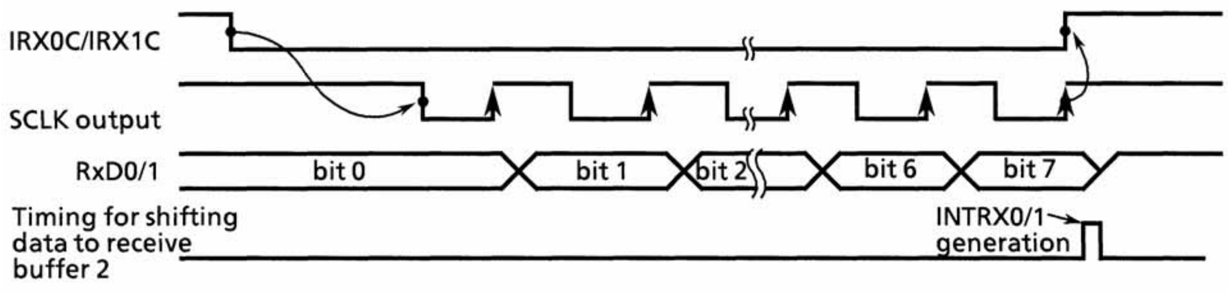


Figure 3.10.19 Data Receive in I/O Interface Mode (SCLK output mode)

In SCLK input mode, if SCLK is input after the CPU reads the received data and clears the receive interrupt flag INTES0<IRX0C>/INTES1<IRX1C>, the next data frame is shifted into receive buffer 1. When an 8-bit data frame is received, the data are shifted to receive buffer 2 (SC0BUF/SC1BUF) and INTES0<IRX0C>/ INTES1<IRX1C> is set again, generating interrupt INTRX0/1.

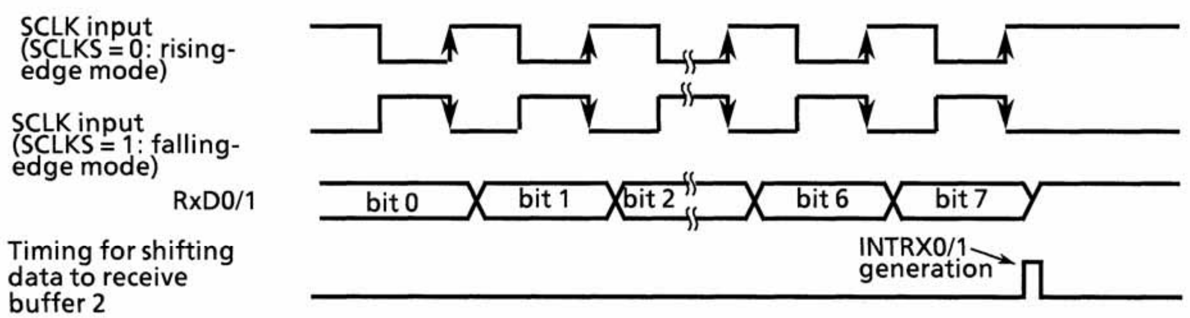


Figure 3.10.20 Data Receive in I/O Interface Mode (SCLK input mode)

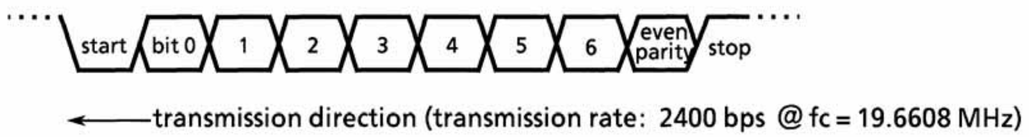
Note: To receive data in either SCLK input mode or SCLK output mode, first enable receive (SC0MOD/SC1MOD<RXE> = “1”).

(2) Mode 1 (7-bit UART mode)

Setting the serial channel mode register SC0MOD<SM1:0>/SC1MOD<SM1:0> to "01" specifies 7-bit UART mode.

A parity bit can be added in this mode. Enable or disable the addition of a parity bit by the serial channel control register SC0CR<PE>/SC1CR<PE> bit. With <PE> set to "1" (Parity enabled), select even or odd parity using SC0CR<EVEN>/SC1CR<EVEN>.

Example: When data are transmitted in the following format, the control registers are set as follows. The example shows channel 0.



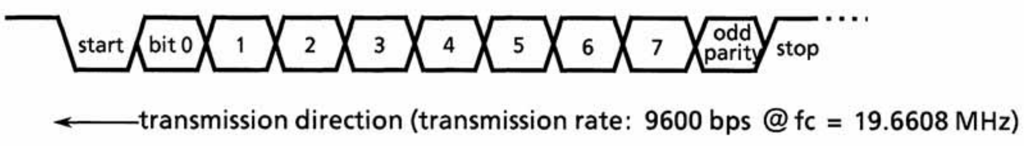
		7	6	5	4	3	2	1	0	
PFCR	←	–	–	–	–	–	–	–	1	Sets PF0 as TxD0 pin.
PFFC	←	–	–	X	–	–	–	X	1	
SC0MOD	←	X	0	–	0	0	1	0	1	Sets 7-bit UART mode.
SC0CR	←	X	1	1	X	X	X	0	0	Adds even parity.
BR0CR	←	0	X	1	0	1	0	0	0	Sets transmission rate to 2400 bps.
T16RUN	←	1	X	–	–	–	–	–	–	Starts prescaler for baud rate generator.
INTES0	←	X	1	0	0	–	–	–	–	Enables interrupt INTTX0 and sets interrupt level 4.
SC0BUF	←	*	*	*	*	*	*	*	*	Sets transmit data.

Note : X; Don't care –; No change

(3) Mode 2 (8-bit UART mode)

Setting serial channel mode register SC0MOD<SM1:0>/SC1MOD<SM1:0> to "10" selects 8-bit UART mode. A parity bit can be added in this mode. Enable or disable the addition of a parity bit by the serial channel control register SC0CR<PE>/SC1CR<PE> bit. With <PE> set to "1" (parity enabled), select even or odd parity using SC0CR<EVEN>/SC1CR<EVEN>.

Example: When data are transmitted in the following format, the control registers are set as follows. The example shows channel 0.



Main routine settings:

		7	6	5	4	3	2	1	0	
PFCR	←	–	–	–	–	–	–	0	–	Sets PF1 (RxD0) as input pin.
SC0MOD	←	–	0	1	0	1	0	0	1	Sets 8-bit UART mode and enables reception.
SC0CR	←	X	0	1	X	X	X	0	0	Adds odd parity.
BR0CR	←	0	X	0	1	1	0	0	0	Sets transmission rate to 9600 bps.
T16RUN	←	1	X	–	–	–	–	–	–	Starts prescaler for baud rate generator.
INTES0	←	–	–	–	–	X	1	0	0	Enables interrupt INTRX0 and sets interrupt level 4.

Interrupt routine processing example:

Acc ← SC0CR AND 00011100 If Acc ≠ 0 then ERROR	Checks for errors.
Acc ← SC0BUF	Reads data received.

Note : X; Don’t care –; No change

(4) Mode 3 (9-bit UART mode)

Setting the serial channel mode register SC0MOD<SM1:0>/SC1MOD<SM1:0> to “11” selects 9-bit UART mode. A parity bit cannot be added in this mode.

At transmission, the most significant bit (9th bit) is written to <TB8> of the serial channel mode register. At receiving, the most significant bit is saved in <RB8> of the serial channel control register.

When data are written to or read from the buffer, the most significant bit is always read or written first, followed by the SC0BUF/SC1BUF register.

3.11 Analog/Digital Converter

TMP94C251A incorporates a high-speed, high-precision 10-bit analog/digital converter (AD converter) with 8-channel analog input.

Figure 3.11.1 is a block diagram of the AD converter. The 8-channel analog input pins (AN0 to AN7) are also used as input-only port G pins and can be also used as input ports.

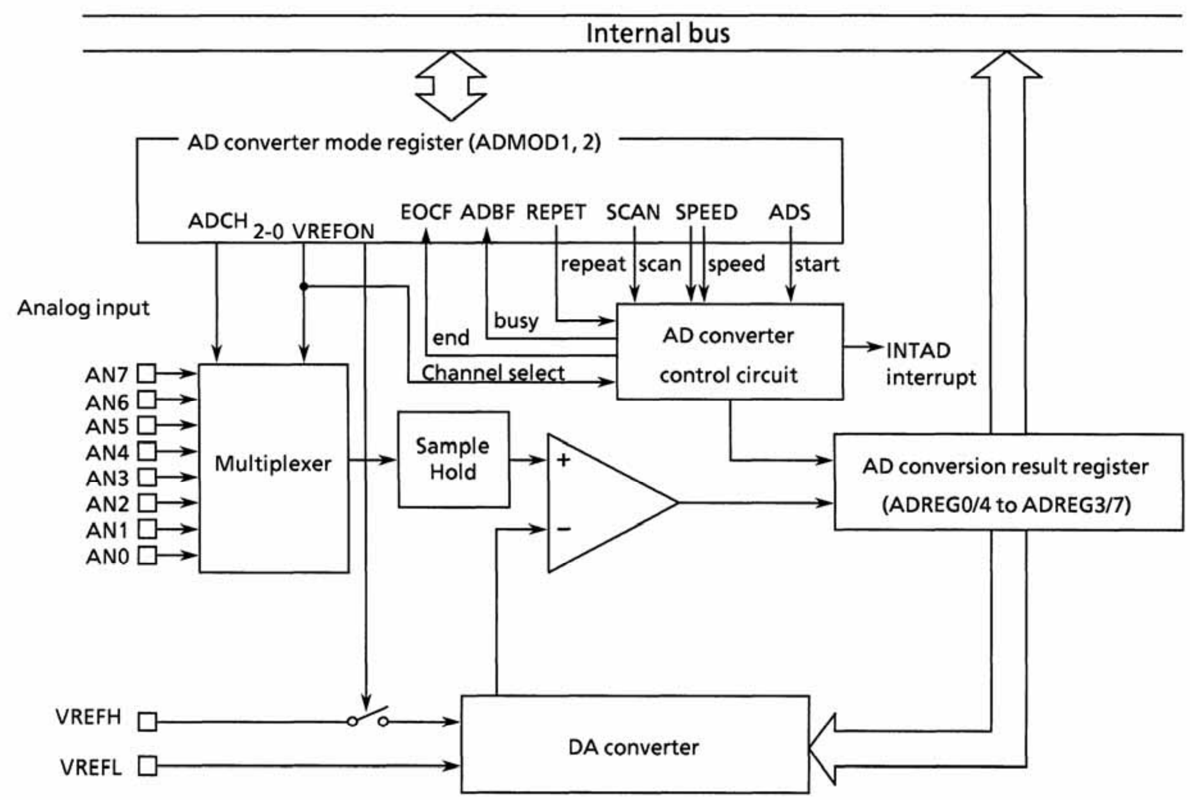


Figure 3.11.1 Block Diagram for AD Converter

ADMOD1
(0128H)

	7	6	5	4	3	2	1	0
bit Symbol	EOCF	ADBF	–	–	–	REPET	SCAN	ADS
Read/Write	R					R/W		
After reset	0	0	0	–	–	0	0	0
Function	AD Conversion End Flag 1: END	AD Conversion BUSY Flag 1: BUSY	Fix to "0"			Repeat Mode 0: Single Mode 1: Repeat Mode	Scan Mode 0: Fixed channel 1: Channel Scan Mode	AD Conversion start 1: START Always read as "0"

AD conversion Start

0	–
1	Start Conversion

Note) Always "0" when data is read

AD Scan Mode

0	Fixed channel mode
1	Channel scan mode

AD Repeat Mode

0	Single mode
1	Repeat mode

AD Busy flag

0	NOT BUSY
1	BUSY

AD END flag

0	NOT END
1	END

Figure 3.11.2 AD Control Register

ADMOD2 (0129H)

	7	6	5	4	3	2	1	0
bit Symbol	VREFON	–	SPEED1	SPEED0	–	ADCH2	ADCH1	ADCH0
Read/Write	R/W							
After reset	1	–	0	0	–	0	0	0
Function	Ladder Resistance ON / OFF		Conversion Speed 00: 160 states 01: 320 states 10: 640 states 11: 1280 states			Analog Input Channel Selection		

Analog Input Channel Selection

<ADCH2,1,0> \ <SCAN>	0	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

Conversion Speed

00	160 states (8 μs at 20 MHz)
01	320 states (16 μs at 20 MHz)
10	640 states (32 μs at 20 MHz)
11	1280 states (64 μs at 20 MHz)

Ladder Resistance ON / OFF Selection

0	Ladder Resistance OFF
1	Ladder Resistance ON

Note: Set the <VREFON> register to 1 before starting conversion (setting <ADS> to 1).

Figure 3.11.3 AD Control Register

ADREG04L (0120H)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR01	ADR00	–	–	–	–	–	–
Read/Write	R							
After Reset	Undefined		–	–	–	–	–	–
Function	Lower 2 bits of AD result for AN0 or AN4 are stored.							

ADREG04H (0121H)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
Read/Write	R							
After Reset	Undefined							
Function	Upper 8 bits of AD result for AN0 or AN4 are stored.							

ADREG15L (0122H)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR11	ADR10	–	–	–	–	–	–
Read/Write	R							
After Reset	Undefined		–	–	–	–	–	–
Function	Lower 2 bits of AD result for AN1 or AN5 are stored.							

ADREG15H (0123H)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
Read/Write	R							
After Reset	Undefined							
Function	Upper 8 bits of AD result for AN1 or AN5 are stored.							

Note: The result registers are used both as AN0 and AN4, AN1 and AN5, AN2 and AN6, and AN3 and AN7. They are stored in ADREG04, ADREG15, ADREG26 and ADREG37.

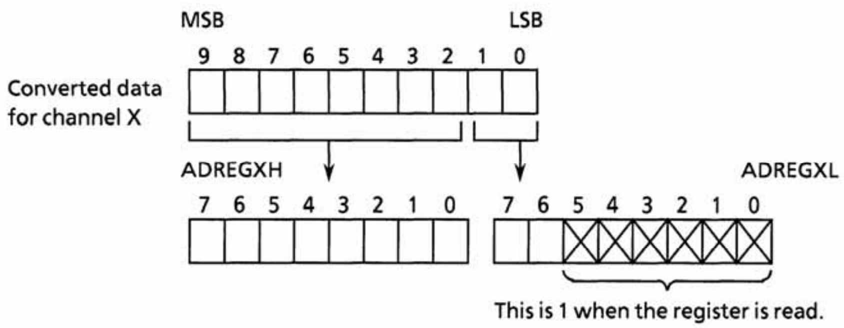


Figure 3.11.4 AD Conversion Result Register (ADREG04, ADREG15)

ADREG26L (0124H)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR21	ADR20	–	–	–	–	–	–
Read/Write	R							
After Reset	Undefined		–	–	–	–	–	–
Function	Lower 2 bits of AD result for AN2 or AN6 are stored.							

ADREG26H (0125H)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
Read/Write	R							
After Reset	Undefined							
Function	Upper 8 bits of AD result for AN2 or AN6 are stored.							

ADREG37L (0126H)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR31	ADR30	–	–	–	–	–	–
Read/Write	R							
After Reset	Undefined		–	–	–	–	–	–
Function	Lower 2 bits of AD result for AN3 or AN7 are stored.							

ADREG37H (0127H)

	7	6	5	4	3	2	1	0
Bit Symbol	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
Read/Write	R							
After Reset	Undefined							
Function	Upper 8 bits of AD result for AN3 or AN7 are stored.							

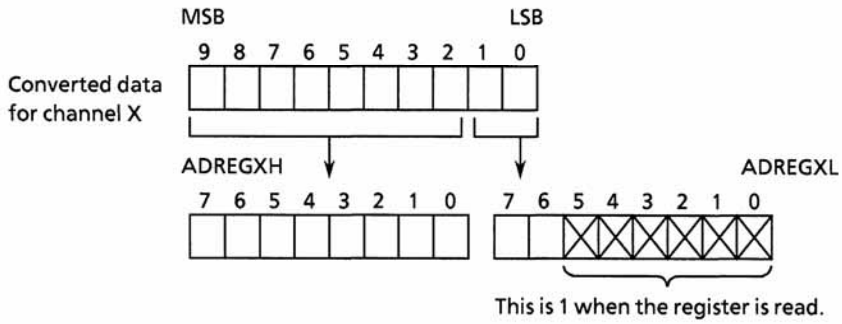


Figure 3.11.5 AD Conversion Result Register (ADREG26, ADREG37)

3.11.1 Operation

(1) Analog reference voltage

The high analog reference voltage is applied to the VREFH pin, and the low analog reference voltage is applied to the VREFL pin.

The reference voltage between VREFH and VREFL is divided by 1024 (using ladder resistance) and compared with the analog input voltage for AD conversion.

The switch between VREFH and VREFL can be turned off by writing 0 to ADMOD2 <VREFON>.

When <VREFON> = 0, before the conversion can start, must be written to <VREFON> and a 3 μs period must be allowed so that the internal reference voltage can stabilize (Regardless of fc) before 1 is written to ADMOD1 to <ADS>.

(2) Analog input channels

The analog input channel is selected by ADMOD2<ADCH2:0>. However, the channel which should be selected depends on the operation mode of the AD converter.

In Fixed Analog Input mode, one channel is selected out of eight pins, AN0 to AN7, by <ADCH2:0>.

In analog input channel scan mode, the number of channels to be scanned is specified by ADMOD2<ADCH2:0>, e.g., AN0 only, AN0→AN1, AN0→AN1→AN2, AN0→AN1→AN2→AN3, AN4→AN5, AN4→AN5→AN6 or AN4→AN5→AN6→AN7.

When reset the AD conversion channel register will be initialized to ADMOD2<ADCH2:0> = 000, so that the AN0 pin is selected.

The pins which are not used as analog input channels can be used as ordinary input port pins for port G.

(3) Starting AD conversion

AD conversion starts when 1 is written to the AD conversion register ADMOD1 <ADS>. When conversion starts, the conversion busy flag ADMOD1<ADBF>, which indicates that conversion is in progress, is set to 1.

(4) AD conversion mode

Both fixed AD conversion channel mode and conversion channel scan mode include two conversion modes; single and repeat conversion mode.

In fixed channel repeat mode, conversion of the specified single channel is executed repeatedly.

In scan repeat mode, scanning is executed repeatedly.

The AD conversion mode is selected by ADMOD1<REPET, SCAN>.

(5) AD conversion speed selection

There are four AD conversion speed modes. The selection is made by the ADMOD2 <SPEED1:0> register.

When reset, <SPEED1:0> is initialized to 00, selecting 160-state conversion mode (8 μs at 20 MHz).

(6) AD conversion end and interrupt

- AD conversion single mode

  When AD conversion of the specified channel has finished (in fixed channel conversion mode) or when AD conversion of the last channel has finished (in channel scan mode), ADMOD<EOCF> is set to 1, the ADMOD<ADBF> flag is reset to 0, and the INTAD interrupt is generated.

- AD conversion repeat mode

  For both fixed conversion channel mode and conversion channel scan mode, INTAD should be disabled in Repeat mode. Always set INTE0AD to 000, to disable the interrupt request.
  Write 0 to ADMOD1<REPET> to terminate Repeat mode. Repeat mode will be exited as soon as the conversion in progress is completed.

(7) Storing the AD conversion result

The results of AD conversion are stored in the registers ADREG04 to ADREG37 for each channel. The result registers are used as AN0 and AN4, AN1 and AN5, AN2 and AN6 and AN3 and AN7.

However, the contents of the registers do not indicate which channel's data has been converted.

In repeat mode, the registers are updated as soon as conversion ends.

ADREG04 to ADREG37 are read-only registers.

(8) Reading the AD conversion result

The results of AD conversion are stored in the registers ADREG04 to ADREG37.

When the one of the registers ADREG04, ADREG15, ADREG26 or ADREG37 are read, ADMOD1<EOCF> is cleared to 0.

Setting example: [1] When the analog input voltage on the AN3 pin is AD-converted at 160-state speed and the result is transferred to the memory address 0100H by the AD interrupt INTAD routine.

Main setting

```
INTE0AD   ← X 1 0 0 - - - -    Enable INTAD and set interrupt level 4.
ADMOD2    ← 1 X 0 0 X 0 1 1    Specify AN3 pin as an analog input channel and
ADMOD1    ← X X 0 X X 0 0 1    start AD conversion in 160-atate speed mode.
```

INTAD routine

```
WA          ← ADREG37          Read ADREG37L and ADREG37H values and write
                               to WA (16 bits).
WA          >> 6               Right-shift WA six times and write 0 in upper bits.

(000100H)←    WA               Write contents of WA in memory at 0100H
```

[2] When the analog input voltage of the four pins AN4 to AN7 are AD converted at 320-state speed and the channel is set to scan and repeat mode.

Main setting

```
INTE0AD   ← X 0 0 0 - - - -    Disable INTAD.
ADMOD2    ← 1 X 0 1 X 1 1 1    Specify AN4 to AN7 pins as input channel, select
ADMOD1    ← X X 0 X X 1 1 1    Scan & Repeat mode and start AD conversion.
```

Note: X: Don't care –: No change

3.12 8-Bit Voltage Output-type DA Converter

TMP94C251A incorporates a 2-channel, 8-bit resolution DA converter with the following features.

- String resistor method buffer output-type 8-bit resolution DA converter with two internal channels
- Registers DAREG0 and DAREG1 to control the analog voltage output

Figure 3.12.1 is a block diagram of the DA converter.

