TOSHIBA Original CMOS 16-Bit Microcontroller

TLCS-900/L Series

TMP93CS32

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 f_{FPH}) with IDLE1 or STOP mode (IDLE2/RUN are 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.

Low Voltage/Low Power

CMOS 16-Bit Microcontrollers

TMP93CS32F

1. Outline and Device Characteristics

The TMP93CS32 is high-speed, advanced 16-bit microcontroller developed for controlling medium to large-scale equipment.

The TMP93CS32 is housed in 64-pin flat package (P-QFP64-1414-0.80A).

The device characteristics are as follows:

(1) Original 16-bit CPU (900/L CPU)
- TLCS-90 instruction mnemonic upward compatible
- 16-Mbyte linear address space
- General-purpose registers and register bank system
- 16-bit multiplication/division and bit transfer/arithmetic instructions
- Micro DMA: 4 channels (1.6 μs per 2 bytes at 20 MHz)

(2) Minimum instruction execution time: 200 ns at 20 MHz

(3) Internal RAM: 2 Kbytes
Internal ROM: 64 Kbytes

(4) External memory expansion
- Can be expanded up to16 Mbytes (for both programs and data).
- AM8/$\overline{\text{AM16}}$ pin (Select the external data bus width)
- Can mix 8- and 16-bit external data buses.
(Dynamic bus sizing)

(5) 8-bit timer: 4 channels

(6) 16-bit timer: 2 channels

(7) Serial interface: 2 channels

(8) 10-bit AD converter: 6 channels

(9) High current output: 2 ports

030619EBP1

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

(10) Watchdog timer

(11) Bus width/wait controller: 3 blocks

(12) Interrupt functions: 31

- 9 CPU interrupts (SWI instruction, and Illegal instruction)
- 16 internal interrupts
- 6 external interrupts

7-level priority can be set. (except $\overline{\text{NMI}}$, INTWD)

(13) I/O ports

49 pins for TMP93CS32

(14) Standby function: 4 HALT modes (RUN, IDLE2, IDLE1, STOP)

(15) Clock-gear function

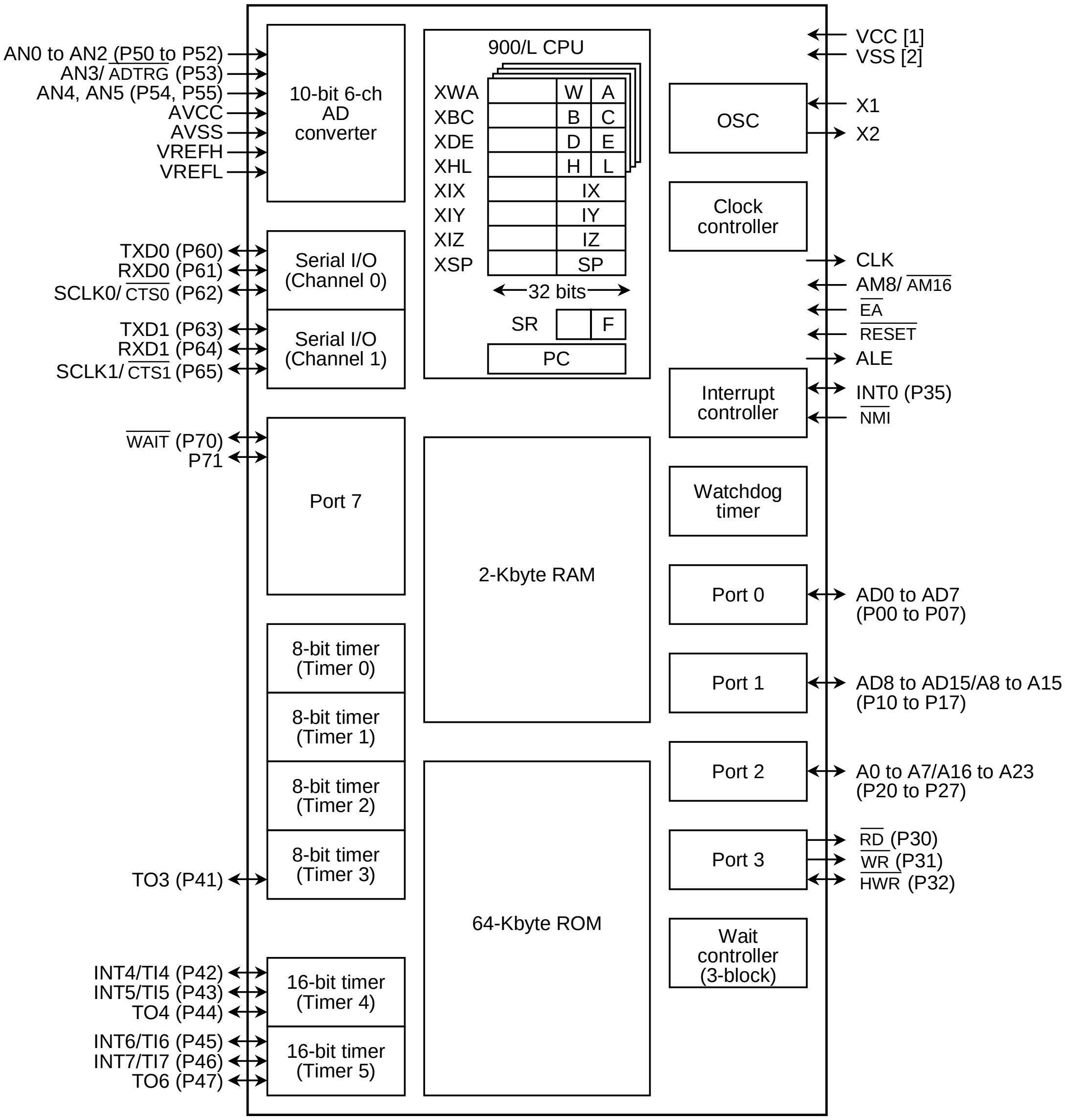
- Clock can be changed from fc to fc/16.

(16) Wide range of operating voltage

- V_{CC} = 2.7 to 5.5 V

(17) Package

- P-QFP64-1414-0.80A



Note: The items in parentheses ( ) are the initial setting after reset.

Figure 1.1 TMP93CS32 Block Diagram

2. Pin Assignment and Functions

The assignment of input and output pins for the TMP93CS32, their names and functions are described below.

2.1 Pin Assignment

Figure 2.1.1 shows pin assignment of the TMP93CS32.

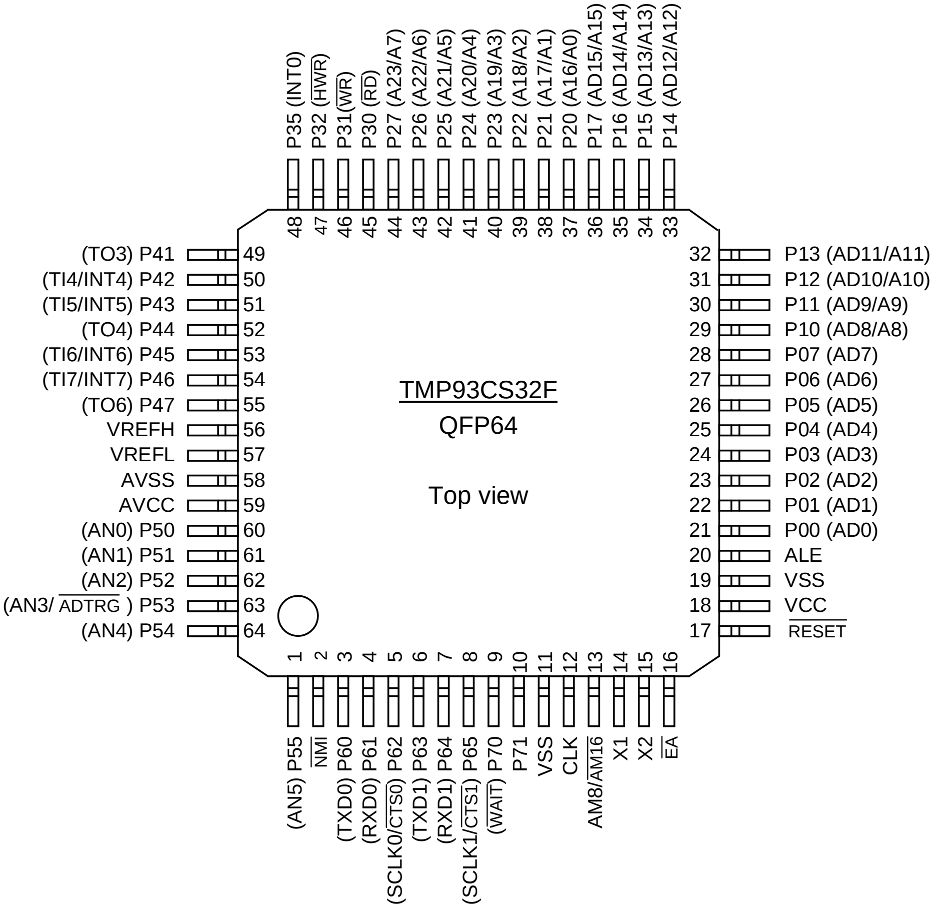


Figure 2.1.1 Pin Assignment (64-Pin QFP)

2.2 Pin Names and Functions

The names of input/output pins and their functions are described below Table 2.2.1 to Table 2.2.2 Pin Names and Functions.

Table 2.2.1 Pin Names and Functions (1/2)

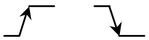
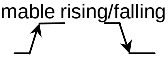
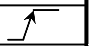
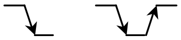
Pin Name	Number of Pins	I/O	Functions
P00 to P07	8	I/O	Port 0: I/O port that allows selection of I/O on a bit basis
AD0 to AD7		3-state	Address/Data (lower): Bits 0 to 7 for address/data bus
P10 to P17	8	I/O	Port 1: I/O port that allows selection of I/O on a bit basis
AD8 to AD15		3-state	Address/Data (upper): Bits 8 to 15 for address/data bus
A8 to A15		Output	Address: Bits 8 to 15 for address bus
P20 to P27	8	I/O	Port 2: I/O port that allows selection of I/O on a bit basis (with pull-up resistor)
A0 to A7		Output	Address: Bits 0 to 7 for address bus
A16 to A23		Output	Address: Bits 16 to 23 for address bus
P30	1	Output	Port 30: Output port
$\overline{RD}$		Output	Read: Strobe signal for reading external memory
P31	1	Output	Port 31: Output port
$\overline{WR}$		Output	Write: Strobe signal for writing data on pins AD0 to AD7
P32	1	I/O	Port 32: I/O port (with pull-up resistor)
$\overline{HWR}$		Output	High write: Strobe signal for writing data on pins AD8 to AD15
P35	1	I/O	Port 35: I/O port
INT0		Input	Interrupt request pin 0: Interrupt request pin with programmable level/rising edge 
P41	1	I/O	Port 41: I/O port
TO3		Output	PWM output 3: 8-bit PWM timer 3 output
P42	1	I/O	Port 42: I/O port
TI4		Input	Timer input 4: Timer 4 input
INT4		Input	Interrupt request pin 4: Interrupt request pin with programmable rising/falling edge 
P43	1	I/O	Port 43: I/O port
TI5		Input	Timer input 5: Timer 4 input
INT5		Input	Interrupt request pin 5: Interrupt request pin with rising edge 
P44	1	I/O	Port 44: I/O port
TO4		Output	Timer output 4: Timer 4 output pin
P45	1	I/O	Port 45: I/O port
TI6		Input	Timer input 6: Timer 5 input
INT6		Input	Interrupt request pin 6: Interrupt request pin with programmable rising/falling edge 
P46	1	I/O	Port 46: I/O port
TI7		Input	Timer input 7: Timer 5 input
INT7		Input	Interrupt request pin 7: Interrupt request pin with rising edge 
P47	1	I/O	Port 47: I/O port
TO6		Output	Timer output 6: Timer 5 output pin
P50 to P52, P54, P55	5	Input	Port 50 to Port 52, Port 54, Port 55: Input port
AN0 to AN2, AN4, AN5		Input	Analog input: Analog signal input for AD converter
P53	1	Input	Port 53: Input port
AN3		Input	Analog input: Analog signal input for AD converter
$\overline{ADTRG}$		Input	AD converter external start trigger input

Table 2.2.2 Pin Names and Functions (2/2)

Pin Name	Number of Pins	I/O	Functions
P60	1	I/O	Port 60: I/O port (with pull-up resistor)
TXD0		Output	Serial send data 0
P61	1	I/O	Port 61: I/O port (with pull-up resistor)
RXD0		Input	Serial receive data 0
P62	1	I/O	Port 62: I/O port (with pull-up resistor)
SCLK0		I/O	Serial data send enable 0 (Clear to Send)
$\overline{\text{CTS0}}$		Input	Serial Clock I/O 0
P63	1	I/O	Port 63: I/O port (with pull-up resistor)
TXD1		Output	Serial send data 1
P64	1	I/O	Port 64: I/O port (with pull-up resistor)
RXD1		Input	Serial receive data 1
P65	1	I/O	Port 65: I/O port (with pull-up resistor)
$\overline{\text{CTS1}}$		Input	Serial data send enable 1 (Clear to send)
SCLK1		I/O	Serial clock I/O 1
P70	1	I/O	Port 70: I/O port (High current output available)
$\overline{\text{WAIT}}$		Input	WAIT: Pin used to request CPU bus wait (It is active in (1 + N) waits mode. Set by the bus-width/wait control register.)
P71	1	I/O	Port 71: I/O port (high current output available)
$\overline{\text{NMI}}$	1	Input	Non-maskable interrupt request pin: Interrupt request pin with falling edge. Can also be operated at falling and rising edges by program. 
CLK	1	Output	Clock output: Outputs "f_{SYS} ÷ 2" clock. Pulled up during reset. Can be disabled for reducing noise.
$\overline{\text{EA}}$	1	Input	"1" should be inputted with TMP93CS32.
AM8/ $\overline{\text{AM16}}$	1	Input	Address mode: Selects external data bus width. "1" should be inputted. The data bus width for external access is set by chip select/wait control register, port 1 control register.
ALE	1	Output	Address latch enable Can be disabled for reducing noise.
$\overline{\text{RESET}}$	1	Input	Reset: Initializes TMP93CS32. (with pull-up resistor)
VREFH	1	Input	Pin for high level reference voltage input to AD converter
VREFL	1	Input	Pin for low level reference voltage input to AD converter
AVCC	1	Input	Power supply pin for AD converter
AVSS	1	Input	GND pin for AD converter (0 V)
X1	1	Input	Oscillator connecting pin
X2	1	Output	Oscillator connecting pin
VCC	1	Input	Power supply pin
VSS	2	Input	GND pin (All VSS pins are connected to the GND (0 V).)

Note: Built-in pull-up resistors can be released from the pins other than the $\overline{\text{RESET}}$ pin by software.

3. Operation

This section describes the functions and basic operational blocks of TMP93CS32 devices.
See the 7. Points of Concern and Restriction for the using notice and restrictions for each block.

3.1 CPU

The TMP93CS32 device has a built-in high-performance 16-bit CPU (900/L CPU). (For CPU operation, see TLCS-900/L CPU in the previous section).

This section describes CPU functions unique to the TMP93CS32 that are not described in the previous section.

3.1.1 Reset

When resetting the TMP93CS32 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 (16 μs at 20 MHz). Thus, when turn on the switch, be set to the power supply voltage is within the operating voltage range, and that the internal high-frequency oscillator has stabilized. Then hold the $\overline{\text{RESET}}$ input to Low level at least for 10 system clocks.

Clock gear is initialized 1/16 mode by Reset operation. It means that the system clock mode f_{SYS} is set to fc/32 (= fc/16 × 1/2).

When reset is accepted, the CPU sets as follows:

- Program Counter (PC) according to Reset Vector that is stored FFFF00H to FFFF02H.
  PC <7:0> ← Data in location FFFF00H
  PC <15:8> ← Data in location FFFF01H
  PC <23:16> ← Data in location FFFF02H
- Stack pointer (XSP) for system mode to 100H.
- IFF2 to 0 bits of status register to 111. (Sets mask register to interrupt level 7.)
- MAX bit of status register to 1. (Sets to maximum mode)
- Bits RFP2 to 0 of status register to 000. (Sets register banks to 0.)

When reset is released, instruction execution starts from PC (reset vector). CPU internal registers other than the above are not changed.

When reset is accepted, processing for built-in I/Os, ports, and other pins is as follows:

- Initializes built-in I/O registers as per specifications.
- Sets port pins (Including pins also used as built-in I/Os) to general-purpose input/output port mode.
- Pulls up the CLK pin to "High" level.
- Sets the ALE pin to high impeadance (High-Z).

Note 1: By resetting, register in the CPU except program counter (PC), status register (SR) and stack pointer (XSP) and the data in internal RAM are not changed.

Note 2: The CLK pin is pulled up to "High" level during reset. When the voltage is put down externally, there is possible to cause malfunctions.

Figure 3.1.1 shows the reset timing chart of TMP93CS32.

3.1.2 AM8/$\overline{\text{AM16}}$ Pin

Set this pin to "H". After reset, the CPU accesses the internal ROM with 16-bit bus width. The bus width when the CPU accesses an external area is set by bus width/wait control registers and the registers of Port 1. (The value "H"of this pin is ignored and the value set by register is active.) For details, see the bus width/wait control registers in section 3.6.3.

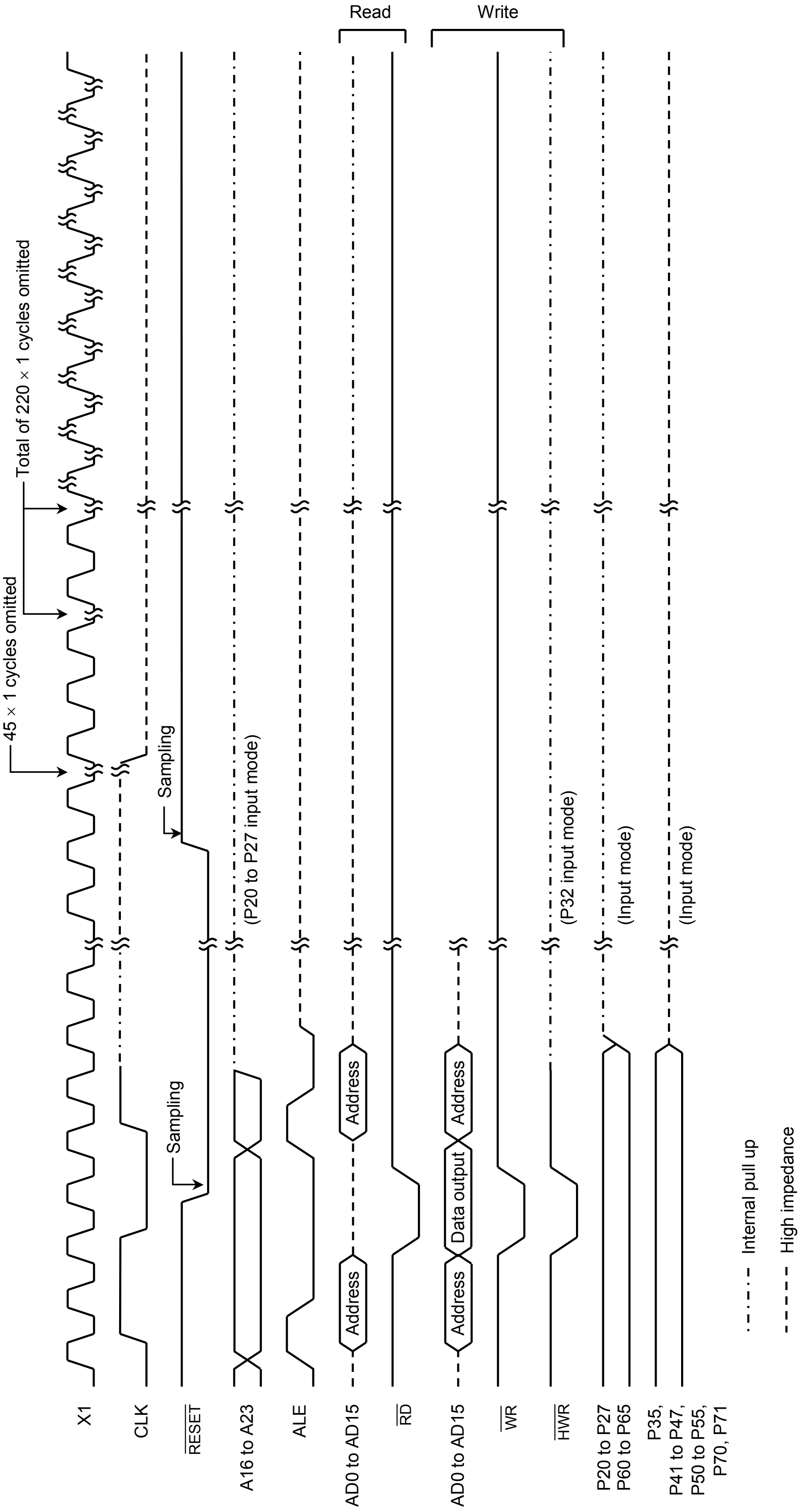


Figure 3.1.1 TMP93CS32 Reset Timing Chart

3.2 Memory Map

Figure 3.2.1 is a memory map of the TMP93CS32.

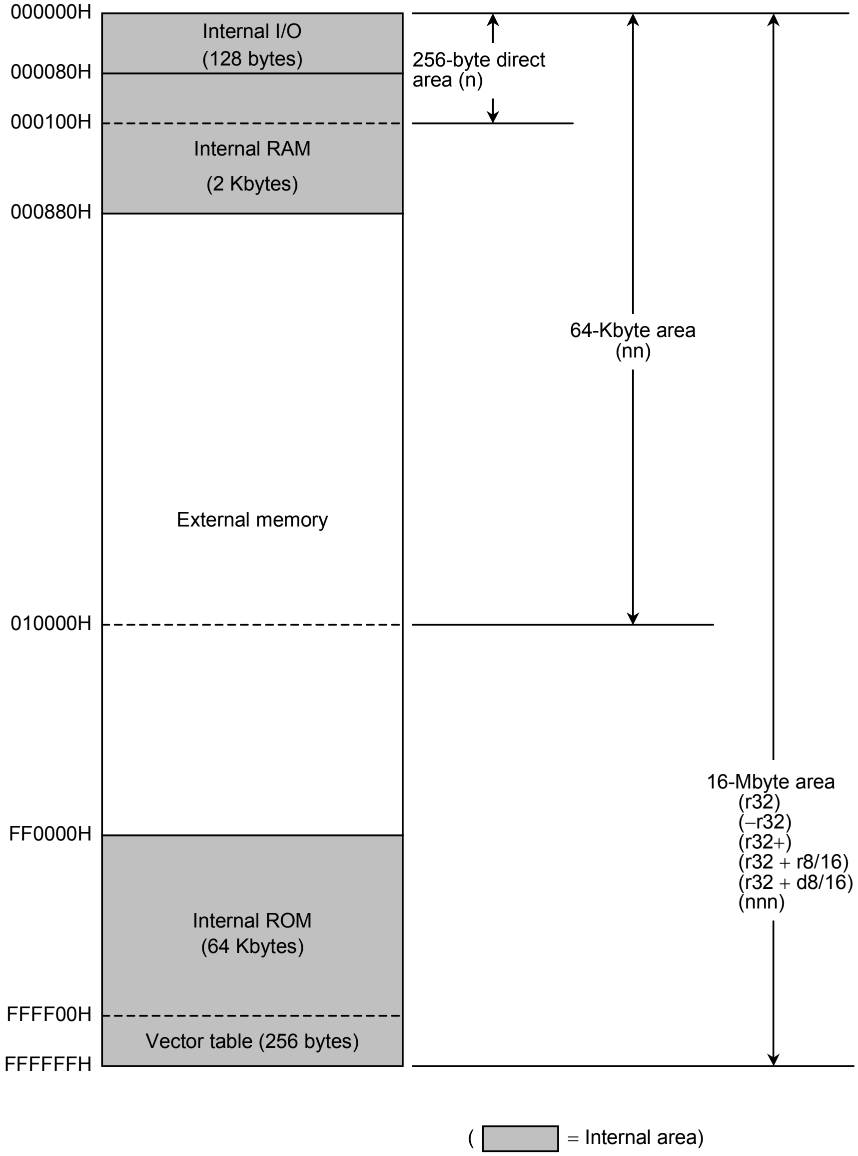


Figure 3.2.1 Memory Map

3.3 Standby Function

Standby control circuits consist of (1) System clock controller, (2) Prescaler clock controller and (3) Standby controller.

Figure 3.3.1 shows a transition figure. Figure 3.3.2 shows the block diagram.

Figure 3.3.3 shows I/O registers. Table 3.3.1 shows the internal operation and system clock.

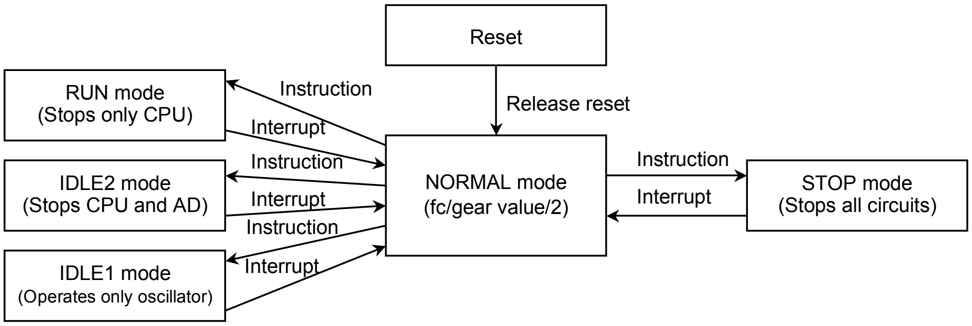


Figure 3.3.1 Transition Figure

The clock frequency input from X1, X2 pin is called fc. The clock frequency selected by SYSCR1<GEAR2:0> is called system clock f_{FPH}. The devided clock of f_{FPH} is called system clock f_{SYS}, and the 1 cycle of f_{SYS} is called 1 state operating mode.

Table 3.3.1 Internal Operation and System Clock

Operating Mode	Oscillator fc	CPU	Internal I/O	System Clock f_{SYS}
RESET	Oscillation	Reset	Reset	fc/32
NORMAL		Operate	Operate	Programmable (fc/2, fc/4, fc/8, fc/16, fc/32)
RUN		Stop		
IDLE2			Stop only AD	
IDLE1			Stop	
STOP	Stop			Stop

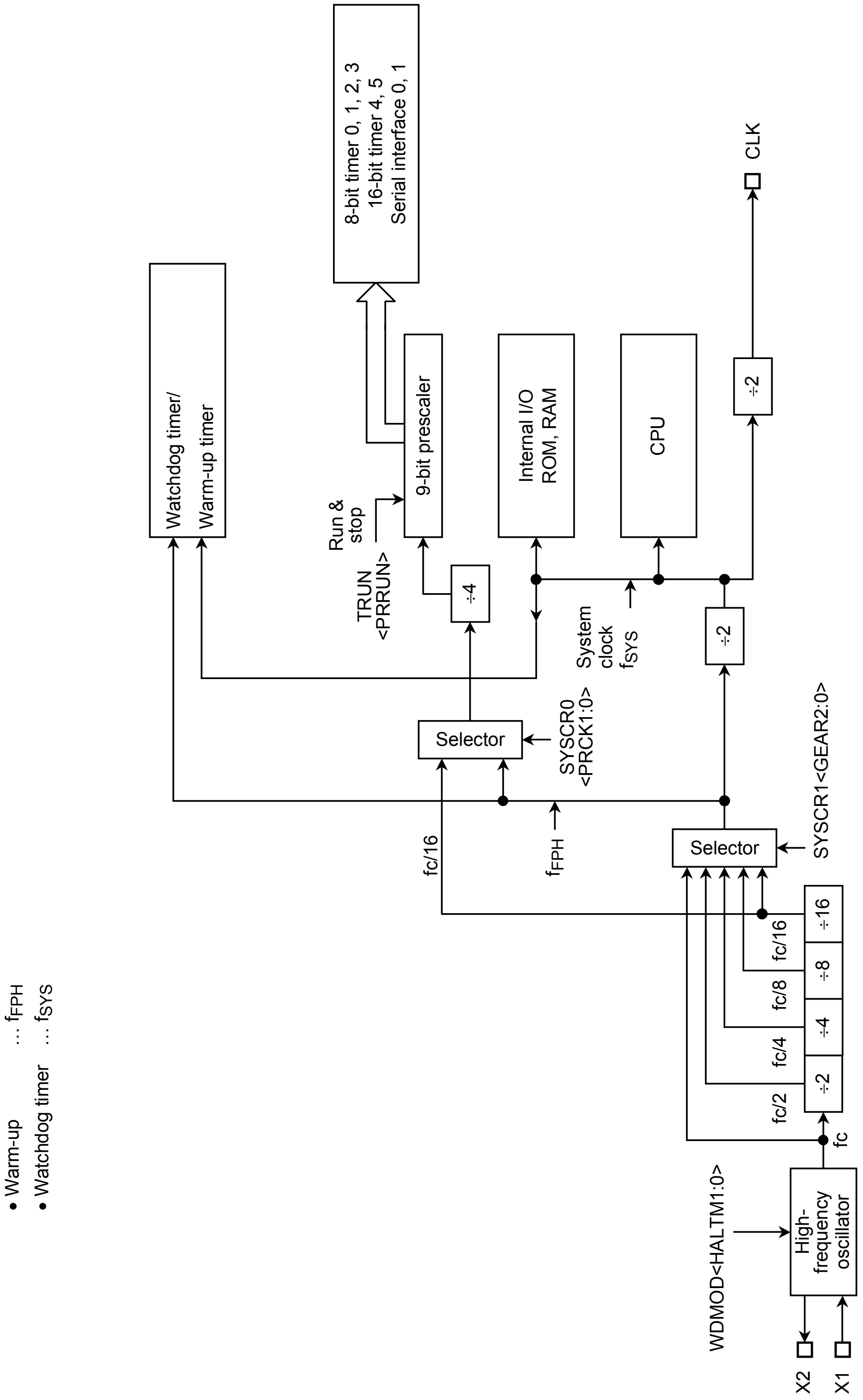


Figure 3.3.2 Block Diagram of Standby Circuits

		7	6	5	4	3	2	1	0
SYSCR0 (006EH)	Bit symbol	–	–	–	–	–	–	PRCK1	PRCK0
	Read/Write	R/W							
	After reset	1	0	1	0	0	0	0	0
	Function	Always write “1” (This bit is read as “1”.)	Always write “0” (This bit is read as “0”.)	Always write “1” (This bit is read as “1”.)	Always write “0” (This bit is read as “0”.)	Always write “0” (This bit is read as “0”.)	Always write “0” (This bit is read as “0”.)	Select prescaler clock 00: f_{FPH} 01: (Reserved) 10: fc/16 11: (Reserved)	

		7	6	5	4	3	2	1	0
SYSCR1 (006FH)	Bit symbol					–	GEAR2	GEAR1	GEAR0
	Read/Write					R/W			
	After reset					0	1	0	0
	Function					Always write “0” (This bit is read as “0”.)	Select gear value 000: fc 001: fc/2 010: fc/4 011: fc/8 100: fc/16 101: (Reserved) 110: (Reserved) 111: (Reserved)		

		7	6	5	4	3	2	1	0
CKOCR (006DH)	Bit symbol							ALEEN	CLKEN
	Read/Write							R/W	
	After reset							0	0
	Function							ALE pin output control 0: High-Z output 1: ALE output	CLK pin output control 0: High-Z output 1: CLK output

		7	6	5	4	3	2	1	0
WDMOD (005CH)	Bit symbol	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
	Read/Write	R/W							
	After reset	1	0	0	0	0	0	0	0
	Function	WDT control 0: Disable 1: Enable	WDT detection time 00: $2^{15}/f_{SYS}$ 01: $2^{17}/f_{SYS}$ 10: $2^{19}/f_{SYS}$ 11: $2^{21}/f_{SYS}$		Warm-up timer 0: 2^{14}/clock frequency input 1: 2^{16}/clock frequency input	HALT mode 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		0: Don’t care 1: Connects WDT output to $\overline{\text{RESET}}$ pin internally.	Pin state control in STOP mode 0: I/O off 1: Remains the state before HALT

Note 1: SYSCR1<Bit7:4> are read as “1”.