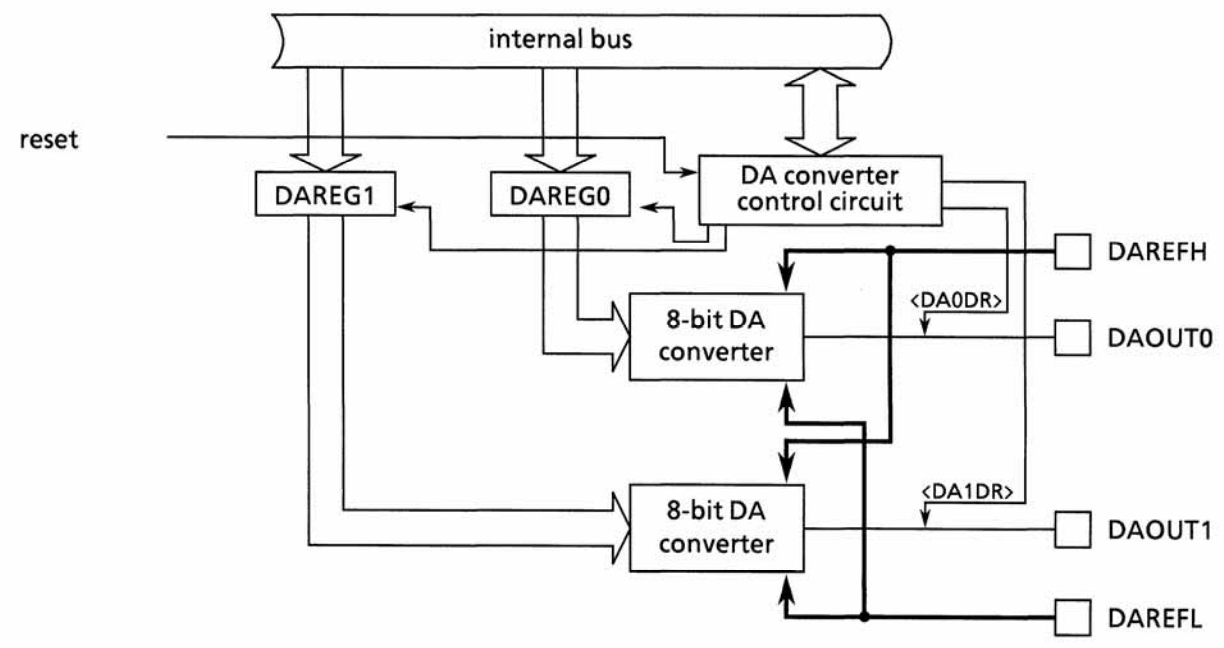


Figure 3.12.1 DA Converter Block Diagram

DA converter drive register

DADRV (0132H)		7	6	5	4	3	2	1	0
	bit Symbol	—	—	—	—	—	—	DA1DR	DA0DR
	Read/Write							R/W	
	After reset	—	—	—	—	—	—	0	
	Function							0: Outputs 0 V 1: Outputs register conversion value.	

DA output drive setting

0	Outputs 0 V
1	Outputs register conversion value.

DA conversion register 0

DAREG0 (0130H) (No RMW)		7	6	5	4	3	2	1	0
	bit Symbol	—							
	Read/Write	W							
	After reset	Undefined							
	Function	Starts DA conversion at register write, outputs to DAOUT0.							

DA conversion register 1

DAREG1 (0131H) (No RMW)		7	6	5	4	3	2	1	0
	bit Symbol	—							
	Read/Write	W							
	After reset	Undefined							
	Function	Starts DA conversion at a register write, outputs to DAOUT1.							

These registers are used for the DA converter digital input data. The relationship between the register values and the output voltages is as follows: output voltage V = (DAREFH - DAREFL) x N/256 (where N is the register value).

Note: Read-modify-write is prohibited for registers DAREG0, DAREG1.

Figure 3.12.2 DA Converter Registers

3.12.1 Operation

When DA converter drive register DADRV<DA1DR, DA0DR> is set to “1”, the internal DA converter converts digital values in DA converter registers DAREG1 or DAREG0 to analog values, and outputs these values as voltages from pins DAOUT1 and DAOUT0. Figure 3.12.2 shows the relationship between input data and output voltage.

As a reset clears <DA1DR> and <DA0DR> to “0”, DAOUT1 and DAOUT0 pins output 0 V. After a reset, DAREG1 and DAREG0 are undefined. To output the relevant analog value using the DA converter, write input data in DAREG1 and DAREG0, then write “1” to the DADRV bit of the channel to be used. Be sure to write data to DAREG1 and DAREG0 first. If, after a reset, DADRV is set to “1” before the input data are written to DAREG1 and DAREG0, DAREG1 and DAREG0 are undefined, and the converter outputs undefined analog values.

If the HALT instruction is executed after specifying STOP mode (WDMOD<HALTM1:0> = “01”), the DAOUT output varies with WDMOD<DRVE> and DADRV<DAnDR> as shown in Table 3.12.1.

Table 3.12.1 DAOUT Output in STOP Mode

WDMOD <DRVE>	DADRV <DAnDR>	DAOUTn OUTPUT
0	0	High-Z
	1	High-Z
1	0	0 V output
	1	Output the converted register value

Example: Set DAREFH = Vcc, DAREFL = GND.

		7	6	5	4	3	2	1	0	
DAREG1	←	1	1	1	1	1	1	1	1	Writes FFH. $DAOUT1 = Vcc \times \frac{255}{256} \doteq Vcc$
DAREG0	←	1	0	0	0	0	0	0	0	Writes 80H. $DAOUT0 = Vcc \times \frac{128}{256} = \frac{Vcc}{2}$
DADRV	←	X	X	X	X	X	X	1	1	Outputs DAOUT1/DAOUT0.
DAREG1	←	1	0	0	0	0	0	0	0	Writes 80H. Outputs Vcc / 2 to DAOUT1.
DAREG0	←	1	1	1	1	1	1	1	1	Writes FFH. Outputs Vcc to DAOUT0.

3.13 Watchdog Timer (Runaway detection timer)

TMP94C251A incorporates a watchdog timer for detecting runaways.

The watchdog timer (WDT) returns the CPU to its normal state after the watchdog timer detects the start of a CPU malfunction (Runaway) due to noise, for example. When the watchdog timer detects a runaway, it generates a non-maskable interrupt to notify the CPU of the runaway and outputs a "0" signal from the watchdog timer out pin ($\overline{\text{WDTOUT}}$) to notify any peripheral devices of the runaway.

Connecting the watchdog timer output to the RESET pin (within the chip) forces a reset.

3.13.1 Configuration

Figure 3.13.1 is a block diagram of the watchdog timer (WDT).

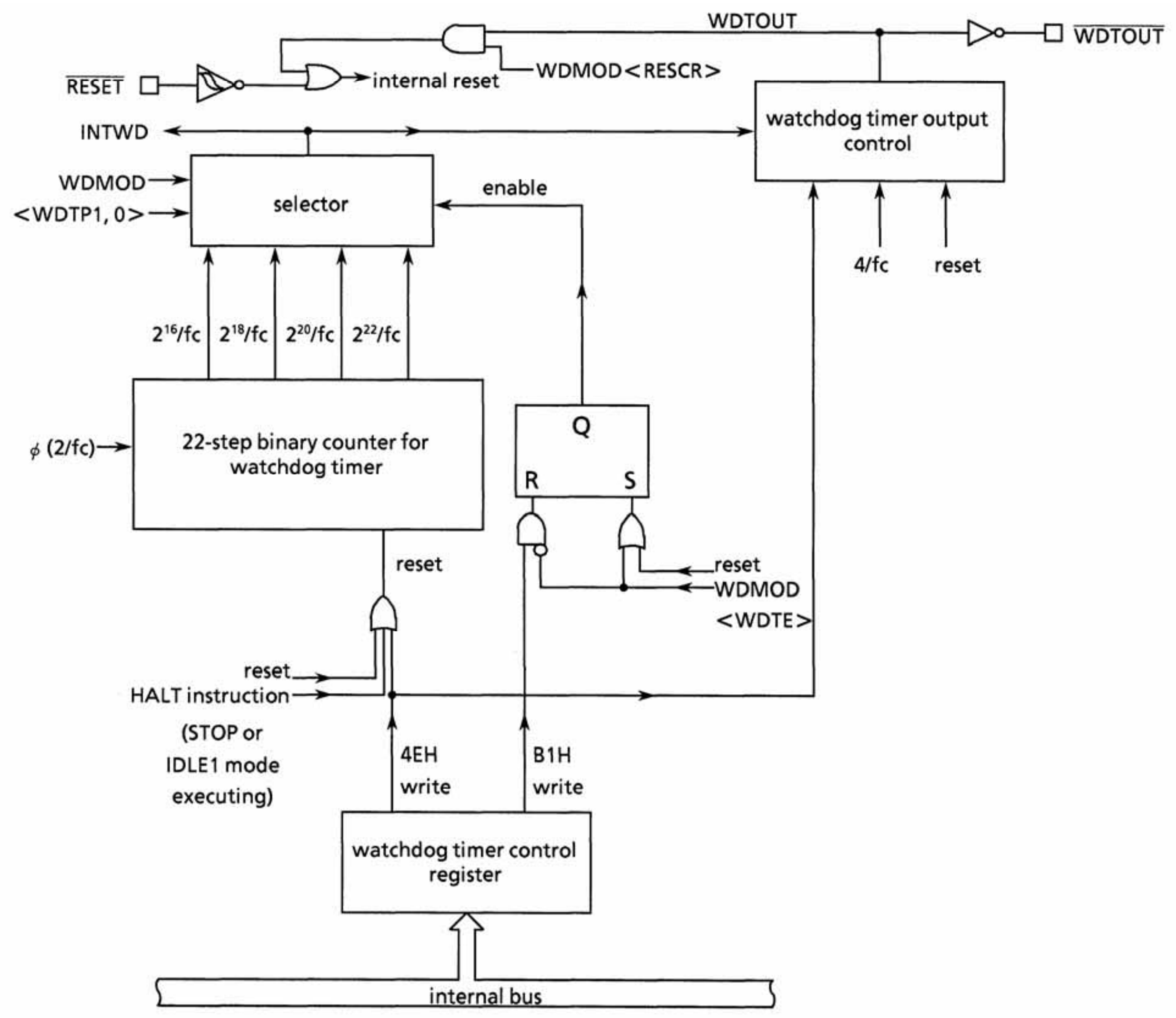


Figure 3.13.1 Watchdog Timer Block Diagram

The watchdog timer is a 22-step binary counter, which uses ϕ (2/fc) as the input clock.

The WDMOD register selects the output of one of four binary counters: 2^{16}/fc, 2^{18}/fc, 2^{20}/fc, or 2^{22}/fc. Overflow from the selected counter generates a watchdog timer interrupt and outputs a signal to the watchdog timer out pin.

As a result of watchdog timer overflow, the watchdog timer out pin ($\overline{\text{WDTOUT}}$) outputs “0”, which can be used as a reset signal for peripheral devices.

Clearing the watchdog timer (writing the clear code (4EH) to the WDCR register) sets the WDTOUT pin to “1”. In normal mode, the $\overline{\text{WDOUT}}$ pin continually outputs “0” until the clear code is written to the WDCR register.

The watchdog timer output can also be connected to the RESET pin internally. In this case, the watchdog timer out pin ($\overline{\text{WDTOUT}}$) outputs “0” for 16/fc to 40/fc (800 ns - 2 μs at 20 MHz), resetting itself at the same time.

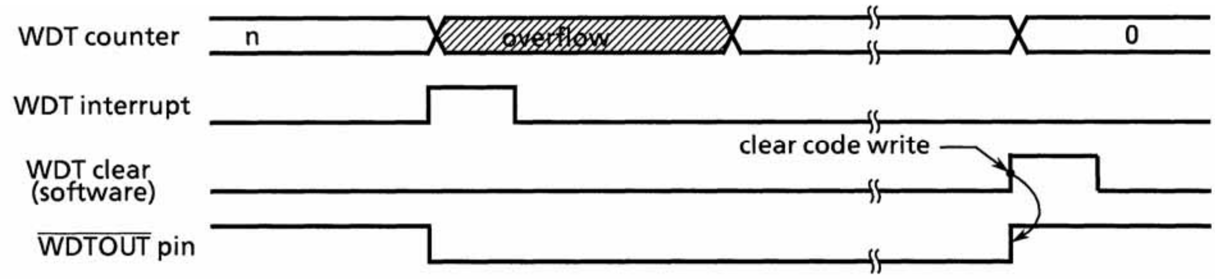


Figure 3.13.2 Normal Mode

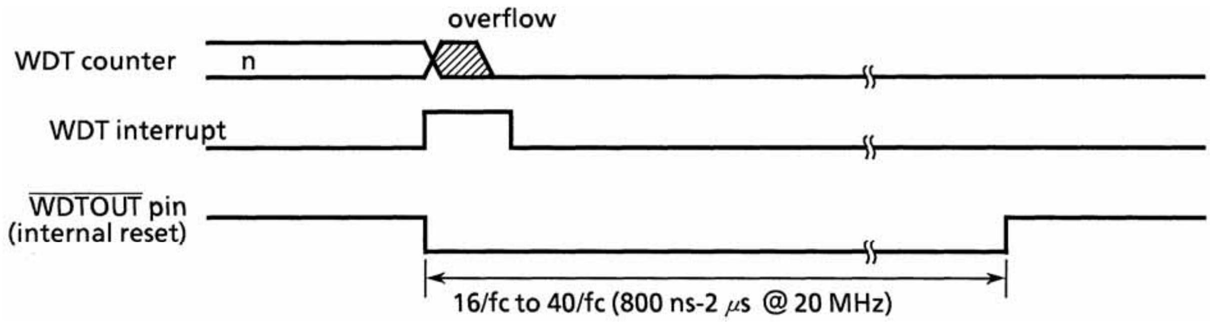


Figure 3.13.3 Reset Mode

3.13.2 Control Registers

The watchdog timer (WDT) is controlled by three control registers: WDMOD, WDCR and CLKMOD.

(1) Watchdog timer mode register WDMOD

[1] Setting watchdog timer detection time <WDTP>

This 2-bit register is used to set the watchdog timer interrupt time for detecting runaways. At reset, this register is initialized to “00” (WDMOD<WDTP1:0> is set to “00”), setting a detection time of 2^{16}/fc [s]. (The number of states is approximately 32,768.)

[2] Watchdog timer enable/disable control <WDTE>

At reset, the WDMOD<WDTE> bit is initialized to “1”, enabling the watchdog timer.

Disabling the watchdog timer requires both clearing WDTE to 0 and writing disable code B1H in the WDCR register. This two-step process makes it difficult for a runaway to disable the watchdog timer.

To return from the disable state to the enable state, simply set the <WDTE> bit to “1”.

[3] Connection of watchdog timer output to reset pin <RESCR>

This register determines whether or not the watchdog timer resets itself after a runaway is detected.

At reset, WDMOD<RESCR> is initialized to 0, and the watchdog timer will therefore not trigger a reset.

(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 the WDMOD<WDTE> register to 0, writing the disable code “B1H” to the WDCR register disables the watchdog timer.

```
WDMOD ← 0 - - - - - X X      Clears WDTE to 0.
WDCR  ← 1 0 1 1 0 0 0 1      Writes disable code B1H.
```

- Enable control

  Set WDMOD7<WDTE> to 1.

- Clear control

  Writing clear code 4EH to the WDCR register clears the binary counter and resumes the count.

```
WDCR  ← 0 1 0 0 1 1 1 0      Writes clear code 4EH.
```

(3) Clock mode register CLKMOD

This register is used to set the warm-up time after the stop mode ends.

Writing "0" to the CLKMOD <WARM> bit, 2^{15}/fc (Approximately 1.6 ms at 20 MHz) is selected and writing "1", 2^{17}/fc (Approximately 6.6 ms at 20 MHz) is selected.

Also, the system clock output can be disabled by writing 0 to CLKMOD<CLKOE>.

WDMOD (0110H)

	7	6	5	4	3	2	1	0
bit Symbol	WDTE	WDTP1	WDTP0	–	HALTM1	HALTM0	RESCR	DRVE
Read/Write	R/W							
After reset	1	0	0	–	0	0	0	0
Function	WDT control 1: enable	WDT detection time select 00: 2^{16}/fc 01: 2^{18}/fc 10: 2^{20}/fc 11: 2^{22}/fc		Reserved	Standby mode 00: RUN mode 01: STOP mode 10: IDLE mode 11: Don't care		1: Internally connects WDTOUT pin to RESET pin.	1: Drives pins in STOP mode

DREVE (See description of STOP mode in the section on standby function.)

Controls watchdog timer output.

0	–
1	Connects WDTOUT pin to reset pin.

Selects standby mode by HALT instruction

00	RUN mode (halts CPU only)
01	STOP mode (all circuits shut down)
10	IDLE mode (oscillator only in operation)
11	Don't care

Selects watchdog timer detection time.

00	2^{16}/fc (approximately 3.28 ms @ 20 MHz)
01	2^{18}/fc (approximately 13.1 ms @ 20 MHz)
10	2^{20}/fc (approximately 52.4 ms @ 20 MHz)
11	2^{22}/fc (approximately 210 ms @ 20 MHz)

Controls watchdog timer enable/disable.

0	Disable
1	Enable

Figure 3.13.4 Watchdog Timer Mode Register

WDCR (0111H)

	7	6	5	4	3	2	1	0
bit Symbol	–							
Read/Write	W							
After reset	–							
Function	B1H: WDT disable code 4EH: WDT clear code							

Disables and clears WDT.

B1H	Disable code
4EH	Clear code
Others	–

Figure 3.13.5 Watchdog Timer Control Register

CLKMOD (10AH)

	7	6	5	4	3	2	1	0
bit Symbol	–	–	–	WARM	–	CLKOE	–	–
Read/Write				R/W		R/W		
After reset	–	–	–	0	–	1	0	0
Function				Warming up time 0: 2^{15}/fc 1: 2^{17}/fc		CLK Output Enable 0: High-Z 1: Out	Leave these bits fixed to 0.	

System clock output control

0	Disabled (High-Z)
1	Output

Selects Warming up time after stop Mode ends

0	2^{15}/fc (approximately 1.6 ms @ 20 MHz)
1	2^{17}/fc (approximately 6.6 ms @ 20 MHz)

Figure 3.13.6 CLOCK Mode Register

3.13.3 Operation

After the detection time set by the WDMOD<WDTP1:0> register is reached, the watchdog timer generates interrupt INTWD and outputs a low signal to the watchdog timer out pin $\overline{\text{WDTOUT}}$. The binary counter for the watchdog timer must be cleared to 0 by software (Instruction) before INTWD is generated. If the CPU malfunctions (Runaway) due to causes such as noise and does not execute an instruction to clear the binary counter, the binary counter overflows and generates INTWD.

The CPU interprets INTWD as a malfunction detection signal, which can be used to start the malfunction recovery program to return the system to normal. A CPU malfunction can also be fixed by connecting the watchdog timer output to a reset pin for peripheral devices.

The watchdog timer begins operating immediately on release of the watchdog timer reset.

The watchdog timer is reset and halted in IDLE and STOP modes. The watchdog counter continues counting during bus release ($\overline{\text{BUSAK}}$ = Low).

The watchdog timer operates in RUN mode; it can be disabled when RUN mode is entered.

Examples:

[1] Clear the binary counter.

```
WDCR   ← 0 1 0 0 1 1 1 0     Writes clear code (4EH).
```

[2] Set the watchdog timer detection time to 2^{18}/fc.

```
WDMOD  ← 1 0 1 X – – – –
```

[3] Disable the watchdog timer.

```
WDMOD  ← 0 – – X – – – –     Clears WDTE to "0".
WDCR   ← 1 0 1 1 0 0 0 1     Writes disable code (B1H).
```

[4] Select IDLE mode.

```
WDMOD  ← 0 – – X 1 0 – –     Disables WDT and set IDLE mode.
WDCR   ← 1 0 1 1 0 0 0 1
Executes HALT instruction.    Sets to standby mode.
```

[5] Select STOP mode. (Warm-up time 2^{17}/fc)

```
WDMOD  ← – – – X 0 1 – –     Sets to STOP mode.
CLKMOD ← X X X 1 X – – –
Executes HALT instruction.    Sets to standby mode.
```

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

3.14 Bus Release Function

TMP94C251A has a bus request pin ($\overline{\text{BUSRQ}}$ also used as P75) for releasing the bus, and a bus acknowledge pin ($\overline{\text{BUSAK}}$ also used as P76). Set these pins using P7CR and P7FC.

3.14.1 Operation

When the bus release request pin ($\overline{\text{BUSRQ}}$) is set to active (Low), TMP94C251A acknowledges a bus release request.

When the operand cycle completes, TMP94C251A first sets the address bus (A23 to A0) and the bus control signals ($\overline{\text{RD}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\text{CS0}}$ to $\overline{\text{CS5}}$) simultaneously to high, sets these signals and the output buffer for the data bus (D31 to D0) to off, and sets the $\overline{\text{BUSAK}}$ pin to low, indicating that the bus is released.

When using as input port or output port modes, the bus release is not executed for the port, and the output buffer is not turned off.

During bus release, TMP94C251A disables all access to the internal I/O registers, although the internal I/O functions are not affected. As 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.

When inputting “low” into $\overline{\text{BUSRQ}}$ terminal, continue “low” input until $\overline{\text{BUSAK}}$ terminal outputs “low”. If the request is released before $\overline{\text{BUSAK}}$ terminal outputs “low”, the memory controller may malfunction.

4. Electrical Characteristics

4.1 Maximum Ratings

Symbol	Parameter	Rating	Unit
V cc	Power Supply Voltage	– 0.5 to 6.5	V
V IN	Input Voltage	– 0.5 to Vcc + 0.5	V
Σ IOL	Output Current (total)	120	mA
Σ IOH	Output Current (total)	– 120	mA
P D	Power Dissipation (Ta = 70°C)	600	mW

Note: The 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 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 maximum rating value will ever be exceeded.

4.2 DC Electrical Characteristics

Vcc = 5 V ± 10%, TA = –20 to 70°C
X1 = 8 to 10 MHz (Internal operation = 16 to 20 MHz)

Symbol	Parameter	Min	Max	Unit	Test Condition
V IL0	Input Low Voltage P00 to P07 (D0 to 7) P10 to P17 (D8 to 15) P20 to P27 (D16 to 23) P30 to P37 (D24 to 31)	– 0.3	0.8	V	
V IL1	Input Low Voltage P40 to P47 P50 to P57 P60 to P67 P75 P86 PC0, PC1 PD0 to PD2, PD4 to PD6 PE0 to PE2, PE4 to PE6 PF0 to PF2, PF4 to PF6 PG0 to PG7 PH0 to PH3 PZ0 to PZ7	– 0.3	0.3∗Vcc	V	
V IL2	Input Low Voltage PH4 (INT0) $\overline{\text{NMI}}$ $\overline{\text{RESET}}$	– 0.3	0.25∗Vcc	V	
V IL3	Input Low Voltage AM0, AM1 TEST0, TEST1	– 0.3	0.3	V	
V IL4	Input Low Voltage X1	– 0.3	0.2∗Vcc	V	
V IH0	Input High Voltage P00 to P07 (D0 to 7) P10 to P17 (D8 to 15) P20 to P27 (D16 to 23) P30 to P37 (D24 to 31)	2.2	Vcc + 0.3	V	

Note: Typical value are for Ta = 25 °C and Vcc = 5 V unless otherwise noted.

Symbol	Parameter	Min	Max	Unit	Test Condition
V IH1	Input High Voltage P40 to P47 P50 to P57 P60 to P67 P75 P86 PC0, PC1 PD0 to PD2, PD4 to PD6 PE0 to PE2, PE4 to PE6 PF0 to PF2, PF4 to PF6 PG0 to PG7 PH0 to PH3 PZ0 to PZ7	0.7*Vcc	Vcc + 0.3	V	
V IH2	Input High Voltage PH4 (INT0) $\overline{\text{NMI}}$ $\overline{\text{RESET}}$	0.75*Vcc	Vcc + 0.3	V	
V IH3	Input High Voltage AM0, AM1 TEST0, TEST1	Vcc − 0.3	Vcc + 0.3	V	
V IH4	Input High Voltage X1	0.8*Vcc	Vcc + 0.3	V	
V OL	Output Low Voltage		0.45	V	IOL = 1.6 mA
V OH0	Operating Cururent (NORMAL)	2.4		V	IOH = −400μA
V OH1	Output High Voltage	0.75*Vcc		V	IOH = −100μA
V OH2	Output High Voltage	0.9*Vcc		V	IOH = −20μA
I LI	Input Leakage Current	0.02 (typ.)	±5	μA	0.0V ≦ Vin ≦ Vcc
I L0	Output Leakage Current	0.05 (typ.)	±10	μA	0.2V ≦ Vin ≦ Vcc − 0.2 V
I cc0	Operating Current (NORMAL)	90	108	mA	X1 = 10 MHz (Internal 20 MHz)
I cc1	RUN	50	70	mA	X1 = 10 MHz (Internal 20 MHz)
I cc2	IDLE	5	20	mA	X1 = 10 MHz (Internal 20 MHz)
I cc3	STOP	0.5	50	μA	0.2 V ≦ Vin ≦ Vcc − 0.2 V Ta = −20∼70°C
I cc4	STOP		10	μA	0.2 V ≦ Vin ≦ Vcc − 0.2 V Ta = 0∼50°C
V STOP	Power Down Voltage @ STOP (for internal RAM back-up)	2.0	6.0	V	VIL2 = 0.2*Vcc VIH2 = 0.8*Vcc
RRST	Pull Up Registance $\overline{\text{RESET}}$	50	150	kΩ	
CIO	Pin Capacitance		10	pF	fc = 1 MHz
VTH	Schmitt Width PH4 (INT0) $\overline{\text{NMI}}$ $\overline{\text{RESET}}$	0.4	1.0 (typ)	V	

4.3 AC Electrical Characteristics

4.3.1 Basic Bus Cycle

(1) Read cycle

No.	Symbol	Parameter	Min	Max	at 20 MHz	at 16 MHz	Unit
1	t_{OSC}	OSC period (X1/X2)	100	125	100	125	ns
2	t_{CYC}	System Clock Period (= T)	50	62.5	50	62.5	ns
3	t_{CL}	CLK Low Width	0.5 × T – 15		10	16	ns
4	t_{CH}	CLK High Width	0.5 × T – 15		10	16	ns
5-1	t_{AD}	A0 to A23 → D0 to D31 Input at 0 waits		2.0 × T – 50	50	75	ns
5-2	t_{AD3}	A0 to A23 → D0 to D31 Input at 1 wait		3.0 × T – 50	100	138	ns
6-1	t_{RD}	$\overline{RD}$ Fall → D0 to D31 Input at 0 waits		1.5 × T – 45	30	49	ns
6-2	t_{RD3}	$\overline{RD}$ Fall → D0 to D31 Input at 1 wait		2.5 × T – 45	80	111	ns
7-1	t_{RR}	$\overline{RD}$ Low Width at 0 waits	1.5 × T – 20		55	74	ns
7-2	t_{RR3}	$\overline{RD}$ Low Width at 1 wait	2.5 × T – 20		105	136	ns
8	t_{AR}	A0 to A23 Valid → $\overline{RD}$ Fall	0.5 × T – 20		5	11	ns
9	t_{RK}	$\overline{RD}$ Fall → CLK Fall	0.5 × T – 20		5	11	ns
10	t_{HA}	A0 to A23 Invalid → D0 to D31 Hold	0		0	0	ns
11	t_{HR}	$\overline{RD}$ Rise → D0 to D31 Hold	0		0	0	ns
12	t_{APR}	A0 to A23 Valid → PORT Input		2.0 × T – 120	–20	5	ns
13	t_{APH}	A0 to A23 Valid → PORT Hold	2.0 × T		100	125	ns
14	t_{TK}	$\overline{WAIT}$ Setup Time	15		15	15	ns
15	t_{KT}	$\overline{WAIT}$ Hold Time	5		5	5	ns

(2) Write cycle

No.	Symbol	Parameter	Min	Max	at 20 MHz	at 16 MHz	Unit
1	t_{OSC}	OSC Period (X1/X2)	100	125	100	125	ns
2	t_{CYC}	System Clock Period (= T)	50	62.5	50	62.5	ns
3	t_{CL}	CLK Low Width	0.5 × T – 15		10	16	ns
4	t_{CH}	CLK High Width	0.5 × T – 15		10	16	ns
5-1	t_{DW}	D0 to D31 Valid → $\overline{WRx}$ Rise at 0 waits	1.25 × T – 35		28	43	ns
5-2	t_{DW3}	D0 to D31 Valid → $\overline{WRx}$ Rise at 1 wait	2.25 × T – 35		78	106	ns
6-1	t_{WW}	$\overline{WRx}$ Low Width at 0 waits	1.25 × T – 30		33	48	ns
6-2	t_{WW3}	$\overline{WRx}$ Low Width at 1 wait	2.25 × T – 30		83	111	ns
7	t_{AW}	A0 to A23 Valid → $\overline{WRx}$ Fall	0.5 × T – 20		5	11	ns
8	t_{WK}	$\overline{WRx}$ Fall → CLK Fall	0.5 × T – 20		5	11	ns
9	t_{WA}	$\overline{WRx}$ Rise → A0 to A23 Hold	0.25 × T – 5		8	11	ns
10	t_{WD}	$\overline{WRx}$ Rise → D0 to D31 Hold	0.25 × T – 5		8	11	ns
11	t_{APW}	A0 to A23 Valid → PORT Output		2.0 × T + 70	170	195	ns
12	t_{TK}	$\overline{WAIT}$ Setup Time	15		15	15	ns
13	t_{KT}	$\overline{WAIT}$ Hold Time	5		5	5	ns
14	t_{RDO}	$\overline{RD}$ Rise → D0 to D15 Output	0.5 × T – 5		20	26	ns

AC Condition

Output: P0 to P3 (D0 to D31), P4 to P6 (A0 to A23), P70 ($\overline{RD}$), P71 to P74 ($\overline{WRx}$)

High = 2.0 V, Low = 0.8 V, CL = 50 pF

Others

High = 2.0 V, Low = 0.8 V, CL = 50 pF

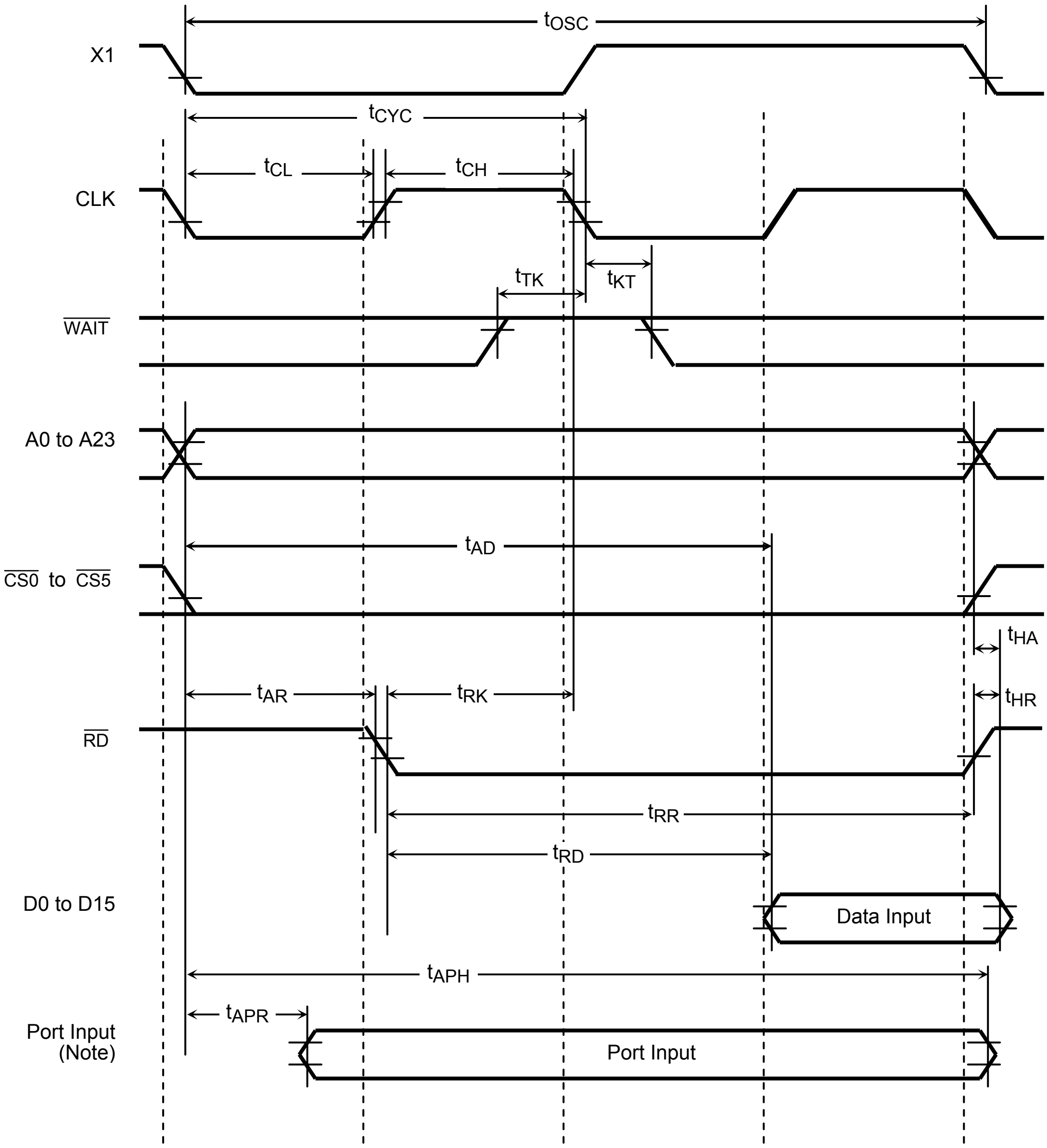
Input: P0 to P3 (D0 to D31)

High = 2.4 V, Low = 0.45 V

Others

High = 0.8 Vcc, Low = 0.2 Vcc

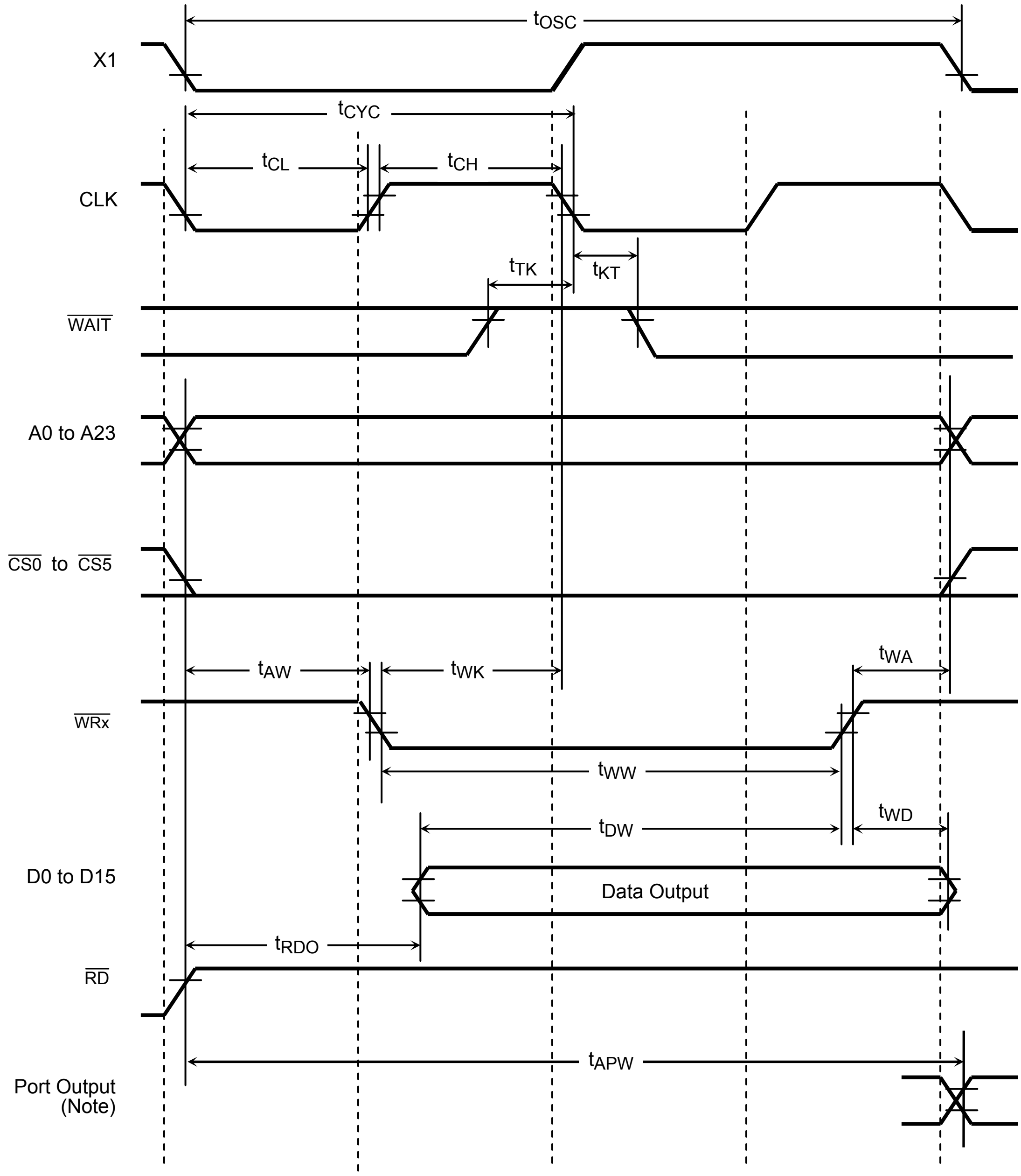
(1) Read cycle (0 waits)



Note 1: The phase relation between X1 input signal and the other signals is unsettled. The timing chart above is an example.

Note 2: Since the CPU accesses the internal area to read data from a port, the control signals of external pins such as $\overline{RD}$ and $\overline{CS}$ are not enabled. Therefore, the above waveform diagram should be regarded as depicting internal operation. Please also note that the timing and AC characteristics of port input/output shown above are typical representation. For details, contact your local Toshiba sales representative.

(2) Write cycle (0 waits)

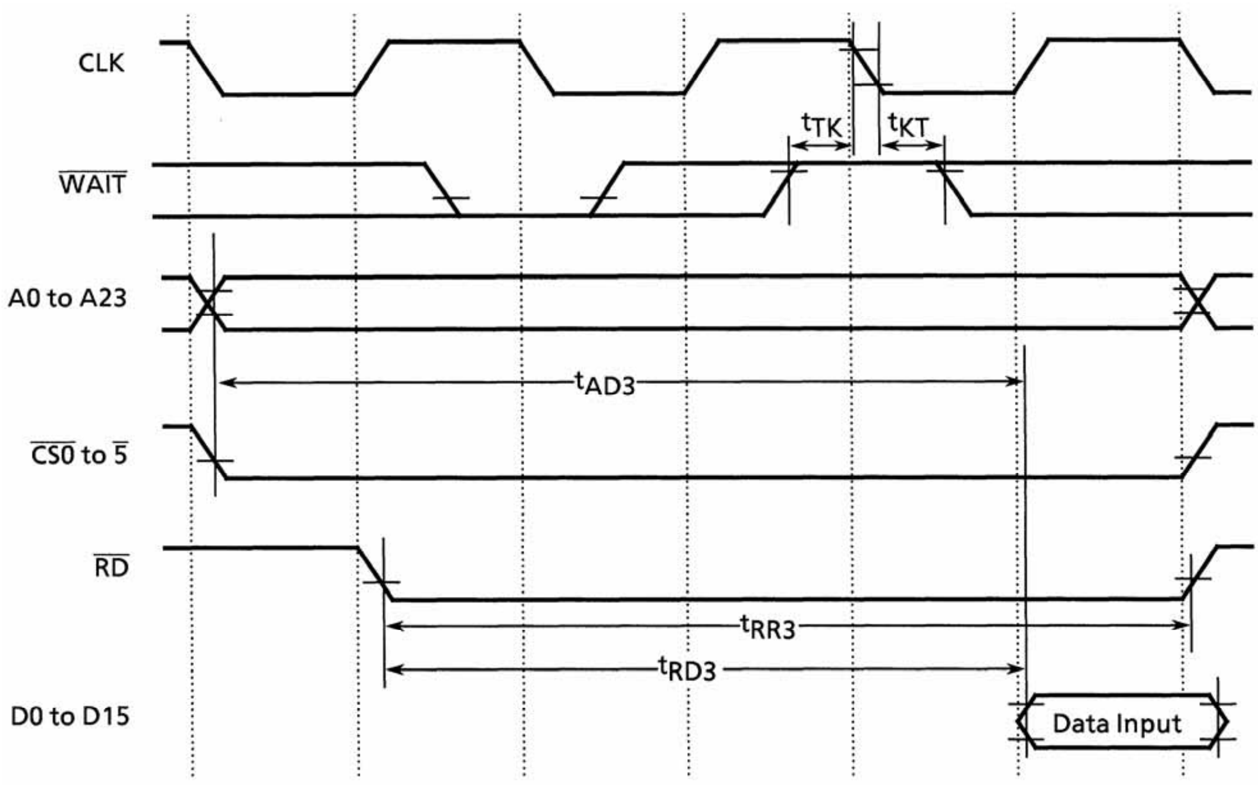


Note 1: The phase relation between X1 input signal and the other signals is unsettled. The timing chart above is an example.

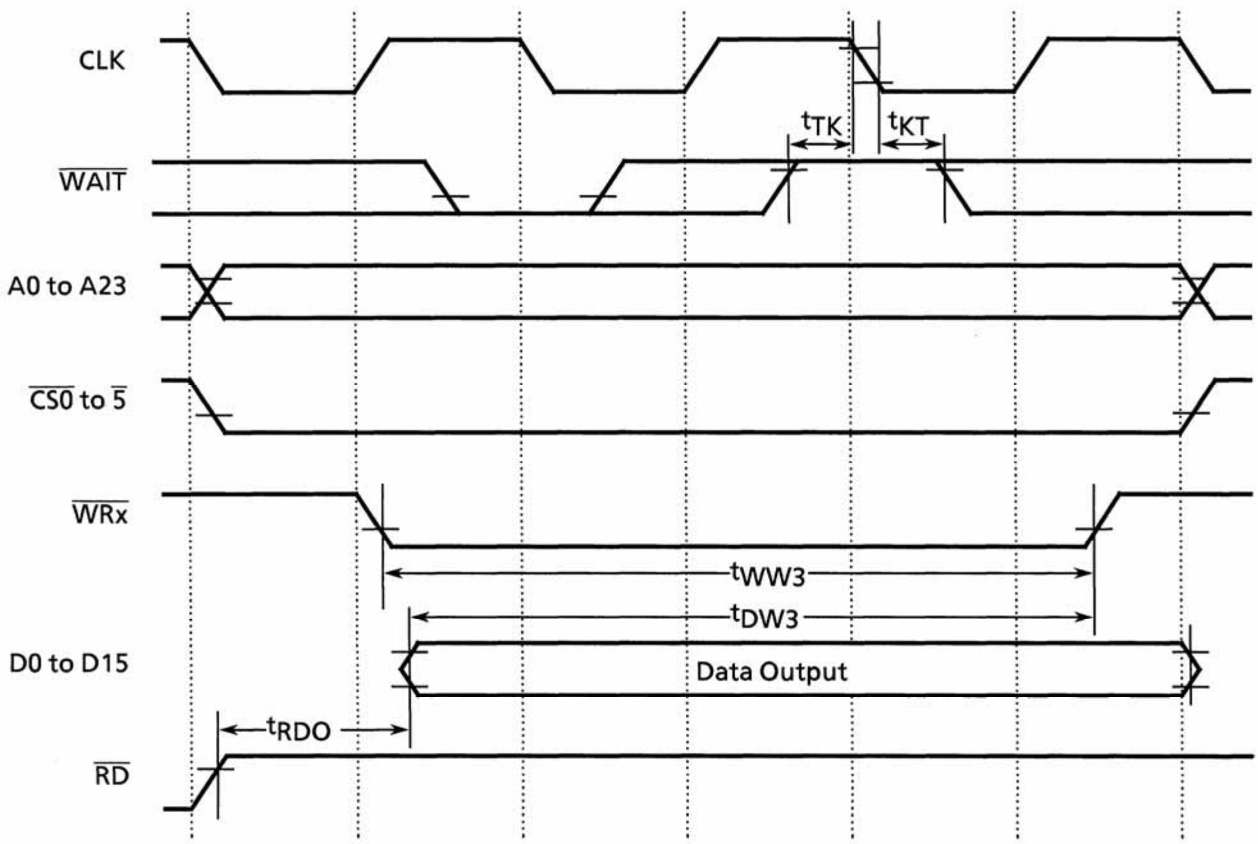
Note 2: $\overline{WRx}$ shows $\overline{WRL}$, $\overline{WRH}$.

Note 3: Since the CPU accesses the internal area to write data to a port, the control signals of external pins such as $\overline{WR}$ and $\overline{CS}$ are not enabled. Therefore, the above waveform diagram should be regarded as depicting internal operation. Please also note that the timing and AC characteristics of port input/output shown above are typical representation. For details, contact your local Toshiba sales representative.