Note 2: Resetting clears <ALEEN>, <CLKEN>bit to “0”. The CLK pin is internally pulled up during reset.

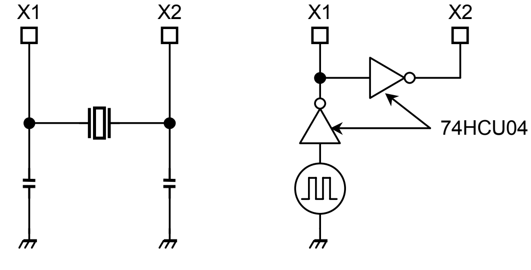
Figure 3.3.3 I/O Registers about Standby

3.3.1 System Clock Controller

The system clock controller generates system clock (f_{SYS}) for CPU core and internal I/O. It contains a oscillation circuit and clock gear circuit. The register SYSCR1<GEAR2:0> changes clock gear to either 1, 2, 4, 8, or 16 (fc, fc/2, fc/4, fc/8, or fc/16), and these functions can reduce the power consumption of the equipment in which the device is installed.

The system clock (f_{SYS}) is set to fc/32 (fc/16 × 1/2) because of <GEAR2:0> = "100" by resetting. For example, f_{SYS} is set to 0.625 MHz by resetting the case of 20 MHz oscillator is connected to X1, X2 pins.

The fc clock can be easily obtained by connecting a resonator to the X1/X2 pins respectively. Clock input from an external oscillator is also possible.



(a) Crystal/Ceramic resonator (b) External oscillator

Figure 3.3.4 Examples of Resonator Connection

* Accurate adjustment of the oscillation frequency

The CLK pin outputs at 1/2 the system clock frequency (f_{SYS}/2) is used to monitor the oscillation clock.

With a system requiring adjustment of the oscillation frequency, an adjusting program must be written.

(1) Clock gear controller

The clock gear select register SYSCR1<GEAR2:0> sets f_{FPH} to any one of fc, fc/2, fc/4, fc/8, fc/16. Switching f_{FPH} with the clock gear reduces the power consumption.

Clock setting example:

Changing gear value of the high-frequency clock

```
SYSCR1    EQU     006FH
          LD      (SYSCR1), XXXX0000B    ;    Changes fSYS to fc/2.
          LD      (SYSCR1), XXXX0100B    ;    Changes fSYS to fc/32.

          X: Don’t care
```

(High-frequency clock gear changing)

To change the frequency of the clock gear, write the value to SYSCR1<GEAR2:0> register. It is necessary to continue the warm-up time until changing after writing the register value.

There is a possibility that the instruction next to the clock-gear-changing instruction is executed by the clock gear before changing. To execute the instruction next to the clock-gear-changing instruction by the clock gear after changing, input the dummy instruction (instruction to execute the write cycle) as follows.

Example:

```
SYSCR1    EQU     006FH
          LD      (SYSCR1), XXXX 0001B   ;    Changes fSYS to fc/4.
          LD      (Dummy), 00H           ;    Dummy instruction.
          Instruction to be executed by the clock gear after changing
          X: Don’t care
```

3.3.2 Prescaler Clock Controller

The 9-bit prescaler provides a clock to 8-bit Timer 0, 1, 2, 3, 16-bit Timer 4, 5, and serial interface 0, 1.

The clock input to the 9-bit prescaler is selected either f_{FPH} or fc/16 by SYSCR0<PRCK1:0> register.

<PRCK1:0> register is initialized to "00" by resetting.

When the IDLE1 mode (Operates only oscillator) is used, set TRUN<PRRUN> to "0" to reduce the power consumption of 9-bit prescaler before "HALT" instruction is executed.

3.3.3 Internal Clock Pin Output Function

CLK pin outputs f_{SYS} divided by 2 internal clocks.

Outputs are specified by the clock output control register CKOCR<CLKEN>. Writing "1" sets clock output, and writing "0" sets high impedance.

During reset, CLK pin is internally pulled up regardless of the value of <CLKEN> register. See TMP93CS32 reset timing chart in Figure 3.1.1.

Note: To set <CLKEN> = "0" and set CLK pin to high impedance, pull up externally to prevent through current which follows to the input buffer of CLK pin.

3.3.4 Standby Controller

(1) HALT mode

When the HALT instruction is executed, the operating mode changes RUN, IDLE2, IDLE1, or STOP mode depending on the contents of the HALT mode setting register WDMOD<HALTM1:0>. Figure 3.3.5 shows the alternative states of the watchdog timer mode registers.

Watchdog Timer Mode Register

WDMOD (005CH)

	7	6	5	4	3	2	1	0
Bit symbol	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
Read/Write	R/W							
After reset	1	0	0	0	0	0	0	0
Function	Watchdog timer control 0: Disable 1: Enable	Watchdog timer detect time selection 00: $2^{15}/f_{SYS}$ 01: $2^{17}/f_{SYS}$ 10: $2^{19}/f_{SYS}$ 11: $2^{21}/f_{SYS}$		Warm-up timer 0: 2^{14}/clock frequency selection 1: 2^{16}/clock frequency selection	Halt mode selection 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		Runaway detection internal reset control 1: Executes internal reset by runaway detection	STOP mode pin control 1: Drive pins in STOP mode

Pin state control in STOP mode

0	I/O off
1	Retains the state before halt

HALT mode setting

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

Warm-up time selection at returning from the stop mode (see Table 3.3.4)

0	2^{14}/select clock frequency
1	2^{16}/select clock frequency

Figure 3.3.5 Watchdog Timer Mode Register

The futures of RUN, IDLE2, IDLE1, and STOP modes are as follows.

1. RUN: Only the CPU halts; power consumption remains unchanged.
2. IDLE2: The built-in oscillator and the specified I/O operates.
   The power consumption is redced to 1/2 than that during NORMAL operation.
3. IDLE1: Only the built-in oscillator operates, while all other built-in circuits stop.
   The power consumption is reduced to 1/5 or less than that during NORMAL operation.
4. STOP: All internal circuits including the built-in oscillator stop. This greatly reduces power consumption.

The operations in the halt state is described in Table 3.3.2.

Table 3.3.2 I/O Operation during HALT Mode

HALT Mode		RUN	IDLE2	IDLE1	STOP
WDMOD<HALTM1:0>		00	11	10	01
Block	CPU	Stop			
	I/O port	Keep the state when the "HALT" instruction was executed.			See Table 3.3.5
	8-Bit timer	Operate	Operate	Stop	Stop
	16-Bit timer	Operate	Operate	Stop	Stop
	Serial channel	Operate	Operate	Stop	Stop
	AD converter	Operate	Stop	Stop	Stop
	Watchdog timer	Operate	Operate	Stop	Stop
	Interrupt controller	Operate	Operate	Stop	Stop

(2) How to release the HALT mode

These halt states can be released by resetting or requesting an interrupt. The halt release sources are determined by the combinations between the states of interrupt mask register <IFF2:0> and the HALT modes. The details for releasing the halt status are shown in Table 3.3.3.

- Released by requesting an interrupt

  The operating released from the HALT mode depends on the interrupt enabled status. When the interrupt request level set before executing the HALT instruction exceeds the value of the interrupt mask register, the interrupt due to the source is processed after releasing the HALT mode, and CPU starts executing an instruction that follows the HALT instruction. When the interrupt request level set before executing the HALT instruction is less than the value of the interrupt mask register, releasing the HALT mode is not executed. (In non-maskable interrupts, interrupt processing is processed after releasing the HALT mode regardless of the value of the mask register.)

  However only for INT0 interrupts, even if the interrupt request level set before executing the HALT instruction is less than the value of the interrupt mask register, releasing the HALT mode is executed. In this case, interrupt processing is not processed, and CPU starts executing the instruction next to the HALT instruction, but the interrupt request flag is held at "1".

  Note: Usually, interrupts can release all halts status. However, the interrupts = ($\overline{\text{NMI}}$, INT0) which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 3 clocks of f_{FPH}) with IDLE1 or STOP mode (IDLE2/RUN are 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 resetting

  Releasing all halt status is executed by resetting.

  When the STOP mode is released by RESET, it is necessary enough resetting time (3 ms or more) to set the operation of the oscillator to be stable.

  When releasing the HALT mode by resetting, the internal RAM data keeps the state before the HALT instruction is executed. However the other setting contents are initialized. (Releasing due to interrupts keep the state before the HALT instruction is executed.)

Table 3.3.3 Halt Releasing Source and Halt Releasing Operation

Interrupt Receiving Status			Interrupt Enable (Interrupt level) ≥ (Interrupt mask)				Interrupt Disable (Interrupt level) < (Interrupt mask)			
HALT Mode			RUN	IDLE2	IDLE1	STOP	RUN	IDLE2	IDLE1	STOP
Halt releasing source	Interrupt	NMI	♦	♦	♦	♦*1	–	–	–	–
		INTWDT	♦	×	×	×	–	–	–	–
		INT0	♦	♦	♦	♦*1	○	○	○	○*1
		INT4 to INT7	♦	♦	×	×	×	×	×	×
		INTT0 to INTT3	♦	♦	×	×	×	×	×	×
		INTTR4 to INTTR7	♦	♦	×	×	×	×	×	×
		INTO4, INTO5	♦	♦	×	×	×	×	×	×
		INTRX0, TX0	♦	♦	×	×	×	×	×	×
		INTRX1, TX1	♦	♦	×	×	×	×	×	×
		INTAD	♦	×	×	×	×	×	×	×
		RESET	♦	♦	♦	♦	♦	♦	♦	♦

♦: After releasing the HALT mode, CPU starts interrupt processing. (RESET initializes LSI.)

○: After releasing the HALT mode, CPU starts executing an instruction that follows the HALT instruction.

×: It can not be used to release the HALT mode.

–: This combination type does not exist because the priority level (interrupt request level) of non-maskable interrupts is fixed to highest priority level "7".

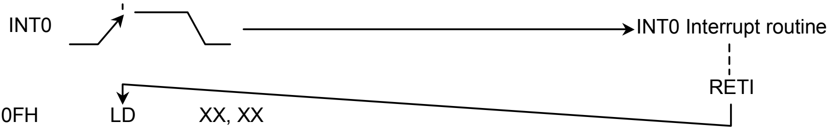
*1: Releasing the HALT mode is executed after passing the warm-up time.

Note: When releasing the HALT mode is executed by INT0 interrupt of the level mode in the interrupt enabled status, hold level "H" until starting interrupt processing. If level "L" is set, interrupt processing is correctly started.

(Example releasing "RUN" mode)

INT0 interrupt releases halt state when the RUN mode is on.

```
Address
8203H    LD      (IIMC), 00H          ;   Selects INT0 interrupt rising edge.
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.
```



```
820FH    LD      XX, XX
```

(3) Operation

1. RUN mode

In the RUN mode, the system clock continues to operate even after a HALT instruction is executed. Only the CPU stops executing the instruction.

In the halt state, an interrupt request is sampled with the falling edge of the "CLK" signal.

Releasing the RUN mode is executed by the external/internal interrupts. (See Table 3.3.3 Halt Releasing Source and Halt Releasing Operation.)

Figure 3.3.6 shows the interrupt timing for releasing the halt state by interrupts in the RUN/IDLE2 mode.

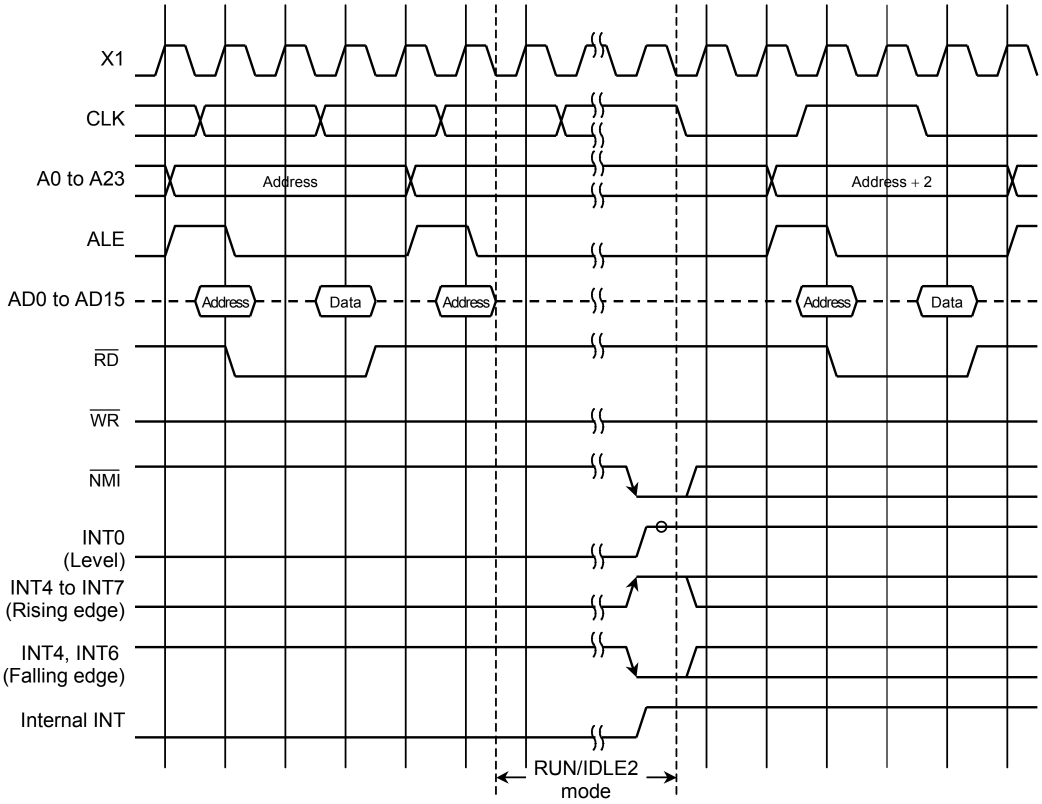


Figure 3.3.6 Timing Chart for Releasing the Halt State by Interrupt in RUN/IDLE2 Modes

2. IDLE2 mode

In the IDLE2 mode, the system clock is supplied to only specific internal I/O devices, and the CPU stops executing the current instruction.

In the IDLE2 mode, the halt state is released by an interrupt with the same timing as in the RUN mode. The IDLE2 mode is released by external/internal interrupt, except INTWDT/INTAD interrupts. (See Table 3.3.3 Halt Releasing Source and Halt Releasing Operation.)

In the IDLE2 mode, the watchdog timer should be disabled before entering the halt status to prevent the watchdog timer interrupt occurring just after releasing the HALT mode.

3. IDLE1 mode

In the IDLE1 mode, only the internal oscillator operates. The system clock in the MCU stops, the CLK pin is fixed at the level "H" in the output enable (CKOCR<CLKEN> = "1").

In the halt state, and interrupt request is sampled aynchronunsly with the system clock, however the halt release (restart of operation) is performed synchronously with it.

IDLE1 mode is released by external interrupts ($\overline{\text{NMI}}$, INT0). (See Table 3.3.3 Halt Releasing Source and Halt Releasing Operation.)

When the IDLE1 mode is used, setting TRUN<PRRUN> to "0" to stop 9, 5-bit prescaler before "HALT" instruction reduces the power consumption.

Figure 3.3.7 illustrates the timing for releasing the halt state by interrupts in the IDLE1 mode.

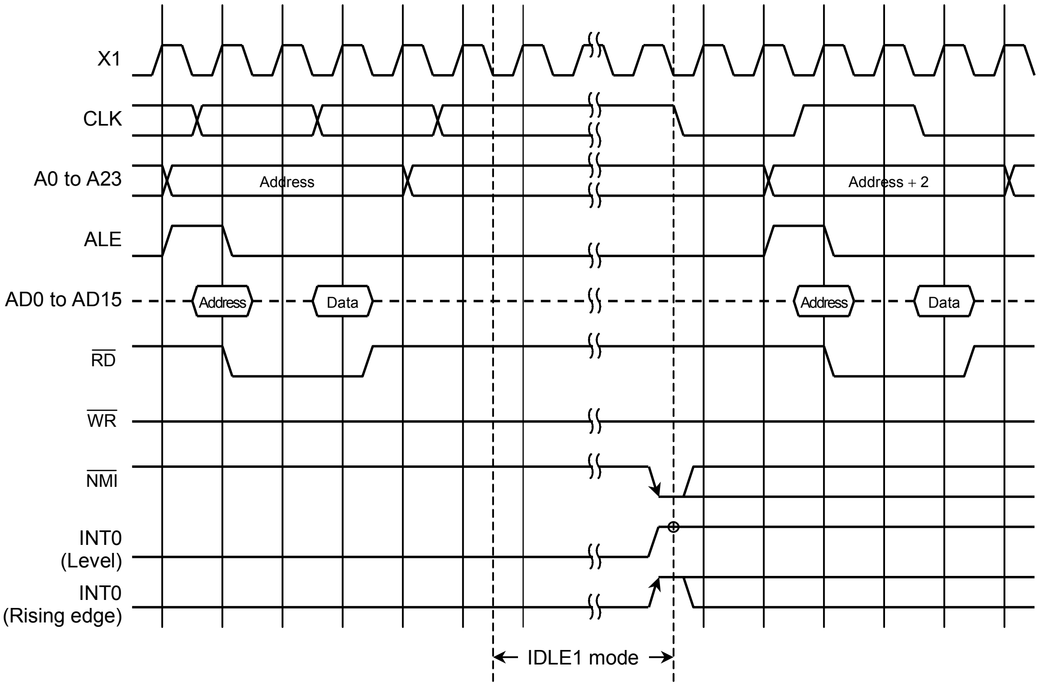


Figure 3.3.7 Timing Chart of Halt Released by Interrupts in IDLE1 Mode

4. STOP mode

The STOP mode is selected to stop all internal circuits including the internal oscillator. The pin status in the STOP mode depends on setting of a bit in the watchdog timer mode register WDMOD<DRVE>. (See Figure 3.3.5 for setting of WDMOD<DRVE>.) Table 3.3.5 summarizes the state of these pins in the STOP mode.

The STOP mode is released by external interrupts ($\overline{\text{NMI}}$, INT0). When the STOP mode is released, the system clock output starts after the warm-up time required to attain stable oscillation. The warm-up time can be set using WDMOD<WARM>. See the example of warm-up time (Table 3.3.4).

In a system which supplies stable clock generated by an external oscillator, the warm-up time can be reduced by using the setting of T45CR<QCU>.

Figure 3.3.8 illustrates the timing for releasing the halt state by interrupts during the STOP mode.

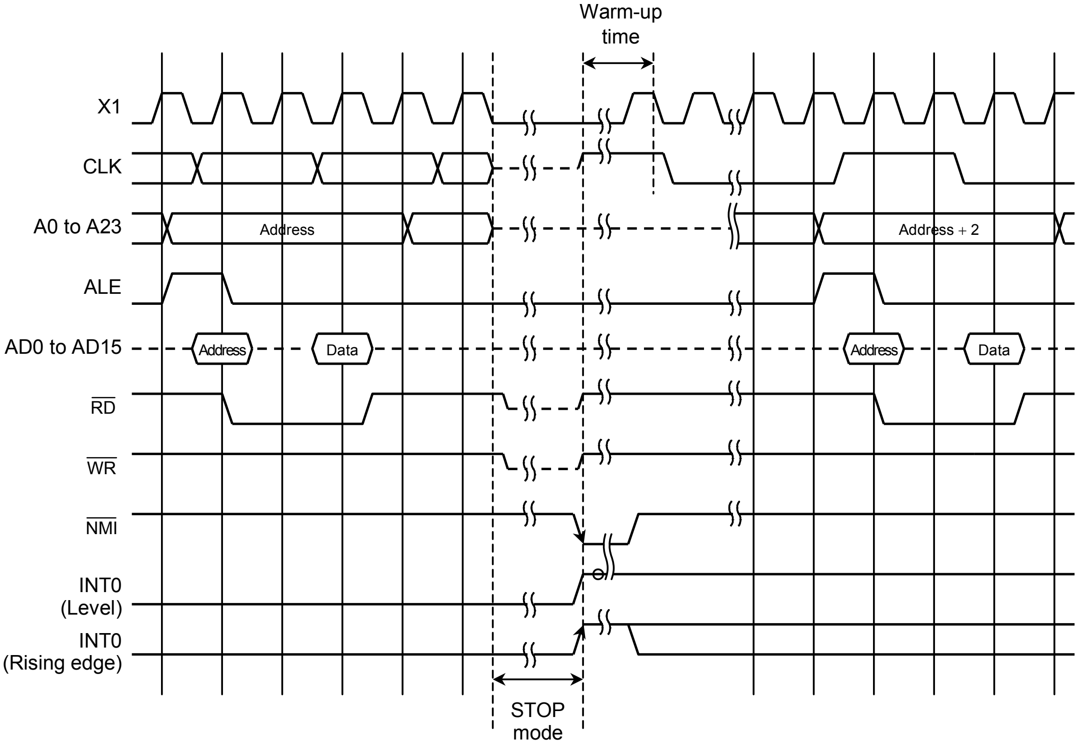


Figure 3.3.8 Timing Chart of Halt State Release by Interrupts in STOP Mode

Table 3.3.4 The Example of Warm-up Time after Releasing the STOP Mode

Clock Operation Frequency after the STOP Mode	Warm-up Time [ms]		Remark
	WDMOD<WARM> = 0	WDMOD<WARM> = 1	
fc	0.8192	3.2768	fc = 20 MHz
fc/2	1.6384	6.5536	
fc/4	3.2768	13.1072	
fc/8	6.5536	26.2144	
fc/16	13.1072	52.4288	

How to calculate the warm-up time

WDMOD<WARM> = “0”: 2^{14}/(Clock operation frequency after releasing the HALT in STOP mode)
WDMOD<WARM> = “1”: 2^{16}/(Clock operation frequency after releasing the HALT in STOP mode)

Table 3.3.5 Pin States in STOP Mode

Pin Name	I/O	<DRVE> = 0	<DRVE> = 1
P00 to P07	Input mode Output mode AD0 to AD7	Δ High-Z High-Z	Δ Output High-Z
P10 to P17	Input mode Output mode/A8 to A15 AD8 to AD15	Δ High-Z High-Z	Δ Output High-Z
P20 to P27	Input mode Output mode, A0 to A7/A16 to A23	Δ Δ	Δ Output
P30 ($\overline{\text{RD}}$), P31 ($\overline{\text{WR}}$)	Output	High-Z	Output
P32 ($\overline{\text{HWR}}$)	Input mode Output mode	PU* PU*	PU Output
P35	Input mode Output mode	Invalid High-Z	Invalid Output
P41 to P47	Input mode Output mode	Invalid High-Z	Invalid Output
P50 to P55	Input	Δ	Δ
P60 to P65	Input mode Output mode	PU* PU*	PU Output
P70, P71	Input mode Output mode	Invalid High-Z	Invalid Output
$\overline{\text{NMI}}$	Input	Input	Input
ALE	Output (<ALEEN> = 1)	"L" level output	"L" level output
CLK	Output (<CLKEN> = 1)	High-Z	"H" level output
$\overline{\text{RESET}}$	Input	Input	Input
$\overline{\text{EA}}$	Input	"H" level fix	"H" level fix
AM8/$\overline{\text{AM16}}$	Input	"H" level fix	"H" level fix
X1	Input	Invalid	Invalid
X2	Output	"H" level output	"H" level output

Input: Input gate in operation. Fix input voltage to 0 or 1 so that the input pin stays constant.

Output: Output state

Invalid: Input is not accepted.

High-Z: Output is at high impedance.

PU: Programmable pull-up pin in input gate in operation. Fix the pin to avoid through current since the input gate operates when a pull-up pin resistance is not set.

PU*: Programmable pull-up pin in input gate disable state. No through current even if the pin is set to high impedance.

Δ: When a HALT instruction is executed and the CPU stops at the address of the port register, an input gate operates. Fix the pin to avoid through current, and change the program.
In all other cases, input is not accepted.

Note: Port registers are used for controlling programmable pull up. If a pin is also used for an output function (e.g. TO3) and the output function is specified, whether pull up is selected depends on the output function data. If a pin is also used for an input function, whether pull up is selected depends on the port register setting value only.

3.4 Interrupts

TLCS-900 interrupts are controlled by the CPU interrupt mask flip-flop (IFF2 to IFF0) and the built-in interrupt controller.

Altogether the TMP93CS32 has the following 31 interrupt sources:

- Internal interrupts … 9
  SWI instruction, Illegal instruction execution
- Interrupts from external pins ($\overline{\text{NMI}}$, INT0, INT4 to INT7) … 6
- Interrupts from built-in I/Os … 16

A fixed individual interrupt vector number is assigned to each interrupt source; six levels of priority can also be assigned to each maskable interrupt. Non-maskable interrupts have a fixed priority of 7.

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

The CPU compares the value of the priority sent with the value in the CPU interrupt mask register <IFF2:0>. If the value is greater than that the CPU interrupt mask register, the interrupt is accepted. The value in the CPU interrupt mask register <IFF2:0> can be changed using the EI instruction (Executing EI n changes the contents of <IFF2:0> to n). For example, programming EI 3 enables acceptance of maskable interrupts with a priority of 3 or greater, and non-maskable interrupts which are set in the interrupt controller. The DI instruction (<IFF2:0> = 7) operates in the same way as the EI 7 instruction. Since the priority values for maskable interrupts are 0 to 6, the DI instruction is used to disable acceptance of maskable interrupts. The EI instruction becomes effective immediately after execution (With the TLCS-90, the EI instruction becomes effective after execution of the subsequent instruction).

In addition to the general-purpose interrupt processing mode described above, there is also a Micro DMA processing mode. Micro DMA is a mode used by the CPU to automatically transfer byte or word data. It enables the CPU to process interrupts such as data saves to built-in I/Os at high speed.

Figure 3.4.1 is a flowchart showing overall interrupt processing.

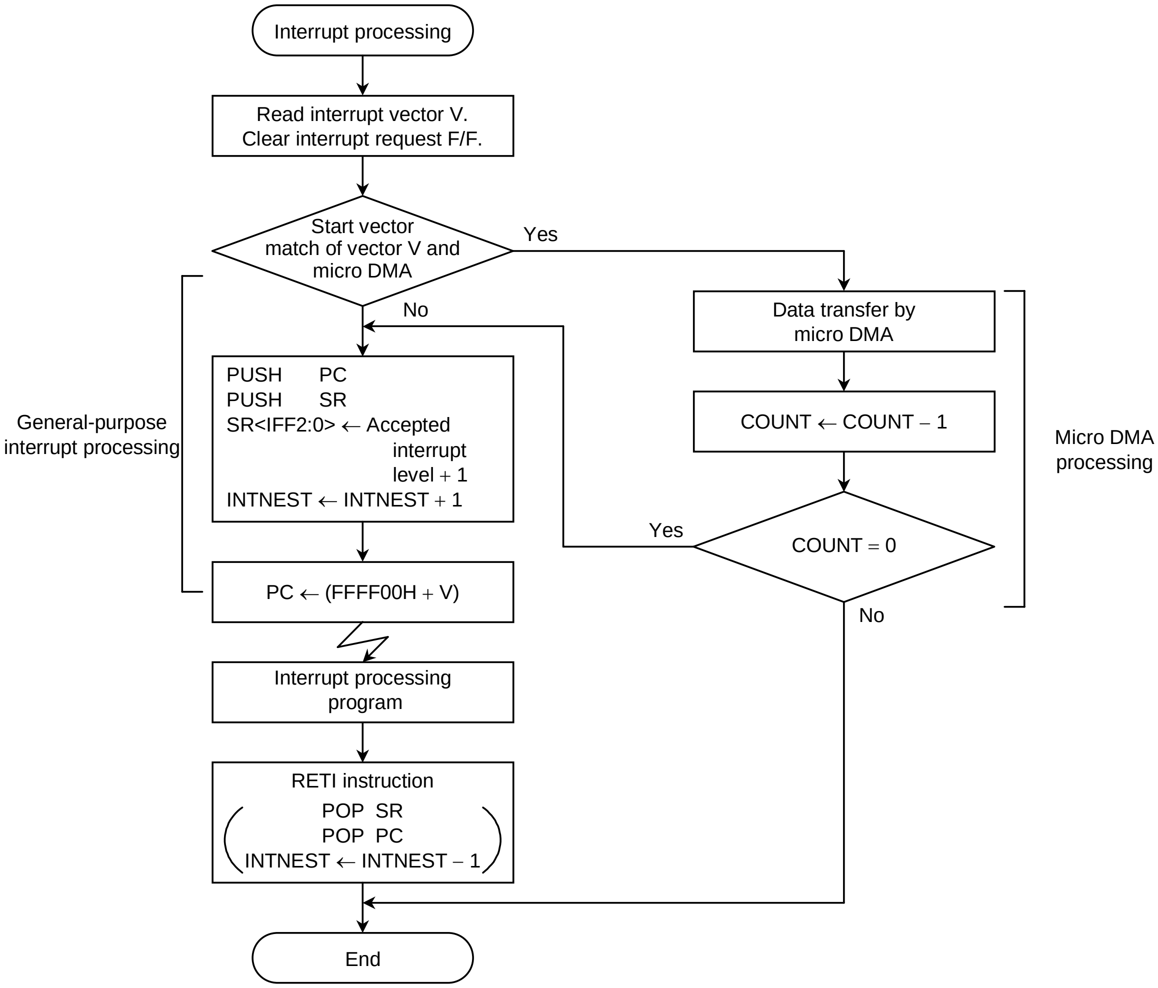


Figure 3.4.1 Interrupt Processing Flowchart

3.4.1 General-Purpose Interrupt Processing

When accepting an interrupt, the CPU operates as follows. In the cases of software interrupts or interrupts generated by the CPU because of attempts to execute illegal instructions, the following steps (1) and (3) are not executed.