(3) Read cycle (1 wait)



(4) Write cycle (1 wait)



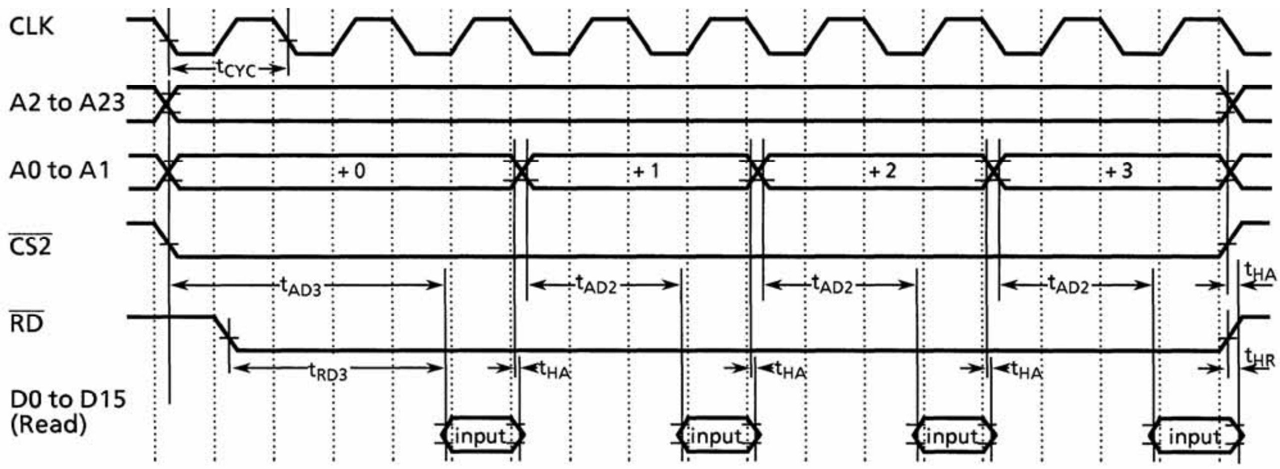
4.3.2 Page ROM Read Cycle

(1) 3-2-2-2 mode

No.	Symbol	Parameter	Min	Max	@20 MHz	@16 MHz	Unit
1	t_{CYC}	System Clock Period (= T)	50	62.5	50	62.5	ns
2	t_{AD2}	A0, A1 → D0 to D15 Input		1.0 × T-50	50	75	ns
3	t_{AD3}	A2 to A23 → D0 to D15 Input		3.0 × T-50	100	138	ns
4	t_{RD3}	$\overline{RD}$ Fall → D0 to D15 Input		2.5 × T-45	80	111	ns
5	t_{HA}	A0 to A23 Invalid → D0 to D15 Hold	0		0	0	ns
6	t_{HR}	$\overline{RD}$ Rise → D0 to D15 Hold	0		0	0	ns

AC Condition
Output: P4 to P6 (A0 to A23), P70 ($\overline{RD}$)
High = 2.0 V, Low = 0.8 V, CL = 50 pF
CLK, P82 ($\overline{CS2}$)
High = 2.0 V, Low = 0.8 V, CL = 50 pF
Input: P0 to P1 (D0 to D15)
High = 2.4 V, Low = 0.45 V

(2) Page ROM read cycle (3-2-2-2 mode)



4.3.3 DRAM Bus Cycle

No.	Symbol	Parameter	Min	Max	@ 20 MHz	@ 16 MHz	Unit
1	t_{CYC}	System Clock Period (= T)	50	62.5	50	62.5	ns
2	t_{RC}	$\overline{RAS}$ Cycle Time	3.00 × T		150	188	ns
3	t_{PC}	Page Mode Cycle Time	2.00 × T		100	125	ns
4-1	t_{RAC}	$\overline{RAS}$ Access Time		1.75 × T – 45	43	64	ns
4-2	t_{RAC4}	$\overline{RAS}$ Access Time @ 4 Clock Access		2.75 × T – 45	93	127	ns
5	t_{CAC}	$\overline{CAS}$ Access Time		1.00 × T – 40	10	23	ns
6-1	t_{AA}	Column Address Access Time		1.25 × T – 45	18	33	ns
6-2	t_{AA2}	Column Address Access Time @ Page Mode		2.00 × T – 45	55	80	ns
6-3	t_{AA4}	Column Address Access Time @ 4 Clock Access		2.25 × T – 45	68	96	ns
7	t_{CPA}	$\overline{CAS}$ Pre-charge Access Time		2.00 × T – 45	55	80	ns
8	t_{OFF}	Input Data Hold Time	0		0	0	ns
9	t_{RP}	$\overline{RAS}$ Pre-charge Time	1.25 × T – 20		43	58	ns
10-1	t_{RAS}	$\overline{RAS}$ Width	1.75 × T – 20		68	89	ns
10-2	t_{RAS4}	$\overline{RAS}$ Width @ 4 Clock Access	2.75 × T – 20		118	152	ns
11	t_{RSH}	$\overline{RAS}$ Hold Time	1.00 × T – 20		30	43	ns
12	t_{RHCP}	$\overline{CAS}$ Pre-charge to $\overline{RAS}$ Hold Time	2.00 × T – 20		80	105	ns
13-1	t_{CSH}	$\overline{CAS}$ Hold Time	1.75 × T – 20		68	89	ns
13-2	t_{CSH4}	$\overline{CAS}$ Hold Time @ 4 Clock Access	2.75 × T – 20		118	152	ns
14	t_{CAS}	$\overline{CAS}$ Width	1.00 × T – 20		30	43	ns
15	t_{RCD}	$\overline{RAS}$ – $\overline{CAS}$ Delay Time	0.75 × T – 17		21	30	ns
16	t_{RAD}	$\overline{RAS}$ – Column Address Delay Time		0.50 × T + 20	45	51	ns
17	t_{CRP}	$\overline{CAS}$ – $\overline{RAS}$ Pre-charge Time	1.25 × T – 20		43	58	ns
18-1	t_{CP}	$\overline{CAS}$ Pre-charge Time @ Refresh	0.50 × T – 15		10	16	ns
18-2	t_{CP2}	$\overline{CAS}$ Pre-charge Time @ Page Mode	1.00 × T – 20		30	43	ns
19	t_{ASR}	Row Address Set-up Time	1.25 × T – 40		23	38	ns
20	t_{RAH}	Row Address Hold Time	0.50 × T – 15		10	16	ns
21-1	t_{ASC}	Column Address Set-up Time	0.25 × T – 12		1	4	ns
21-2	t_{ASC2}	Column Address Set-up Time @ Page Mode	1.00 × T – 20		30	43	ns
22	t_{CAH}	Column Address Hold Time	1.00 × T – 20		30	43	ns
23	t_{AR}	Column Address Hold Time from $\overline{RAS}$	1.75 × T – 20		68	89	ns
24	t_{RAL}	Column Address $\overline{RAS}$ Read Time	1.25 × T – 20		43	58	ns
25	t_{RCS}	Read Command Set-up Time	2.00 × T – 40		60	85	ns
26	t_{RCH}	Read Command Hold Time from $\overline{CAS}$	0.50 × T – 20		5	11	ns
27	t_{RRH}	Read Command Hold Time from $\overline{RAS}$	0.50 × T – 20		5	11	ns
28	t_{WCH}	Write Command Hold Time	1.00 × T – 20		30	43	ns
29	t_{WCR}	Write Command Hold Time from $\overline{RAS}$	1.75 × T – 20		68	89	ns
30	t_{WP}	Write Command Time	1.50 × T – 20		55	74	ns
31	t_{RWL}	Write Command $\overline{RAS}$ Read Time	1.50 × T – 20		55	74	ns
32	t_{CWL}	Write Command $\overline{CAS}$ Read Time	1.50 × T – 20		55	74	ns
33	t_{DS}	Data Output Set-up Time	1.50 × T – 30		45	58	ns

No.	Symbol	Parameter	Min	Max	@20 MHz	@16 MHz	Unit
34	t_{DH}	Data Output Hold Time	1.00×T-25		25	38	ns
35	t_{DHR}	Data Output Hold Time from $\overline{RAS}$	1.75×T-5		83	104	ns
36	t_{WCS}	Write Command Set-up Time	0.50×T-20		5	11	ns
37	t_{CSR}	$\overline{CAS}$ Set-up Time	0.75×T-20		18	27	ns
38	t_{CHR}	$\overline{CAS}$ Hold Time	1.75×T-20		68	89	ns
39	t_{RPC}	$\overline{RAS}$ Pre-charge $\overline{CAS}$ Active Time	0.50×T-20		5	11	ns
40	t_{ROH}	$\overline{RAS}$ Hold Time from $\overline{OE}$	1.00×T-20		30	43	ns
41	t_{OEA}	$\overline{OE}$ Access Time		1.00×T-40	10	23	ns
42	t_{OEZ}	Input Data Hold Time from $\overline{OE}$	0		0	0	ns
43	t_{RPS}	$\overline{RAS}$ Pre-charge Time @ Release Self Refresh Cycle	2.25×T-20		93	121	ns
44	t_{CHS}	$\overline{CAS}$ Hold Time @ Release Self Refresh Cycle	– 15		– 15	– 15	ns

AC Condition

Output: P0 to P1 (D0 to D15), P4 to P6 (A0 to A23), P70 ($\overline{RD}$), P71 to P74 ($\overline{WRx}$)

High = 2.0 V, Low = 0.8 V, CL = 50 pF

Others

High = 2.0 V, Low = 0.8 V, CL = 50 pF

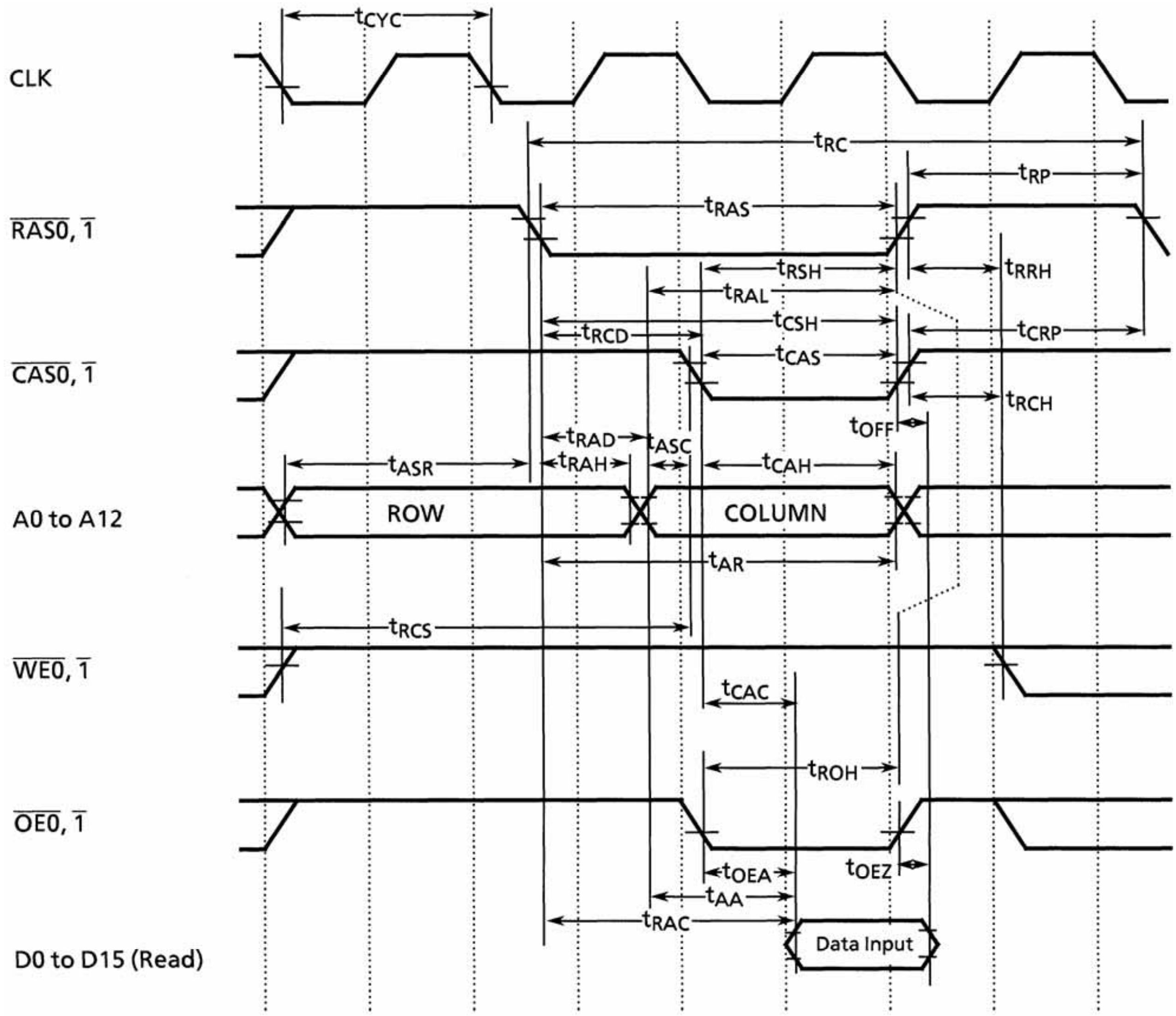
Input: P0 to P1 (D0 to D15)

High = 2.4 V, Low = 0.45 V

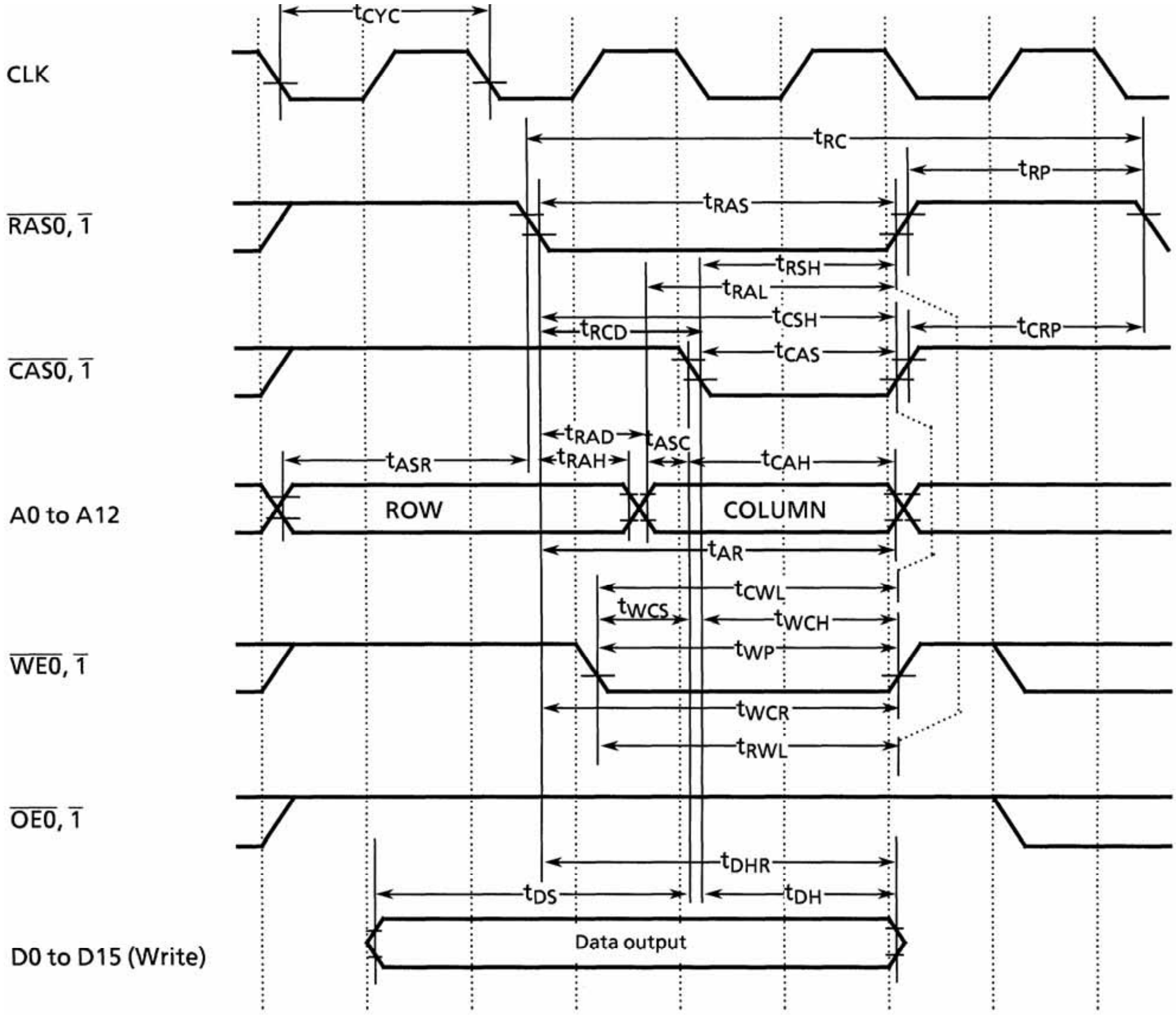
Others

High = 0.8 Vcc, Low = 0.2 Vcc

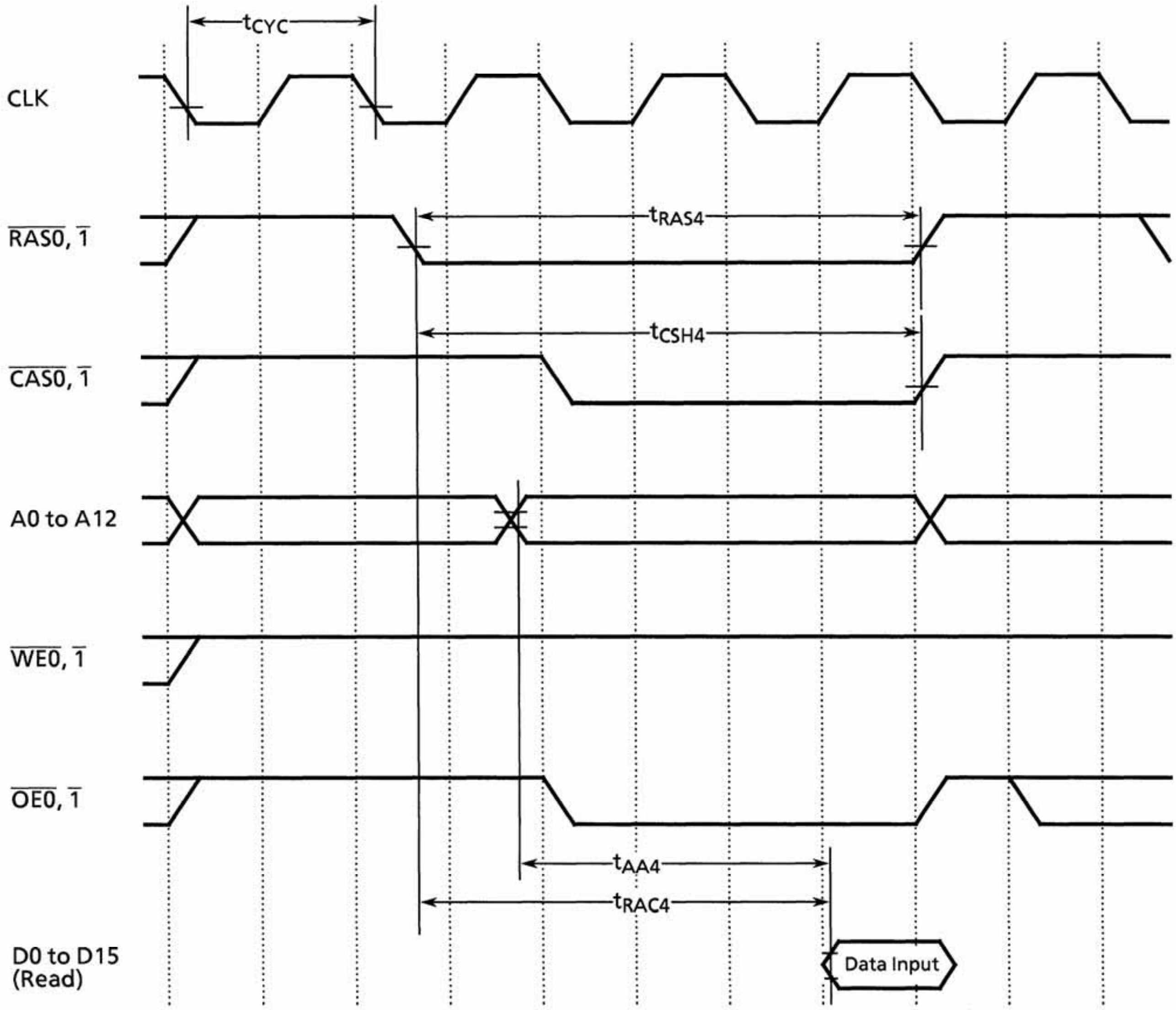
(1) DRAM read cycle (3 clock access)



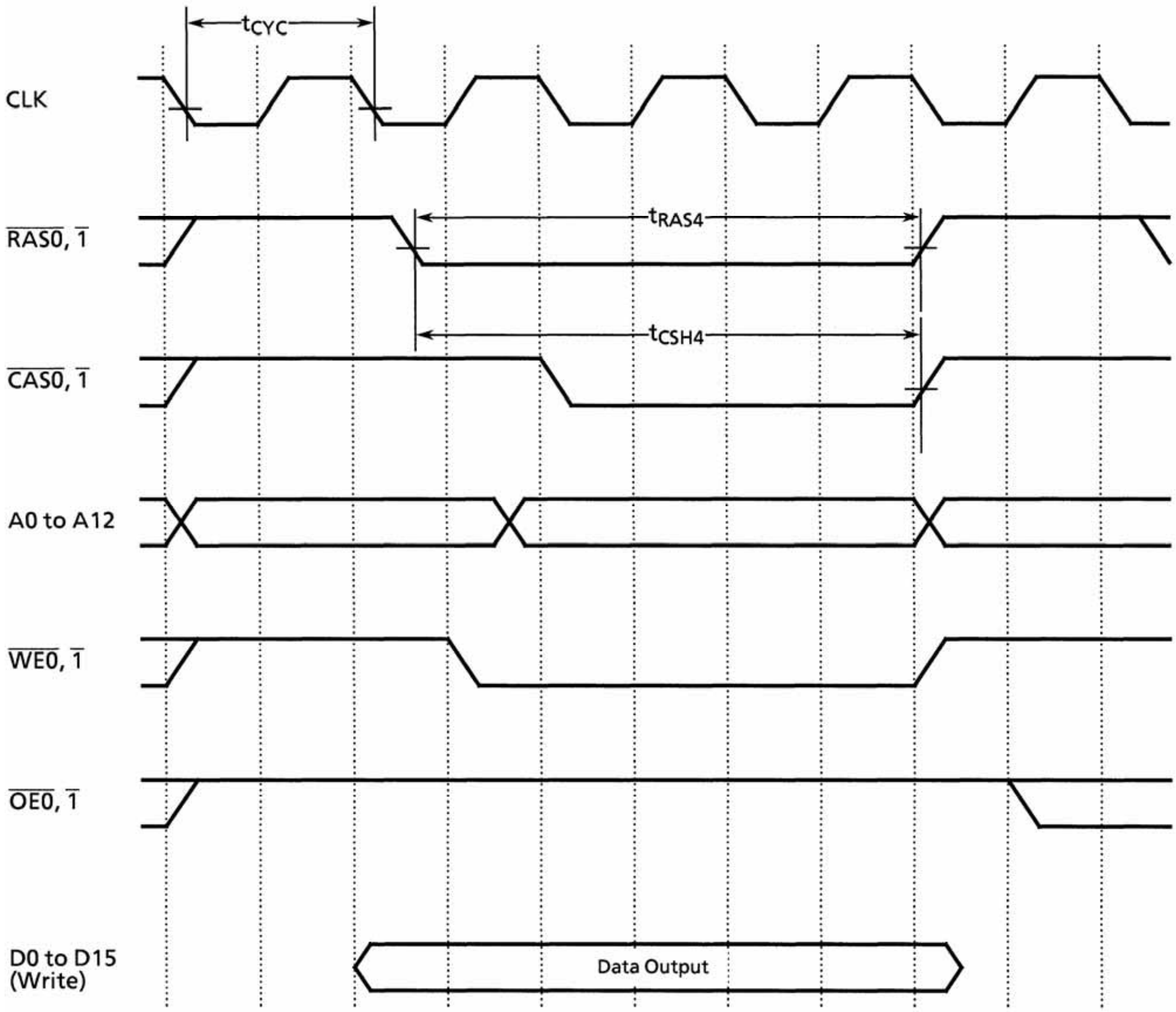
(2) DRAM write cycle (3 clock access)