(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 controller generates 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 and the status register to the system stack area (Area indicated by the system mode stack pointer (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 increments the INTNEST (Interrupt nesting counter).

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

The following diagram shows all the above processing state number.

Bus Width of Stack Area	Bus Width of Interrupt Vector Area	Interrupt Processing State Number
8 bits	8 bits	35
	16 bits	31
16 bits	8 bits	29
	16 bits	25

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 INTNEST (Interrupt nesting counter).

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.

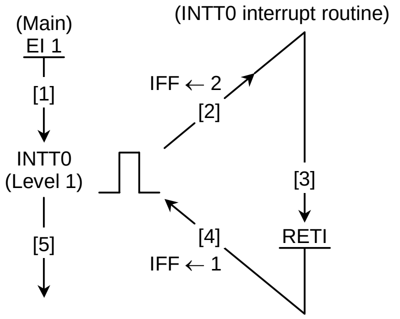
The interrupt request with a priority higher than the accepted now interrupt during the CPU is processing above (1) to (5) is accepted before the 1'st instruction in the interrupt processing routine, causing interrupt processing to nest. (This is the same case of over lapped each Non-maskable interrupt (level 7).) (Non-maskable interrupts (level 7) can be accepted, causing interrupt processing to rest.)

The CPU does not accept an interrupt request of the same level as that of the interrupt being processed.

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

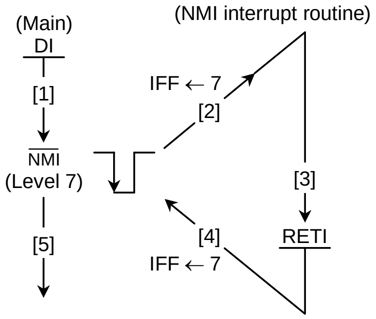
The following (1) to (5) show a flowchart of interrupt processing.

(1) Maskable



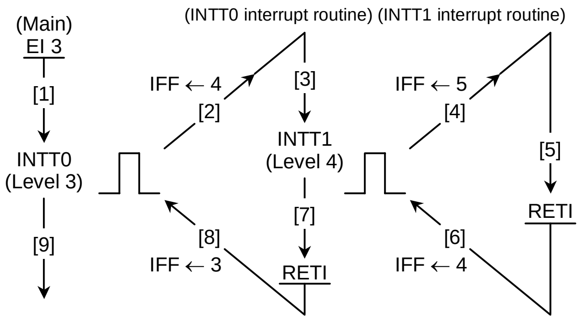
During execution of the main program, the CPU accepts an interrupt request. The CPU increments the IFF so that the interrupts of level 1 are not accepted during processing the interrupt routine.

(2) Non-maskable interrupt



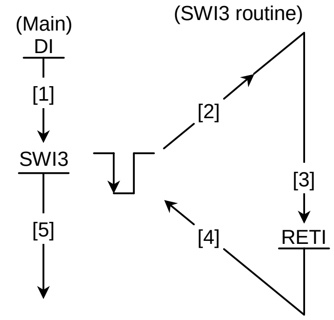
DI instruction is executed in the main program, so that the interrupts of only level 7 are accepted. The CPU does not increment the IFF even if the CPU accepts an interrupt request of level 7.

(3) Interrupt nesting



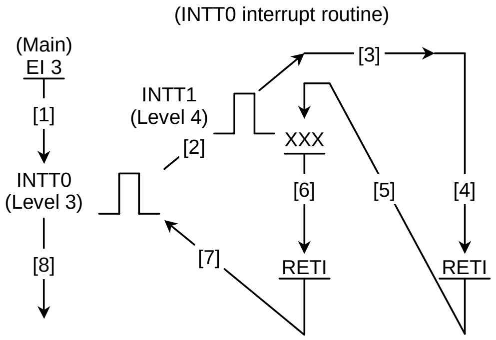
During processing the interrupts of level 3, the IFF is set to 4. When an interrupt with a level higher than level 4 is generated, the CPU accepts the interrupt with the higher level, causing interrupt processing to nest.

(4) Software interrupt



The CPU accepts the software interrupt request during DI status(IFF = 7) because of the level 7. The IFF is not changed by the software interrupts.

(5) Interrupt sampling timing



If an interrupt with a level higher than the interrupt being processed is generated, the CPU accepts the interrupt with the higher level. The program counter which returns at e is the start address of INTT0 interrupt routine.

Example: ___ (Underline): Instruction
[1], [2] …: Execution flow

The addresses FFFF00H to FFFFFFH (256 bytes) of the TMP93CS32 are assigned for interrupt vector area.

Table 3.4.1 TMP93CS32 Interrupt Table

Default Priority	Type	Interrupt Source	Vector Value "V"	Address Refer to Vector	Micro DMA Start Vector
1	Non-maskable	Reset or SWI0 instruction	0000H	FFFF00H	–
2		SWI 1 instruction	0004H	FFFF04H	–
3		Illegal instruction or SWI2	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		NMI: $\overline{\text{NMI}}$ pin input	0020H	FFFF20H	08H
10		INTWD: Watchdog timer	0024H	FFFF24H	09H
11	Maskable	INT0 : INT0 pin input	0028H	FFFF28H	0AH
12		(Reserved)	002CH	FFFF2CH	–
13		INT4: INT4 pin input	0030H	FFFF30H	0CH
14		INT5: INT5 pin input	0034H	FFFF34H	0DH
15		INT6: INT6 pin input	0038H	FFFF38H	0EH
16		INT7: INT7 pin input	003CH	FFFF3CH	0FH
17		INTT0: 8-bit timer 0	0040H	FFFF40H	10H
18		INTT1: 8-bit timer 1	0044H	FFFF44H	11H
19		INTT2: 8-bit timer 2	0048H	FFFF48H	12H
20		INTT3: 8-bit timer 3	004CH	FFFF4CH	13H
21		INTTR4: 16-bit timer4 (TREG4)	0050H	FFFF50H	14H
22		INTTR5: 16-bit timer 4 (TREG5)	0054H	FFFF54H	15H
23		INTTR6: 16-bit timer 5 (TREG6)	0058H	FFFF58H	16H
24		INTTR7: 16-bit timer 5 (TREG7)	005CH	FFFF5CH	17H
25		INTTO4: 16-bit timer 4 (Overflow)	0060H	FFFF60H	18H
26		INTTO5: 16-bit timer 5 (Overflow)	0064H	FFFF64H	19H
27		INTRX0: Serial receive (Channel 0)	0068H	FFFF68H	1AH
28		INTTX0: Serial send (Channel 0)	006CH	FFFF6CH	1BH
29		INTRX1: Serial receive (Channel 1)	0070H	FFFF70H	1CH
30		INTTX1: Serial send (Channel 1)	0074H	FFFF74H	1DH
31		INTAD: AD conversion completion	0078H	FFFF78H	1EH
–		(Reserved)	007CH	FFFF7CH	–
to		to	to	to	to
–		(Reserved)	00FCH	FFFFFCH	–

Setting to reset/interrupt vector

1. Reset vector

FFFF00H	PC<7:0>
FFFF01H	PC<15:8>
FFFF02H	PC<23:16>
FFFF03H	XX

The vector base addresses are depended on the products.

Type No.	Vector Base Address	PC Setting Sequence after Reset	Notes
TMP93CS32 TMP93PW32	FFFF00H	PC<7:0> ← Address FFFF00H PC<15:8> ← Address FFFF01H PC<23:16> ← Address FFFF02H	P27 to P20/A23 to A16 pins input ports with pull-up due to reset. The logic data is "FFH". When Port 2 is used as A23 to A16 pins to access the program ROM, set PC <23:16> to "FFH" and the reset vector to "FF0000H to FFFFFFH" (for mainly products without ROM).

2. Interrupt vector (Except reset vector)

Address refer to vector

+0	PC<7:0>
+1	PC<15:8>
+2	PC<23:16>
+3	XX

XX: Don't care

(Setting Example)

Sets the RESET vector: FF0000H, NMI vector: FF9ABCH, INTAD vector: 123456H.

```
ORG     FFFF00H
DL      FF0000H                 ;   RESET = FF0000H
ORG     FFFF20H
DL      FF9ABCH                 ;   NMI = FF9ABCH
ORG     FFFF78H
DL      123456H                 ;   INTAD = 123456H
ORG     FF0000H
LD      A, B
     ¦
ORG     FF9ABCH
LD      B, C
     ¦
ORG     123456H
LD      C, A
     ¦
```

Note:
ORG, DL are assembler directives.

- ORG: Control location counter
- DL: Defines long word (32-bit) data

3.4.2 Micro DMA

In addition to the conventional interrupt processing, the TLCS-900 also has a micro DMA function. When an interrupt is accepted, in addition to an interrupt vector, the CPU receives data indicating whether processing is micro DMA mode or general-purpose interrupt. If micro DMA mode is requested, the CPU performs micro DMA processing.

The TLCS-900 can process at very high speed because it has transfer parameters in dedicated registers in the CPU. Since those dedicated registers are assigned as CPU control registers, they can only be accessed by the LDC instruction.

(1) Micro DMA operation

Micro DMA operation starts when the accepted interrupt vector value matches the micro DMA start vector value. The micro DMA has four channels so that it can be set for up to four types of interrupt source.

When a micro DMA interrupt is accepted, data is automatically transferred from the transfer source address to the transfer destination address set in the control register, and the transfer counter is decremented. If the value in the counter after decrementing is other than 0, micro DMA processing is completed; if the value in the counter after decrementing is 0, general-purpose interrupt processing is performed.

32-bit control registers are used for setting transfer source/destination addresses. However, the TLCS-900 has only 24 address pins for output. A 16-Mbyte space is available for the micro DMA.

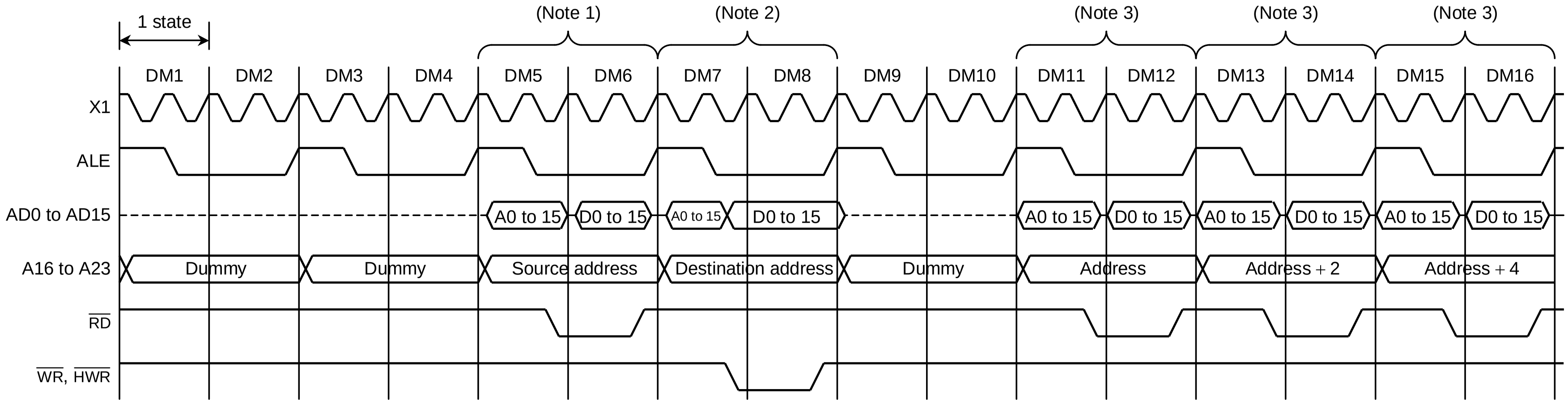
There are two data transfer modes: one-byte mode and one-word mode. Incrementing, decrementing, and fixing the transfer source/destination address after transfer can be done in both modes. Therefore data can easily be transferred between I/O and memory and between I/Os. For details of transfer modes, see the description of transfer mode registers.

The transfer counter has 16 bits, so up to 65536 transfers (The maximum when the initial value of the transfer counter is 0000H.) can be performed for one interrupt source by micro DMA processing.

When the transfer counter is decremented to “0” after data is transferred with micro DMA, general-purpose interrupt processing is performed. After processing the general-purpose interrupt, starting the interrupts of the same channel restarts the transfer counter from 65536. If necessary, reset the transfer counter.

Interrupt sources processed by micro DMA processing are those with the Micro DMA start vectors listed in Table 3.4.1.

The following timing chart is a micro DMA cycle of the transfer address INC (increment) mode (Condition: MAX mode, 16-bit bus width for 16 Mbytes, 0 waits).

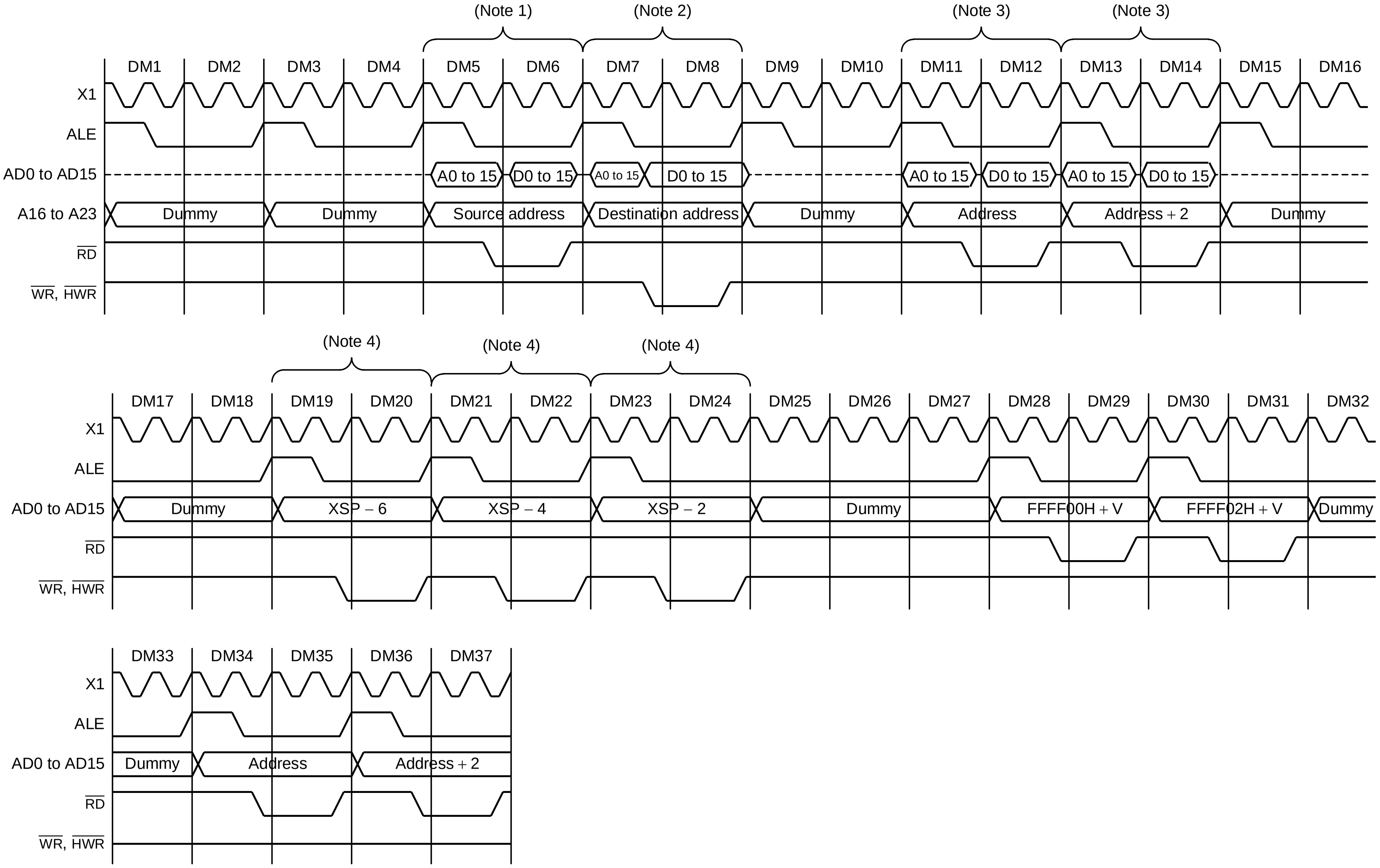


Note 1: These 2 states are added in the case that the bus width of the source address area is 8 bits or the address starts from an odd number.

Note 2: These 2 states are added in the case that the bus width of the destination address area is 8 bits or the address starts from an odd number.

Note 3: This may be a dummy cycle with an instruction queue buffer.

Figure 3.4.2 Micro DMA Cycle (COUNT ≠ 0)



Note 1: These 2 states are added in the case that the bus width of the source address area is 8 bits or the address starts from an odd number.

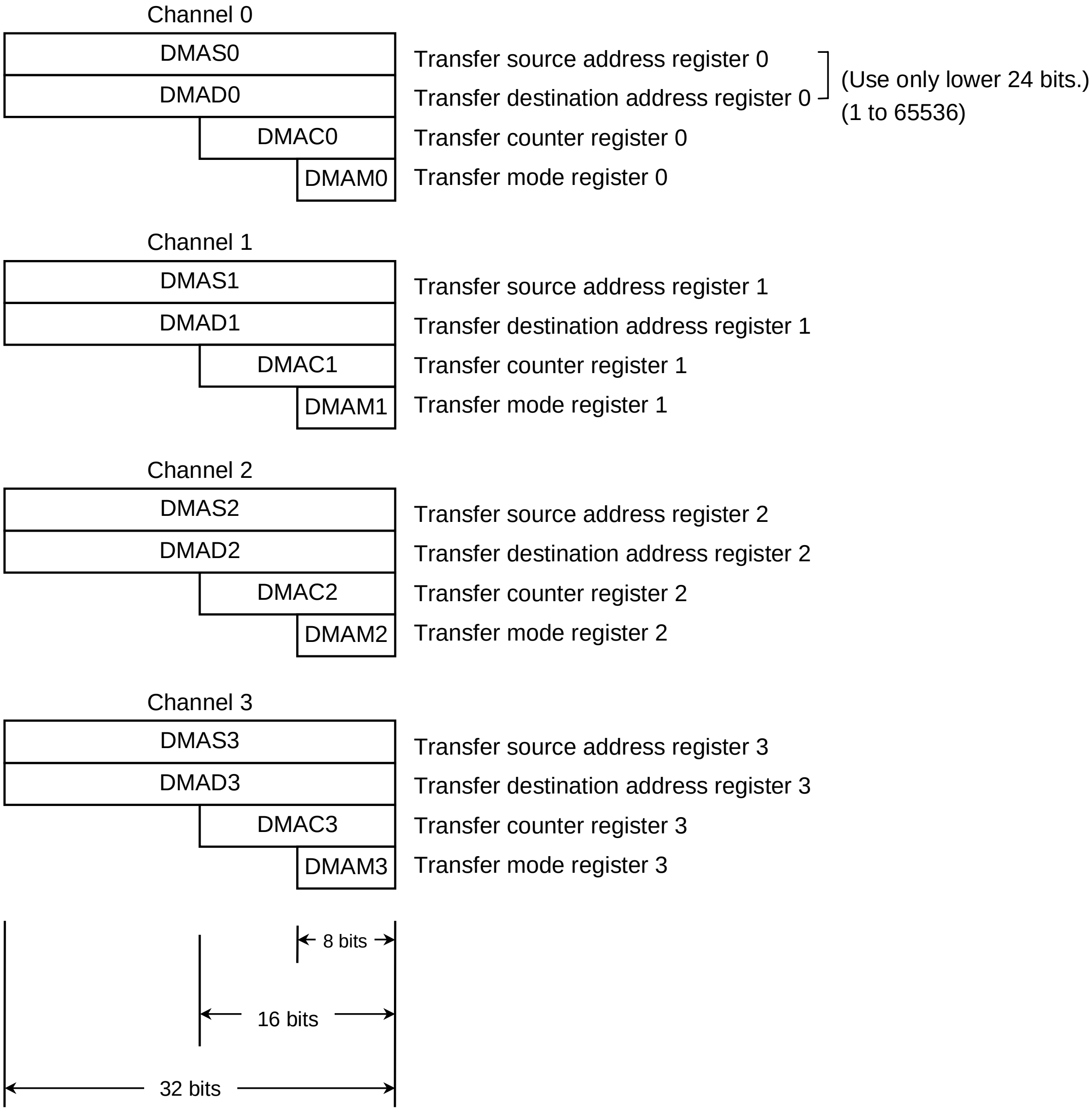
Note 2: These 2 states are added in the case that the bus width of the destination address area is 8 bits or the address starts from an odd number.

Note 3: This will be a dummy cycle with an instruction queue buffer.

Note 4: These 2 states are added in the case that the bus width of the stack address area is 8 bits or the stack pointer starts from an odd number.

Figure 3.4.3 Micro DMA Cycle (COUNT = 0)

(2) Register configuration (CPU control register)

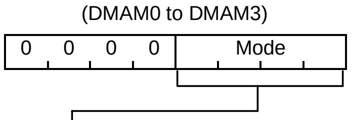


These control register can not be set only “LDC cr, r” instruction.

Example:

```
LD      XWA, 100H
LDC     DMAS0, XWA
LD      XWA, 50H
LDC     DMAD0, XWA
LD      WA, 40H
LDC     DMAC0, WA
LD      A, 05H
LDC     DMAM0, A
```

(3) Transfer mode register details



Note: When setting values for this register, clear the upper 4 bits to 0.

Execution time (Min) at 20 MHz

Z: 0 = Byte transfer, 1 = Word transfer

0	0	0	Z	Transfer destination address INC mode for I/O to memory (DMADn+) ← (DMASn) DMACn ← DMACn − 1 if DMACn = 0 then INT.	16 states (1.6 μs)
0	0	1	Z	Transfer destination address DEC mode for I/O to memory (DMADn−) ← (DMASn) DMACn ← DMACn − 1 if DMACn = 0 then INT.	16 states (1.6 μs)
0	1	0	Z	Transfer source address INC mode...........for memory to I/O (DMADn) ← (DMASn+) DMACn ← DMACn − 1 if DMACn = 0 then INT.	16 states (1.6 μs)
0	1	1	Z	Transfer source address DEC mode ....for memory to I/O (DMADn) ← (DMASn−) DMACn ← DMACn − 1 if DMACn = 0 then INT.	16 states (1.6 μs)
1	0	0	Z	Fixed address mode...I/O to I/O (DMADn) ← (DMASn) DMACn ← DMACn − 1 if DMACn = 0 then INT.	16 states (1.6 μs)
1	0	1	1	Counter mode .... for interrupt counter DMASn ← DMASn + 1 DMACn ← DMACn − 1 if DMACn = 0 then INT.	11 states (1.1 μs)

(1 states = 100 ns at 20 MHz, high frequency mode)

Note 1: n: Corresponds to micro DMA channels 0 to 3.
DMADn+/DMASn+: Post-increment (Increments register value after transfer.)
DMADn−/DMASn−: Post-decrement (Decrement register value after transfer.)

Note 2: Execution time: When setting source address/destination address area to 16-bit bus, 0 waits.
Clock condition: fc = 20 MHz, clock gear: 1(fc)

Note 3: Do not use the codes other than the above mentioned codes for transfer mode register.

3.4.3 Interrupt Controller

Figure 3.4.4 is a block diagram of the interrupt circuits. The left half of the diagram shows the interrupt controller; the right half includes the CPU interrupt request signal circuit and the halt release signal circuit.

Each interrupt channel (Total of 22 channels) in the interrupt controller has an interrupt request flip-flop, interrupt priority setting register, and a register for storing the micro DMA start vector. The interrupt request fip-flop is used to latch interrupt requests from peripheral devices.

The flip-flop is cleared to 0 at reset, when the CPU reads the interrupt channel vector after the acceptance of interrupt, or when the CPU executes an instruction that clears the interrupt of that channel (Writes 0 in the clear bit of the interrupt priority setting register).

For example, to clear the INT0 interrupt request, set the register after the DI instruction as follows.

```
          LD        (INTE0AD), - - - - 0 - - - B
```

The status of the interrupt request flip-flop is detected by reading the clear bit. Detects whether there is an interrupt request for an interrupt channel.

The interrupt priority can be set by writing the priority in the interrupt priority setting register (e.g., INTE0AD, INTE45, etc.) provided for each interrupt source. Interrupt levels to be set are from 1 to 6. Writing 0 or 7 as the interrupt priority disables the corresponding interrupt request. The priority of the non-maskable interrupt ($\overline{\text{NMI}}$ pin, watchdog timer, etc.) is fixed to 7. If interrupt requests with the same interrupt level are generated simultaneously, interrupts are accepted in accordance with the default priority (The smaller the vector value, the higher the priority).

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 sent; 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 requests 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 four registers used to store the Micro DMA start vector. These are I/O registers; unlike other micro DMA registers (DMAS, DMAD, DMAM, and DMAC). Writing the start vector of the interrupt source for the micro DMA processing (see Table 3.4.1), enables the corresponding interrupt to be processed by micro DMA processing. The values must be set in the micro DMA parameter registers (e.g., DMAS and DMAD) prior to the micro DMA processing.

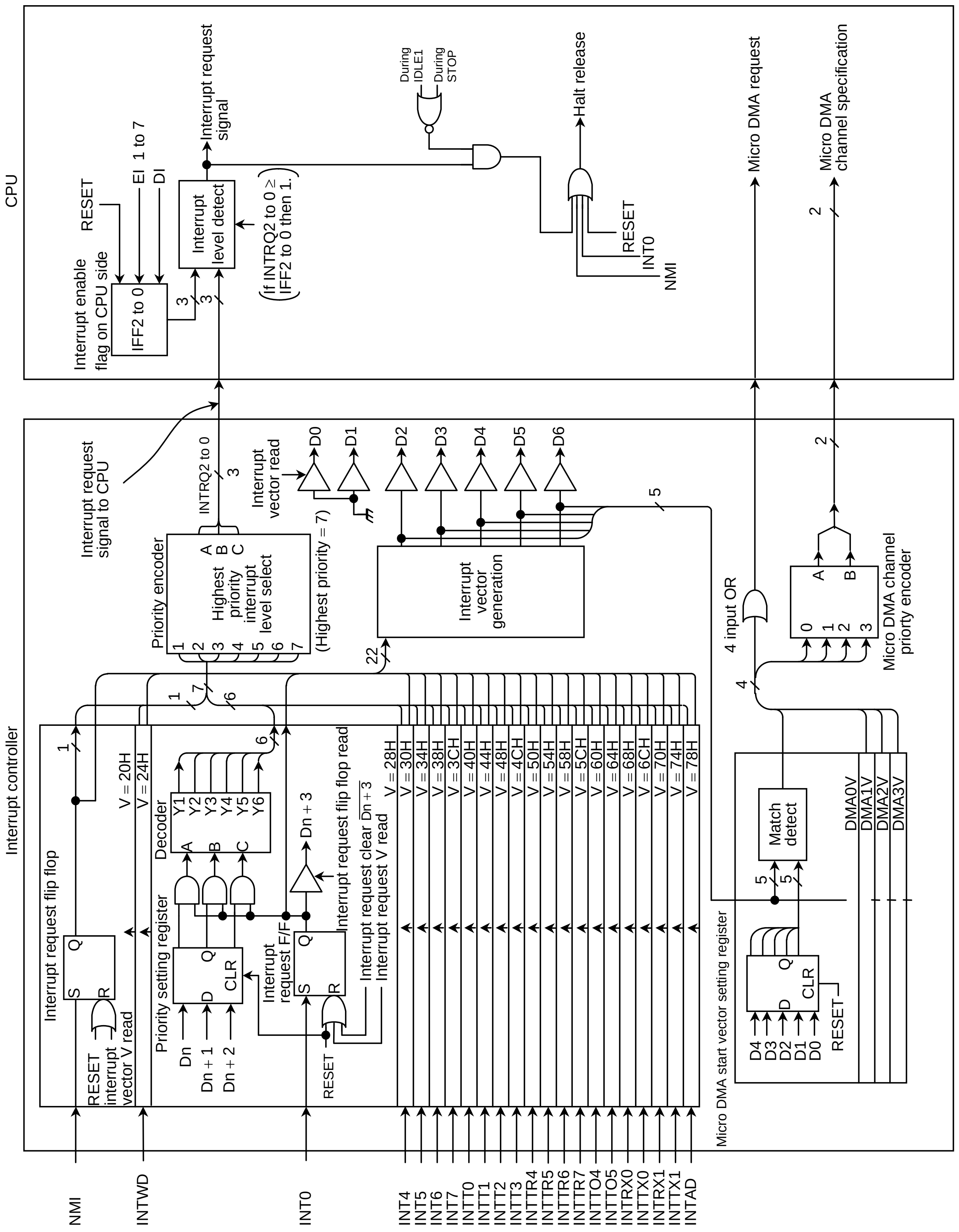


Figure 3.4.4 Block Diagram of Interrupt Controller

(1) Interrupt priority setting register

Symbol	Address	7	6	5	4	3	2	1	0	
INTE0AD	0070H	INTAD				INT0				←Interrupt source
		IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0	←Bit symbol
		R/W	W			R/W	W			←Read/Write
		0	0	0	0	0	0	0	0	←After reset
INTE45	0071H	INT5				INT4				
		I5C	I5M2	I5M1	I5M0	I4C	I4M2	I4M1	I4M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTE67	0072H	INT7				INT6				
		I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTET10	0073H	INTT1 (Timer1)				INTT0 (Timer0)				
		IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTET32	0074H	INTT3 (Timer3)				INTT2 (Timer2)				
		IT3C	IT3M2	IT3M1	IT3M0	IT2C	IT2M2	IT2M1	IT2M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTET54	0075H	INTTR5 (TREG5)				INTTR4 (TREG4)				
		IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTET76	0076H	INTTR7 (TREG7)				INTTR6 (TREG6)				
		IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTEO54	0077H	INTTO5				INTTO4				
		ITO5C	ITO5M2	ITO5M1	ITO5M0	ITO4C	ITO4M2	ITO4M1	ITO4M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTES0	0078H	INTTX0				INTRX0				
		ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	
INTES1	0079H	INTTX1				INTRX1				
		ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0	
		R/W	W			R/W	W			
		0	0	0	0	0	0	0	0	

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	Prohibits interrupt request.