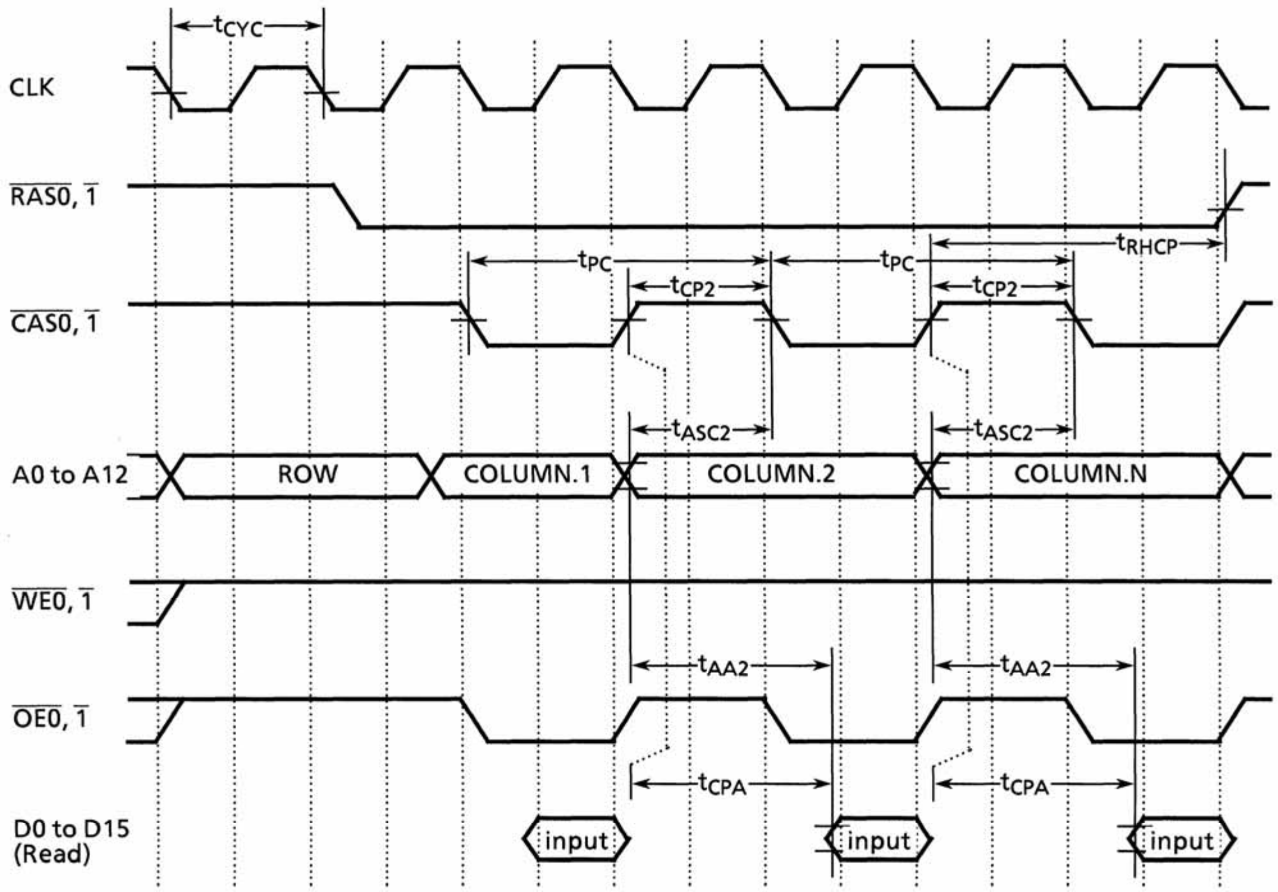
(3) DRAM read cycle (4 clock access)



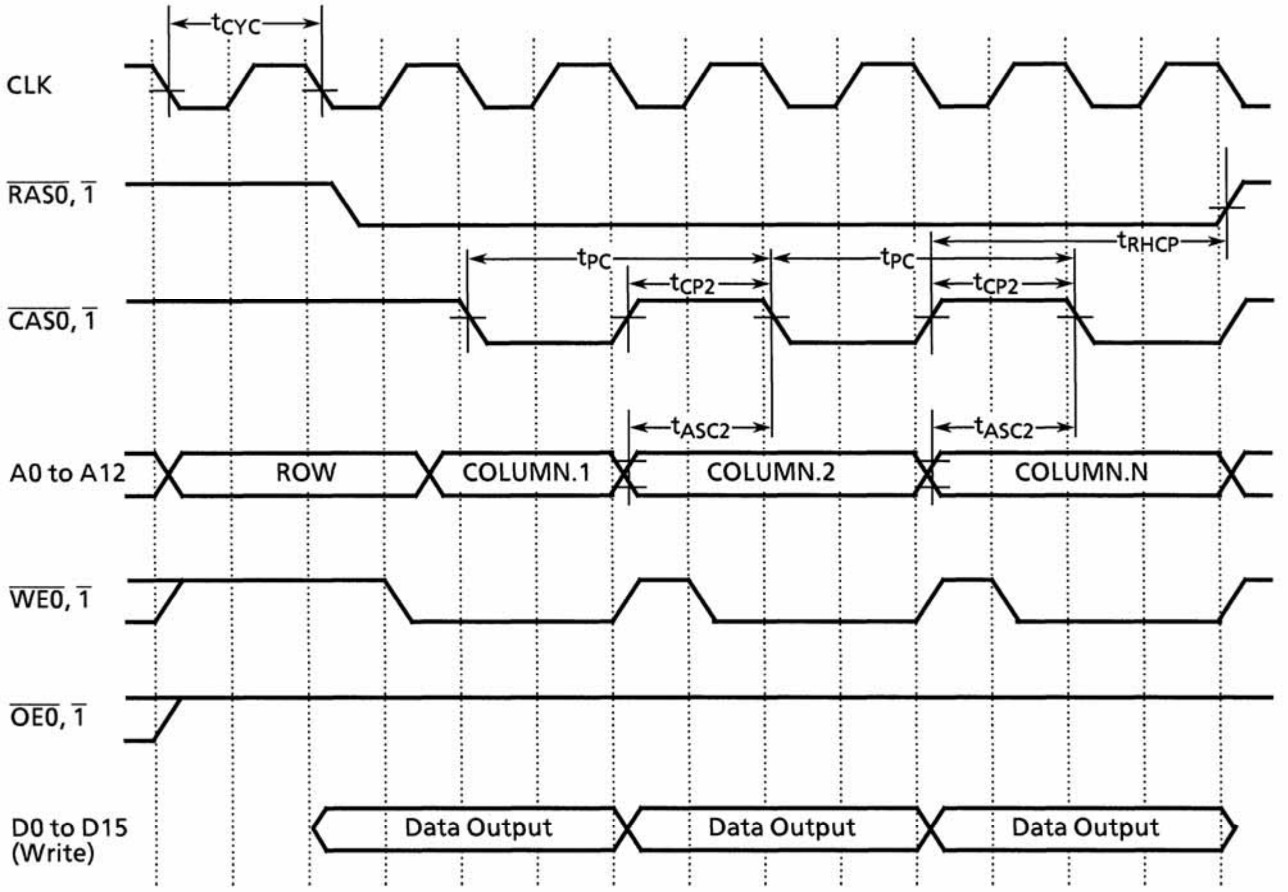
(4) DRAM write cycle (4 clock access)



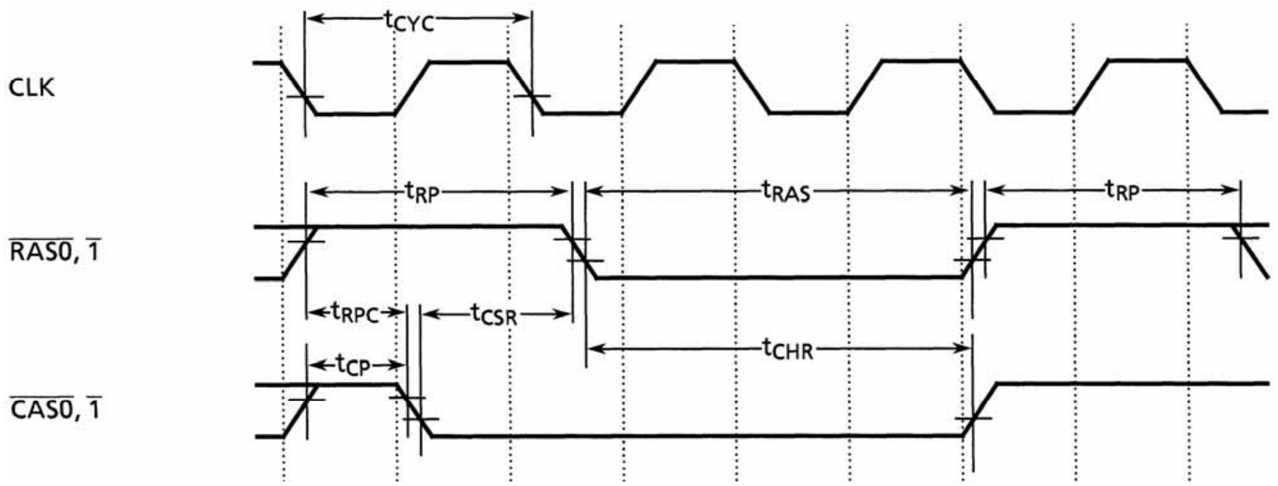
(5) DRAM page mode read cycle (3-2-2-2 mode)



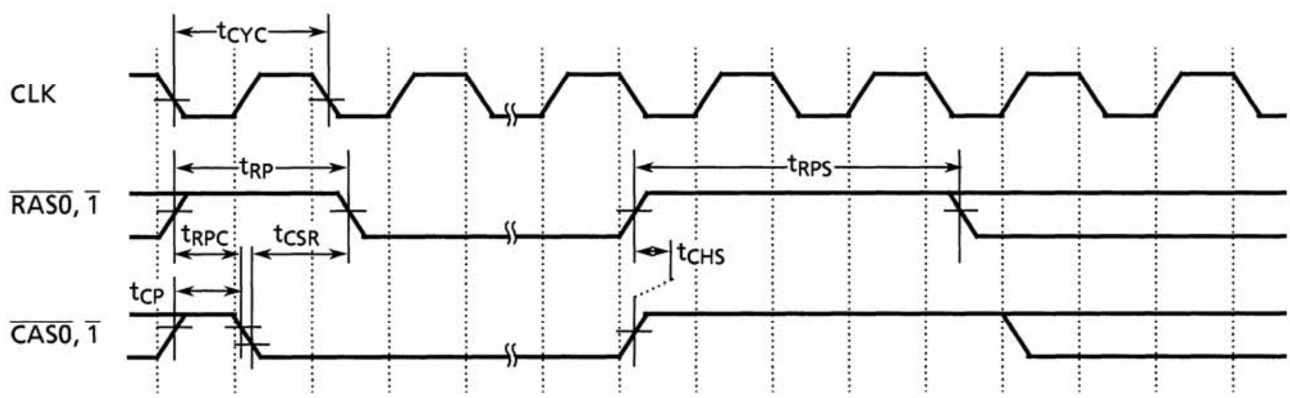
(6) DRAM page mode write cycle (3-2-2-2 mode)



(7) DRAM $\overline{\text{CAS}}$ before $\overline{\text{RAS}}$ interval refresh cycle (3 cycle mode)



(8) DRAM $\overline{\text{CAS}}$ before $\overline{\text{RAS}}$ self refresh cycle



4.4 Event Counter (TI4, TI5, TI6, TI7, TI8, TI9, TIA, TIB)

V_{CC} = 5 V ± 10%, TA = −20 to 70℃ (Internal 16 to 20 MHz)

Symbol	Parameter	Variable		20 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{VCK}	Clock cycle	8T + 100		500		600		ns
t_{VCKL}	Clock low-level pulse width	4T + 40		240		290		ns
t_{VCKH}	Clock high-level pulse width	4T + 40		240		290		ns

4.5 Serial Channel Timing

(1) SCLK input mode (I/O interface mode)

V_{CC} = 5 V ± 10%, TA = −20 to 70℃ (Internal 16 to 20 MHz)

Symbol	Parameter	Variable		20 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{SCY}	SCLK cycle	16T		0.8		1.0		μs
t_{OSS}	Output Data → Rising edge of SCLK	t_{SCY}/2 − 5T − 50		100		138		ns
t_{OHS}	SCLK rising edge → Output Data hold	5T − 100		150		213		ns
t_{HSR}	SCLK rising edge → Input Data hold	0		0		0		ns
t_{SRD}	SCLK rising edge → effective data input		t_{SCY} − 5T − 100		450		588	ns

(2) SCLK output mode (I/O interface mode)

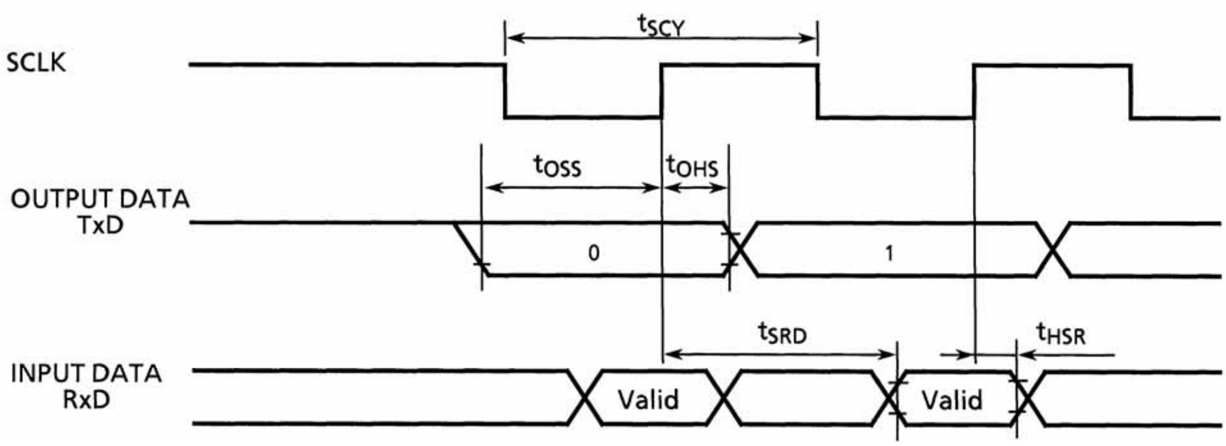
V_{CC} = 5 V ± 10%, TA = −20 to 70℃ (Internal 16 to 20 MHz)

Symbol	Parameter	Variable		20 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{SCY}	SCLK cycle (programmable)	16T	8192T	0.8	409.6	1.0	512	μs
t_{OSS}	Output Data → SCLK rising edge	t_{SCY} − 2T − 150		550		725		ns
t_{OHS}	SCLK rising edge → Output Data hold	2T − 80		20		45		ns
t_{HSR}	SCLK rising edge → Input Data hold	0		0		0		ns
t_{SRD}	SCLK rising edge → effective data input		t_{SCY} − 2T − 150		550		725	ns

(3) SCLK input mode (UART mode)

V_{CC} = 5 V ± 10%, TA = −20 to 70℃ (Internal 16 to 20 MHz)

Symbol	Parameter	Variable		20 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{SCY}	SCLK cycle	4T + 20		220		270		ns
t_{SCYL}	SCLK Low level Pulse width	2T + 5		105		130		ns
t_{SCYH}	SCLK High level Pulse width	2T + 5		105		130		ns



4.6 10-Bit AD Conversion Characteristics

Vcc = 5 V ± 10%, TA = – 20 to 70°C (Internal 16 to 20 MHz)

Symbol	Parameter		Min	Typ	Max	Unit
VREFH	Analog reference voltage (High)		V_{CC}-0.2 V	V_{CC}	V_{CC}	V
VREFL	Analog reference voltage (Low)		V_{SS}	V_{SS}	V_{SS} + 0.2 V	
VAIN	Analog input voltage range		VREFL		VREFH	
I_{REF} (VREFL = 0 V)	Analog current for analog reference voltage V_{CC} = 5V ± 10%	<VREFON> = 1		0.5	1.5	mA
	V_{CC} = 5V ± 10%	<VREFON> = 0		0.02	5.0	μA
Error (Quantize error of ± 0.5 LSB not included)	V_{CC} = 5V ± 10%	Total error		± 3.0	± 6	LSB

Note 1: 1LSB = (VREFH – VREFL)/1024 [V]

Note 2: Power supply current ICC from the digital power supply includes the power supply from the AVCC pin.

4.7 8-Bit DA Conversion Characteristics

Vcc = 5 V ± 10%, TA = –20 to 70°C (Internal 16 to 20 MHz)

Symbol	Parameter	Condition	Min	Typ	Max	Unit
DAREFH	Analog reference voltage (+)		4.0		V_{CC}	V
DAREFL	Analog reference voltage (–)		V_{SS}		V_{SS}	
	Total error	RL = 2.4 KΩ		2.0	4.0	LSB
	Output voltage range	RL = 2.4 KΩ	V_{SS} + 0.5		V_{SS} – 0.5	V
	Settling time	RL = 2.4 KΩ, CL = 100 pF			5	μs
DAC output mode	Output impedance				5	Ω
	Resistive load	V_{SS} + 0.5 ≤ DAOUT ≤ V_{CC} – 0.5	2.4			kΩ

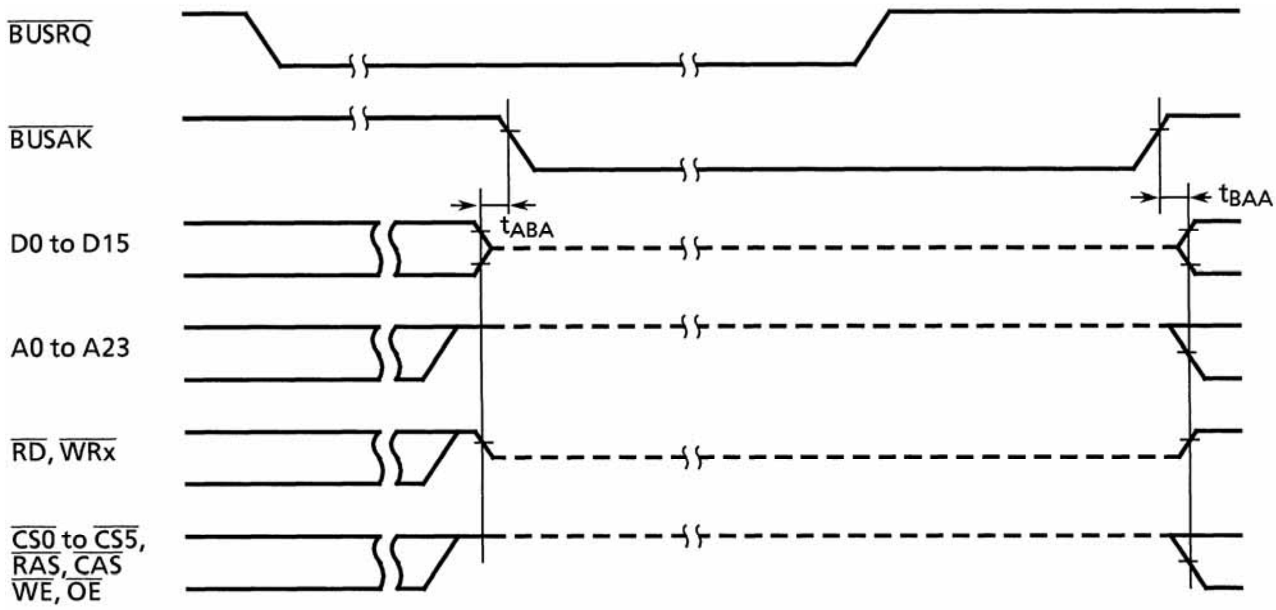
Note: RL is the resistance load of the DA converter output in.

4.8 Interrupt Operation

Vcc = 5 V ± 10%, TA = – 20 to 70°C (Internal 16 to 20 MHz)

Symbol	Parameter	Variable		20 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{INTAL}	$\overline{NMI}$, INT0 Low level Pulse width	4T		200		250		ns
t_{INTAH}	$\overline{NMI}$, INT0 High level Pulse width	4T		200		250		ns
t_{INTBL}	INT4 toINTB Low level Pulse width	8T + 100		500		600		ns
t_{INTBH}	INT4 to INTB High level Pulse width	8T + 100		500		600		ns

4.9 Bus Request/Bus Acknowledge Timing



Vcc = 5 V ± 10%, TA = − 20 to 70°C (Internal 16 to 20 MHz)

Symbol	Parameter	Variable		20 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{ABA}	Floating time to $\overline{BUSAK}$ fall	0	80	0	80	0	80	ns
t_{BAA}	Floating time to $\overline{BUSAK}$ rise	0	80	0	80	0	80	ns

Note: The bus will be released after the WAIT request is inactive, when the $\overline{BUSRQ}$ is set to "Low" during "wait" cycle.

5. Table of Special Function Registers (SFRs) (SFR: Special function register)

The special function registers (SFRs) include the I/O ports and peripheral control registers allocated to the 1024 byte addresses from 000000H to 0003FFH.

(1) I/O port

(2) Timer control

(3) Watchdog timer control

(4) Clock control

(5) Serial channel control

(6) AD converter control

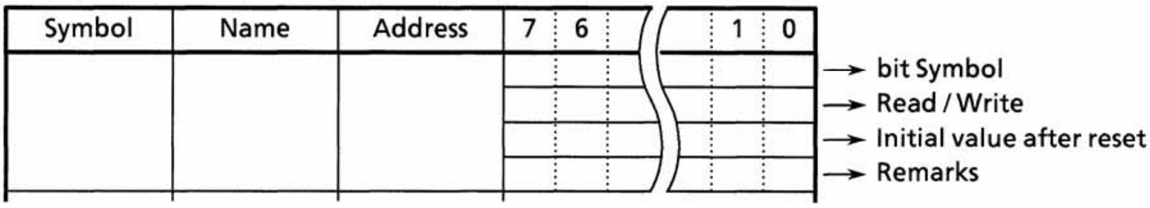
(7) DA converter control

(8) Interrupt control

(9) Memory control

(10) DRAM control

Configuration of the table



Explanations of symbols

R/W: Either read or write is possible

R: Only read is possible

W: Only write is possible

W*: Either read or write is possible (Always read as “1”)

1*: Always read as “1”

No RMW: Prohibit read-modify-write.
(Prohibit RES/SET/TSET/CHG/STCF/ANDCF/ORCF/XORCF etc.)

Table 5.1 I/O Register Address Map

Address	Name	Address	Name	Address	Name	Address	Name
TLCS-900/H2 type 8 bit I/O							
00h	P0	01h	••••	02h	P0CR	03h	P0FC
04h	P1	05h	••••	06h	P1CR	07h	P1FC
08h	••••	09h	••••	0Ah	••••	0Bh	••••
0Ch	••••	0Dh	••••	0Eh	••••	0Fh	••••
10h	P4	11h	••••	12h	P4CR	13h	P4FC
14h	P5	15h	••••	16h	P5CR	17h	P5FC
18h	P6	19h	••••	1Ah	P6CR	1Bh	P6FC
1Ch	P7	1Dh	••••	1Eh	P7CR	1Fh	P7FC
20h	P8	21h	••••	22h	P8CR	23h	P8FC
24h	••••	25h	••••	26h	••••	27h	••••
28h	PA	29h	••••	2Ah	••••	2Bh	PAFC
2Ch	PB	2Dh	••••	2Eh	••••	2Fh	PBFC
30h	PC	31h	••••	32h	PCCR	33h	PCFC
34h	PD	35h	••••	36h	PDCR	37h	PDFC
38h	PE	39h	••••	3Ah	PECR	3Bh	PEFC
3Ch	PF	3Dh	••••	3Eh	PFCR	3Fh	PFFC
40h	PG	41h	••••	42h	••••	43h	••••
44h	PH	45h	••••	46h	PHCR	47h	PHFC
68h	PZ	69h	••••	6Ah	PZCR	6Bh	••••
TLCS-90 type I/O							
80h	T8RUN	81h	TRDC	82h	T02FFCR	83h	••••
84h	T01MOD	85h	T23MOD	86h	••••	87h	••••
88h	TREG0	89h	TREG1	8Ah	TREG2	8Bh	TREG3
8Ch	••••	8Dh	••••	8Eh	••••	8Fh	••••
90h	TREG4L	91h	TREG4H	92h	TREG5L	93h	TREG5H
94h	CAP4L	95h	CAP4H	96h	CAP5L	97h	CAP5H
98h	T4MOD	99h	T4FFCR	9Ah	••••	9Bh	••••
9Ch	••••	9Dh	••••	9Eh	T16RUN	9Fh	T16CR
A0h	TREG6L	A1h	TREG6H	A2h	TREG7L	A3h	TREG7H
A4h	CAP6L	A5h	CAP6H	A6h	CAP7L	A7h	CAP7H
A8h	T6MOD	A9h	T6FFCR	AAh	••••	ABh	••••
ACh	••••	ADh	••••	AEh	••••	AFh	••••
B0h	TREG8L	B1h	TREG8H	B2h	TREG9L	B3h	TREG9H
B4h	CAP8L	B5h	CAP8H	B6h	CAP9L	B7h	CAP9H
B8h	T8MOD	B9h	T8FFCR	BAh	••••	BBh	••••
BCh	••••	BDh	••••	BEh	••••	BFh	••••
C0h	TREGAL	C1h	TREGAH	C2h	TREGBL	C3h	TREGBH
C4h	CAPAL	C5h	CAPAH	C6h	CAPBL	C7h	CAPBH
C8h	TAMOD	C9h	TAFFCR	CAh	••••	CBh	••••
CCh	••••	CDh	••••	CEh	••••	CFh	••••
D0h	SC0BUF	D1h	SC0CR	D2h	SC0MOD	D3h	BR0CR
D4h	SC1BUF	D5h	SC1CR	D6h	SC1MOD	D7h	BR1CR
D8h	••••	D9h	••••	DAh	••••	DBh	••••
DCh	••••	DDh	••••	DEh	••••	DFh	••••
TLCS-900/H2 type 8 bit I/O							
E0h	INTE45	E1h	INTE67	E2h	INTE89	E3h	INTEAB
E4h	INTET01	E5h	INTET23	E6h	INTET45	E7h	INTET67
E8h	INTET89	E9h	INTETAB	EAh	INTES0	EBh	INTES1
ECh	INTETC01	EDh	INTETC23	EEh	INTETC45	EFh	INTETC67
F0h	INTE0AD	F1h	••••	F2h	••••	F3h	••••
F4h	••••	F5h	••••	F6h	IIMC	F7h	INTNMWDT
F8h	INTCLR	F9h	(reserved)	FAh	••••	FBh	••••
FCh	(reserved)	FDh	(reserved)	FEh	(reserved)	FFh	(reserved)
100h	DMA0V	101h	DMA1V	102h	DMA2V	103h	DMA3V
104h	DMA4V	105h	DMA5V	106h	DMA6V	107h	DMA7V
108h	DMAB	109h	DMAR	10Ah	CLKMOD	10Bh	(reserved)
10Ch	••••	10Dh	••••	10Eh	••••	10Fh	••••

Address	Name	Address	Name	Address	Name	Address	Name
TLCS-90 type I/O							
110h	WDMOD	111h	WDCR	112h	······	113h	······
114h	······	115h	······	116h	······	117h	······
118h	······	119h	······	11Ah	······	11Bh	······
11Ch	······	11Dh	······	11Eh	······	11Fh	······
120h	ADREG04L	121h	ADREG04H	122h	ADREG15L	123h	ADREG15H
124h	ADREG26L	125h	ADREG26H	126h	ADREG37L	127h	ADREG37H
128h	ADMOD1	129h	ADMOD2	12Ah	(reserved)	12Bh	······
12Ch	······	12Dh	······	12Eh	······	12Fh	······
130h	DAREG0	131h	DAREG1	132h	DADRV	133h	······
134h	······	135h	······	136h	······	137h	······
138h	······	139h	······	13Ah	······	13Bh	······
13Ch	······	13Dh	······	13Eh	······	13Fh	······
TLCS-900/H2 type 8 bit I/O							
140h	B0CSL	141h	B0CSH	142h	MAMR0	143h	MSAR0
144h	B1CSL	145h	B1CSH	146h	MAMR1	147h	MSAR1
148h	B2CSL	149h	B2CSH	14Ah	MAMR2	14Bh	MSAR2
14Ch	B3CSL	14Dh	B3CSH	14Eh	MAMR3	14Fh	MSAR3
150h	B4CSL	151h	B4CSH	152h	MAMR4	153h	MSAR4
154h	B5CSL	155h	B5CSH	156h	MAMR5	157h	MSAR5
158h	······	159h	······	15Ah	······	15Bh	······
15Ch	DRAMEXT	15Dh	······	15Eh	······	15Fh	······
160h	DRAM0CRL	161h	DRAM0CRH	162h	DRAM1CRL	163h	DRAM1CRH
164h	DRAM0REF	165h	DRAM1REF	166h	PMEMCR	167h	······

Note 1: TLCS-900/H2 type I/Os are always accessed by two clocks (100 ns at 20 MHz)

Note 2: TLCS-90 type I/Os are accessed by five clocks min. (250 ns at 20 MHz) and eight clocks max. (400 ns at 20 MHz)

(1) Input/output ports

Port 0

Symbol	Name	Address	7	6	5	4	3	2	1	0
P0	PORT0	00h	P07	P06	P05	P04	P03	P02	P01	P00
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
P0CR	PORT0 Control Register	02h (no RMW)	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P0FC	PORT0 Function Register	03h (no RMW)	–	–	–	–	–	–	–	P0F
			W							
			–	–	–	–	–	–	–	1
			0: PORT 1: Data Bus (D7 to D0)							

Port 1

Symbol	Name	Address	7	6	5	4	3	2	1	0
P1	PORT1	04h	P17	P16	P15	P14	P13	P12	P11	P10
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
P1CR	PORT1 Control Register	06h (no RMW)	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P1FC	PORT1 Function Register	07h (no RMW)	–	–	–	–	–	–	–	P1F
			W							
			–	–	–	–	–	–	–	0/1
			0: PORT 1: Data Bus (D15 to D8)							

Port 4

Symbol	Name	Address	7	6	5	4	3	2	1	0
P4	PORT4	10h	P47	P46	P45	P44	P43	P42	P41	P40
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
P4CR	PORT4 Control Register	12h (no RMW)	P47C	P46C	P45C	P44C	P43C	P42C	P41C	P40C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P4FC	PORT4 Function Register	13h (no RMW)	P47F	P46F	P45F	P44F	P43F	P42F	P41F	P40F
			W							
			1	1	1	1	1	1	1	1
			0: PORT 1: Address Bus (A7 to A0)							

Port 5

Symbol	Name	Address	7	6	5	4	3	2	1	0
P5	PORT5	14h	P57	P56	P55	P54	P53	P52	P51	P50
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
P5CR	PORT5 Control Register	16h	P57C	P56C	P55C	P54C	P53C	P52C	P51C	P50C
			W							
			0	0	0	0	0	0	0	0
		(no RMW)	0: Input 1: Output							
P5FC	PORT5 Function Register	17h	P57F	P56F	P55F	P54F	P53F	P52F	P51F	P50F
			W							
			1	1	1	1	1	1	1	1
		(no RMW)	0: PORT 1: Address Bus (A15 to A8)							

Port 6

Symbol	Name	Address	7	6	5	4	3	2	1	0
P6	PORT6	18h	P67	P66	P65	P64	P63	P62	P61	P60
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
P6CR	PORT6 Control Register	1Ah	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C
			W							
			0	0	0	0	0	0	0	0
		(no RMW)	0: Input 1: Output							
P6FC	PORT6 Function Register	1Bh	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
			W							
			1	1	1	1	1	1	1	1
		(no RMW)	0: PORT 1: Address Bus (A23 to A16)							

Port 7

Symbol	Name	Address	7	6	5	4	3	2	1	0
P7	PORT7	1Ch	–	P76	P75	P74	P73	P72	P71	P70
				R/W						
			–	1	1	1	1	1	1	1
				Output	In/Out	Output				
P7CR	PORT7 Control Register	1Eh	–	–	P75C	–	–	–	–	–
					W					
			–	–	0	–	–	–	–	–
		(no RMW)	0: Input 1: Output							
P7FC	PORT7 Function Register	1Fh	–	P76F	P75F	P74F	P73F	P72F	P71F	P70F
				W						
			–	0	0	0	0	0	0	1
		(no RMW)		0: PORT 1: $\overline{BUSAK}$	0: PORT 1: $\overline{BUSRQ}$	0: PORT 1: reserved	0: PORT 1: reserved	0: PORT 1: $\overline{WRH}$	0: PORT 1: $\overline{WRL}$	0: PORT 1: $\overline{RD}$

Port 8

Symbol	Name	Address	7	6	5	4	3	2	1	0
P8	PORT8	20h	–	P86	P85	P84	P83	P82	P81	P80
				R/W						
			–	0	1	1	1	0	1	1
				In/Out	Output					
P8CR	PORT7 Control Register	22h (no RMW)	–	P86C	–	–	–	–	–	–
				W						
			–	0	–	–	–	–	–	–
			0: Input 1: Output							
P8FC	PORT8 Function Register	23h (no RMW)	–	P86F	P85F	P84F	P83F	P82F	P81F	P80F
				W						
			–	0	0	0	0	0	0	0
				0: PORT 1: $\overline{\text{WAIT}}$	0: PORT 1: $\overline{\text{CS5}}$	0: PORT 1: $\overline{\text{CS4}}$	0: PORT 1: $\overline{\text{CS3}}$ $\overline{\text{RAS1}}$	0: PORT 1: $\overline{\text{CS2}}$	0: PORT 1: $\overline{\text{CS1}}$ $\overline{\text{RAS0}}$	0: PORT 1: $\overline{\text{CS0}}$

Port A

Symbol	Name	Address	7	6	5	4	3	2	1	0
PA	PORTA	28h	–	–	–	PA4	PA3	PA2	PA1	PA0
						R/W				
			–	–	–	1	1	1	1	1
						Output				
PAFC	PORTA Function Register	2Bh (no RMW)	–	–	–	PA4F	PA3F	PA2F	PA1F	PA0F
						W				
			–	–	–	0	0	0	0	0
						0: PORT 1: $\overline{\text{WE0}}$	0: PORT 1: $\overline{\text{OE1}}$	0: PORT 1: $\overline{\text{OE0}}$	0: PORT 1: $\overline{\text{UCAS0}}$	0: PORT 1: $\overline{\text{CAS0}}$ $\overline{\text{LCAS0}}$

Port B

Symbol	Name	Address	7	6	5	4	3	2	1	0
PB	PORTB	2Ch	–	–	–	PB4	PB3	PB2	PB1	PB0
						R/W				
			–	–	–	1	1	1	1	1
						Output				
PBFC	PORTB Function Register	2Fh (no RMW)	–	–	–	PB4F	PB3F	PB2F	PB1F	PB0F
						W				
			–	–	–	0	0	0	0	0
						0: PORT 1: $\overline{\text{WE1}}$	0: PORT 1: reserved	0: PORT 1: reserved	0: PORT 1: $\overline{\text{UCAS1}}$	0: PORT 1: $\overline{\text{CAS1}}$ $\overline{\text{LCAS1}}$

Port C

Symbol	Name	Address	7	6	5	4	3	2	1	0
PC	PORTC	30h	–	–	–	–	–	–	PC1	PC0
									R/W	
			–	–	–	–	–	–	0	0
									Input/Output	
PCCR	PORTC Control Register	32h (no RMW)	–	–	–	–	–	–	PC1C	PC0C
									W	
			–	–	–	–	–	–	0	0
									(See below)	
PCFC	PORTC Function Register	33h (no RMW)	–	–	–	–	–	–	PC1F	PC0F
									W	
			–	–	–	–	–	–	0	0
									(See below)	

PCFC	PCCR	function	
		PC1	PC0
0	0	Input Port	
0	1	Output Port	
1	0	TO3	TO1
1	1	TOB	TO7

Port D

Symbol	Name	Address	7	6	5	4	3	2	1	0
PD	PORTD	34h	–	PD6	PD5	PD4	–	PD2	PD1	PD0
				R/W				R/W		
			–	0	0	0	–	0	0	0
				Input/Output				Input/Output		
PDCR	PORTD Control Register	36h (no RMW)	–	PD6C	PD5C	PD4C	–	PD2C	PD1C	PD0C
				W				W		
			–	0	0	0	–	0	0	0
				0: Input 1: Output				0: Input 1: Output		
PDFC	PORTD Function Register	37h (no RMW)	–	PD6F	PD5F	PD4F	–	PD2F	PD1F	PD0F
				W				W		
			–	0	0	0	–	0	0	0
				0: PORT 1: TI7 INT7	0: PORT 1: TI6 INT6	0: PORT 1: TO6		0: PORT 1: TI5 INT5	0: PORT 1: TI4 INT4	0: PORT 1: TO4

Port E

Symbol	Name	Address	7	6	5	4	3	2	1	0
PE	PORTE	38h	–	PE6	PE5	PE4	–	PE2	PE1	PE0
				R/W				R/W		
			–	0	0	0	–	0	0	0
				Input/Output				Input/Output		
PECR	PORTE Control Register	3Ah (no RMW)	–	PE6C	PE5C	PE4C	–	PE2C	PE1C	PE0C
				W				W		
			–	0	0	0	–	0	0	0
				0: Input 1: Output				0: Input 1: Output		
PEFC	PORTE Function Register	3Bh (no RMW)	–	PE6F	PE5F	PE4F	–	PE2F	PE1F	PE0F
				W				W		
			–	0	0	0	–	0	0	0
				0: PORT 1: TIB INTB	0: PORT 1: TIA INTA	0: PORT 1: TOA		0: PORT 1: TI9 INT9	0: PORT 1: TI8 INT8	0: PORT 1: TO8

Port F

Symbol	Name	Address	7	6	5	4	3	2	1	0
PF	PORTF	3Ch	–	PF6	PF5	PF4	–	PF2	PF1	PF0
				R/W				R/W		
				0	0	0	–	0	0	0
			–	Input/Output				Input/Output		
PFCR	PORTF Control Register	3Eh (no RMW)	–	PF6C	PF5C	PF4C	–	PF2C	PF1C	PF0C
				W				W		
			–	0	0	0	–	0	0	0
				0: Input 1: Output				0: Input 1: Output		
PFFC	PORTF Function Register	3Fh (no RMW)	–	PF6F	PF5F	PF4F	–	PF2F	PF1F	PF0F
				W				W		
			–	0	0	0	–	0	0	0
				0: PORT 1: $\overline{\text{CTS1}}$ SCLK1	0: PORT 1: RxD1	0: PORT 1: TxD1		0: PORT 1: $\overline{\text{CTS0}}$ SCLK0	0: PORT 1: RxD0	0: PORT 1: TxD0

Port G

Symbol	Name	Address	7	6	5	4	3	2	1	0
PG	PORTG	40h	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
			R							
			Input							

Port H

Symbol	Name	Address	7	6	5	4	3	2	1	0
PH	PORTH	44h	–	–	–	PH4	PH3	PH2	PH1	PH0
						R/W				
			–	–	–	0	0	0	0	0
						Input/Output				
PHCR	PORTH Control Register	46h (no RMW)	–	–	–	PH4C	PH3C	PH2C	PH1C	PH0C
						W				
			–	–	–	0	0	0	0	0
						0: Input 1: Output				
PHFC	PORTH Function Register	47h (no RMW)	–	–	–	PH4F	PH3F	PH2F	PH1F	PH0F
						W				
			–	–	–	0	0	0	0	0
						0: PORT 1: INT0	0: PORT 1: TC3	0: PORT 1: TC2	0: PORT 1: TC1	0: PORT 1: TC0

Port Z

Symbol	Name	Address	7	6	5	4	3	2	1	0
PZ	PORTZ	68h	PZ7	PZ6	PZ5	PZ4	PZ3	PZ2	PZ1	PZ0
			R/W							
			0	0	0	0	0	0	0	0
			Input/Output							
PZCR	PORTZ Control Register	6Ah (no RMW)	PZ7C	PZ6C	PZ5C	PZ4C	PZ3C	PZ2C	PZ1C	PZ0C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							

(2) Timer

8 Bit Timer 01, 23

Symbol	Name	Address	7	6	5	4	3	2	1	0
T8RUN	8 Bit Timer Control	80h	–	–	–	–	T3RUN	T2RUN	T1RUN	T0RUN
							R/W			
			–	–	–	–	0	0	0	0
							8 Bit Timer Run/Stop Control 0: Stop&Clear 1: Run (Countup)			
TREG0	8 Bit Timer Reg. 0	88h (no RMW)	–							
			W							
			Undefined							
TREG1	8BitTimer Reg. 1	89h (no RMW)	–							
			W							
			Undefined							
T01MOD	8 Bit Timer 0, 1 Source CLK & MODE	84h (no RMW)	T01M1	T01M0	PWM01	PWM00	T1CLK1	T1CLK0	T0CLK1	T0CLK0
			W							
			0	0	0	0	0	0	0	0
			00: 8 bit Timer 01: 16 bit Timer 10: 8 bit PPG 11: 8 bit PWM		00: – 01: 2^6-1 10: 2^7-1 11: 2^8-1	PWM cycle	00: TO0TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: Reserved 01: ϕT1 10: ϕT4 11: ϕT16	
TFFCR	8BitTimer Flip-Flop Control	82h **(no RMW)**	TFF3C1	TFF3C0	TFF3IE	TFF3IS	TFF1C1	TFF1C0	TFF1IE	TFF1IS
			W		R/W		W		R/W	
			–	–	0	0	–	–	0	0
			00: Invert TFF3 01: SetT FF3 10: Clear TFF3 11: Don't care		0: Don't 1: TFF3 Invert Enable	Invert 0: T2 1: T3	00: Invert TFF1 01: Set TFF1 10: Clear TFF1 11: Don't care		0: Don't 1: TFF1 Invert Enable	Invert 0: T0 1: T1
TREG2	8 Bit Timer Reg. 2	8Ah (no RMW)	–							
			W							
			Undefined							
TREG3	8 Bit Timer Reg. 3	8Bh (no RMW)	–							
			W							
			Undefined							
T23MOD	8 Bit Timer 2, 3 Source CLK & MODE	85h (no RMW)	T23M1	T23M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
			W							
			0	0	0	0	0	0	0	0
			00: 8 bit Timer 01: 16 bit Timer 10: 8 bit PPG 11: 8 bit PWM		00: – 01: 2^6-1 10: 2^7-1 11: 2^8-1	PWM cycle	00: TO2TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: Reserved 01: ϕT1 10: ϕT4 11: ϕT16	
TRDC	Timer Reg. Double Buffer Control Reg	81h	–	–	–	–	–	–	TR2DE	TR0DE
									R/W	
			–	–	–	–	–	–	0	0
									0: Double Buffer Disable 1: Double Buffer Enable	

16 Bit Timer Control

Symbol	Name	Address	7	6	5	4	3	2	1	0
T16RUN	16 Bit Timer Control	9Eh	PRRUN	–	–	–	TARUN	T8RUN	T6RUN	T4RUN
			R/W				R/W			
			0	–	–	–	0	0	0	0
			Prescaler 0: Stop 1: Run				16 BitT imer Run / Stop Control 0: Stop&Clear 1: Run (Countup)			
T16CR	T4, T6, T8, TA Control	9Fh	–	–	–	–	DBAEN	DB8EN	DB6EN	DB4EN
							R/W			
			–	–	–	–	0	0	0	0
							1: Double Buffer Enable			

16 Bit Timer 4

Symbol	Name	Address	7	6	5	4	3	2	1	0
TREG4L	16 Bit Timer Reg. 4L	90h (no RMW)	–							
			W							
			Undefined							
TREG4H	16 Bit Timer Reg. 4H	91h (no RMW)	–							
			W							
			Undefined							
TREG5L	16 Bit Timer Reg. 5L	92h (no RMW)	–							
			W							
			Undefined							
TREG5H	16 Bit Timer Reg. 5H	93h (no RMW)	–							
			W							
			Undefined							
CAP4L	Capture Reg. 4L	94h	–							
			R							
			Undefined							
CAP4H	Capture Reg. 4H	95h	–							
			R							
			Undefined							
CAP5L	Capture Reg. 5L	96h	–							
			R							
			Undefined							
CAP5H	Capture Reg. 5H	97h	–							
			R							
			Undefined							
T4MOD	16 Bit Timer 4 Source CLK & MODE	98h **(no RMW)**	–	–	CAP4IN	CAP45M1	CAP45M0	CLE	T4CLK1	T4CLK0
					W	R/W				
			–	–	–	0	0	0	0	0
					0: Soft Capture 1: Don't care	Capture Timing 00: Disable 01: TI4 $\uparrow$ TI5 $\uparrow$ 10: TI4 $\uparrow$ TI4 $\downarrow$ 11: TFF1 $\uparrow$ TFF1 $\downarrow$		1: UC4 Clear Enable	Source Clock 00: TI4 01: ϕT1 10: ϕT4 11: ϕT16	
T4FFCR	16BitTimer4 Flip-Flop Control	99h **(no RMW)**	–	–	CAP5T4	CAP4T4	EQ5T4	EQ4T4	TFF4C1	TFF4C0
					R/W				W	
			–	–	0	0	0	0	–	–
					TFF4 Invert Trigger 0: Trigger Disable 1: Trigger Enable				00: Invert TFF4 01: Set TFF4 10: Clear TFF4 11: Don't care	