IxxC	Function (Read)	Function (Write)
0	Indicates no interrupt request.	Clears interrupt request flag.
1	Indicates interrupt request.	- - - - - Don’t care - - - - -

Note 1: Read-modify-write is prohibited.

Note 2: Note about clearing interrupt request flag
The interrupt request flag of INTRX0, INTRX1 are not cleared by writing “00” to IXXC because of they are level interrupts. They can be cleared only by resetting or reading SCBUFn.

Figure 3.4.5 Interrupt Priority Setting Register

(2) External interrupt control

Interrupt Input Mode Control Register

		7	6	5	4	3	2	1	0
IIMC (007BH)	Bit symbol			−			I0IE	I0LE	NMIREE
	Read/Write			W			W		
	After reset			0			0	0	0
Read-modify-write instruction prohibited	Function			(Note) Always write “0”.			1: INT0 input enable	0: INT0 edge mode 1: INT0 level mode	1: Can be accepted in $\overline{\text{NMI}}$ rising edge.

INT0 input enable (Note 1)

0	INT0 disable (P35 function only)
1	Input enable

$\overline{\text{NMI}}$ rising edge enable

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

INT0 level enable (Note 2)

0	Rising edge detect interrupt
1	High level interrupt

Note 1: The INT0 pin can also be used for standby release as described later. Even if the pin is not used for standby release, setting this register to “0” maintains the port function during standby mode.

Note 2: Case of changing from level to edge for INT0 pin mode (<I0LE> “1” → “0”)

Execution example:

```
LD     (INTE0AD), XXXX0000B   ;   INT0 disable, clean the request flag.
LD     (IIMC), XXXXX10XB      ;   Change from level to edge.
LD     (INTE0AD), XXXX0nnnB   ;   Set  interrupt level “n” for INT0, clear the request flag.
```

Note 3: IIMC<bit7:3> is always read as “1”.

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

Figure 3.4.6 Interrupt Input Mode Control Register

Table 3.4.2 Setting of External Interrupt Pin Functions

Interrupt	Shared Pin	Mode	Setting Method
NMI	$\overline{\text{NMI}}$ (Dedicated pin)	Falling edge	IIMC<NMIREE> = 0
		Falling and rising edges	IIMC<NMIREE> = 1
INT0	P35	Rising edge	IIMC<I0LE> = 0, <I0IE> = 1
		Level	IIMC<I0LE> = 1, <I0IE> = 1
INT4	P42	Rising edge	T4MOD<CAP12M1:0> = 0, 0 or 0, 1 or 1, 1
		Falling edge	T4MOD<CAP12M1:0> = 1, 0
INT5	P43	Rising edge	− − −
INT6	P45	Rising edge	T5MOD<CAP34M1:0> = 0, 0 or 0, 1 or 1, 1
		Falling edge	T5MOD<CAP34M1:0> = 1, 0
INT7	P46	Rising edge	− − −

(3) Micro DMA start vector

When the CPU reads the interrupt vector after accepting an interrupt, it simultaneously compares the interrupt vector (Bits 2 to 6 of the interrupt vector) with each channel's Micro DMA start vector. When the two match, the interrupt from the channel whose value matched is processed in Micro DMA mode.

If the interrupt vector matches more than one channel, the channel with the lower channel number has a higher priority.

Micro DMA0 Start Vector

DMA0V (007CH) (Note)

	7	6	5	4	3	2	1	0
Bit symbol				DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
Read/Write				W				
After reset				0	0	0	0	0
Function	Micro DMA channel 0 processed by matching bits 2 to 6 of the interrupt vector.							

Micro DMA1 Start Vector

DMA1V (007DH) (Note)

	7	6	5	4	3	2	1	0
Bit symbol				DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
Read/Write				W				
After reset				0	0	0	0	0
Function	Micro DMA channel 1 processed by matching bits 2 to 6 of the interrupt vector.							

Micro DMA2 Start Vector

DMA2V (007EH) (Note)

	7	6	5	4	3	2	1	0
Bit symbol				DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
Read/Write				W				
After reset				0	0	0	0	0
Function	Micro DMA channel 2 processed by matching bits 2 to 6 of the interrupt vector.							

Micro DMA3 Start Vector

DMA3V (007FH) (Note)

	7	6	5	4	3	2	1	0
Bit symbol				DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
Read/Write				W				
After reset				0	0	0	0	0
Function	Micro DMA channel 3 processed by matching bits 2 to 6 of the interrupt vector.							

Note: Read-modify-write instruction is prohibited.

Figure 3.4.7 Micro DMA Start Vector Register

(4) Notes

The instruction execution unit and the bus interface unit of this CPU operate independently of each other. Therefore, if the instruction used to clear an interrupt request flag of an interrupt is fetched before the interrupt is generated, it is possible that the CPU might execute the fetched instruction to clear the interrupt request flag while reading the interrupt vector after accepting the interrupt.

To avoid the above occurring, clear the interrupt request flag by entering the instruction to clear the flag after the DI instruction. In the case of setting an interrupt enable again by EI instruction after the execution of clearing instruction, execute EI instruction after clearing instruction and following more than one instruction are executed. When EI instruction is placed immediately after clearing instruction, an interrupt becomes enable before interrupt request flags are cleared.

In the case of changing the value of the interrupt mask register<IFF2:0> by execution of POP SR instruction, disable an interrupt by DI instruction before execution of POP SR instruction.

3.5 Functions of Ports

The TMP93CS32 has 49 bits for I/O ports.

These port pins have I/O functions for the built-in CPU and internal I/Os as well as general-purpose I/O port functions. Table 3.5.1 lists the function of each port pin. Table 3.5.2 lists I/O registers and specification.

Resetting makes the port pins listed below function as general-purpose I/O ports.

I/O pins programmable for input or output are set to input ports.

To set port pins for built-in functions, a program is required.

Table 3.5.1 Functions of Ports

(PU = with programmable pull-up resistor)

Port Name	Pin Name	Pin No.	Direction	R	Direction Setting Unit	Pin Name for Built-in Function
Port 0	P00 to P07	8	I/O	–	Bit	AD0 to AD7
Port 1	P10 to P17	8	I/O	–	Bit	AD8 to AD15/A8 to A15
Port 2	P20 to P27	8	I/O	PU	Bit	A0 to A7/A16 to A23
Port 3	P30	1	Output	–	(Fixed)	$\overline{\text{RD}}$
	P31	1	Output	–	(Fixed)	$\overline{\text{WR}}$
	P32	1	I/O	PU	Bit	$\overline{\text{HWR}}$
	P35	1	I/O	–	Bit	INT0
Port 4	P41	1	I/O	–	Bit	TO3
	P42	1	I/O	–	Bit	TI4/INT4
	P43	1	I/O	–	Bit	TI5/INT5
	P44	1	I/O	–	Bit	TO4
	P45	1	I/O	–	Bit	TI6/INT6
	P46	1	I/O	–	Bit	TI7/INT7
	P47	1	I/O	–	Bit	TO6
Port 5	P50 to P52	3	Input	–	(Fixed)	AN0 to AN2,
	P53	1	Input	–	(Fixed)	AN3/ $\overline{\text{ADTRG}}$
	P54, P55	2	Input	–	(Fixed)	AN4, AN5
Port 6	P60	1	I/O	PU	Bit	TXD0
	P61	1	I/O	PU	Bit	RXD0
	P62	1	I/O	PU	Bit	SCLK0/ $\overline{\text{CTS0}}$
	P63	1	I/O	PU	Bit	TXD1
	P64	1	I/O	PU	Bit	RXD1
	P65	1	I/O	PU	Bit	SCLK1/ $\overline{\text{CTS1}}$
Port 7	P70	1	I/O	–	Bit	$\overline{\text{WAIT}}$ /(High current output)
	P71	1	I/O	–	Bit	(High current output)

Table 3.5.2 I/O Registers and Specification (1/2)

Port	Name	Specification	I/O register		
			Pn	PnCR	PnFC
Port 0	P00 to P07	Input port	X	0	None
		Output port	X	1	
		AD (0 to 7) bus	X	X	
Port 1	P10 to P17	Input port	X	0	0
		Output port	X	1	0
		AD (8 to 15) bus	X	0	1
		A (8 to 15) output	X	1	1
Port 2	P20 to P27	Input port (without pull up)	0	0	0
		Input port (with pull up)	1	0	0
		Output port	X	1	0
		A (0 to 7) Output	1	0	1
		A (16 to 23) output	1	1	1
Port 3	P30	Output port	X	None	0
		Outputs $\overline{\text{RD}}$ only when accessing external space	1		1
		Always outputs $\overline{\text{RD}}$	0		1
	P31	Output port	X	None	0
		Outputs $\overline{\text{WR}}$ only when accessing external space	X		1
	P32	Input port (without pull up)	0	0	0
		Input port (with pull up)	1	0	0
		Output port	X	1	0
		$\overline{\text{HWR}}$ output	X	1	1
	P35	Input port/INT0 input (Note 1)	X	0	None
		Output port	X	1	
Port 4	P41	Input port	X	0	0
		Output port	X	1	0
		TO3 output	X	1	1
	P42	Input port/TI4/INT4 input	X	0	None
		Output port	X	1	
	P43	Input port/TI5/INT5 input	X	0	
		Output port	X	1	
	P44	Input port	X	0	0
		Output port	X	1	0
		TO4 output	X	1	1
	P45	Input port/TI6/INT6 input	X	0	None
		Output port	X	1	
	P46	Input port/TI7/INT7 input	X	0	
		Output port	X	1	
	P47	Input port	X	0	0
		Output port	X	1	0
		TO6 output	X	1	1
Port 5	P50 to P57	Input port	X	None	
		AN0 to AN5 input (Note 2)	X		

X: Don't care

Note 1: Using P35 pin as INT0, IIMC register has to be set enable interrupt.

Note 2: Using P50 to P55 pins as input channels for the AD converter, the channels are selected by ADMOD1<ADCH2:0>.

Table 3.5.3 I/O Registers and Specification (2/2)

Port	Name	Specification	I/O Register		
			Pn	PnCR	PnFC
Port 6	P60	Input port (without pull up)	0	0	0
		Input port (with pull up)	1	0	0
		Output port	X	1	0
		TXD0 output	X	1	0
	P61	Input port/RXD0 input (without pull up)	0	0	None
		Input port/RXD0 input (with pull up)	1	0	
		Output port	X	1	
	P62	Input port/SCLK0/ $\overline{\text{CTS0}}$ input (without pull up)	0	0	0
		Input port/SCLK0/ $\overline{\text{CTS0}}$ input (with pull up)	1	0	0
		Output port	X	1	0
		SCLK0 output	X	1	1
	P63	Input port (without pull up)	0	0	0
		Input port (with pull up)	1	0	0
		Output port	X	1	0
		TXD1 output	X	1	1
	P64	Input port/RXD1 (without pull up)	0	0	None
		Input port/RXD1 (with pull up)	1	0	
		Output port	X	1	
	P65	Input port/SCLK1/ $\overline{\text{CTS1}}$ input (without pull up)	0	0	0
		Input port/SCLK1/ $\overline{\text{CTS1}}$ input (with pull up)	1	0	0
		Output port	X	1	0
		SCLK1 output	X	1	1
Port 7	P70	Input port/ $\overline{\text{WAIT}}$ input	X	0	None
		Output port	X	1	
	P71	Input port	X	0	
		Output port	X	1	

X: Don’t care

3.5.1 Port 0 (P00 to P07)

Port 0 is an 8-bit general-purpose I/O port. I/O can be set on a bit basis using the control register P0CR. Resetting clears all bits of P0CR to 0 and sets Port 0 to input mode. Figure 3.5.1 shows the registers for Port 0.

In addition to functioning as a general-purpose I/O port, Port 0 also functions as an address data bus (AD0 to AD7). To access external memory, Port 0 functions as an address data bus (AD0 to AD7) and all bits of the control register P0CR are cleared to 0.

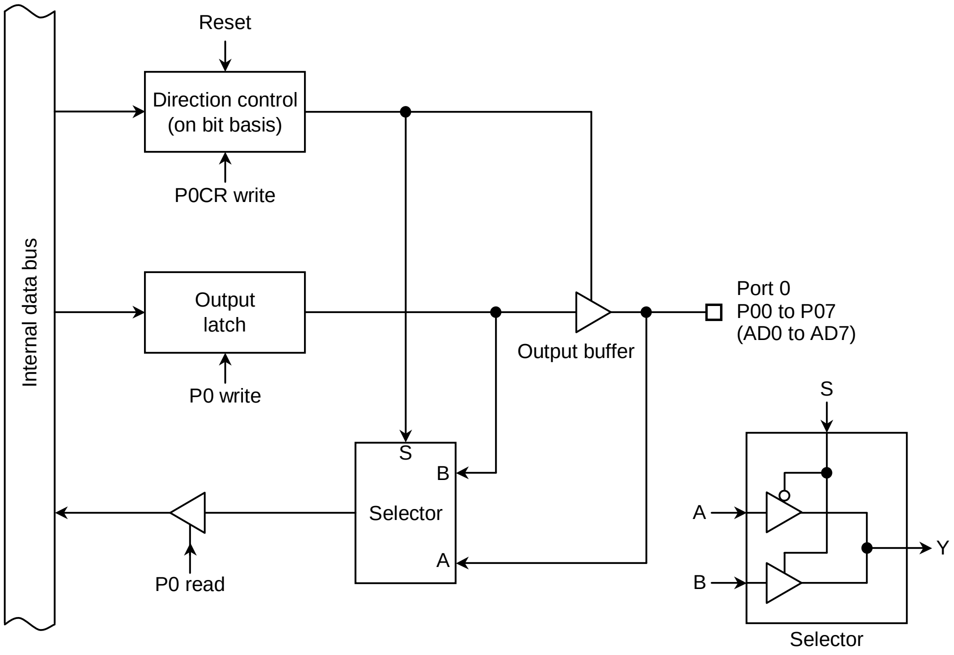


Figure 3.5.1 Port 0

3.5.2 Port 1 (P10 to P17)

Port 1 is an 8-bit general-purpose I/O port. I/O can be set on a bit basis using control register P1CR and function register P1FC. Resetting sets all bits of output latch P1, control register P1CR, and function register P1FC to 0 and sets Port 1 to input mode.

Figure 3.5.3 shows the registers for Port 1.

In addition to functioning as a general-purpose I/O port, Port 1 also shares functions as an address data bus (AD8 to AD15) or an address bus (A8 to A15).

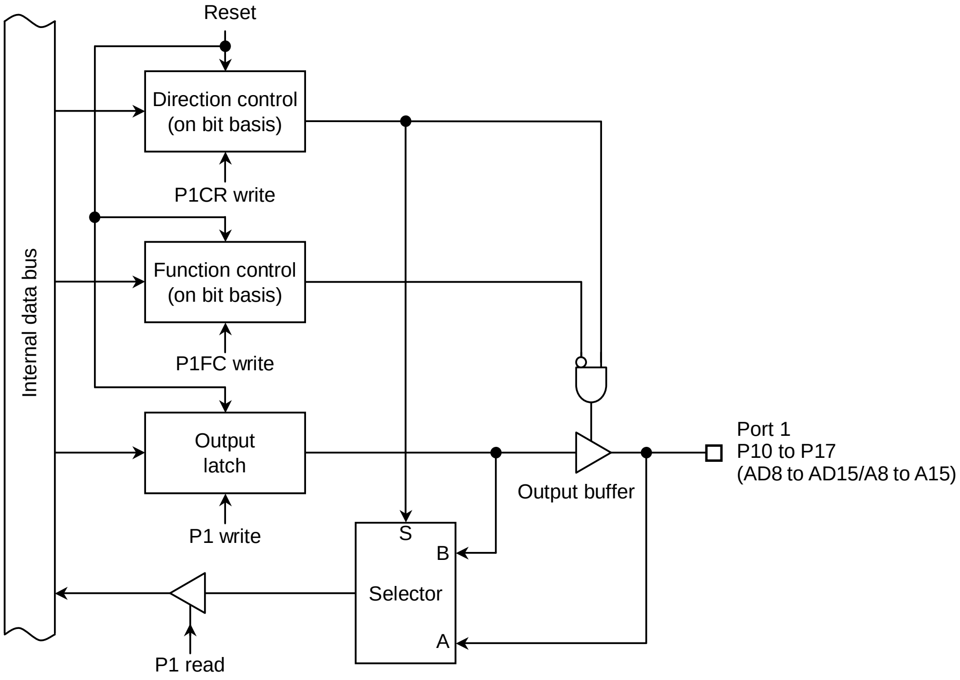


Figure 3.5.2 Port 1

Port 0 Register

P0
(0000H)

	7	6	5	4	3	2	1	0
Bit symbol	P07	P06	P05	P04	P03	P02	P01	P00
Read/Write	R/W							
After reset	Input mode (Output latch register becomes undefined.)							

Port 0 Control Register

P0CR
(0002H)

	7	6	5	4	3	2	1	0
Bit symbol	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0:Input 1:Output (At external access, Port 0 becomes AD7 to AD0 and P0CR is cleared to 0.)							

Port 0 I/O setting

0	Input
1	Output

Note: Read-modify-write instruction is prohibited for P0CR.

Port 1 Register

P1
(0001H)

	7	6	5	4	3	2	1	0
Bit symbol	P17	P16	P15	P14	P13	P12	P11	P10
Read/Write	R/W							
After reset	Input mode (Output latch register is cleared to “0”.)							

Port 1 Control Register

P1CR
(0004H)

	7	6	5	4	3	2	1	0
Bit symbol	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	<<See P1FC below.>>							

Note: Read-modify-write instruction is prohibited for P1CR.

Port 1 Function Register

P1FC
(0005H)

	7	6	5	4	3	2	1	0
Bit symbol	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	P1FC/P1CR = 00: Input, 01: Output, 10: AD15 to AD8, 11: A15 to A8							

Port 1 function setting

P1FC<P1XF> P1CR<P1XC>	0	1
0	Input port	Address data bus (AD15 to AD8)
1	Output port	Address bus (A15 to A8)

Note 1: Read-modify-write instruction is prohibited for P1FC.

Note 2: <P1XF> is bit X in register P1FC; <P1XC>, in register P1CR.

Figure 3.5.3 Registers for Ports 0 and 1

3.5.3 Port 2 (P20 to P27)

Port 2 is an 8-bit general-purpose I/O port. I/O can be set on bit basis using the control register P2CR and function register P2FC. All bits of the output latch P2 is set to "1" by reset, and all bits of P2CR and P2FC are cleared to "0". Port 2 becomes the input mode with the pull-up resistor.

In addition to functioning as a general-purpose I/O port, Port 2 also shares functions as an address bus (A0 to A7) and an address bus (A16 to A23). To use Port 2 as address bus (A0 to A7 or A16 to A23), write "0" to P2 output latch and turn off the programmable pull-up resistor.

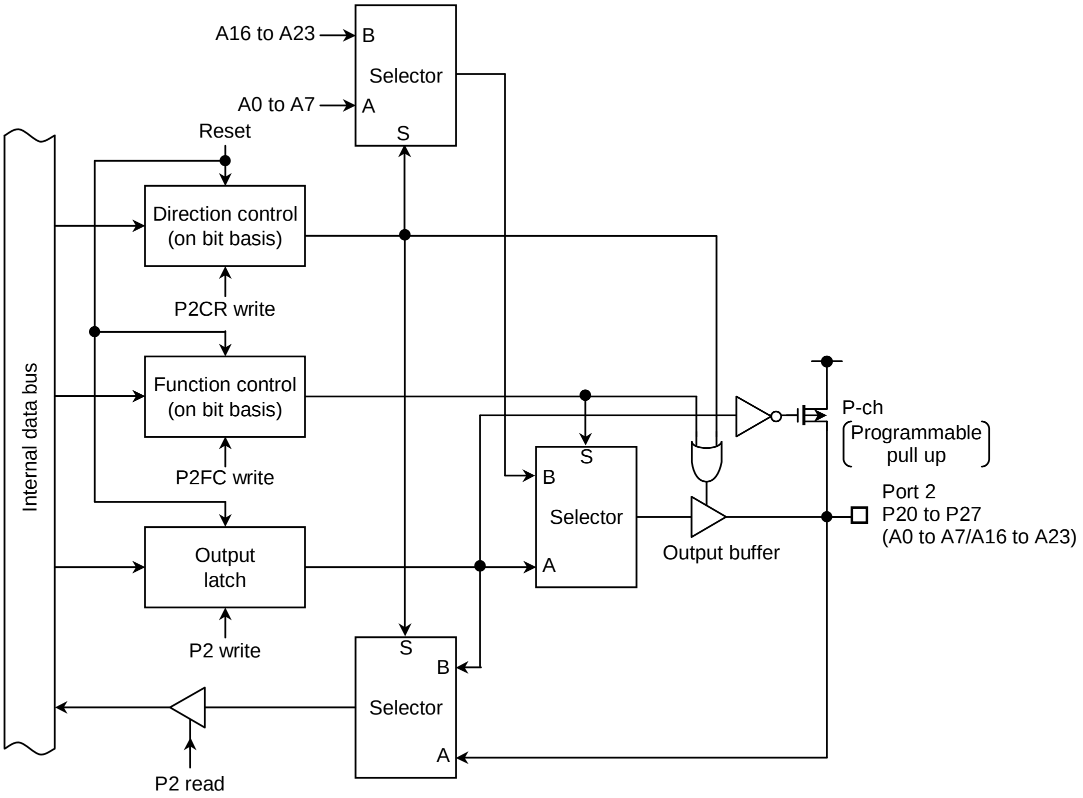


Figure 3.5.4 Port 2

Port 2 Register

P2
(0006H)

	7	6	5	4	3	2	1	0
Bit symbol	P27	P26	P25	P24	P23	P22	P21	P20
Read/Write	R/W							
After reset	Input mode (Output latch register is set to “1”.)							

Note: When port P2 is used in the input mode, P2 register controls the built-in pull-up resistor. Read-modify-write is prohibited in the input mode or the I/O mode.
Setting the built-in pull-up resistor may be depended on the states of the input pin.

Port 2 Control Register

P2CR
(0008H)

	7	6	5	4	3	2	1	0
Bit symbol	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	<<See P2FC below.>>							

Note: Read-modify-write instruction is prohibited for P2CR.

Port 2 Function Register

P2FC
(0009H)

	7	6	5	4	3	2	1	0
Bit symbol	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	P2FC/P2CR = 00: Input, 01: Output, 10: A7 to A0, 11: A23 to A16							

Port 2 function setting

P2FC<P2XF> P2CR<P2XC>	0	1
0	Input port	Address bus (A7 to A0)
1	Output port	Address bus (A23 to A16)

Note 1: Read-modify-write instruction is prohibited for P2FC.

Note 2: <P2XF> is bit X in register P2FC; <P2XC>; in register P2CR.
To set as an address bus A23 to A16, set P2FC after setting P2CR.

Figure 3.5.5 Registers for Port 2

3.5.4 Port 3 (P30 to P32, P35)

Port 3 is an 4-bit general-purpose I/O port. I/O can be set on a bit basis, but note that P30 and P31 are used for output only.

I/O is set using control register P3CR and function register P3FC. Resetting resets all bits of output latch P3, control register P3CR (bits 0 and 1 are unused), and function register P3FC to 0. Resetting also outputs 1 from P30 and P31.

In addition to functioning as a general-purpose I/O port, Port 3 also shares functions as an I/O for the CPU's control/status signal.

With the TMP93CS32, when P30 pin is defined as $\overline{\text{RD}}$ signal output mode (<P30F> = 1), clearing the output latch register <P30> to 0 outputs the $\overline{\text{RD}}$ strobe (used for the pseudo static RAM) from the P30 pin even when the internal address area is accessed. If the output latch register <P30> remains 1, the $\overline{\text{RD}}$ strobe signal is output only when the external address area is accessed.

(1) P30 ($\overline{\text{RD}}$) and P31 ($\overline{\text{WR}}$)

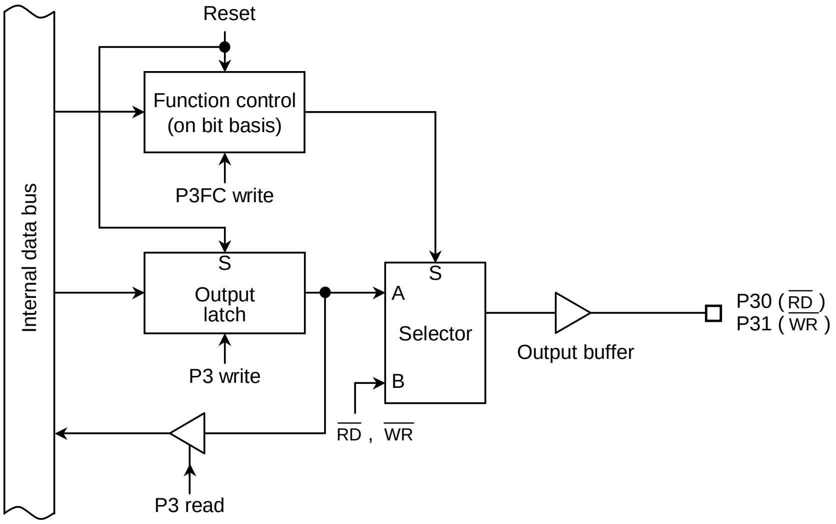


Figure 3.5.6 Port 3 (P30 and P31)

(2) P32 ($\overline{\text{HWR}}$)

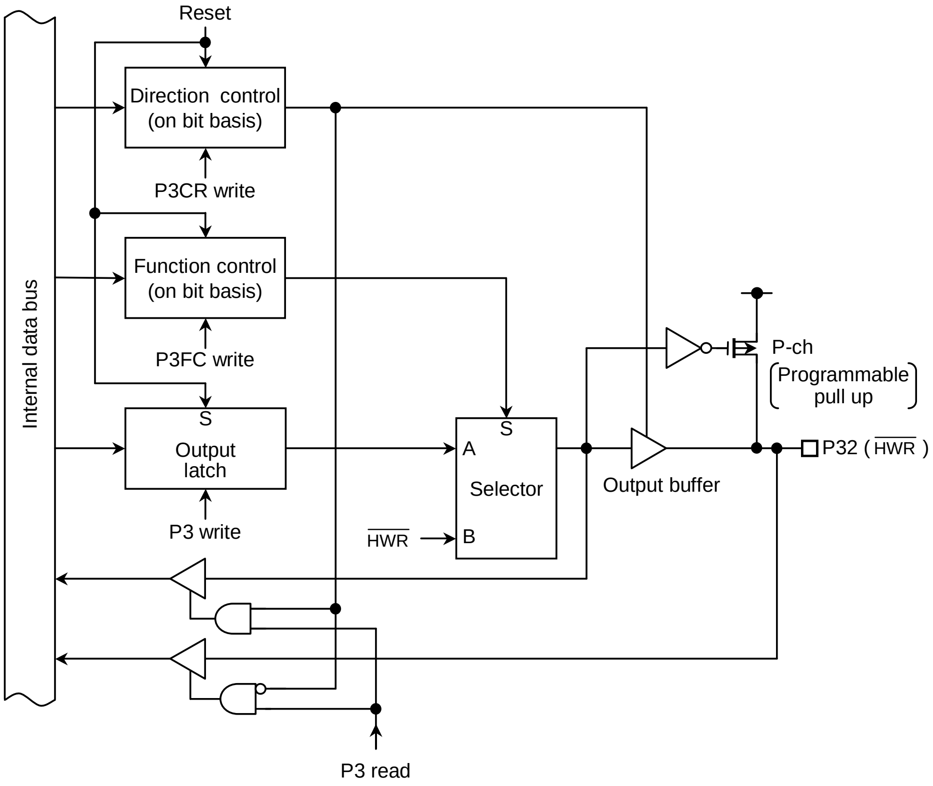


Figure 3.5.7 Port 3 (P32)

(3) P35 (INT0)

Port 35 is a general-purpose I/O port, and also used as an INT0 pin for external interrupt request input.

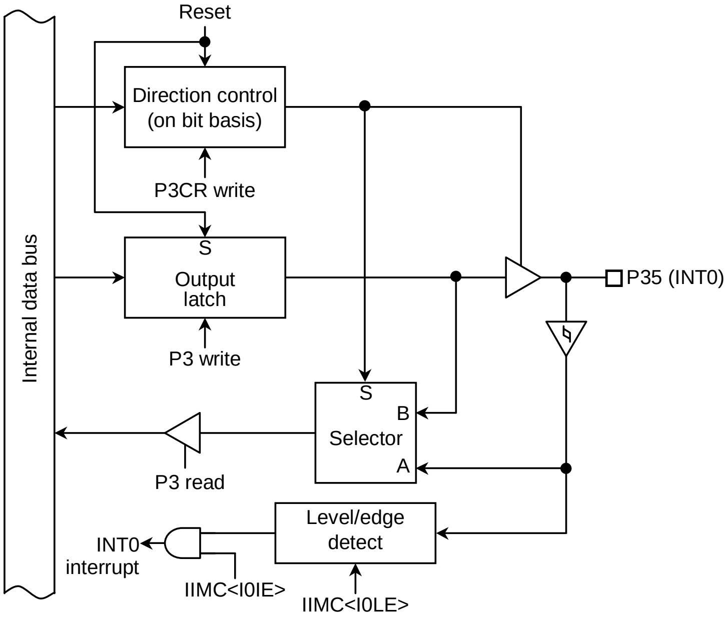


Figure 3.5.8 Port 35

Port 3 Register

		7	6	5	4	3	2	1	0
P3 (0007H)	Bit symbol			P35	–	–	P32	P31	P30
	Read/Write			R/W					
	After reset			1	–	–	1	1	1
	Function			Input mode	(Note) Always write "1" (This bit is read as "1").		Input mode	Output mode	

Note: When port 32 is used in the input mode, P3 register controls the built-in pull-up resistor. Read-modify-write is prohibited in the input mode or the I/O mode.
Setting the built-in pull-up resistor may be depended on the states of the input pin.

Port 3 Control Register

		7	6	5	4	3	2	1	0
P3CR (000AH)	Bit symbol			P35C	–	–	P32C		
	Read/Write			W					
	After reset			0	–	–	0		
	Function			0: Input 1: Output	(Note) Always write "1" (This bit is read as "1").		0: Input 1: Output		

I/O setting

0	Input
1	Output

Note: Read-modify-write instruction is prohibited for P3CR.

Port 3 Function Register

		7	6	5	4	3	2	1	0
P3FC (000BH)	Bit symbol		–		–	–	P32F	P31F	P30F
	Read/Write		W		W				
	After reset		–		–	–	0	0	0
	Function		(Note) Always write "0" (This bit is read as "1").		(Note) Always write "0" (This bit is read as "1").		0: Port 1: $\overline{\text{HWR}}$	0: Port 1: $\overline{\text{WR}}$	0: Port 1: $\overline{\text{RD}}$

P32 ($\overline{\text{HWR}}$) function setting

<P32> <P32F>	0	1
0	"0" output	"1" output
1	$\overline{\text{HWR}}$ output only for external access	

P30 ($\overline{\text{RD}}$) function setting

<P30> <P30F>	0	1
0	"0" output	"1" output
1	Always $\overline{\text{RD}}$ output (for pseudo SRAM)	$\overline{\text{RD}}$ output only for external access

P31 ($\overline{\text{WR}}$) function setting

<P31> <P31F>	0	1
0	"0" output	"1" output
1	$\overline{\text{WR}}$ output only for external access	

Note: Read-modify-write instruction is prohibited for P3FC.

Figure 3.5.9 Registers for Port 3

3.5.5 Port 4 (P41 to P47)

Port 4 is a 7-bit general-purpose I/O port. I/O can be set on bit basis. Resetting sets Port 4 to the input port. In addition to functioning as a general-purpose I/O port, Port 4 also shares functions as 16-bit timer 4 and 5 clocks, an output for 8-bit timer F/F3, 16-bit timer F/F4 and 5. Writing 1 in the corresponding bit of the Port 4 function register (P4FC) enables output of the timer.

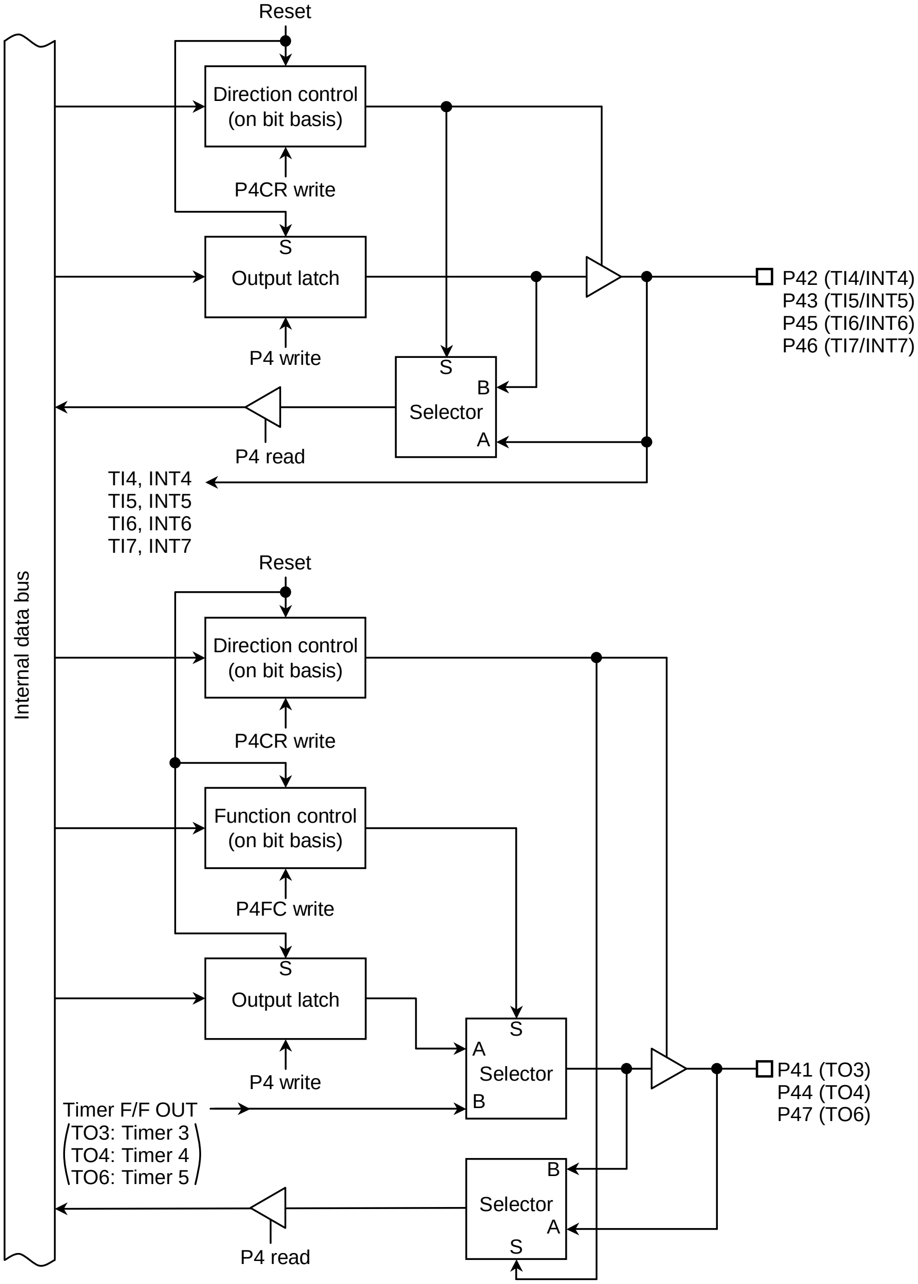


Figure 3.5.10 Port 4 (P41 to P47)

Port 4 Register

P4
(000CH)

	7	6	5	4	3	2	1	0
Bit symbol	P47	P46	P45	P44	P43	P42	P41	–
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	–
Function	Input mode							(Note) Always write “1” (This bit is read as “1”).

Port 4 Control Register

P4CR
(000EH)

	7	6	5	4	3	2	1	0
Bit symbol	P47C	P46C	P45C	P44C	P43C	P42C	P41C	–
Read/Write	W							
After reset	0	0	0	0	0	0	0	–
Function	0: Input 1: Output							(Note) Always write “1” (This bit is read as “1”).

Port 4 I/O setting

0	Input
1	Output

Note: Read-modify-write instruction is prohibited for P4CR.

Port 4 Function Register

P4FC
(0010H)

	7	6	5	4	3	2	1	0
Bit symbol	P47F			P44F			P41F	
Read/Write	W			W			W	
After reset	0			0			0	
Function	0: Port 1: TO6			0: Port 1: TO4			0: Port 1: TO3	

Setting P47 as TO6

P4FC<P47F>	1
P4CR<P47C>	1

Setting P44 as TO4

P4FC<P44F>	1
P4CR<P44C>	1

Setting P41 as TO3

P4FC<P41F>	1
P4CR<P41C>	1

Note 1: Read-modify-write instruction is prohibited for P4FC.

Note 2: P42/TI4, P43/TI5, P45/TI6, P46/TI7 pin does not have a register changing port/function.
For example, when it is used as an input port, the input signal for port is inputted to 8-/16-bit timer as a timer input.

Figure 3.5.11 Register for Port 4

3.5.6 Port 5 (P50 to P55)

Port 5 is an 6-bit input port, also used as an analog input pin for the internal AD Converter. Additionally, P53 is also used as an analog conversion external trigger input pin ($\overline{\text{ADTRG}}$).

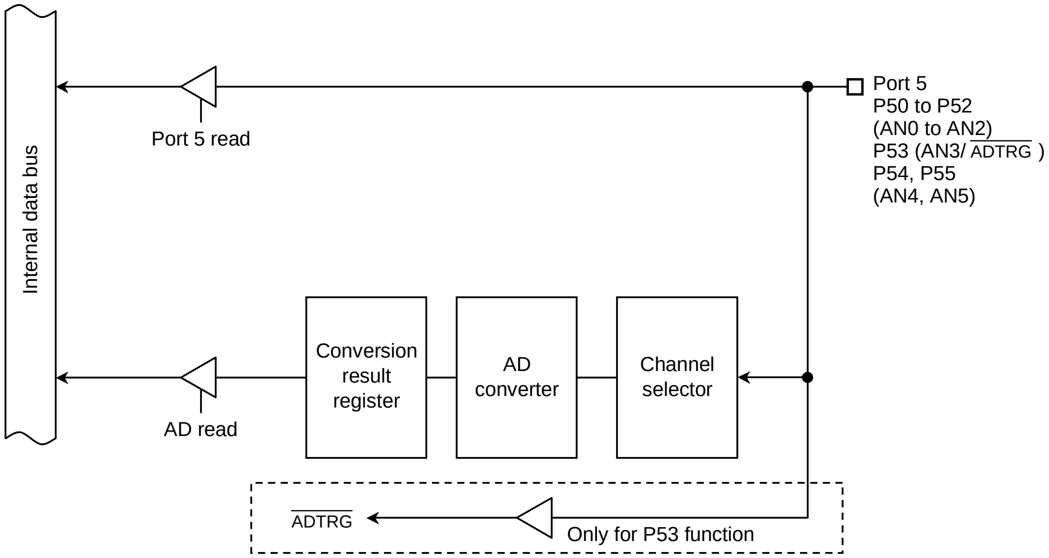


Figure 3.5.12 Port 5

Port 5 Register

P5 (000DH)		7	6	5	4	3	2	1	0
	Bit symbol			P55	P54	P53	P52	P51	P50
	Read/Write			R					
	After reset			Undefined					
	Function			Input mode					

Note: The input channel selection of AD Converter is set by AD Converter mode register ADMOD1.

Figure 3.5.13 Registers for Port 5

3.5.7 Port 6 (P60 to P65)

Port 60 to 65 is a 6-bit general-purpose I/O port. I/Os can be set on a bit basis.

Resetting sets P60 to 65 to an input port and connects a pull-up resistor.

It also sets all bits of the output latch register to 1.

In addition to functioning as a general-purpose I/O port, P60 to P65 can also share function as an I/O for serial channels 0 and 1. Writing "1" in the corresponding bit of the Port 6 function register (P6FC) enables this function.

Resetting sets the function register value to "0" and sets all bits to input ports.

(1) Ports 60 (TXD0) and 63 (TXD1)

Ports 60 and 63 also function as serial channel TXD output pins in addition to I/O ports.

They have a programmable open-drain function.

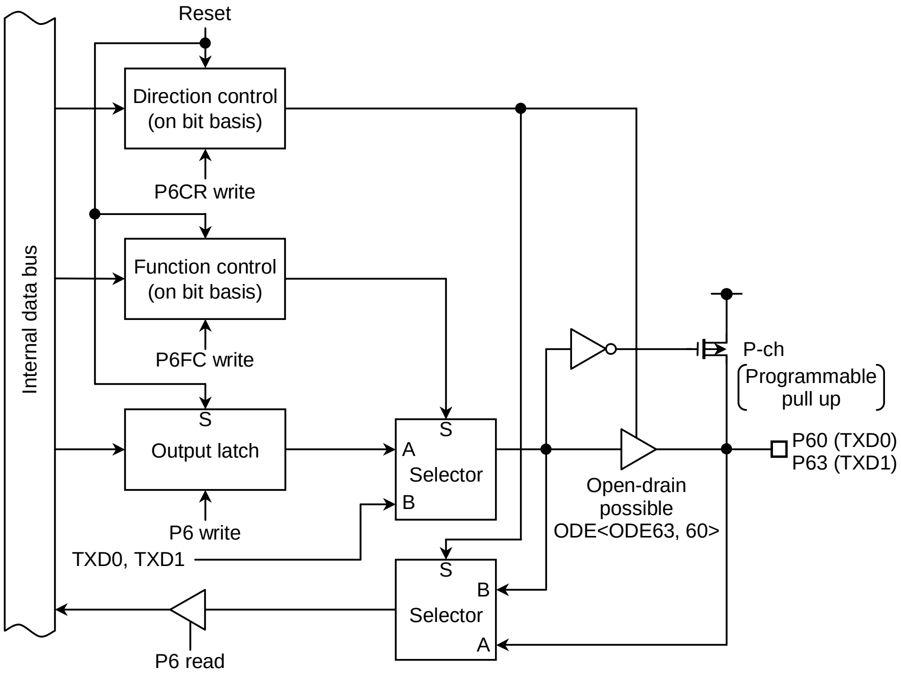


Figure 3.5.14 Ports 60 and 63

(2) Ports 61 (RXD0) and 64 (RXD1)

Ports 61 and 64 are I/O ports, and also used as RXD input pins for serial channels.

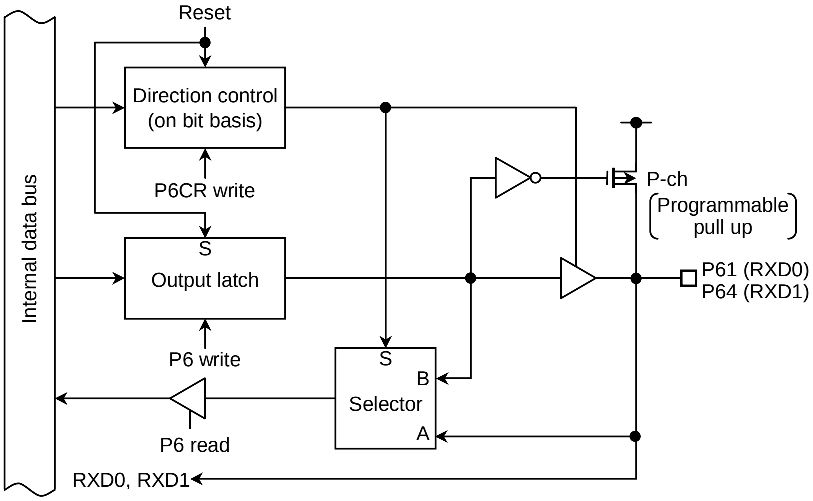


Figure 3.5.15 Ports 61 and 64

(3) Ports 62 ($\overline{\text{CTS0}}$/SCLK0) and 65 ($\overline{\text{CTS1}}$/SCLK1)

Ports 62 and 65 are an I/O port, and also used as a $\overline{\text{CTS}}$ input pin and as a SCLK I/O pin for serial channels.

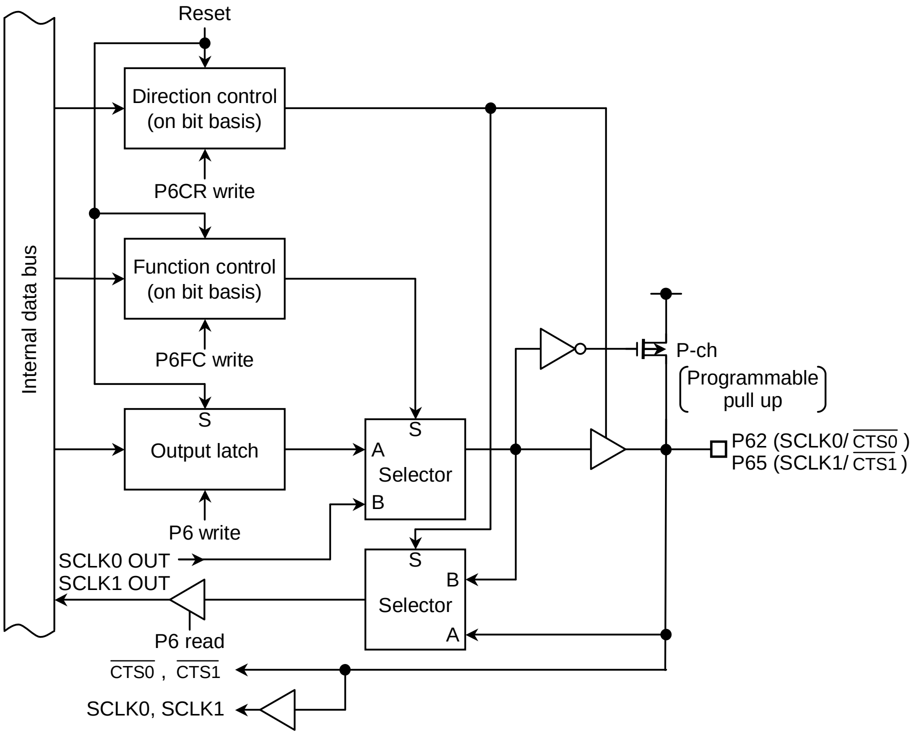


Figure 3.5.16 Ports 62 and 65

Port 6 Register

P6
(0012H)

	7	6	5	4	3	2	1	0
Bit symbol	–	–	P65	P64	P63	P62	P61	P60
Read/Write	R/W							
After reset	–	–	1	1	1	1	1	1
Function	(Note) Always write "1" (This bit is read as "1").		Input mode					

Note: When port P6 is used in the input mode, P6 register controls the built-in pull-up resistor. Read-modify-write is prohibited in the input mode or the I/O mode. Setting the built-in pull-up resistor may be depended on the states of the input pin.

Port 6 Control Register

P6CR
(0014H)

	7	6	5	4	3	2	1	0
Bit symbol	–	–	P65C	P64C	P63C	P62C	P61C	P60C
Read/Write	W							
After reset	–	–	0	0	0	0	0	0
Function	(Note) Always write "1" (This bit is read as "1").		0: Input 1: Output					

Port 6 I/O setting

0	Input
1	Output

Note: Read-modify-write instruction is prohibited for P6CR.

Port 6 Function Register

P6FC
(0016H)

	7	6	5	4	3	2	1	0
Bit symbol			P65F		P63F	P62F		P60F
Read/Write			W		W			W
After reset			0		0	0		0
Function			0: Port 1: SCLK1		0: Port 1: TXD1	0: Port 1: SCLK0		0: Port 1: TXD0

P63 TXD1 output setting (Note 2)

P6FC<P63F>	1
P6CR<P63C>	1

P60 TXD0 output setting (Note 2)

P6FC<P60F>	1
P6CR<P60C>	1

P65 SCLK1 output setting

P6FC<P65F>	1
P6CR<P65C>	1

P62 SCLK0 output setting

P6FC<P62F>	1
P6CR<P62C>	1

Note 1: Read-modify-write instruction is prohibited for P6FC.

Note 2: To set the TXD pin to open drain, write "1" in bit0 (for TXD0 pin) or bit1 (for TXD1 pin) of the ODE register.
P61/RXD0, P64/RXD1 pins do not have a register changing port/function.
When using as input ports, the serial receive data is input to SIO.

Figure 3.5.17 Register for Port 6

3.5.8 Port 7 (P70 and P71)

Port 7 is an 2-bit general-purpose I/O port. I/O can be set on a bit basis. Port 7 can output large current and drive LED directly. In addition to I/O port, Port 70 also shares functions as $\overline{\text{WAIT}}$ input pin. Resetting sets the function register P7CR to 0, and all bits to input ports.

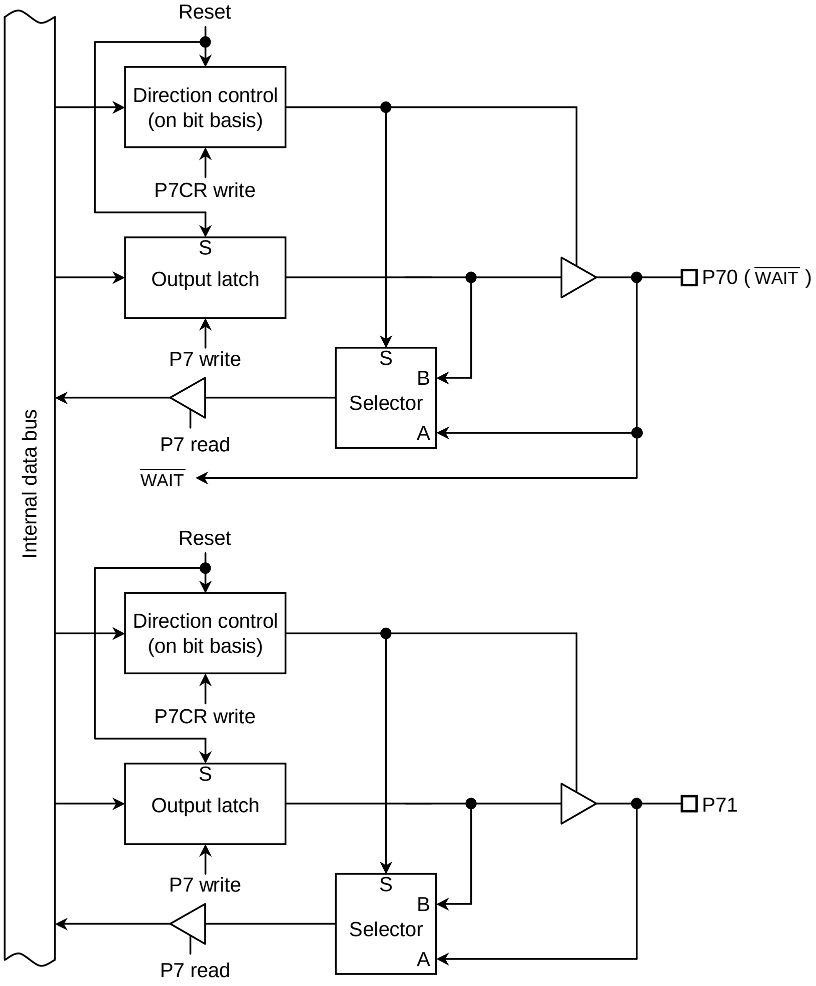


Figure 3.5.18 Port 71 to 77

Port 7 Register

P7
(0013H)

	7	6	5	4	3	2	1	0
Bit symbol	–	–	–	–	–	–	P71	P70
Read/Write	R/W							
After reset	–	–	–	–	–	–	1	1
Function	(Note) Always write "1" (This bit is read as "1").						Input mode	

Port 7 Control Register

P7CR
(0015H)

	7	6	5	4	3	2	1	0
Bit symbol	–	–	–	–	–	–	P71C	P70C
Read/Write	W							
After reset	–	–	–	–	–	–	0	0
Function	(Note) Always write "1" (This bit is read as "1").						0: Input 1: Output	

Port 7 I/O setting

0	Input
1	Output

Note 1: Read-modify-write instruction is prohibited for P7CR.

Note 2: P70/ $\overline{\text{WAIT}}$ pin does not have a register changing port/function.
For example, when it is used as an input port, the input signal is inputted as $\overline{\text{WAIT}}$ input.
When it is used as $\overline{\text{WAIT}}$ input pin, bit <BmWn> of bus width wait control register must be specified.

Figure 3.5.19 Registers for Port 7

3.6 Bus Width/Wait Controller, AM8/$\overline{\text{AM16}}$ Pin

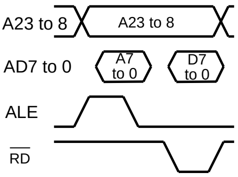
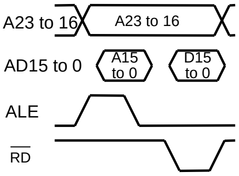
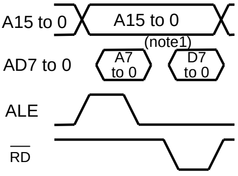
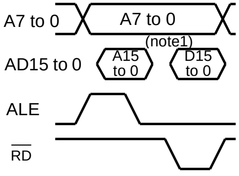
TMP93CS32 has a built-in controller used to control wait ($\overline{\text{WAIT}}$ pin) and data bus size (8 or 16 bits) for any of the three block address areas.

3.6.1 AM8/$\overline{\text{AM16}}$ Pin

Set this pin to "H". After reset, the CPU accesses the internal ROM with 16-bit bus width. The bus width when the CPU accesses an external area is set by the bus width/wait cntrol registers (Described at 3.6.3) and the registers of Port 1. (The value "1" of this pin is ignored and the value set by register is active.)

3.6.2 Address/Data Bus Pins

Port 0/AD0 to AD7, Port 1/AD8 to AD15/A8 to A15 and Port 2/A16 to A23/A0 to A7 function as address/data bus for connecting the external memories.

		1	2	3	4
Number of address bus pins		Max 24 (to 16 Mbytes)	Max 24 (to 16 Mbytes)	Max 16 (to 64 Kbytes)	Max 8 (to 256 bytes)
Number of data bus pins		8	16	8	16
Number of multiplexed pins		8	16	0	0
Mode pins	$\overline{\text{EA}}$	V_{IH}			
	AM8/$\overline{\text{AM16}}$	V_{IH}			
Port function	Port 0	AD0 to AD7	AD0 to AD7	AD0 to AD7	AD0 to AD7
	Port 1	A8 to A15	AD8 to AD15	A8 to A15	AD8 to AD15
	Port 2	A16 to A23	A16 to A23	A0 to A7	A0 to A7
Timing chart					

Note 1: In case of 3 and 4, the data bus signals output the addresses since the signals are also used as the address bus. Writing "0" to bit CKOCR<ALEEN>, ALE signal can be stopped outputting.

Note 2: After reset operation, Port 0, Port 1, and Port 2 function as Input ports, not as address, data bus signals.

3.6.3 Bus Width/Wait Control Registers

Figure 3.6.1 shows control registers.

One block address areas are controlled by 1-byte bus width/wait control registers (WAITC0, WAITC1, WAITC2).

(1) Data bus size select

Bit4 (<B0BUS>, <B1BUS>, <B2BUS>) of the control register is used to specify data bus size. Setting this bit to 0 accesses the memory in 16-bit data bus mode; setting it to 1 accesses the memory in 8-bit data bus mode.

Changing data bus size depending on the access address is called dynamic bus sizing. Table 3.6.1 shows the details of the bus operation.

(2) Wait control

Control register bits 3 and 2 (<B0W1:0>, <B1W1:0>, <B2W1:0>) are used to specify the number of waits. These bits execute the following operation by setting.

“00” A2-state wait is inserted regardless of the $\overline{\text{WAIT}}$ pin status.

“01” A1-state wait is inserted regardless of the $\overline{\text{WAIT}}$ pin status.

“10” A1-state wait is inserted and the $\overline{\text{WAIT}}$ pin status is sampled. If the pin is low, inserting the wait maintains the bus cycle until the pin goes high.

“11” The bus cycle is completed without a wait (0 waits) regardless of the $\overline{\text{WAIT}}$ pin status.

After reset operation, clear to “00” (A2-state wait mode).

(3) Address area specification

Control register bits 1 and 0 (<B0C1:0>, <B1C1:0>, <B2C1:0>) are used to specify the target address area. Setting these bits to 00 enables settings (Wait state, Bus size, etc.) as follows:

- WAITC0 setting enabled when 7F00H to 7FFFH is accessed.
- WAITC1 setting enabled when 880H to 7FFFH is accessed.
- WAITC2 setting enabled when 8000H to 3FFFFFH is accessed.

Setting bits to 01 enables setting for each block when 400000H to 7FFFFFH is accessed. Setting bits to 10 enables them when 800000H to BFFFFFH is accessed. Setting bits to 11 enables them when C00000H to FFFFFFH is accessed.

		7	6	5	4	3	2	1	0
WAITC0 (0068H)	Bit symbol				B0BUS	B0W1	B0W0	B0C1	B0C0
	Read/Write				W				
	After reset				0	0	0	0	0
	Function				0: 16-bit bus 1: 8-bit bus	00: 2 waits 01: 1 wait 10: (1 + n) waits 11: 0 waits		00: 7F00H to 7FFFH 01: From 400000H 10: From 800000H 11: From C00000H	
WAITC1 (0069H)	Bit symbol				B1BUS	B1W1	B1W0	B1C1	B1C0
	Read/Write				W				
	After reset				0	0	0	0	0
	Function				0: 16-bit bus 1: 8-bit bus	00: 2 waits 01: 1 wait 10: (1 + n) waits 11: 0 waits		00: 880H to 7FFFH 01: From 400000H 10: From 800000H 11: From C00000H	
WAITC2 (006AH)	Bit symbol				B2BUS	B2W1	B2W0	B2C1	B2C0
	Read/Write				W				
	After reset				0	0	0	1	1
	Function				0: 16-bit bus 1: 8-bit bus	00: 2 waits 01: 1 wait 10: (1 + n) waits 11: 0 waits		00: From 8000H 01: From 400000H 10: From 800000H 11: From C00000H	

Note: Read-modify-write instruction is prohibited for WAITC0, WAITC1, and WAITC2.