16 Bit Timer 6

Symbol	Name	Address	7	6	5	4	3	2	1	0
TREG6L	16 Bit Timer Reg. 6L	A0h (no RMW)	–							
			W							
			Undefined							
TREG6H	16 Bit Timer Reg. 6H	A1h (no RMW)	–							
			W							
			Undefined							
TREG7L	16 Bit Timer Reg. 7L	A2h (no RMW)	–							
			W							
			Undefined							
TREG7H	16 Bit Timer Reg. 7H	A3h (no RMW)	–							
			W							
			Undefined							
CAP6L	Capture Reg. 6L	A4h	–							
			R							
			Undefined							
CAP6H	Capture Reg. 6H	A5h	–							
			R							
			Undefined							
CAP7L	Capture Reg. 7L	A6h	–							
			R							
			Undefined							
CAP7H	Capture Reg. 7H	A7h	–							
			R							
			Undefined							
T6MOD	16 Bit Timer 6 Source CLK & MODE	A8h (no RMW)	CAP7T7	EQ7T7	CAP6IN	CAP67M1	CAP67M0	CLE	T6CLK1	T6CLK0
			R/W		W	R/W				
			0	0	0	0	0	0	0	0
			TFF7 INV TRG 0: TRG Disable 1: TRG Enable		0: Soft Capture 1: Don't care	Capture Timing 00: Disable 01: TI6↑ TI7↑ 10: TI6↑ TI6↓ 11: TFF1↑ TFF1↓		1: UC6 Clear Enable	Source Clock 00: TI6 01: ϕT1 10: ϕT4 11: ϕT16	
T6FFCR	16 Bit Timer6 Flip-Flop Control	A9h (no RMW)	TFF7C1	TFF7C0	CAP7T6	CAP6T6	EQ7T6	EQ6T6	TFF6C1	TFF6C0
			W		R/W				W	
			0	0	0	0	0	0	–	–
			00: InvertT FF7 01: Set TFF7 10: Clear TFF7 11: Don't care		TFF6 Invert Trigger 0: Trigger Disable 1: Trigger Enable				00: Invert TFF6 01: Set TFF6 10: Clear TFF6 11: Don't care	

16 Bit Timer 8

<table>
<tr><th>Symbol</th><th>Name</th><th>Address</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td rowspan="3">TREG8L</td><td rowspan="3">16 BitTimer Reg. 8L</td><td rowspan="3">B0h
(no RMW)</td><td colspan="8">–</td></tr>
<tr><td colspan="8">W</td></tr>
<tr><td colspan="8">Undefined</td></tr>
<tr><td rowspan="3">TREG8H</td><td rowspan="3">16 BitTimer Reg. 8H</td><td rowspan="3">B1h
(no RMW)</td><td colspan="8">–</td></tr>
<tr><td colspan="8">W</td></tr>
<tr><td colspan="8">Undefined</td></tr>
<tr><td rowspan="3">TREG9L</td><td rowspan="3">16 BitTimer Reg. 9L</td><td rowspan="3">B2h
(no RMW)</td><td colspan="8">–</td></tr>
<tr><td colspan="8">W</td></tr>
<tr><td colspan="8">Undefined</td></tr>
<tr><td rowspan="3">TREG9H</td><td rowspan="3">16 BitTimer Reg. 9H</td><td rowspan="3">B3h
(no RMW)</td><td colspan="8">–</td></tr>
<tr><td colspan="8">W</td></tr>
<tr><td colspan="8">Undefined</td></tr>
<tr><td rowspan="3">CAP8L</td><td rowspan="3">Capture Reg. 8L</td><td rowspan="3">B4h</td><td colspan="8">–</td></tr>
<tr><td colspan="8">R</td></tr>
<tr><td colspan="8">Undefined</td></tr>
<tr><td rowspan="3">CAP8H</td><td rowspan="3">Capture Reg. 8H</td><td rowspan="3">B5h</td><td colspan="8">–</td></tr>
<tr><td colspan="8">R</td></tr>
<tr><td colspan="8">Undefined</td></tr>
<tr><td rowspan="3">CAP9L</td><td rowspan="3">Capture Reg. 9L</td><td rowspan="3">B6h</td><td colspan="8">–</td></tr>
<tr><td colspan="8">R</td></tr>
<tr><td colspan="8">Undefined</td></tr>
<tr><td rowspan="3">CAP9H</td><td rowspan="3">Capture Reg. 9H</td><td rowspan="3">B7h</td><td colspan="8">–</td></tr>
<tr><td colspan="8">R</td></tr>
<tr><td colspan="8">Undefined</td></tr>
<tr><td rowspan="4">T8MOD</td><td rowspan="4">16 Bit Timer 8 Source CLK & MODE</td><td rowspan="4">B8h
(no RMW)</td><td>–</td><td>–</td><td>CAP8IN</td><td>CAP89M1</td><td>CAP89M0</td><td>CLE</td><td>T8CLK1</td><td>T8CLK0</td></tr>
<tr><td></td><td></td><td>W</td><td colspan="5">R/W</td></tr>
<tr><td>–</td><td>–</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td></td><td></td><td>0: Soft Capture
1: Don't care</td><td colspan="2">Capture Timing
00: Disable
01: TI8↑ TI9↑
10: TI8↑ TI8↓
11: TFF1↑ TFF1↓</td><td>1: UC8 Clear Enable</td><td colspan="2">Source Clock
01: ϕT1
10: ϕT4
11: ϕT16</td></tr>
<tr><td rowspan="4">T8FFCR</td><td rowspan="4">16 BitTimer8 Flip-Flop Control</td><td rowspan="4">B9h
(no RMW)</td><td>–</td><td>–</td><td>CAP9T8</td><td>CAP8T8</td><td>EQ9T8</td><td>EQ8T8</td><td>TFF8C1</td><td>TFF8C0</td></tr>
<tr><td></td><td></td><td colspan="4">R/W</td><td colspan="2">W</td></tr>
<tr><td>–</td><td>–</td><td>0</td><td>0</td><td>0</td><td>0</td><td>–</td><td>–</td></tr>
<tr><td></td><td></td><td colspan="4">TFF8 Invert Trigger
0: Trigger Disable
1: Trigger Enable</td><td colspan="2">00: Invert TFF8
01: Set TFF8
10: Clear TFF8
11: Don't care</td></tr>
</table>

16 Bit Timer A

Symbol	Name	Address	7	6	5	4	3	2	1	0
TREGAL	16 Bit Timer Reg. AL	C0h (no RMW)	–							
			W							
			Undefined							
TREGAH	16 Bit Timer Reg. AH	C1h (no RMW)	–							
			W							
			Undefined							
TREGBL	16 Bit Timer	C2h (no RMW)	–							
			W							
			Undefined							
TREGBH	16 Bit Timer Reg. BH	C3h (no RMW)	–							
			W							
			Undefined							
CAPAL	Capture Reg. AL	C4h	–							
			R							
			Undefined							
CAPAH	Capture Reg. AH	C5h	–							
			R							
			Undefined							
CAPBL	Capture Reg. BL	C6h	–							
			R							
			Undefined							
CAPBH	Capture Reg. BH	C7h	–							
			R							
			Undefined							
TAMOD	16 Bit Timer A Source CLK & MODE	C8h (no RMW)	CAPBTB	EQBTB	CAPAIN	CAPABM1	CAPABM0	CLE	TACLK1	TACLK0
			R/W		W	R/W				
			0	0	0	0	0	0	0	0
			TFFB INV TRG 0: TRG Disable 1: TRG Enable		0: Soft Capture 1: Don't care	Capture Timing 00: Disable 01: TIA↑ TIB↑ 10: TIA↑ TIA↓ 11:TFF1↑ TFF1↓		1: UCA Clear Enable	Source Clock 00: TIA 01: ϕT1 10: ϕT4 11: ϕT16	
TAFFCR	16 Bit Timer A Flip-Flop Control	C9h (no RMW)	TFFBC1	TFFBC0	CAPBTA	CAPATA	EQBTA	EQATA	TFFAC1	TFFAC0
			W		R/W				W	
			0	0	0	0	0	0	–	–
			00: Invert TFFB 01: SetT FFB 10: Clear TFFB 11: Don't care		TFFA Invert Trigger 0: Trigger Disable 1: Trigger Enable				00: Invert TFFA 01: Set TFFA 10: Clear TFFA 11: Don't care	

(3) Watchdog timer

Symbol	Name	Address	7	6	5	4	3	2	1	0
WDMOD	Watch Dog Timer Mode	110h	WDTE	WDTP1	WDTP0	–	HALTM1	HALTM0	RESCR	DRVE
			R/W				R/W			
			1	0	0	0	0	0	0	0
			1: WDT Enable	00: 2^{16}/fc 01: 2^{18}/fc 10: 2^{20}/fc 11: 2^{22}/fc		Fix to "0"	Standby Mode 00: Run Mode 01: Stop Mode 10: IDLE Mode 11: (Reserved)		1: Internal WDT out connect to Reset Pin.	1: Drive pin in STOP mode.
WDCR	WatchDog TimerControl Register	111h	–							
			W							
			–							
			B1h: WDT Disable Code 4Eh: WDT Clear Code							

(4) Clock control

Symbol	Name	Address	7	6	5	4	3	2	1	0
CLKMOD	Clock Mode	10Ah	–	–	–	WARM	–	CLKOE	–	–
						R/W		R/W		
			–	–	–	0	–	1	0	0
						Warming up time 0: 2^{15}/fc 1: 2^{17}/fc		CLK Output Enable 0: High Z 1: out	Fix to "0"	

(5) Serial channels

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC0BUF	Serial Channel 0 Buffer	D0h (no RMW)	RB7 TB7	RB6 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 TB1	RB0 TB0
			R (Receiving)/W (Transmission)							
			Undefined							
SC0CR	Serial Channel 0 Control	D1h	RB8	EVEN	PE	OERR	PERR	FERR	SCLK	IOC
			R	R/W		R (Clear 0 after reading)			R/W	
			Undefi.	0	0	0	0	0	0	0
			Receive Data bit 8	Parity 0: Odd 1: Even	Parity Addition 0: Disable 1: Enable	1: Error Overrun	Parity	Framing	0: SCLK0 ↑ 1: SCLK0 ↓	0: Baud rate genera. 1: SCLK0 pin input
SC0MOD	Serial Channel 0 Mode	D2h	TB8	CTSE	RXE	–	SM1	SM0	SC1	SC0
			R/W				R/W			
			Undefi.	0	0	0	0	0	0	0
			Trans-mission Data bit 8	0: CTS Disable 1: CTS Enable	0: Receive Disable 1: Receive Enable	Fix to "0"	00: I/O interface 01: UART 7 bit 10: UART 8 bit 11: UART 9 bit		00: TO2 Trigger 01: baudrate generator 10: Internal clock ph1 11: External clock SCLK0	
BR0CR	Baud Rate Channel 0	D3h	–	–	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
					R/W					
			0	–	0	0	0	0	0	0
			Fix to "0"		00: ϕT0 (4/fc) 01: ϕT2 (16/fc) 10: ϕT8 (64/fc) 11: ϕT32 (256/fc)		Set of the Divided frequency 0000: 16 divisions 0001: Don't set 0010 → 1111: 2 to 15 divisions			
SC1BUF	Serial Channel 1 Buffer	D4h	RB7 TB7	RB6 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 TB1	RB0 TB0
			R (Receiving)/W (Transmission)							
			Undefined							
SC1CR	Serial Channel 1 Control	D5h	RB8	EVEN	PE	OERR	PERR	FERR	SCLK	IOC
			R	R/W		R (Clear 0 after reading)			R/W	
			Undefi.	0	0	0	0	0	0	0
			Receive Data bit 8	Parity 0: Odd 1: Even	Parity Addition 0: Disable 1: Enable	1: Error Overrun	Parity	Framing	0:SCLK1 ↑ 1: SCLK1 ↓	0: Baud rate genera. 1: SCLK0 Pin input
SC1MOD	Serial Channel 1 Mode	D6h	TB8	CTSE	RXE	–	SM1	SM0	SC1	SC0
			R/W				R/W			
			Undefi.	0	0	0	0	0	0	0
			Trans-mission Data bit 8	1: CTS Enable	1: Receive Enable	Fix to "0"	00: I/O interface 01: UART 7 bit 10: UART 8 bit 11: UART 9 bit		00: TO2 Trigger 01: baudrate generator 10: Internal clock ph1 11: External clock SCLK1	
BR1CR	Baud rate Channel 1	D7h	–	–	BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
					R/W					
			0	–	0	0	0	0	0	0
			Fix to "0"		00: ϕT0 (4/fc) 01: ϕT2 (16/fc) 10: ϕT8 (64/fc) 11: ϕT32 (256/fc)		Set of the Divided frequency 0000: 16 divisions 0001: Don't set 0010 → 1111: 2 to 15 divisions			

(6) AD converter

Symbol	Name	Address	7	6	5	4	3	2	1	0
ADMOD1	AD Mode Reg. 1	128h	EOCF	ADBF	–	–	–	RPT	SCAN	ADS
			R					R/W		
			0	0	0	–	–	0	0	0
			0: Busy or Stop 1: End	0: Stop 1: Busy	Fix to "0"			Repeat Mode 0: Once 1: Repeat	Scan Mode 0: Settle 1: Scan	0: – 1: Run Conver.
ADMOD2	AD Mode Reg. 2	129h	VREFON	–	SPEED1	SPEED0	–	ADCH2	ADCH1	ADCH0
			R/W		R/W			R/W		
			1	–	0	0	–	0	0	0
			0: Ladder Resistance off 1: Ladder Resistance on		SpeedSelect 00: 160 state 01: 320 state 10: 640 state 11: 1280 state			(See below)		
ADREG04L	AD Result Reg 0/4 Low	120h	ADR01	ADR00	–	–	–	–	–	–
			R							
			Undefined		–	–	–	–	–	–
ADREG04H	AD Result Reg 0/4 High	121h	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
			R							
			Undefined							
ADREG15L	AD Result Reg 1/5 Low	122h	ADR11	ADR10	–	–	–	–	–	–
			R							
			Undefined		–	–	–	–	–	–
ADREG15H	AD Result Reg 1/5 High	123h	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
			R							
			Undefined							
ADREG26L	AD Result Reg 2/6 Low	124h	ADR21	ADR20	–	–	–	–	–	–
			R							
			Undefined		–	–	–	–	–	–
ADREG26H	AD Result Reg 2/6 High	125h	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
			R							
			Undefined							
ADREG37L	AD Result Reg 3/7 Low	126h	ADR31	ADR30	–	–	–	–	–	–
			R							
			Undefined		–	–	–	–	–	–
ADREG37H	AD Result Reg 3/7 High	127h	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
			R							
			Undefined							

ADCH [2:0] \ SCAN	0	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

(7) DA converter

Symbol	Name	Address	7	6	5	4	3	2	1	0
DAREG0	DA Conversion Reg. 0	130h	–							
			W							
			Undefined							
		(no RMW)	DA conversion startat DAREG0 input, and intime the datais send to DAOUT0.							
DAREG1	DA Conversion Reg. 1	131h	–							
			W							
			Undefined							
		(no RMW)	DA conversion startat DAREG1 input, and intime the datais send to DAOUT1.							
DADRV	DA Drive Register	132h	–	–	–	–	–	–	DA1DR	DA0DR
									R/W	
			–	–	–	–	–	–	0	0
									0: High-Z 1: Conversion Data Output	

(8) Interrupt control

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE0AD	INT0 & INTAD Enable	F0h	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE45	INT4 & INT5 Enable	E0h	INT5				INT4			
			I5C	I5M2	I5M1	I5M0	I4C	I4M2	I4M1	I4M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE67	INT6 & INT7 Enable	E1h	INT7				INT6			
			I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE89	INT8 & INT9 Enable	E2h	INT9				INT8			
			I9C	I9M2	I9M1	I9M0	I8C	I8M2	I8M1	I8M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTEAB	INTA & INTB Enable	E3h	INTB				INTA			
			IBC	IBM2	IBM1	IBM0	IAC	IAM2	IAM1	IAM0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTET01	INTT0 & INTT1 Enable	E4h	INTT1 (Timer1)				INTT0 (Timer0)			
			IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTET23	INTT2 & INTT3 Enable	E5h	INTT3 (Timer3)				INTT2 (Timer2)			
			IT3C	IT3M2	IT3M1	IT3M0	IT2C	IT2M2	IT2M1	IT2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTET45	INTTR4 & INTTR5 Enable	E6h	INTTR5 (TREG5)				INTTR4 (TREG4)			
			IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTET67	INTTR6 & INTTR7 Enable	E7h	INTTR7 (TREG7)				INTTR6 (TREG6)			
			IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	
INTET89	INTTR8 & INTTR9 Enable	E8h	INTTR9 (TREG9)				INTTR8 (TREG8)			
			IT9C	IT9M2	IT9M1	IT9M0	IT8C	IT8M2	IT8M1	IT8M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETAB	INTTRA & INTTRB Enable	E9h	INTTRB (TREGB)				INTTRA (TREGA)			
			ITBC	ITBM2	ITBM1	ITBM0	ITAC	ITAM2	ITAM1	ITAM0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTES0	INTRX0 & INTTX0 Enable	EAh	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTES1	INTRX1 & INTTX1 Enable	EBh	INTTX1				INTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC01	INTTC0 & INTTC1 Enable	ECh	INTTC1				INTTC0			
			ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC23	INTTC2 & INTTC3 Enable	EDh	INTTC3				INTTC2			
			ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC45	INTTC4 & INTTC5 Enable	EEh	INTTC5				INTTC4			
			ITC5C	ITC5M2	ITC5M1	ITC5M0	ITC4C	ITC4M2	ITC4M1	ITC4M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC67	INTTC6 & INTTC7 Enable	EFh	INTTC7				INTTC6			
			ITC7C	ITC7M2	ITC7M1	ITC7M0	ITC6C	ITC6M2	ITC6M1	ITC6M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTNMWDT	NMI & INTWD Enable	F7h	NMI				INTWD			
			ITCNM	–	–	–	ITCWD	–	–	–
			R				R			
			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.

Symbol	Name	Address	7	6	5	4	3	2	1	0
IIMC	Interrupt Input Mode Control	F6h (no RMW)	–	–	–	–	–	–	IOLE	NMIREE
									R/W	
			–	–	–	–	–	–	0	0
									0: INT0 edge mode 1: INT0 level mode	1: Operate even at /NMI rise edge

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTCLR	Interrupt Clear Control	F8h (no RMW)	Interrupt Vector							
			0	0	0	0	0	0	0	0
			W							
DMA0V	DMA 0 Start Vector	100h			DMA0 Start Vector					
			0	0	DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
					R/W					
					0	0	0	0	0	0
DMA1V	DMA 1 Start Vector	101h			DMA1 Start Vector					
			0	0	DMA1V5	DMA1V4	DMA1V3	DMA0V2	DMA1V1	DMA1V0
					R/W					
					0	0	0	0	0	0
DMA2V	DMA 2 Start Vector	102h			DMA2 Start Vector					
			0	0	DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
					R/W					
					0	0	0	0	0	0
DMA3V	DMA 3 Start Vector	103h			DMA3 Start Vector					
			0	0	DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
					R/W					
					0	0	0	0	0	0
DMA4V	DMA 4 Start Vector	104h			DMA4 Start Vector					
			0	0	DMA4V5	DMA4V4	DMA4V3	DMA4V2	DMA4V1	DMA4V0
					R/W					
					0	0	0	0	0	0
DMA5V	DMA 5 Start Vector	105h			DMA5 Start Vector					
			0	0	DMA5V5	DMA5V4	DMA5V3	DMA5V2	DMA5V1	DMA5V0
					R/W					
					0	0	0	0	0	0
DMA6V	DMA 6 Start Vector	106h			DMA6 Start Vector					
			0	0	DMA6V5	DMA6V4	DMA6V3	DMA6V2	DMA6V1	DMA6V0
					R/W					
					0	0	0	0	0	0
DMA7V	DMA 7 Start Vector	107h			DMA7 Start Vector					
			0	0	DMA7V5	DMA7V4	DMA7V3	DMA7V2	DMA7V1	DMA7V0
					R/W					
					0	0	0	0	0	0
DMAB	DMA Burst	108h	DMA Burst							
			DBST7	DBST6	DBST5	DBST4	DBST3	DBST2	DBST1	DBST0
			R/W							
			0	0	0	0	0	0	0	0
DMAR	DMA Request	109h (no RMW)	DMA Request							
			DREQ7	DREQ6	DREQ5	DREQ4	DREQ3	DREQ2	DREQ1	DREQ0
			R/W							
			0	0	0	0	0	0	0	0

(9) Memory controller

Symbol	Name	Address	7	6	5	4	3	2	1	0
B0CSL	Block 0 CS/WAIT Control reg. L	140h (no RMW)	–	B0WW2	B0WW1	B0WW0	–	B0WR2	B0WR1	B0WR0
				W				W		
			–	0	1	0	–	0	1	0
				001: 0 wait 010: 1 wait 011: N wait others: (Reserved)	101: 2 wait 110: 3 wait			001: 0 wait 010: 1 wait 011: N wait others: (Reserved)	101: 2 wait 110: 3 wait	
B0CSH	Block 0 CS/WAIT Control reg. H	141h (no RMW)	B0E	–	–	B0REC	B0OM1	B0OM0	B0BUS1	B0BUS0
			W			W	W			
			0	–	0	0	0	0	0	0
			CS select 1:enable 0:disable		Fix to "0"	Recovery 0:0 state 1:1 state	00: SRAM/ROM 01: (Reserved) 10: (Reserved) 11: (Reserved)		00: 8 bit 01: 16 bit 10: 32 bit 11: (Reserved)	
MAMR0	Memory Start Address Mask reg. 0	142h	M0V20	M0V19	M0V18	M0V17	M0V16	M0V15	M0V14-V9	M0V8
			R/W							
			1	1	1	1	1	1	1	1
			0: Compare enable 1: Compare disable							
MSAR0	Memory Start Address reg. 0	143h	M0S23	M0S22	M0S21	M0S20	M0S19	M0S18	M0S17	M0S16
			R/W							
			1	1	1	1	1	1	1	1
			Set start address A23 to A16							
B1CSL	Block 1 CS/WAIT Control reg. L	144h (no RMW)	–	B1WW2	B1WW1	B1WW0	–	B1WR2	B1WR1	B1WR0
				W				W		
			–	0	1	0	–	0	1	0
				001: 0 wait 010: 1 wait 011: N wait others: (Reserved)	101: 2 wait 110: 3 wait			001: 0 wait 010: 1 wait 011: N wait others: (Reserved)	101: 2 wait 110: 3 wait	
B1CSH	Block 1 CS/WAIT Control reg. H	145h (no RMW)	B1E	–	–	B1REC	B1OM1	B1OM0	B1BUS1	B1BUS0
			W			W	W			
			0	–	0	0	0	0	0	0
			CS select 1:enable 0:disable		Fix to "0"	Recovery 0:0 state 1:1 state	00: SRAM/ROM 01: (Reserved) 10: DRAM 11: (Reserved)		00: 8 bit 01: 16 bit 10: 32 bit 11: (Reserved)	
MAMR1	Memory Start Address Mask reg. 1	146h	M1V21	M1V20	M1V19	M1V18	M1V17	M1V16	M1V15-V9	M1V8
			R/W							
			1	1	1	1	1	1	1	1
			0: Compare enable 1: Compare disable							
MSAR1	Memory Start Address reg. 1	147h	M1S23	M1S22	M1S21	M1S20	M1S19	M1S18	M1S17	M1S16
			R/W							
			1	1	1	1	1	1	1	1
			Set start address A23 to A16							