Figure 3.6.1 Bus Width/Wait Control Registers

Table 3.6.1 Dynamic Bus Sizing

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

xxxxx: 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-active.

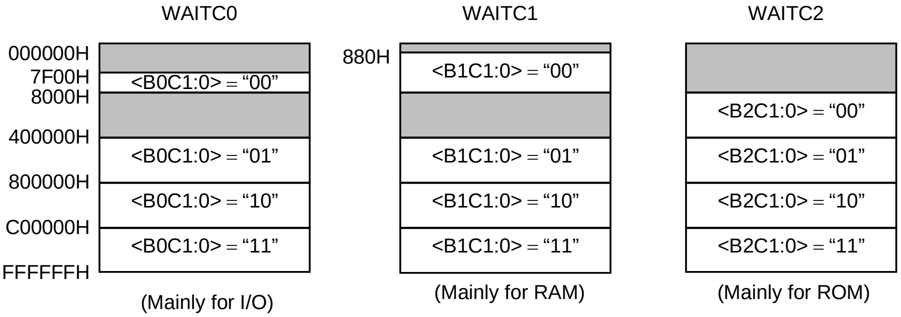
3.6.4 Bus Width/Wait Control

An image of the actual Bus-width/Wait control is shown below. Out of the whole memory area, address areas that can be specified are divided into four parts. Addresses from 000000H to 3FFFFFH are divided differently: 7F00H to 7FFFH is specified for WAITC0; 880H to 7FFFH, for WAITC1; and 8000H to 3FFFFFH, for WAITC2. The reason is that a device other than ROM (e.g., RAM or I/O) might be connected externally.

7F00 to 7FFFH (256 bytes) for WAITC0 are mapped mainly for possible expansions to external I/O.

880H to 7FFFH (Approx. 31 K bytes) for WAITC1 are mapped there mainly for possible extensions to external RAM.

8000H to 3FFFFFH (Approx. 4 M bytes) for WAITC2 are mapped mainly for possible extensions to external ROM. With the TMP93CS32 which has a built-in ROM, addresses from FF0000H to FFFFFFH are used as the internal ROM area; WAITC2 is disabled in this area. After reset, the CPU reads the program from the built-in ROM in 16-bit bus, 0-wait mode.



Note 1: Access priority is highest for built-in I/O, then built-in memory, and lowest for the bus width/wait controller.

Note 2: External areas other than WAITC0 to WAITC2 are accessed in 16-bit data bus (0 waits) mode.
When using the bus width/wait controller, do not specify the same address area more than once.
(However, when addresses 7F00H to 7FFFH for WAITC0 and 880H to 7FFFH for WAITC1 are specified, in other words, specifications overlap, only the WAITC0 setting is active.)

(Example of external memory connection)

Figure 3.6.2 is an example in which an external memory is connected to the TMP93CS32. In this example, 128-Kbyte ROM is connected using 16 bits bus, and 256-Kbyte RAM using 16-bit bus.

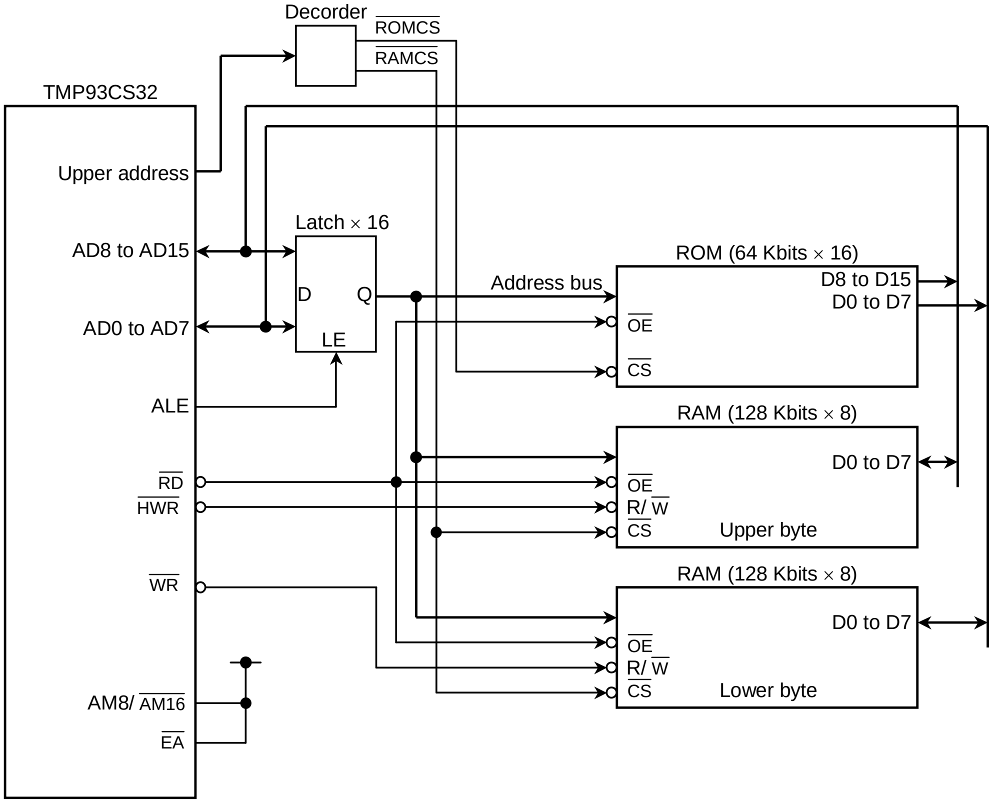


Figure 3.6.2 Example of External Memory Connection (ROM and RAM = 16 Bits)

The TMP93CS32 has built-in ROM and RAM. When ROM and RAM have insufficient capacity, it is possible to connect an external memory as the example of the external memory connection. In this example, the memory configuration is as follows.

Memory		Memory Size	Address	Data Bus
ROM	Internal	64 Kbytes	FF0000H to FFFFFFH	16 bits
	External	128 Kbytes	400000H to 41FFFFH	16 bits
SRAM	Internal	2 Kbytes	000080H to 00087FH	16 bits
	External	256 Kbytes	800000H to 83FFFFH	16 bits

3.7 8-Bit Timers

The TMP93CS32 contains four 8-bit timers (Timers 0, 1, 2, 3), each of which can be operated independently. The cascade connection allows these timers to be used as 16-bit timer. The following four operating modes are provided for the 8-bit timers.

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

} Variable combination (8 bits × 2, 16 bits × 1, etc.)

- 8-bit programmable square wave pulse generation (PPG: Variable duty with variable cycle) output mode (1 timer)
- 8-bit pulse width modulation (PWM: Variable duty with constant cycle) output mode (1 timer)

Figure 3.7.1 shows the block diagram of 8-bit timer (Timer 0, 1), and Figure 3.7.2 shows the block diagram of 8-bit timer (Timer 2, 3).

Each interval timer consists of an 8-bit up counter, 8-bit comparator, and 8-bit timer register. Besides, timer flip-flops (TFF1, TFF3), are provided for pair of timer 0/1 and 2/3.

Among the input clock sources for the interval timers, the internal clocks of φT1, φT4, φT16, and φT256 are obtained from the 9-bit prescaler shown in Figure 3.7.3.

The operation modes and timer flip-flops of the 8-bit timer are controlled by five control registers T10MOD, T32MOD, TFFCR, TRUN, and TRDC.

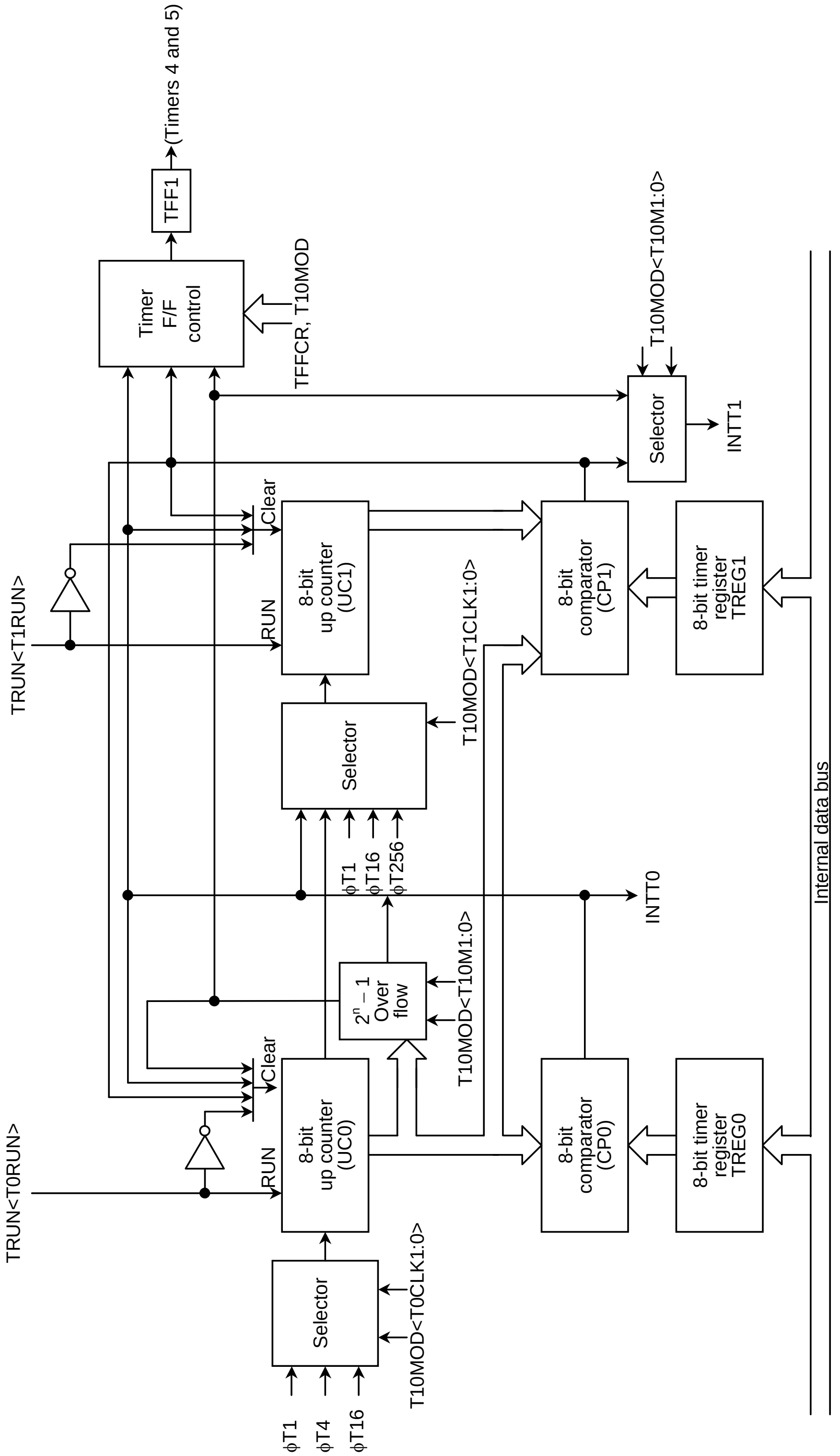


Figure 3.7.1 Block Diagram of 8-Bit Timers (Timers 0 and 1)

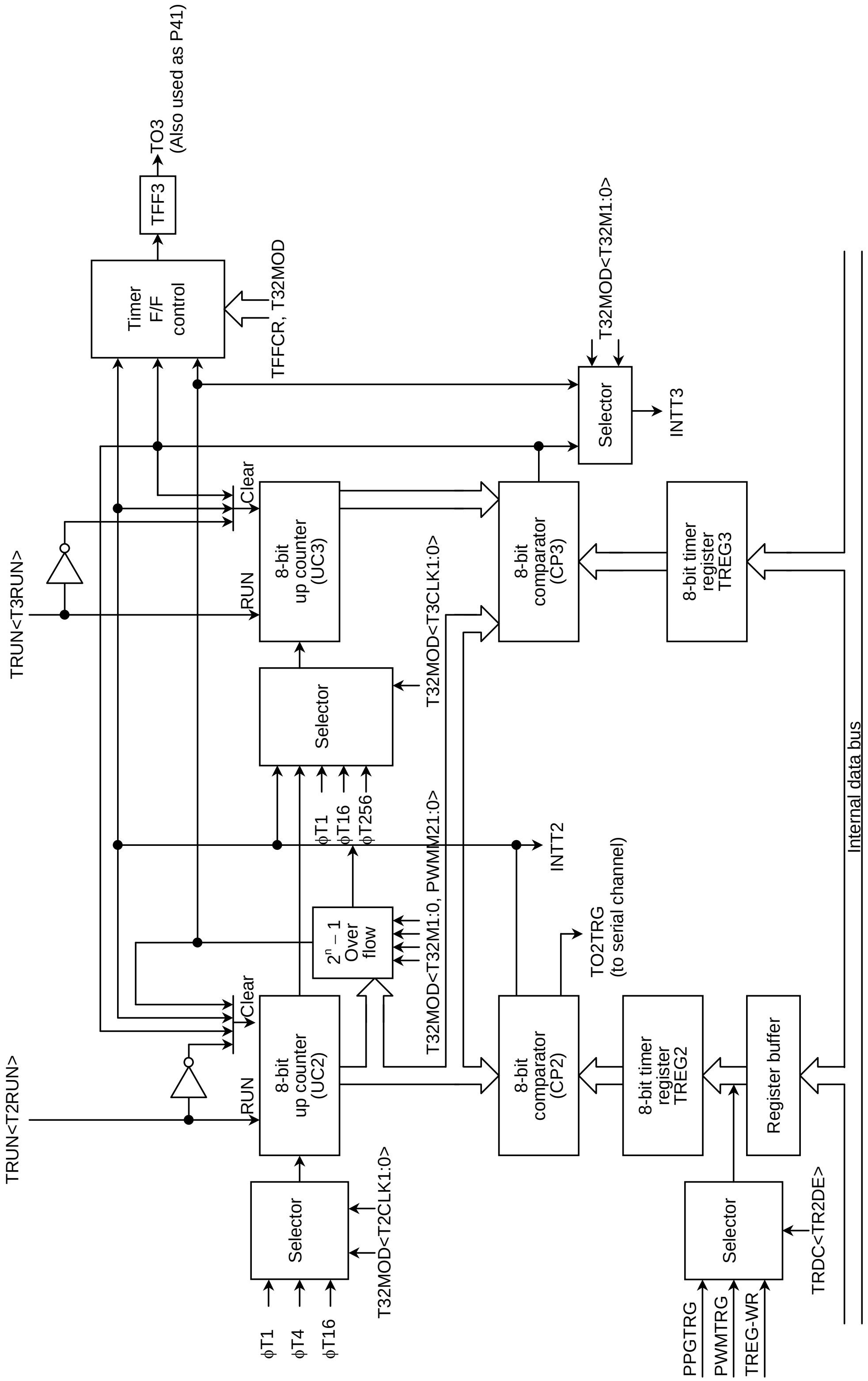


Figure 3.7.2 Block Diagram of 8-Bit Timers (Timers 2 and 3)

(1) Prescaler

There are 9-bit prescaler and prescaler clock selection registers to generate input clock for 8-bit timer 0, 1, 2, and 3, 16-bit timers 4 and 5 and serial interfaces 0 and 1.

Figure 3.7.3 shows the block diagram. Table 3.7.1 shows prescaler clock resolution into 8- and 16-bit timer.

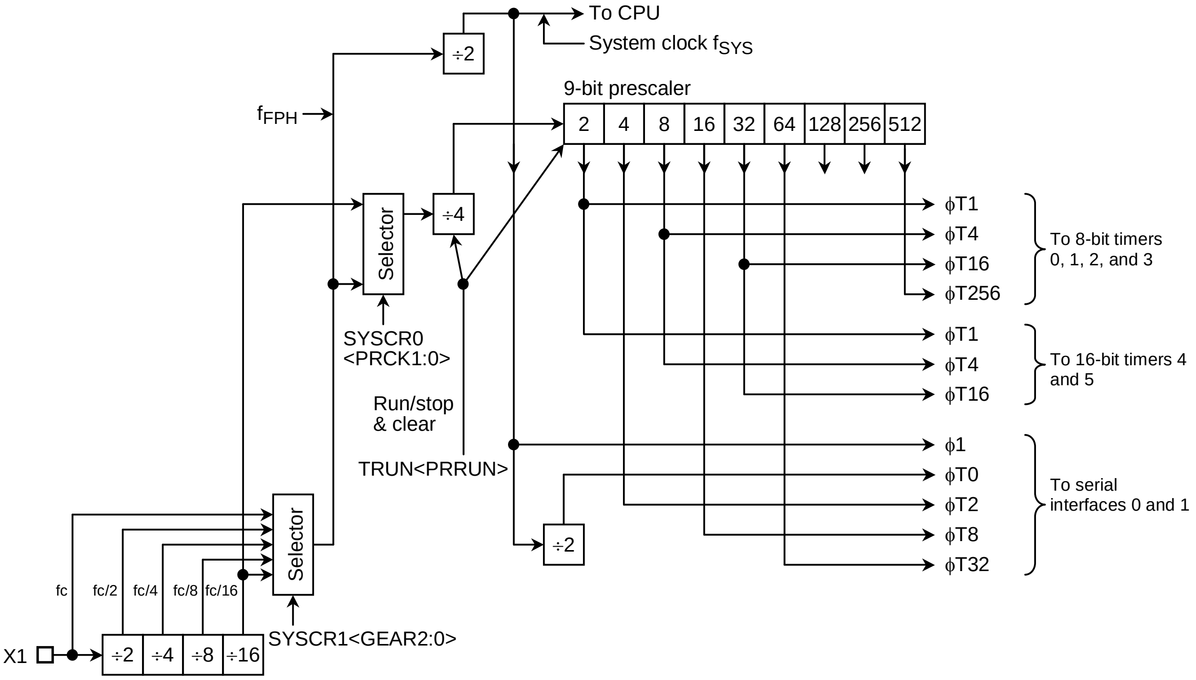


Figure 3.7.3 The Block Diagram of Prescaler

Table 3.7.1 Prescaler Clock Resolution to 8- and 16-Bit Timer

at fc = 20 MHz

Select Prescaler Clock <PRCK1:0>	Gear Value <GEAR2:0>	Prescaler Clock Resolution			
		φT1	φT4	φT16	φT256
00 (f_{FPH})	000 (fc)	fc/2^3 (0.4 μs)	fc/2^5 (1.6 μs)	fc/2^7 (6.4 μs)	fc/2^{11} (102.4 μs)
	001 (fc/2)	fc/2^4 (0.8 μs)	fc/2^6 (3.2 μs)	fc/2^8 (12.8 μs)	fc/2^{12} (204.8 μs)
	010 (fc/4)	fc/2^5 (1.6 μs)	fc/2^7 (6.4 μs)	fc/2^9 (25.6 μs)	fc/2^{13} (409.6 μs)
	011 (fc/8)	fc/2^6 (3.2 μs)	fc/2^8 (12.8 μs)	fc/2^{10} (51.2 μs)	fc/2^{14} (819.2 μs)
	100 (fc/16)	fc/2^7 (6.4 μs)	fc/2^9 (25.6 μs)	fc/2^{11} (102.4 μs)	fc/2^{15} (1.6384 ms)
10 (fc/16 clock)	XXX	fc/2^7 (6.4 μs)	fc/2^9 (25.6 μs)	fc/2^{11} (102.4 μs)	fc/2^{15} (1.6384 ms)

XXX: Don’t care

← 16-bit timer →

← 8-bit timer →

The clock selected among f_{FPH} clock and fc/16 clock is divided by 4 and input to this prescaler. This is selected by prescaler clock selection register SYSCR0<PRCK1:0>.

Resetting sets <PRCK1:0> to “00”, therefore f_{FPH}/4 clock is input.

The 8-bit Timer selects between 4 clock inputs: ϕT1, ϕT4, ϕT16, and ϕT256 among the prescaler output.

This prescaler can be run or stopped by the timer control register TRUN<PRRUN>. Counting starts when <PRRUN> is set to “1”, while the prescaler is cleared to zero and stops operation when <PRRUN> is cleared to “0”.

When the IDLE1 mode (Operates only oscillator) is used, clear TRUN<PRRUN> to “0” to stop this prescaler before “HALT” instruction is executed.

(2) Up counter

This is an 8-bit binary counter which counts up by the input clock pulse specified by T10MOD and T32MOD.

The input clock of timers 0 and 2 is selected from the three internal clocks ϕT1, ϕT4, and ϕT16, according to the set value of T10MOD<T0CLK1:0>/T32MOD<T2CLK1:0> registers.

The input clock of timer 1 and 3 differs depending on the operation mode. When set to 16-bit timer mode, the overflow outputs of timer 0 and 2 are used as the input clock. When set to any other mode than 16-bit timer mode, the input clock is selected from the internal clocks ϕT1, ϕT16, and ϕT256 as well as the comparator output (match detection signal) of timer 0, 2 according to the set value of T10MOD and T32MOD registers.

Example: When T10MOD<T10M1:0> = “01”, the overflow output of timer 0 becomes the input clock of timer 1 (16-bit timer mode).
When T10MOD<T10M1:0> = “00” and T10MOD<T1CLK1:0> = “01”, ϕT1 becomes the input of timer 1 (8-bit timer mode).

Operation mode is also set by T10MOD and T32MOD registers. When reset, it is initialized to T10MOD<T10M1:0> = “00” and T32MOD<T32M1:0> = “00” whereby the up counter is placed in the 8-bit timer mode.

The counting and stop and clear of up counter can be controlled for each interval timer by the timer operation control register TRUN. When reset, all up-counters will be cleared to stop the timers.

(3) Timer register

This is an 8-bit register for setting an interval time. When the set value of timer registers TREG0, TREG1, TREG2, and TREG3, matches the value of up-counter, the comparator match detect signal becomes active. If the set value is 00H, this signal becomes active when the up-counter overflows.

Timer registers TREG2, are double buffer structure, each of which makes a pair with register buffer.

The timer flip-flop controll register TRDC<TR2DE> bits control whether the double buffer structure in the TREG2 should be enabled or disabled. They are disabled when <TR2DE> = 0 and enabled when they are set to 1.

In the condition of double buffer enable state, the data is transfered from the register buffer to the timer register when the $2^n - 1$ overflow occurs in PWM mode, or at the PPG cycle in PPG mode. Therefore, during timer mode, the double buffer can not be used.

When reset, it will be initialized to <TR2DE> = 0 to disable the double buffer. To use the double buffer, write data in the timer register, set <TR2DE> to 1, and write the following data in the register buffer.

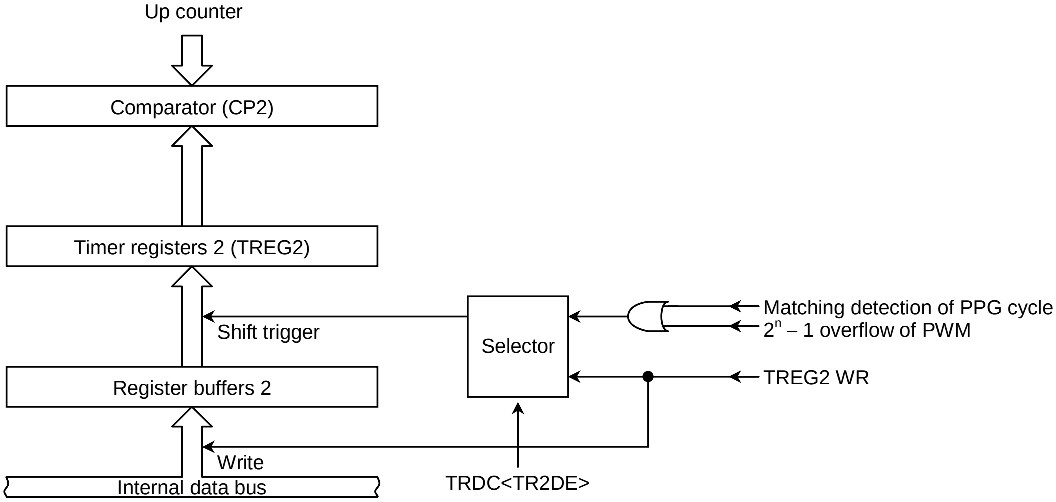


Figure 3.7.4 Configuration of Timer Register 2

Note: Timer register and the register buffer are allocated to the same memory address. When <TR2DE> = 0, the same value is written in the register buffer as well as the timer register, while when <TR2DE> = 1 only the register buffer is written.

The memory address of each timer register is as follows.

TREG0 : 000022H TREG2 : 000026H
TREG1 : 000023H TREG3 : 000027H

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

(4) Comparator

A comparator compares the value in the up counter with the values to which the timer register is set. When they match, the up counter is cleared to zero and an interrupt signal (INTT0, INTT1, INTT2, INTT3) is generated. If the timer flip-flop inversion is enabled, the timer flip-flop is inverted at the same time.

(5) Timer flip-flop (Timer F/F: TFF1, TFF3)

The timer flip-flop (TFF1, TFF3) is a flip-flop inverted by the match detect signal (8-bit comparator output) of each interval timer.

Inverting is disabled or enabled by the timer flip-flop control register TFFCR<TFF3IE, TFF1IE>.

After reset operation, the value of TFF1 and TFF3 is undefined. Writing “01” or “10” to TFFCR<TFF3C1:0, TFF1C1:0> sets “0” or “1” to TFF1, TFF3. Additionally, writing “00” to this bit inverts the value of TFF1, TFF3 (Software inversion).

The signal of TFF3 is output through the TO3 pin (Also used as P41). When using as the timer output, the timer flip-flop should be set by port 4 function register P4FC beforehand. The output pin of TFF1 does not exist.

Timer Operation Control Register

		7	6	5	4	3	2	1	0
TRUN (0020H)	Bit symbol	PRRUN		T5RUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
	Read/Write	R/W		R/W					
	After reset	0		0	0	0	0	0	0
	Function	Prescaler and timer run/stop control 0: Stop and clear 1: Run (count up)							

Count operation

0	Stop and clear
1	Count

PRRUN: Operation of prescaler
T5RUN: Operation of 16-bit timer (Timer 5)
T4RUN: Operation of 16-bit timer (Timer 4)
T3RUN: Operation of 8-bit timer (Timer 3)
T2RUN: Operation of 8-bit timer (Timer 2)
T1RUN: Operation of 8-bit timer (Timer 1)
T0RUN: Operation of 8-bit timer (Timer 0)

Note: TRUN<Bit6> is always read as "1".

System Clock Control Register

		7	6	5	4	3	2	1	0
SYSCR0 (006EH)	Bit symbol	–	–	–	–	–	–	PRCK1	PRCK0
	Read/Write	R/W							
	After reset	1	0	1	0	0	0	0	0
	Function	(Note) Always write "1" (This bit is read as "1").	(Note) Always write "0" (This bit is read as "0").	(Note) Always write "1" (This bit is read as "1").	(Note) Always write "0" (This bit is read as "0").	(Note) Always write "0" (This bit is read as "0").	(Note) Always write "0" (This bit is read as "0").	Select prescaler clock 00: f_{FPH} 01: (Reserved) 10: fc/16 11: (Reserved)	

Select prescaler input clock

00	f_{FPH}
01	(Reserved)
10	fc/16
11	(Reserved)

Figure 3.7.5 8-Bit Timer Related Registers (1/5)

Timers 0 and 1 Mode Control Register

T10MOD
(0024H)

	7	6	5	4	3	2	1	0
Bit symbol	T10M1	T10M0			T1CLK1	T1CLK0	T0CLK1	T0CLK0
Read/Write	R/W				R/W			
After reset	0	0			0	0	0	0
Function	Operation mode 00: 8-bit timer 01: 16-bit timer 10: – 11: –				Source clock of timer 1 00: TO0TRG 01: ϕT1 10: ϕT16 11: ϕT256		Source clock of timer 0 00: Don't set 01: ϕT1 10: ϕT4 11: ϕT16	

Input clock of timer 0

00	Don't set
01	ϕT1 (Prescaler)
10	ϕT4 (Prescaler)
11	ϕT16 (Prescaler)

Input clock of timer 1

	T10MOD<T10M1:0> ≠ 01	T10MOD<T10M1:0> = 01
00	Comparator output of timer 0	Overflow output of timer 0 (16-bit timer mode)
01	Internal clock ϕT1	
10	Internal clock ϕT16	
11	Internal clock ϕT256	

Set the operation mode of timers 0 and 1.

00	Two 8-bit timers (timer 0 and timer 1)
01	16-bit timer
10	–
11	–

Figure 3.7.6 8-Bit Timer Related Register (2/5)

Timers 2 and 3 Mode Control Register

T32MOD
(0028H)

	7	6	5	4	3	2	1	0
Bit symbol	T32M1	T32M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Operation mode 00: 8-bit timer 01: 16-bit timer 10: 8-bit PPG 11: 8-bit PWM		PWM cycle 00: – 01: $2^6 - 1$ 10: $2^7 - 1$ 11: $2^8 - 1$		Source clock of timer 3 00: TO2TRG 01: φT1 10: φT16 11: φT256		Source clock of timer 2 00: Don’t set 01: φT1 10: φT4 11: φT16	

Input clock of timer 2

00	Don’t set
01	φT1 (Prescaler)
10	φT4 (Prescaler)
11	φT16 (Prescaler)

Input clock of timer 3

	T32MOD<T32M1:0> ≠ 01	T32MOD<T32M1:0> = 01
00	Comparator output of timer 2	Overflow output of timer 2
01	Internal clock φT1	
10	Internal clock φT16	(16-bit timer mode)
11	Internal clock φT256	

Select PWM cycle

00	* Don’t care
01	$(2^6 - 1) \times$ input clock frequency of timer 2
10	$(2^7 - 1) \times$ input clock frequency of timer 2
11	$(2^8 - 1) \times$ input clock frequency of timer 2

Set the operation mode of timers 2 and 3.