Symbol	Name	Address	7	6	5	4	3	2	1	0
B2CSL	Block 2 CS/WAIT Control reg. L	148h (no RMW)	–	B2WW2	B2WW1	B2WW0	–	B2WR2	B2WR1	B2WR0
				W				W		
			–	0	1	0	–	0	1	0
				001: 0 wait 101: 2 wait 010: 1 wait 110: 3 wait 011: N wait others: (Reserved)				001: 0 wait 101: 2 wait 010: 1 wait 110: 3 wait 011: N wait others: (Reserved)		
B2CSH	Block 2 CS/WAIT Control reg. H	149h (no RMW)	B2E	B2M	–	B2REC	B2OM1	B2OM0	B2BUS1	B2BUS0
			W			W	W			
			1	0	0	0	0	0	0	0
			CS select 1: enable 0: disable	0: 16MB 1: Sets area.	Fix to "0"	Recovery 0:0 state 1:1 state	00: SRAM/ROM 01: (Reserved) 10: (Reserved) 11: (Reserved)		00: 8 bit 01: 16 bit 10: 32 bit 11: (Reserved)	
MAMR2	Memory Start Address Mask reg. 2	14Ah	M2V22	M2V21	M2V20	M2V19	M2V18	M2V17	M2V16	M2V15
			R/W							
			1	1	1	1	1	1	1	1
			0: Compare enable 1: Compare disable							
MSAR2	Memory Start Address reg. 2	14Bh	M2S23	M2S22	M2S21	M2S20	M2S19	M2S18	M2S17	M2S16
			R/W							
			1	1	1	1	1	1	1	1
			Set start address A23 to A16							
B3CSL	Block 3 CS/WAIT Control reg. L	14Ch (no RMW)	–	B3WW2	B3WW1	B3WW0	–	B3WR2	B3WR1	B0WR0
				W				W		
			–	0	1	0	–	0	1	0
				001: 0 wait 101: 2 wait 010: 1 wait 110: 3 wait 011: N wait others: (Reserved)				001: 0 wait 101: 2 wait 010: 1 wait 110: 3 wait 011: N wait others: (Reserved)		
B3CSH	Block 3 CS/WAIT Control reg. H	14Dh (no RMW)	B3E	–	–	B3REC	B3OM1	B3OM0	B3BUS1	B3BUS0
			W			W	W			
			0	–	0	0	0	0	0	0
			CS select 1: enable 0: disable		Fix to "0"	Recovery 0:0 state 1:1 state	00: SRAM/ROM 01: (Reserved) 10: DRAM 11: (Reserved)		00: 8 bit 01: 16 bit 10: 32 bit 11: (Reserved)	
MAMR3	Memory Start Address Mask reg. 3	14Eh	M3V22	M3V21	M3V20	M3V19	M3V18	M3V17	M3V16	M3V15
			R/W							
			1	1	1	1	1	1	1	1
			0: Compare enable 1: Compare disable							
MSAR3	Memory Start Address reg. 3	14Fh	M3S23	M3S22	M3S21	M3S20	M3S19	M3S18	M3S17	M3S16
			R/W							
			1	1	1	1	1	1	1	1
			Set start address A23 to A16							

Symbol	Name	Address	7	6	5	4	3	2	1	0
B4CSL	Block 4 CS/WAIT Control reg. L	150h	–	B4WW2	B4WW1	B4WW0	–	B4WR2	B4WR1	B4WR0
				W				W		
			–	0	1	0	–	0	1	0
		(no RMW)		001: 0 wait 101: 2 wait 010: 1 wait 110: 3 wait 011: N wait others: (Reserved)				001: 0 wait 101: 2 wait 010: 1 wait 110: 3 wait 011: N wait others: (Reserved)		
B4CSH	Block 4 CS/WAIT Control reg. H	151h	B4E	–	–	B4REC	B4OM1	B4OM0	B4BUS1	B4BUS0
			W			W	W			
			0	–	0	0	0	0	0	0
		(no RMW)	CS select 1: enable 0: disable		Fix to "0"	Recovery 0:0 state 1:1 state	00: SRAM/ROM 01: (Reserved) 10: (Reserved) 11: (Reserved)		00: 8 bit 01: 16 bit 10: 32 bit 11: (Reserved)	
MAMR4	Memory Start Address Mask reg. 4	152h	M3V22	M3V21	M3V20	M3V19	M3V18	M3V17	M3V16	M3V15
			R/W							
			1	1	1	1	1	1	1	1
			0: Compare enable 1: Compare disable							
MSAR4	Memory Start Address reg.4	153h	M3S23	M3S22	M3S21	M3S20	M3S19	M3S18	M3S17	M3S16
			R/W							
			1	1	1	1	1	1	1	1
			Set start address A23 to A16							
B5CSL	Block 5 CS/WAIT Control reg. L	154h	–	B5WW2	B5WW1	B5WW0	–	B5WR2	B5WR1	B5WR0
				W				W		
			–	0	1	0	–	0	1	0
		(no RMW)		001: 0 wait 101: 2 wait 010: 1 wait 110: 3 wait 011: N wait others: (Reserved)				001: 0 wait 101: 2 wait 010: 1 wait 110: 3 wait 011: N wait others: (Reserved)		
B5CSH	Block 5 CS/WAIT Control reg. H	155h	B5E	–	–	B5REC	B5OM1	B5OM0	B5BUS1	B5BUS0
			W			W	W			
			0	–	0	0	0	0	0	0
		(no RMW)	CS select 1: enable 0: disable		Fix to "0"	Recovery 0:0 state 1:1 state	00: SRAM/ROM 01: (Reserved) 10: (Reserved) 11: (Reserved)		00: 8 bit 01: 16 bit 10: 32 bit 11: (Reserved)	
MAMR5	Memory Start Address Mask reg. 5	156h	M5V22	M5V21	M5V20	M5V19	M5V18	M5V17	M5V16	M5V15
			R/W							
			1	1	1	1	1	1	1	1
			0: Compare enable 1: Compare disable							
MSAR5	Memory Start Address reg. 5	157h	M5S23	M5S22	M5S21	M5S20	M5S19	M5S18	M5S17	M5S16
			R/W							
			1	1	1	1	1	1	1	1
			Set start address A23 to A16							
PMEMCR	Page ROM Control reg.	166h	–	–	–	OPGE	OPWR1	OPWR0	PR1	PR0
						R/W				
			–	–	–	0	0	0	1	0
						ROMpage access 0: Disable 1: Enable	Wait number on page 00: 1 CLK (n-1-1-1 mode) 01: 2 CLK (n-2-2-2 mode) 10: 3 CLK (n-3-3-3 mode) 11: (Reserved)		Byte number in a page 00: 64 Byte 01: 32 Byte 10: 16 Byte 11: 8 Byte	

(10) DRAM controller

Symbol	Name	Address	7	6	5	4	3	2	1	0
DRAMEXT	DRAM Select Enable	15Ch	CH0EN	–	–	–	CH1EN	–	–	–
			R/W							
			0	–	–	–	0	–	–	–
			DRAMSEL0 0: Disable 1: Enable				DRAMSEL1 0: Disable 1: Enable			
DRAM0CRL	DRAM 0 Control Register L	160h	SFRC0	–	BRM0	DRAMSEL0	MUXE0	MUXW01	MUXW00	MAC0
			1	–	0	0	0	0	0	0
			Self-refresh 0: Exec. 1: Rele.		Bus release mode control 0: Rele. 1: Not release	DRAM select 0: 16 Mbit 1: 64 Mbit	address multiplex 0: disable 1: Enable	Multiplexed length address 00: 8 bit 01: 9 bit 10: 10 bit 11: 11 bit		memory access control 0: Disable 1: Enable
DRAM0CRH	DRAM 0 Control Register H	161h	P0WW1	P0WW0	P0WR1	P0WR0	PGE0	–	–	–
			R/W							
			1	0	1	0	0	–	–	–
			00: (Reserved) 01: 1 wait (n-2-2-2 mode) 10: 2 wait (n-3-3-3 mode) 11: (Reserved)		00: (Reserved) 01: 1 wait (n-2-2-2 mode) 10: 2 wait (n-3-3-3 mode) 11: (Reserved)		DRAM page access 1:Enable			
DRAM1CRL	DRAM 1 Control Register L	162h	SFRC1	–	BRM1	DRAMSEL1	MUXE1	MUXW11	MUXW10	MAC1
			R/W							
			1	–	0	0	0	0	0	0
			Self-refresh 0: Exec. 1: Rele.		Bus release mode control 0: Rele. 1: Not release	DRAM select 0: 16 Mbit 1: 64 Mbit	address multiplex 0: disable 1: Enable	Multiplexed length address 00: 8 bit 01: 9 bit 10: 10 bit 11: 11 bit		memory access control 0: Disable 1: Enable
DRAM1CRH	DRAM 1 Control Register H	163h	P1WW1	P1WW0	P1WR1	P1WR0	PGE1	–	–	–
			R/W							
			1	0	1	0	0	–	–	–
			00: (Reserved) 01: 1 wait (n-2-2-2 mode) 10: 2 wait (n-3-3-3 mode) 11: (Reserved)		00: (Reserved) 01: 1 wait (n-2-2-2 mode) 10: 2 wait (n-3-3-3 mode) 11: (Reserved)		DRAM page access 1:Enable			
DRAM0REF	DRAM 0 Refresh Control	164h	DM0	RS02	RS01	RS00	RW02	RW01	RW00	RC0
			R/W							
			0	0	0	0	0	0	0	0
			Dummy cycle 0: Prohi bit 1: Execute	Refresh cycle insertion at 000: 78 100: 246 001: 154 101: 302 010: 188 110: 308 011: 226 111: 384			Refresh cycle width 000: 2 100: 6 001: 3 101: 7 010: 4 110: 8 011: 5 111: 9			Refresh cycle 0: Not insert 1: insert
DRAM1REF	DRAM 1 Refresh Control	165h	DM1	RS12	RS11	RS10	RW12	RW11	RW10	RC1
			R/W							
			0	0	0	0	0	0	0	0
			Dummy cycle 0: Prohi bit 1: Execute	Refresh cycle insertion at 000: 78 100: 246 001: 154 101: 302 010: 188 110: 308 011: 226 111: 384			Refresh cycle width 000: 2 100: 6 001: 3 101: 7 010: 4 110: 8 011: 5 111: 9			Refresh cycle 0: Not insert 1: insert

6. Port Section Equivalent Circuit Diagram

- Reading the circuit diagram

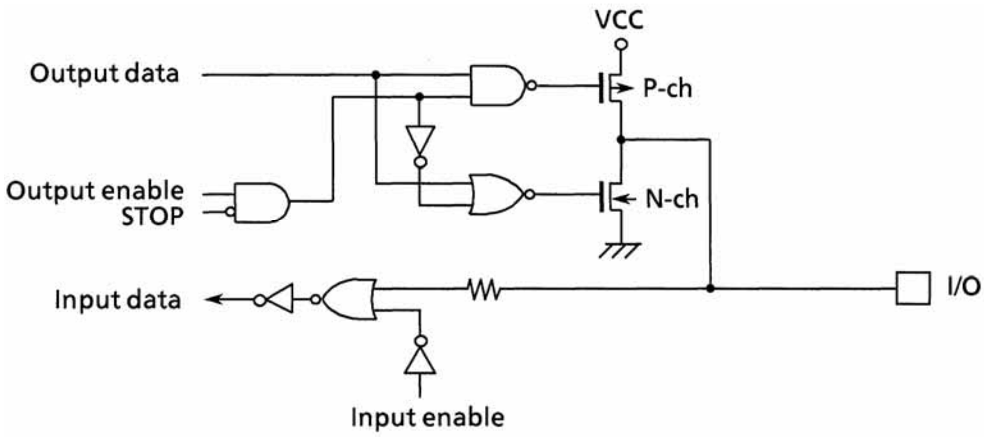
  Basically, the gate symbols written are the same as those used for the standard CMOS logic IC (74HCxx) series.

  The dedicated signal is described below.

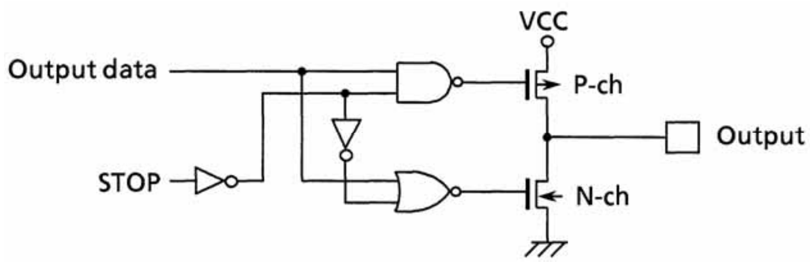
  STOP: This signal becomes active "1" when the HALT mode setting register is set to the STOP mode (WDMOD<HALTM1:0> = 0, 1) and the CPU executes the HALT instruction. When the drive enable bit WDMOD<DRVE> is set to "1", however STOP remains at "0".

- The input protection resistance ranges from several ohms to several hundreds of ohms.

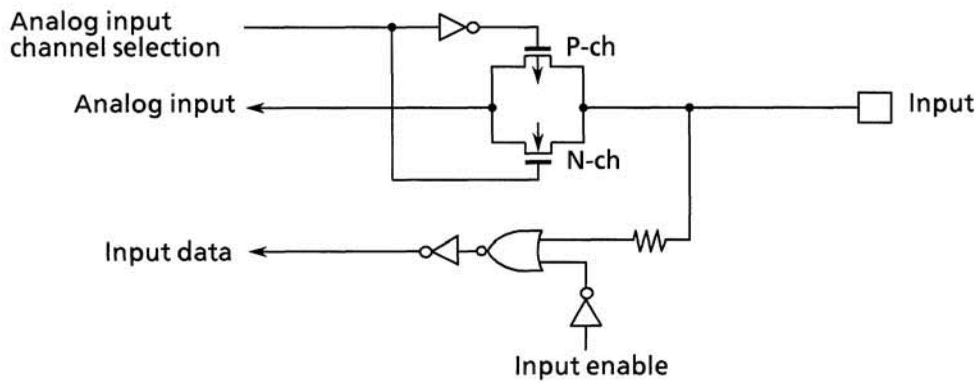
■ P0 (D0 to D7), P1 (D8 to D15), P4 (A0 to A7),
P5 (A8 to A15), P6 (A16 to A23), P75 ($\overline{\text{BUSRQ}}$), P86 ($\overline{\text{WAIT}}$), PC, PD, PE,
PF6 ($\overline{\text{CTS1}}$, SCLK1), PF5 (RXD1), PF4 (TXD1), PF2 (CTS0, SCLK0), PF1 (RXD0), PF0 (TXD0),
PH0 to PH3, PZ



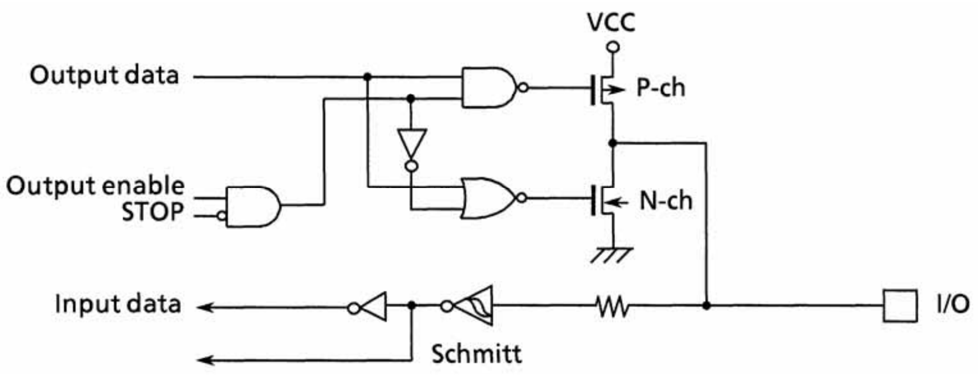
■ P76 ($\overline{\text{BUSAK}}$), P70 to P74, P80 to P85



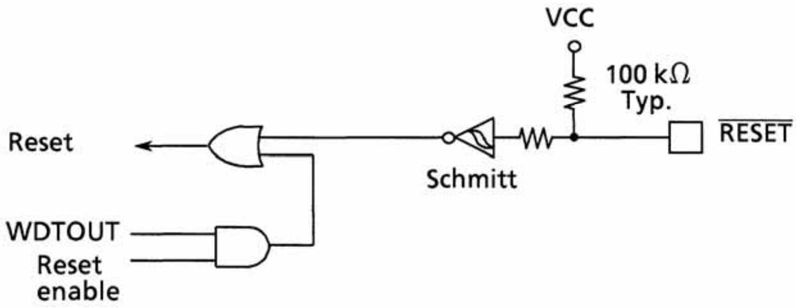
■ PG (AN0 to AN7)



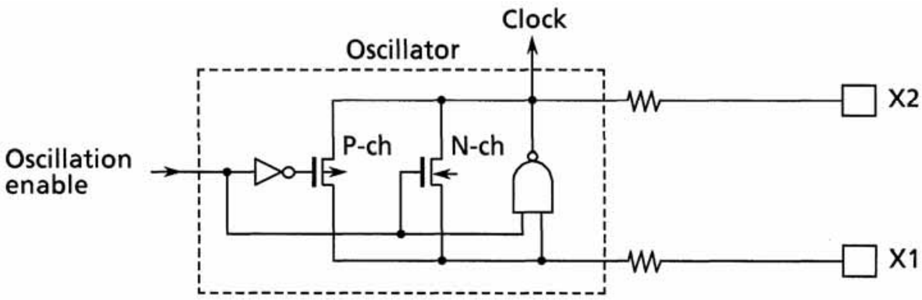
■ PH4 (INT0)



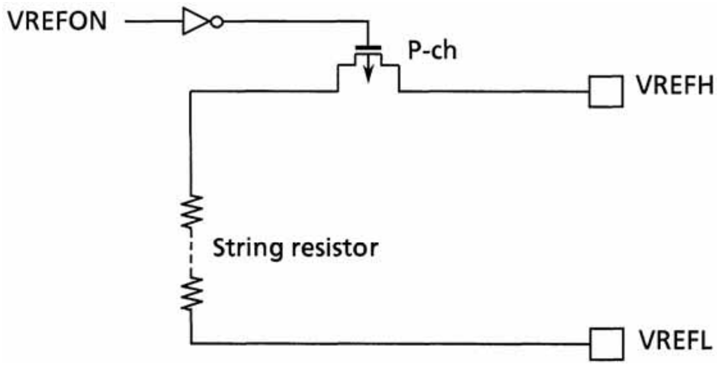
■ $\overline{\text{RESET}}$



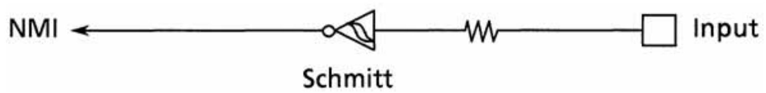
■ X1, X2



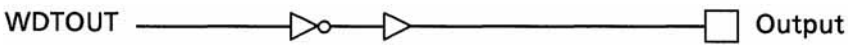
■ VREFH, VREFL



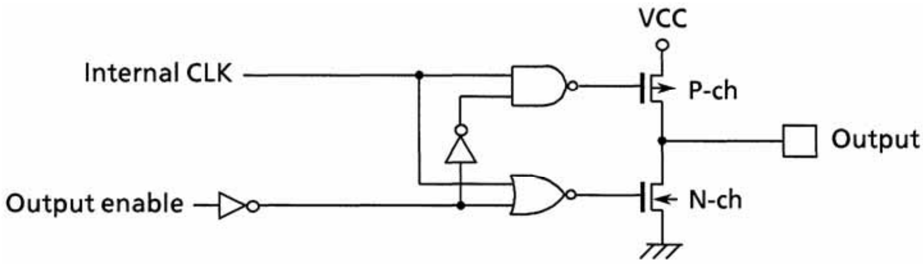
■ $\overline{\text{NMI}}$



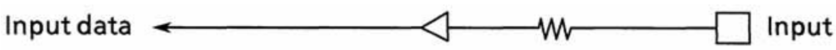
■ $\overline{\text{WDTOUT}}$



■ CLK



■ AM0 to AM1, TEST0 to TEST1



7. Care Points and Restriction

(1) Special expression

[1] Explanation of a built-in I/O register: Register Symbol<Bit symbol>

ex) T8RUN<T0RUN>...Bit T0RUN of Register T8RUN

[2] Read-modify-write Instructions

An instruction which CPU executes following by one instruction.

ex1) SET 3, (T8RUN) ...set bit3 of TRUN

ex2) INC 1, (100H) ...increment the data of 100H

- The read-modify-write Instructions in the TLCS-900

SET	imm, mem	,	RES	imm, mem
CHG	imm, mem	,	TSET	imm, mem
INC	imm, mem	,	DEC	imm, mem
RLD	A, mem	,	ADD	imm, reg

(2) Care points

[1] 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.

[2] When releasing the external reset using "built-in clock doubler" until the internal reset is released, the requiring time to stabilize the circuit is automatically set. See "3.1.2 Reset Operation" for details. Also when releasing standby mode in STOP mode using an interrupt until the internal circuit starts the operation, the stable time of the oscillator is automatically input. See 3.4 "Standby Function, (3) STOP mode" for details.

[3] Undefined bit in the built-in I/O register

When reading the undefined bit in the built-in I/O register, the undefined value is output. Thus, when creating program, it should not be depending on this bit condition.

[4] Setting Data Bus

When starting up with 8-bit data bus by setting AM0 and AM1 pin after the reset is released, the upper data bus is set to input port, thus, when using the upper data bus, change the port control register of its data bus pin.

[5] POP SR instruction

Please execute the POP SR instruction during DI condition.

[6] Releasing the HALT mode by requesting an interruption

Usually, interrupts can release all halts status. However, the interrupts =($\overline{\text{NMI}}$ and 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 IDLE or STOP mode (RUN 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 difficulty. 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.

8. Package Dimensions

P-QFP144-2020-0.50

Unit: mm