00	Two 8-bit timers (timer 2 and timer 3)
01	16-bit timer
10	8-bit PPG output
11	8-bit PWM output (timer 2) + 8-bit timer (timer 3)

Figure 3.7.7 8-Bit Timer Related Register (3/5)

Timer Flip-Flop Control Register

TFFCR
(0025H)

	7	6	5	4	3	2	1	0
Bit symbol	TFF3C1	TFF3C0	TFF3IE	TFF3IS	TFF1C1	TFF1C0	TFF1IE	TFF1IS
Read/Write	W		R/W		W		R/W	
After reset	1	1	0	0	1	1	0	0
Function	00: Invert TFF3 01: Set TFF3 10: Clear TFF3 11: Don't care		TFF3 Inversion trigger 0: Disable 1: Enable	TFF3 Inversion source 0: Timer 2 1: Timer 3	00: Invert TFF1 01: Set TFF1 10: Clear TFF1 11: Don't care		TFF1 Inversion trigger 0: Disable 1: Enable	TFF1 Inversion source 0: Timer 0 1: Timer 1

Select inverse signal of timer F/F3
("Don't care" except in 8-bit timer mode)

0	Inversion by timer 2
1	Inversion by timer 3

Inversion of timer F/F3 (TFF3)

0	Disable invert
1	Enable invert

Control of timer F/F3 (TFF3)

00	Invert the value of TFF3 (software inversion)
01	Set TFF3 to "1".
10	Clear TFF3 to "0".
11	Don't care

Select inverse signal of timer F/F1
("Don't care" except in 8-bit timer mode)

0	Inversion by timer 0
1	Inversion by timer 1

Inversion of Timer F/F1 (TFF1)

0	Disable invert
1	Enable invert

Control of Timer F/F1 (TFF1)

00	Invert the value of TFF1 (software inversion)
01	Set TFF1 to "1".
10	Clear TFF1 to "0".
11	Don't care

Note: TFFCR<TFF3C1:0, TFF1C1:0> is always read as "1".

Figure 3.7.8 8-Bit Timer Related Register (4/5)

Timer Register Double Buffer Control Register

TRDC
(0029H)

	7	6	5	4	3	2	1	0
Bit symbol							TR2DE	–
Read/Write							R/W	
After reset							0	0
Function							0: Double buffer disable 1: Double buffer enable	(Note) Always write "0".

Operation of timer register 2 double butter

0	Disable
1	Enable

Figure 3.7.9 8-Bit Timer Related Register (5/5)

(1) 8-bit timer mode

Four interval timers 0, 1, 2, and 3, can be used independently as 8-bit interval timer.

1. Generating interrupts in a fixed cycle (in case of timer 3)

   To generate timer 3 interrupt at constant intervals using timer 3 (INTT3), first stop timer 3 then set the operation mode, input clock, and a cycle to T32MOD and TREG3 register, respectively. Then, enable interrupt INTT3 and start the counting of timer 3.

Example: To generate timer 3 interrupt every 10 μs at fc = 20 MHz, set each register in the following manner.

* Clock Condition
- Clock gear: 1 (fc)
- Prescaler clock: f_{FPH}

		MSB							LSB	
		7	6	5	4	3	2	1	0	
TRUN	←	–	X	–	–	0	–	–	–	Stop timer 3, and clear it to "0".
T32MOD	←	0	0	X	X	0	1	–	–	Set the 8-bit timer mode, and select ϕT1 (0.4 μs at fc = 20 MHz) as the input clock.
TREG3	←	0	0	0	1	1	0	0	1	Set the timer register 10 μs ÷ ϕT1 = 25 = 19H
INTET32	←	1	1	0	1	–	–	–	–	Enable INTT3, and set it to "Level 5".
TRUN	←	1	X	–	–	1	–	–	–	Start timer 3 counting.

X: Don't care, –: No change

Use the Table 3.7.1 for selecting the input clock.

Note: The input clock of timer 2 and timer 3 are different from as follows.
Timer 2: ϕT1, ϕT4, ϕT16
Timer 3: Match output of timer 2, ϕT1, ϕT16, and ϕT256

2. Generating a 50% duty square wave pulse

The timer flip-flop is included in timers 1 and 3.

The timer flip-flop (TFF3) is inverted at constant intervals, and its status is output to timer output pin (TO3). The output pin of TFF1 does not exist.

Example: To output a 2.4 μs square wave pulse from TO3 pin at fc = 20 MHz, set each register in the following procedures. Either timer 2 or timer 3 may be used, but this example uses timer 3.

* Clock Condition

Clock gear: 1 (fc)
Prescaler clock: f_{FPH}

		7	6	5	4	3	2	1	0	
TRUN	←	–	X	–	–	0	–	–	–	Stop timer 3, and clear it to “0”.
T32MOD	←	0	0	X	X	0	1	–	–	Set the 8-bit timer mode, and select ϕT1 (0.4 μs at fc = 20 MHz) as the input clock.
TREG3	←	0	0	0	0	0	0	1	1	Set the timer register at 2.4 μ s ÷ ϕT1 ÷ 2 = 3.
TFFCR	←	1	0	1	1	–	–	–	–	Set TFF3 to “0”, and set to invert by the match detect signal from timer 3.
P4CR	←	X	X	X	X	–	–	1	–	Select P41 as TO3 pin.
P4FC	←	X	X	X	X	–	–	1	X	
TRUN	←	1	X	–	–	1	–	–	–	Start timer 3 counting.

X: Don’t care, –: No change

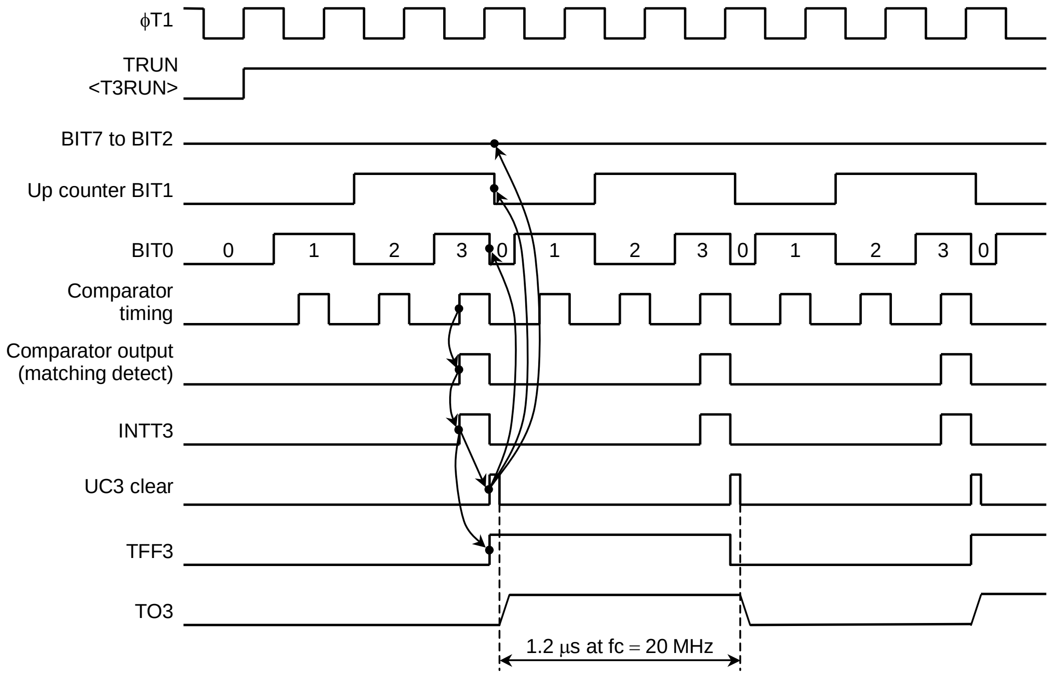


Figure 3.7.10 Square Wave (50% duty) Output Timing Chart

3. Making timer 1 count up by match signal from timer 0 comparator
   (Same function is achieved by using timer 3 and timer 2)

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

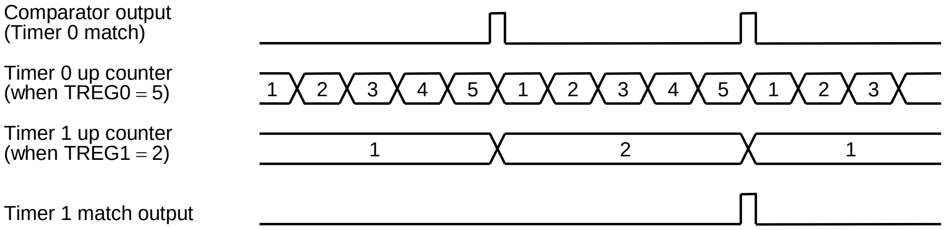


Figure 3.7.11 Timer 1 Count up by Timer 0

(2) 16-bit timer mode

A 16-bit interval timer is configured by using the pair of timer 0 and timer 1 or timer 2 and timer 3.

To make a 16-bit interval timer by cascade connecting timer 0 and timer 1, set timer 1/0 mode register T10MOD<T10M1:0> to "01".

When set in 16-bit timer mode, the overflow output of timer 0 and 2 will become the input clock of timers 1 and 3, regardless of the set value of T10MOD<T1CLK1:0>and T32MOD<T3CLK1:0>. Table 3.7.1 shows the relation between the cycle of timer (interrupt) and the selection of input clock.

The lower 8 bits of the timer (Interrupt) cycle are set by the timer register TREG0 or TREG2, and the upper 8 bits are set by TREG1 or TREG3. Note that TREG0 and TREG2 always must be set first. (Writing data into TREG0 and TREG2 disables the comparator temporarily, and the comparator is restarted by writing data into TREG1 and TREG3.)

Setting example: To generate an interrupt INTT3 every 0.4 seconds at fc = 20 MHz, set the following values for timer registers TREG2 and TREG3.

* Clock Condition

Clock gear: 1 (fc)
Prescaler clock: f_{FPH}

When counting with input clock of ϕT16 (6.4 μs at 20 MHz)

0.4 s ÷ 6.4 μs = 62500 = F424H

Therefore, set TREG3 = F4H and TREG2 = 24H, respectively.

The comparator match signal is output from timer 2 each time the up counter UC2 matches TREG2, where the up counter UC2 is not be cleared, and the interrupt INTT2 is not generated.

With the timer 3 comparator, the match detect signal is output at each comparator timing when up counter UC3 and TREG3 values match. When the match detect signal is output simultaneously from both comparators of timer 2 and timer 3, the up counters UC2 and UC3 are cleared to "0", and the interrupt INTT3 is generated. If inversion is enabled, the value of the timer flip-flop TFF3 is inverted.

Example: When TREG3 = 04H and TREG2 = 80H

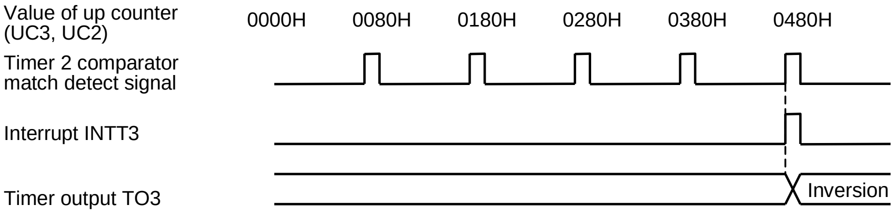


Figure 3.7.12 Timer Output by 16-Bit Timer Mode

(3) 8-bit PPG (Programmable Pulse Generation) output mode

Square wave pulse can be generated at any frequency and duty by timer 2. The output pulse may be either low active or high active. In this mode, timer 3 cannot be used.

Timer 2 outputs pulse to TO3 pin (also used as P41).

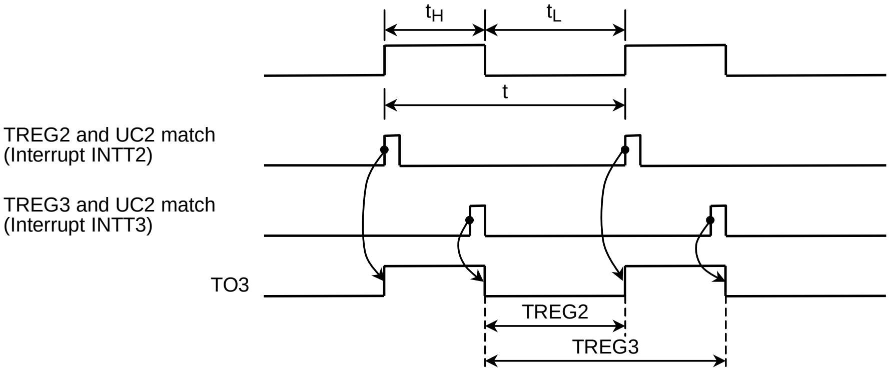


Figure 3.7.13 8-Bit PPG Output Waveforms

In this mode, a programmable square wave is generated by inverting timer output each time the 8-bit up counter (UC2) matches the timer registers TREG2 and TREG3.

However, it is required that the set value of TREG2 is smaller than that of TREG3.

Though the up counter (UC3) of timer 3 is not used in this mode, UC3 should be set for counting by setting TRUN<T3RUN> to 1.

Figure 3.7.14 shows the block diagram for this mode.

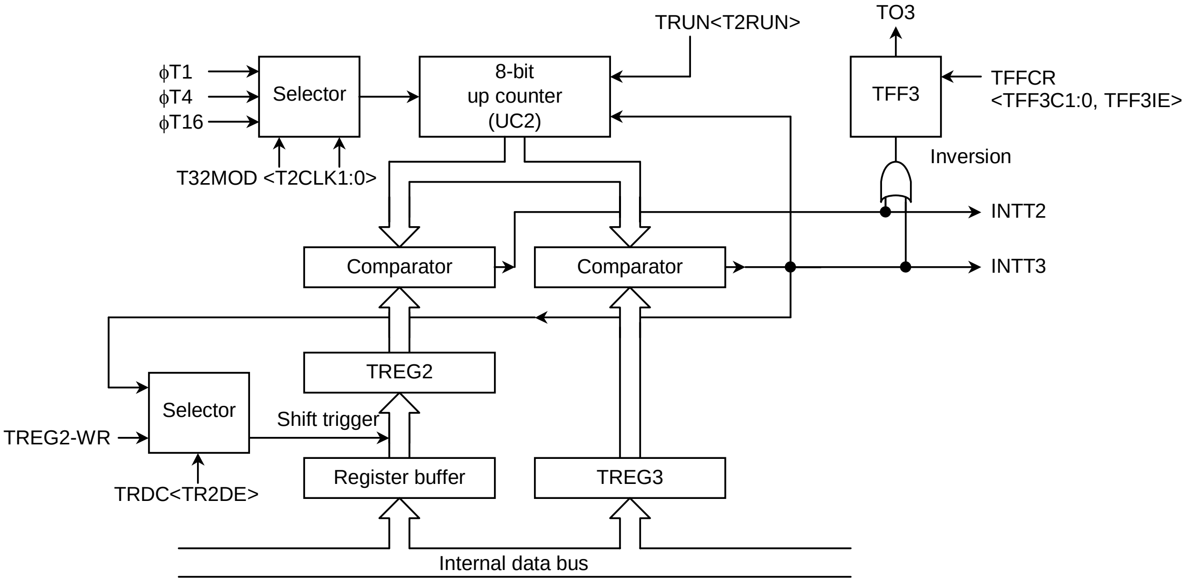


Figure 3.7.14 Block Diagram of 8-Bit PPG Output Mode

When the double buffer of TREG2 is enabled in this mode, the value of register buffer will be shifted in TREG2 each time TREG3 matches UC2.

Use of the double buffer makes easy the handling of low duty waves (when duty is varied).

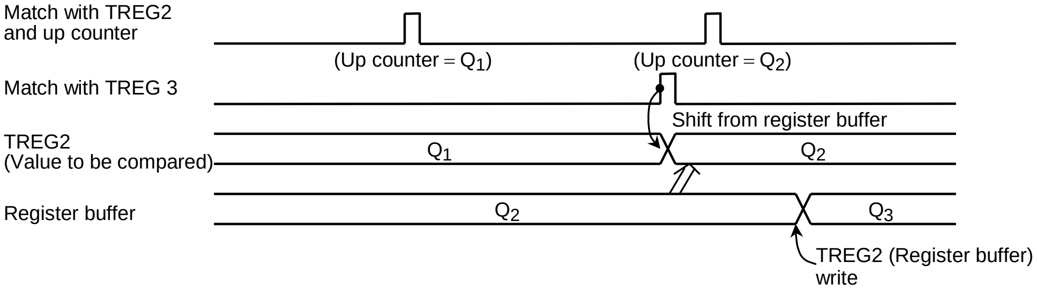
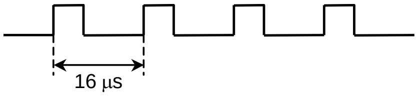


Figure 3.7.15 Operation of Register Buffer

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



* Clock condition

Clock gear: 1 (fc)
Prescaler clock: f_{FPH}

Calculate the value to be set for timer register.

To obtain the frequency 62.5 kHz, the pulse cycle t should be: t = 1/62.5 kHz = 16 µs.

Given $\phi T1 = 0.4$ µs (at 20 MHz),

$16\ \mu s \div 0.4\ \mu s = 40$

Consequently, to set the timer register 3 (TREG3) to TREG3 = 40 = 28H and then duty to 1/4, $t \times 1/4 = 16\ \mu s \times 1/4 = 4\ \mu s$

$4\ \mu s \div 0.4\ \mu s = 10$

Therefore, set timer register 2 (TREG2) to TREG2 = 10 = 0AH.

		7	6	5	4	3	2	1	0	
TRUN	←	–	X	–	–	0	0	–	–	Stop timer 2, 3 and clear it to "0".
T32MOD	←	1	0	X	X	X	X	0	1	Set the 8-bit PPG mode, and select ϕT1 as input clock.
TREG2	←	0	0	0	0	1	0	1	0	Write "0AH".
TREG3	←	0	0	1	0	1	0	0	0	Write "28H".
TFFCR	←	0	1	1	X	–	–	–	–	Sets TFF3 and enable the inversion and double buffer enable. Writing "10" provides negative logic pulse.
P4CR	←	X	X	X	X	–	–	1	–	Set P41 as the TO3 pin.
P4FC	←	X	X	X	X	–	–	1	X	
TRUN	←	1	X	–	–	1	1	–	–	Start timer 2 and timer 3 counting.

X: Don't care –: No change

(4) 8-bit PWM output mode

This mode is valid only for timer 2. In this mode, maximum 8-bit resolution of PWM pulse can be output.

PWM pulse is output to TO3 pin (also used as P41) when using timer 2. Timer 3 can also be used as 8-bit timer.

Timer output is inverted when up counter (UC2) matches the set value of timer register TREG2 or when 2n – 1 (n = 6, 7, or 8; specified by T32MOD<PWM21:20>) counter overflow occurs. Up counter UC2 is cleared when 2n – 1 counter overflow occurs.

To use this PWM mode, the following conditions must be satisfied.

(Set value of timer register) < (Set value of $2^n - 1$ counter overflow)

(Set value of timer register) ≠ 0

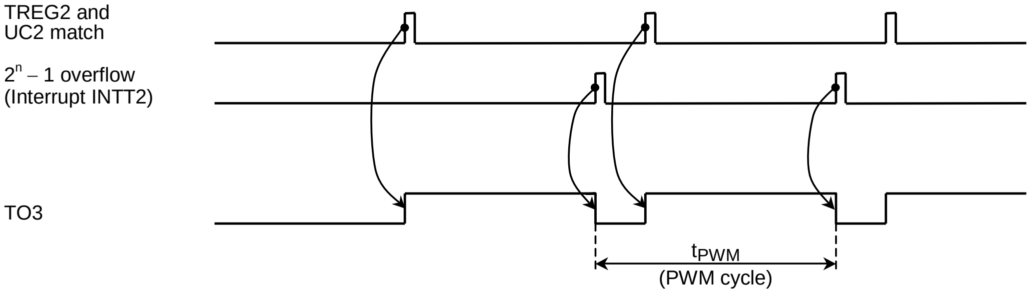


Figure 3.7.16 8-Bit PWM Waveforms

Figure 3.7.17 shows the block diagram of this mode.

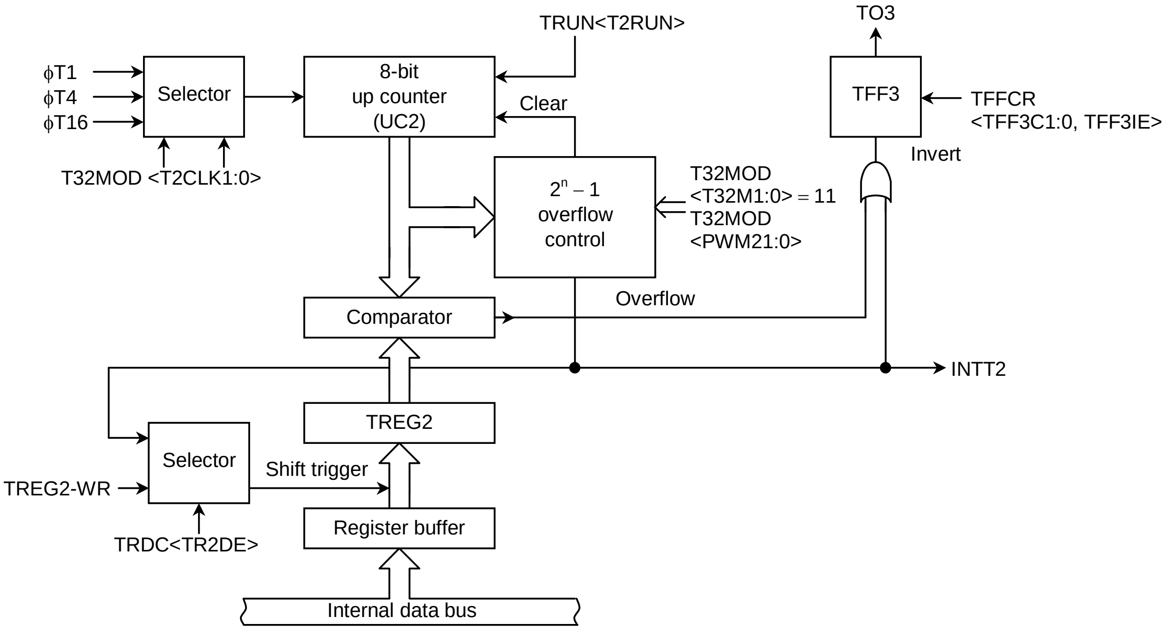


Figure 3.7.17 Block Diagram of 8-Bit PWM Mode

In this mode, the value of register buffer will be shifted in TREG2 if $2^n - 1$ overflow is detected when the double buffer of TREG2 is enabled.

Use of the double buffer makes easy the handling of small duty waves.

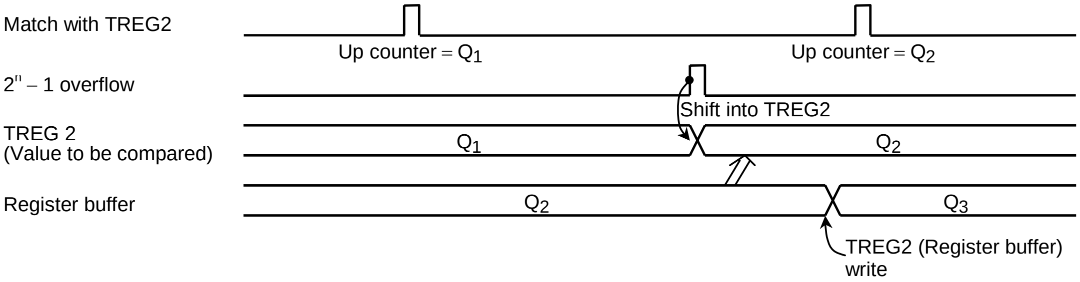
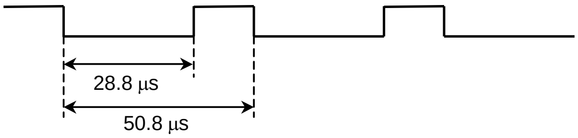


Figure 3.7.18 Operation of Register Buffer

Example: To output the following PWM waves to TO3 pin at fc = 20 MHz.



* Clock condition
Clock gear: 1 (fc)
Prescaler clock: f_{FPH}

To realize 50.8 μs of PWM cycle by ϕT1 = 0.4 μs (at fc = 20 MHz),

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

Consequently, n should be set to 7.

As the period of low level is 28.8 μs, for ϕT1 = 0.4 μs,
set the following value for TREG2.

$$28.8\ \mu s \div 0.4\ \mu s = 72 = 48H$$

		MSB							LSB	
		7	6	5	4	3	2	1	0	
TRUN	←	–	X	–	–	–	0	–	–	Stop timer 2, and clear it to “0”.
T32MOD	←	1	1	1	0	–	–	0	1	Set 8-bit PWM mode (cycle: $2^7 - 1$) and select ϕT1 as the input clock.
TREG2	←	0	1	0	0	1	0	0	0	Writes “48H”.
TFFCR	←	1	0	1	X	–	–	–	–	Clears TFF3, enable the inversion and double buffer.
P4CR	←	X	X	X	X	–	–	1	–	Set P41 as the TO3 pin.
P4FC	←	X	X	X	X	–	–	1	X	
TRUN	←	1	X	–	–	–	1	–	–	Start timer 2 counting.

X: Don’t care, –: No change

Table 3.7.2 PWM Cycle

at fc = 20 MHz

Select Prescaler Clock <PRCK1:0>	Gear Value <GEAR2:0>	PWM Cycle								
		$2^6 - 1$			$2^7 - 1$			$2^8 - 1$		
		ϕT1	ϕT4	ϕT16	ϕT1	ϕT4	ϕT16	ϕT1	ϕT4	ϕT16
00 (f_{FPH})	000 (fc)	25.2 μs	100.8 μs	403.2 μs	50.8 μs	203.2 μs	812.8 μs	102.0 μs	408.0 μs	1.63 ms
	001 (fc/2)	50.4 μs	201.6 μs	806.4 μs	101.6 μs	406.4 μs	1.63 ms	204.0 μs	816.0 μs	3.26 ms
	010 (fc/4)	100.8 μs	403.2 μs	1.61 ms	203.2 μs	812.8 μs	3.26 ms	408.0 μs	1.63 ms	6.53 ms
	011 (fc/8)	201.6 μs	806.4 μs	3.23 ms	406.4 μs	1.63 ms	6.52 ms	816.0 μs	3.26 ms	13.06 ms
	100 (fc/16)	403.2 μs	1.61 ms	6.45 ms	812.8 μs	3.25 ms	13.04 ms	1.63 ms	6.53 ms	26.11 ms
10 (fc/16 clock)	XXX	403.2 μs	1.61 ms	6.45 ms	812.8 μs	3.25 ms	13.04 ms	1.63 ms	6.53 ms	26.11 ms

XXX: Don’t care

(5) Timer mode setting registers

Table 3.7.3 shows the list of 8-bit timer modes.

Table 3.7.3 Timer Mode Setting Registers

Register Name	T10MOD/T32MOD				TFFCR
Name of Function in Register	T10M/T32M	PWM2	T1CLK/T3CLK	T0CLK/T2CLK	TFF1IS/TFF3IS
Function	Timer Mode	PWM Cycle	Upper Timer Input Clock	Lower Timer Input Clock	Timer F/F Invert Signal Select
16-bit timer mode	01	* –	–	ϕT1, ϕT4, ϕT16 (01, 10, 11)	–
8-bit timer × 2 channels	00	* –	Lower timer match: ϕT1, 16, 256 (00, 01, 10, 11)	ϕT1, ϕT4, ϕT16 (01, 10, 11)	0: Lower timer output 1: Upper timer output
8-bit PPG × 1channel	* 10	* –	* –	* ϕT1, ϕT4, ϕT16 (01, 10, 11)	* –
8-bit PWM × 1channel	* 11	* $2^6 - 1$, $2^7 - 1$, $2^8 - 1$ (01, 10, 11)	* –	* ϕT1, ϕT4, ϕT16 (01, 10, 11)	* –
8-bit timer × 1channel	* 11	–	ϕT1, ϕT16, ϕT256 (01, 10, 11)	–	Output disabled

–: Don’t care

*: Don’t set in T10MOD

3.8 16-Bit Timers/Event Counters

The TMP93CS32 contains two (Timer 4 and timer 5) multifunctional 16-bit timer/event counter with the following operation modes.

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

Can be used following operation modes by capture function.

- Frequency measurement mode
- Pulse width measurement mode
- Time differential measurement mode

Timer/event counter consists of 16-bit up counter, two 16-bit timer registers, two 16-bit capture registers (One of them applies double buffer), two comparators, capture input controller, and timer flip-flop and the control circuit.

Timer/event counter is controlled by 4 control registers: T4MOD/T5MOD, T4FFCR/T5FFCR, TRUN and T45CR.

Figure 3.8.1 and Figure 3.8.2 shows the block diagram of 16-bit timer/event counter (timer 4 and timer 5).

Timer 4 and 5 can be used independently.

All timer operate in the same manner, and thus only the operation of Timer 4 will be explained below.

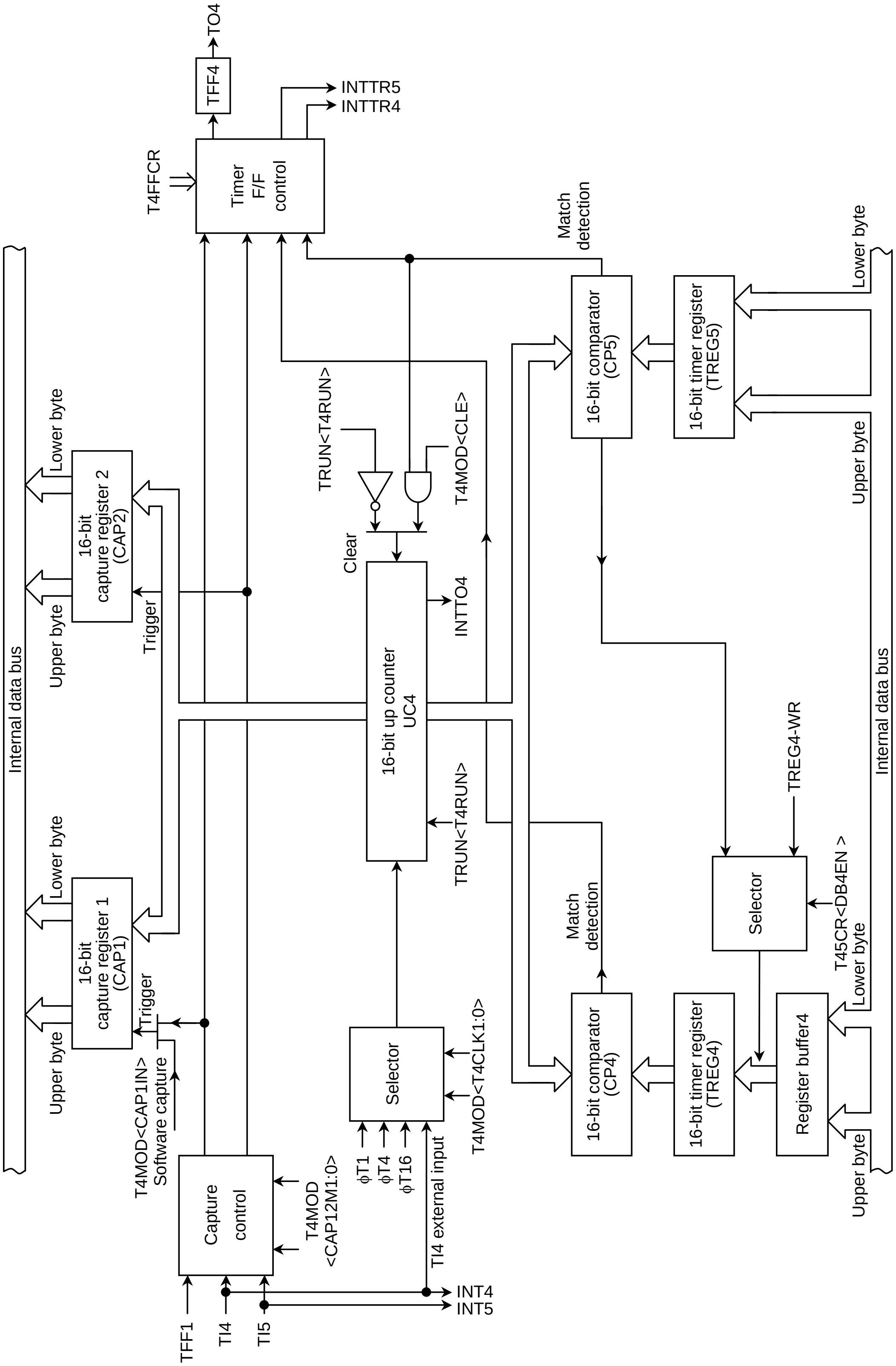


Figure 3.8.1 Block Diagram of 16-Bit Timer (Timer 4)

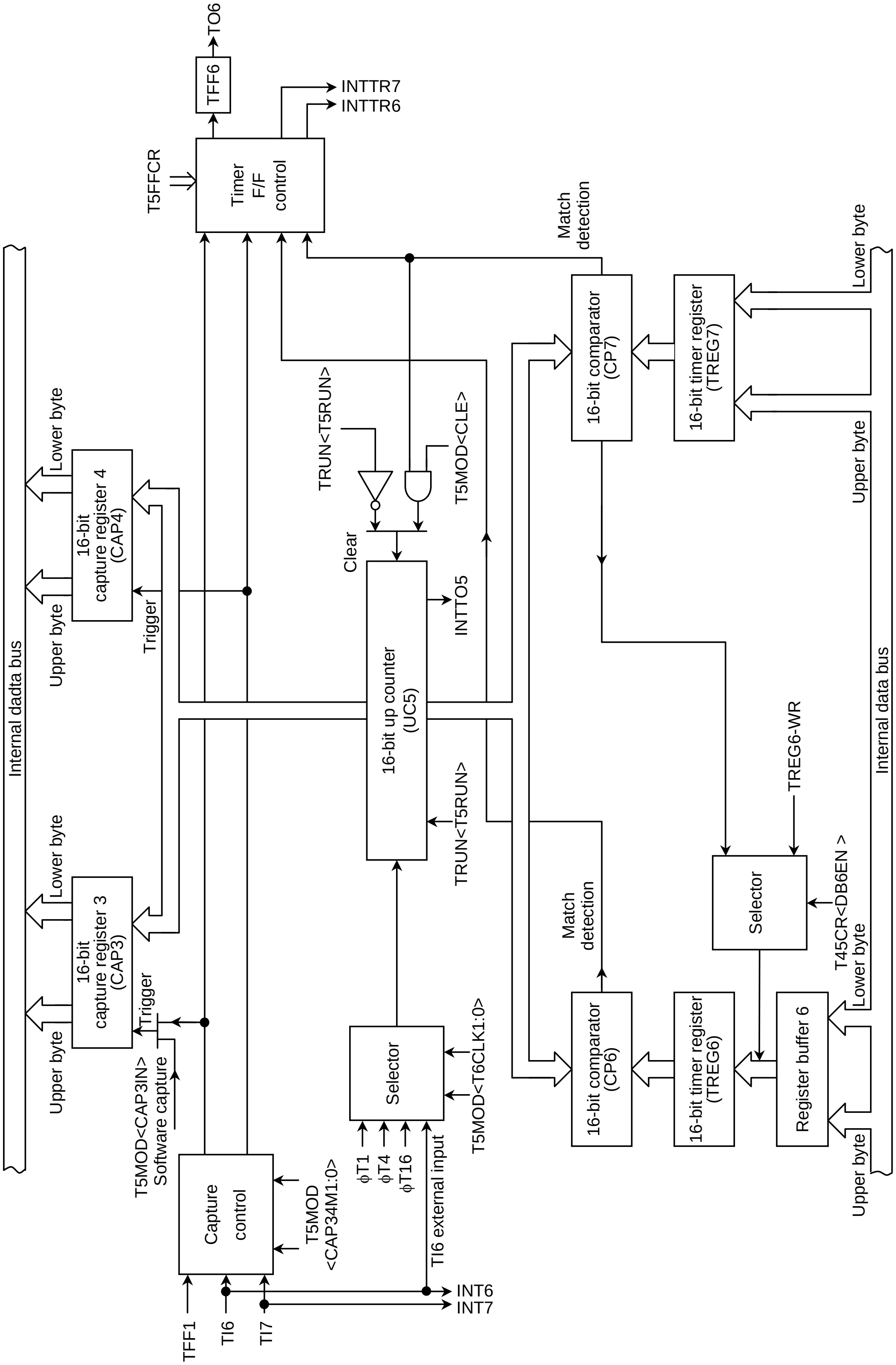


Figure 3.8.2 Block Diagram of 16-Bit Timer (Timer 5)

Timer 4 Mode Control Register

T4MOD
(0038H)

	7	6	5	4	3	2	1	0
Bit symbol			CAP1IN	CAP12M1	CAP12M0	CLE	T4CLK1	T4CLK0
Read/Write			W	R/W				
After reset			1	0	0	0	0	0
Function			0: Software capture 1: Don't care	Capture timing 00: Disable INT4 occurs at rising edge. 01: TI4↑ TI5↑ INT4 occurs at rising edge. 10: TI4↑ TI4↓ INT4 occurs at falling edge. 11: TFF1↑ TFF1↓ INT4 occurs at rising edge.		1: UC4 clear enable	Timer 4 source clock 00: TI4 pin 01: ϕT1 10: ϕT4 11: ϕT16	

Timer 4 input clock

00	External clock (TI4)
01	ϕT1
10	ϕT4
11	ϕT16

Clearing the up counter UC4

0	Clear disable
1	Clear by match with TREG5.

Capture timing of timer 4

	Capture control	INT4 control
00	Capture disable	INT4 occurs on the rising edge of TI4. ↑
01	CAP1 at TI4 rise CAP2 at TI5 rise	
10	CAP1 at TI4 rise CAP2 at TI4 fall	INT4 occurs on the falling edge of TI4. ↓
11	CAP1 at TFF1 rise CAP2 at TFF1 fall	INT4 occurs on the rising edge of TI4. ↑

Software capture

0	The up counter 4 value is loaded to CAP1.
1	Don't care

Figure 3.8.3 16-Bit Timer/Event Counter Related Register (1/6)

Timer 4 Flip-Flop Control Register

T4FFCR
(0039H)

	7	6	5	4	3	2	1	0
Bit symbol			CAP2T4	CAP1T4	EQ5T4	EQ4T4	TFF4C1	TFF4C0
Read/Write			R/W				W	
After reset			0	0	0	0	1	1
Function			TFF4 invert trigger 0: Trigger disable 1: Trigger enable				TFF4 control 00: Invert TFF4 01: Set TFF4 10: Clear TFF4 11: Don’t care	
			Invert when the UC value is loaded to CAP2.	Invert when the UC value is loaded to CAP1.	Invert when the UC matches TREG5.	Invert when the UC matches TREG4.		

Timer flip-flop 4 (TFF4) control

00	Inverts the TFF4 value (Software inversion).
01	Sets TFF4 to “1”.
10	Clear TFF4 to “0”.
11	Don’t care (Always read “11”).

Timer flip-flop 4 (TFF4) invert trigger at the up counter matches TREG4

0	Trigger disable (Invert prohibition)
1	Trigger enable (Invert permission)

Timer flip-flop 4 (TFF4) invert trigger at the up counter matches TREG5

0	Trigger disable (Invert prohibition)
1	Trigger enable (Invert permission)

Timer flip-flop 4 (TFF4) invert trigger at the up counter value is loaded to CAP1

0	Trigger disable (Invert prohibition)
1	Trigger enable (Invert permission)

Timer flip-flop 4 (TFF4) invert trigger at the up counter value is loaded to CAP2

0	Trigger disable (Invert prohibition)
1	Trigger enable (Invert permission)

Figure 3.8.4 16-Bit Timer/Event Counter Related Register (2/6)

Timer 5 Mode Control Register

		7	6	5	4	3	2	1	0
T5MOD (0048H)	Bit symbol			CAP3IN	CAP34M1	CAP34M0	CLE	T5CLK1	T5CLK0
	Read/Write			W	R/W				
	After reset			1	0	0	0	0	0
	Function			Software capture control 0: Software capture 1: Don't care	Capture timing 00: Disable INT6 occurs on the rising edge. 01: TI6↑ TI7↑ INT6 occurs at rising edge. 10: TI6↑ TI6↓ INT6 occurs at falling edge. 11: TFF1↑ TFF1↓ INT4 occurs at rising edge.		Timer 5 UC control 0: Clear disable 1: Clear enable	Timer 5 source clock select 00: TI6 pin 01: φT1 10: φT4 11: φT16	

Timer 5 input clock

00	External clock (TI6 pin input)
01	φT1
10	φT4
11	φT16

Clearing the up counter UC5

0	Clear disable
1	Clear by match with TREG7

Timer 5 capture timing

	Capture control	INT6 control
00	Capture disable	INT6 occurs on the rising edge of TI6.
01	CAP3 at TI6 rise CAP4 at TI7 rise	
10	CAP3 at TI6 rise CAP4 at TI6 fall	INT6 occurs on the falling edge of TI6.
11	CAP3 at TFF1 rise CAP4 at TFF1 fall	INT6 occurs on the rising edge of TI6.

Software capture

0	The up counter 5 value is loaded to CAP3.
1	Don't care

Figure 3.8.5 16-Bit Timer/Event Counter Related Register (3/6)

Timer 5 Flip-Flop Control Register

T5FFCR
(0049H)

		7	6	5	4	3	2	1	0
T5FFCR (0049H)	Bit symbol			CAP4T6	CAP3T6	EQ7T6	EQ6T6	TFF6C1	TFF6C0
	Read/Write			R/W				W	
	After reset			0	0	0	0	1	1
	Function			TFF6 invert trigger 0: Disable trigger 1: Enable trigger				TFF6 control 00: Invert TFF6 01: Set TFF6 10: Clear TFF6 11: Don’t care	
				Invert when the UC value is loaded to CAP4.	Invert when the UC value is loaded to CAP3.	Invert when the UC matches TREG7.	Invert when the UC matches TREG6.		

Timer flip-flop 6 (TFF6) control

00	Inverts the TFF6 value (Software inversion).
01	Sets TFF6 to “1”.
10	Clear TFF6 to “0”.
11	Don’t care (Always read “11”)

Timer flip-flop 6 (TFF6) invert trigger
at up counter matches TREG6

0	Trigger disable (Invert prohibition)
1	Trigger enable (Invert permission)

CAP4T6: Invert when the up counter value is loaded to CAP4
CAP3T6: Invert when the up counter value is loaded to CAP3
EQ7T6: Invert when up counter matches TREG7
EQ6T6: Invert when up counter matches TREG6

Timer flip-flop 6 (TFF6) invert trigger
at up counter matches TREG7

0	Trigger disable (Invert prohibition)
1	Trigger enable (Invert permission)

Figure 3.8.6 16-Bit Timer/Event Counter Related Register (4/6)

Timer 4, 5 Control Register

		7	6	5	4	3	2	1	0
T45CR (003AH)	Bit symbol	QCU						DB6EN	DB4EN
	Read/Write	R/W						R/W	
	After reset	0						0	0
	Function	Watchdog/ Warm-up timer control						Double buffer 0: Disable 1: Enable	
								Double buffer of TREG6	Double buffer of TREG4

Double buffer control

0	Disable
1	Enable

DB6EN: Double buffer of TREG6
DB4EN: Double buffer of TREG4

Watchdog timer/warm-up timer input control

0	Use 7 stage binary counter
1	Not use 7 stage binary counter (Note1)

Note 1: In case of unused 7 state binary counter as a warm-up timer, the stable clock must be input from external circuit.

Note 2: Bit6 to 2 of T45CR are read as "1".

Figure 3.8.7 16-Bit Timer/Event Counter Related Register (5/6)

Timer Operation Control Register

TRUN
(0020H)

	7	6	5	4	3	2	1	0
Bit symbol	PRRUN		T5RUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
Read/Write	R/W		R/W					
After reset	0		0	0	0	0	0	0
Function	Prescaler and timer run/stop control 0: Stop and clear 1: Run (Count up)							

Count operation

0	Stop and clear
1	Count

PRRUN: Operation of prescaler
T5RUN: Operation of 16-bit timer (timer 5)
T4RUN: Operation of 16-bit timer (timer 4)
T3RUN: Operation of 8-bit timer (timer 3)
T2RUN: Operation of 8-bit timer (timer 2)
T1RUN: Operation of 8-bit timer (timer 1)
T0RUN: Operation of 8-bit timer (timer 0)

Note: Bit6 of TRUN is read as “1”.

System Clock Control Register

SYSCR0
(006EH)

	7	6	5	4	3	2	1	0
Bit symbol	–	–	–	–	–	–	PRCK1	PRCK0
Read/Write	R/W							
After reset	1	0	1	0	0	0	0	0
Function	(Note) Always write “1” (This bit is read as “1”).	(Note) Always write “0” (This bit is read as “0”).	(Note) Always write “1” (This bit is read as “1”).	(Note) Always write “0” (This bit is read as “0”).	(Note) Always write “0” (This bit is read as “0”).	(Note) Always write “0” (This bit is read as “0”).	Select prescaler clock 00: f_{FPH} 01: (Reserved) 10: fc/16 11: (Reserved)	

Select gear value of high frequency

00	f_{FPH}
01	(Reserved)
10	fc/16
11	(Reserved)

Figure 3.8.8 16-Bit Timer/Event Counter Related Registers (6/6)

(1) Prescaler

There are 9-bit prescaler and prescaler clock selection registers to generate input clock for 8-bit timer 0, 1, 2, and 3, 16-bit timers 4 and 5 and serial interfaces 0 and 1.

Figure 3.8.9 shows the block diagram. Table 3.8.1 shows prescaler clock resolution into 8- and 16-bit timers.

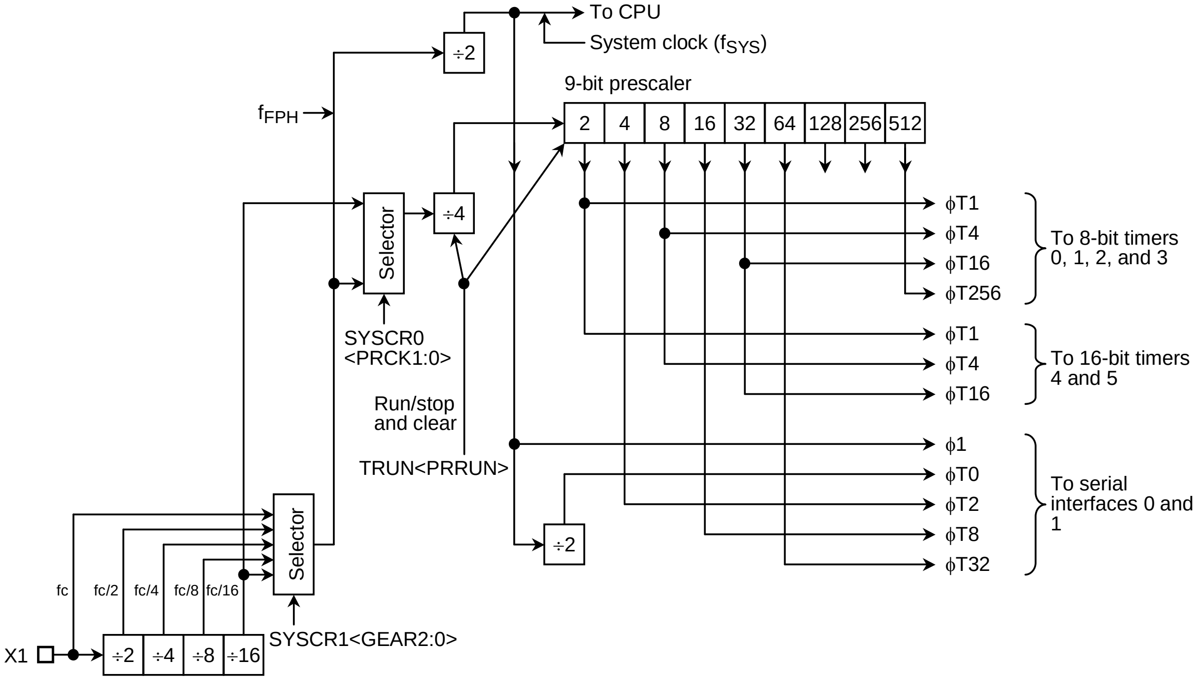


Figure 3.8.9 The Block Diagram of Prescaler

Table 3.8.1 Prescaler Clock Resalation to 8- and 16-Bit Timer

at fc = 20 MHz

Select Prescaler Clock <PRCK1:0>	Gear Value <GEAR2:0>	Prescaler Clock Resolution			
		ϕT1	ϕT4	ϕT16	ϕT256
00 (f$_{FPH}$)	000 (fc)	$fc/2^3$ (0.4 µs)	$fc/2^5$ (1.6 µs)	$fc/2^7$ (6.4 µs)	$fc/2^{11}$ (102.4 µs)
	001 (fc/2)	$fc/2^4$ (0.8 µs)	$fc/2^6$ (3.2 µs)	$fc/2^8$ (12.8 µs)	$fc/2^{12}$ (204.8 µs)
	010 (fc/4)	$fc/2^5$ (1.6 µs)	$fc/2^7$ (6.4 µs)	$fc/2^9$ (25.6 µs)	$fc/2^{13}$ (409.6 µs)
	011 (fc/8)	$fc/2^6$ (3.2 µs)	$fc/2^8$ (12.8 µs)	$fc/2^{10}$ (51.2 µs)	$fc/2^{14}$ (819.2 µs)
	100 (fc/16)	$fc/2^7$ (6.4 µs)	$fc/2^9$ (25.6 µs)	$fc/2^{11}$ (102.4 µs)	$fc/2^{15}$ (1.6384 ms)
10 (fc/16 clock)	XXX	$fc/2^7$ (6.4 µs)	$fc/2^9$ (25.6 µs)	$fc/2^{11}$ (102.4 µs)	$fc/2^{15}$ (1.6384 ms)

XXX: Don’t care

← 16-bit timer →

← 8-bit timer →

The clock selected among f_{FPH} clock and fc/16 clock is divided by 4 and input to this prescaler. This is selected by prescaler clock selection register SYSCR0<PRCK1:0>.

Resetting sets <PRCK1:0> to “00”, therefore f_{FPH}/4 clock is input.

The 16-bit Timers 4 and 5 selects between 3 clock inputs: ϕT1, ϕT4, and ϕT16 among the prescaler outputs.

This prescaler can be run or stopped by the timer operation control register TRUN<PRRUN>. Counting starts when <PRRUN> is set to “1”, while the prescaler is cleared to zero and stops operation when <PRRUN> is cleared to “0”.

When the IDLE1 mode (operates only oscillator) is used, clear TRUN<PRRUN> to “0” to stop this prescaler before “HALT” instruction is executed.

(2) Up counter

UC4 is a 16-bit binary counter which counts up according to the input clock specified by T4MOD<T4CLK1:0> register.

As the input clock, one of the internal clocks ϕT1, ϕT4, and ϕT16 from 9-bit prescaler (also used for 8-bit timer), and external clock from TI4 pin (also used as P42/INT4 pin) can be selected. When reset, it will be initialized to <T4CLK1:0> = 00 to select TI4 input mode. Counting or stop and clear of the counter is controlled by timer operation control register TRUN<T4RUN>.

When clearing is enabled, up counter UC4 will be cleared to zero each time it coincides matches the timer register TREG5. The “clear enable/disable” is set by T4MOD<CLE>.

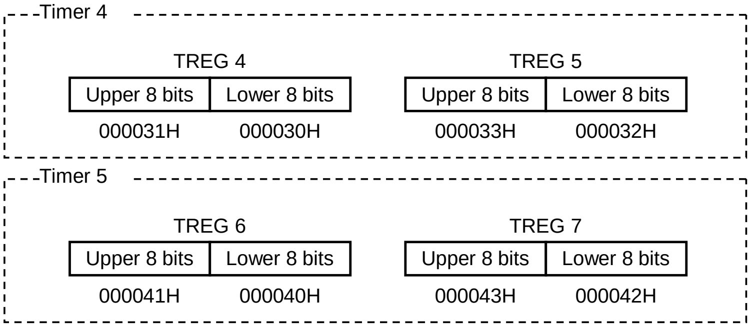
If clearing is disabled, the counter operates as a free-running counter.

A timer overflow interrupt (INTTO4) is generated when UC4 overflow occurs.

(3) Timer registers

These two 16-bit registers are used to set the interval time. When the value of up counter UC4 matches the set value of this timer register, the comparator match detect signal will be active.

Setting data for both upper and lower timer registers (TREG4 and TREG5) is always needed. For example, either using 2-byte date transfer instruction or using 1 byte date transfer instruction twice for lower 8 bits and upper 8 bits in order.



TREG4 to TREG7 are write-only registers, so these registers can not be read by software.

TREG4 timer register is of double buffer structure, which is paired with register buffer. The timer control register T45CR<DB4EN> controls whether the double buffer structure should be enabled or disabled: disabled when <DB4EN> = 0, while enabled when <DB4EN> = 1.

When the double buffer is enabled, the timing to transfer data from the register buffer to the timer register is at the match between the up counter (UC4) and timer register TREG5.

After reset, TREG4 and TREG5 are undefined. To use the 16-bit timer after reset, data should be written beforehand.

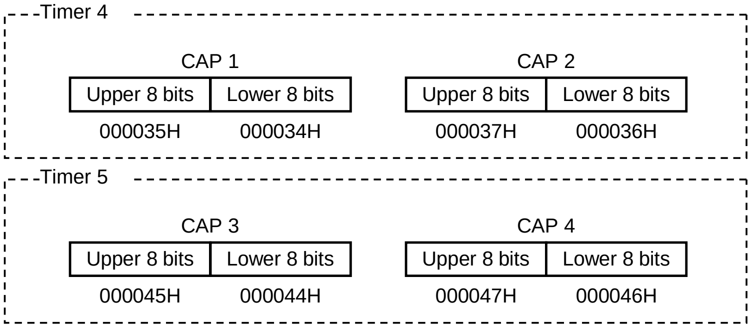
When reset, it will be initialized to <DB4EN> = 0, whereby the double buffer is disabled. To use the double buffer, write data in the timer register, set <DB4EN> = 1, and then write the following data in the register buffer.

TREG4 and register buffer are allocated to the same memory addresses 000030H/000031H. When <DB4EN> = 0, same value will be written in both the timer register and register buffer. When <DB4EN> = 1, the value is written into only the register buffer. To write the initial-value to the timer register, the register buffer should be disabled.

(4) Capture register

These 16-bit registers are used to latch the values of the up counter.

Data in the capture registers should be read all 16 bits. For example, using a 2-byte data load instruction or two 1 byte data load instruction, from the lower 8 bits followed by the upper 8 bits.



CAP 1 to CAP4 are read-only registers, so these registers cannot be written by software.

(5) Capture input control

This circuit controls the timing to latch the value of up counter UC4 into (CAP1 and CAP2). The latch timing of capture register is controlled by register T4MOD<CAP12M 1:0>.

- When T4MOD<CAP12M1:0> = "00"

  Capture function is disabled. Disable is the default on reset.

- When T4MOD<CAP12M1:0> = "01"

  Data is loaded to CAP1 at the rise edge of TI4 pin (also used as P42/INT4) input, while data is loaded to CAP2 at the rise edge of TI5 pin (also used as P43/INT5) input.

- When T4MOD<CAP12M1:0> = "10"

  Data is loaded to CAP1 at the rise edge of TI4 pin input, while to CAP2 at the fall edge. Only in this setting, interrupt INT4 occurs at fall edge.

- When T4MOD<CAP12M1:0> = "11"

  Data is loaded to CAP1 at the rise edge of timer flip-flop TFF1, while to CAP2 at the fall edge.

Besides, the value of up counter can be loaded to capture registers by software. Whenever "0" is written in T4MOD<CAP1IN> the current value of up counter will be loaded to capture register CAP1. It is necessary to keep the prescaler in RUN mode (TRUN<PRRUN> to be "1").

(6) Comparator

These are 16-bit comparators which compare the up counter UC4 value with the set value of (TREG4, TREG5) to detect the match. When a match is detected, the comparators generate an interrupt (INTTR4, INTTR5) respectively. The up counter UC4 is cleared only when UC4 matches TREG5 (The clearing of up counter UC4 can be disabled by setting T4MOD<CLE> = 0).

(7) Timer flip-flop (TFF4)

This flip-flop is inverted by the match detect signal from the comparators and the latch signals to the capture registers. Disable/enable of inversion can be set for each element by T4FFCR<CAP2T4, CAP1T4, EQ5T4, EQ4T4>. After reset, the value of TFF4 is undefined. TFF4 will be inverted when "00" is written in T4FFCR<TFF4C1:0>. Also it is set to "1" when "01" is written, and set to "0" when "10" is written. The value of TFF4 can be output to the timer output pin TO4 (also used as P44). Timer output should be specified by the function register of Port 4. (See Register for Port 4 in Figure 3.5.10.)

(1) 16-bit timer mode

Generating interrupts at fixed intervals

In this example, the interval time is set in the timer register TREG5 to generate the interrupt INTTR5.

		7	6	5	4	3	2	1	0	
TRUN	←	–	X	–	0	–	–	–	–	Stop timer 4.
INTET54	←	1	1	0	0	1	0	0	0	Enable INTTR5 and sets interrupt level 4. Disable INTTR4.
T4FFCR	←	X	X	0	0	0	0	1	1	Disable trigger.
T4MOD	←	0	0	1	0	0	1	*	*	Select internal clock for input and disable the capture function.
		(* * = 01, 10,11)								
TREG5	←	*	*	*	*	*	*	*	*	Set the interval time (16 bits).
		*	*	*	*	*	*	*	*	
TRUN	←	1	X	–	1	–	–	–	–	Start timer 4.

X: Don’t care, –: No change

(2) 16-bit event counter mode

In 16-bit timer mode as described in above, the timer can be used as an event counter by selecting the external clock (TI4 pin input) as the input clock. To read the value of the counter, first perform “software capture” once and read the captured value.

The counter counts at the rise edge of TI4 pin input.

TI4 pin can also be used as P42/INT4.

Since both timers operate in exactly the same way, timer 4 is used for the purposes of explanation.

		7	6	5	4	3	2	1	0	
TRUN	←	–	X	–	0	–	–	–	–	Stop timer 4.
P4CR	←	–	–	–	–	–	0	–	–	Set P42 to input mode.
INTET54	←	1	1	0	0	1	0	0	0	Enable INTTR5 and sets interrupt level 4, while disables INTTR4.
T4FFCR	←	X	X	0	0	0	0	1	1	Disable trigger.
T4MOD	←	0	0	1	0	0	1	0	0	Select TI4 as the input clock.
TREG5	←	*	*	*	*	*	*	*	*	Set the number of counts (16 bits).
TRUN	←	1	X	–	1	–	–	–	–	Start timer 4.

X: Don’t care, –: No change

When used as an event counter, set the prescaler in RUN mode.
(TRUN<PRRUN> = “1”)

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

Square wave pulse can be generated at any frequency and duty by timer 4. The output pulse may be either low-active or high-active.

The PPG mode is obtained by inversion of the timer flip-flop TFF4 that is to be enabled by the match of the up counter UC4 with the timer register TREG4 or TREG5 and to be output to TO4 (also used as P44). In this mode, the following conditions must be satisfied.

(Set value of TREG4) < (Set value of TREG5)

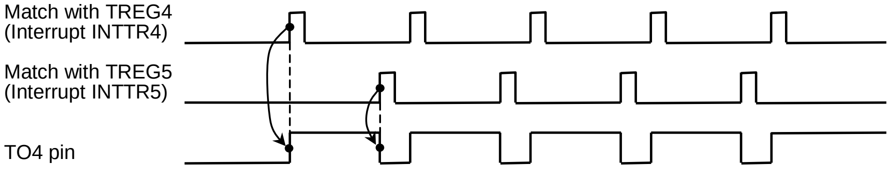


Figure 3.8.10 Programmable Pulse Generation (PPG) Output Waveforms

When the double buffer of TREG4 is enabled in this mode, the value of register buffer 4 will be shifted in TREG4 at match with TREG5. This feature makes easy the handling of low duty waves.

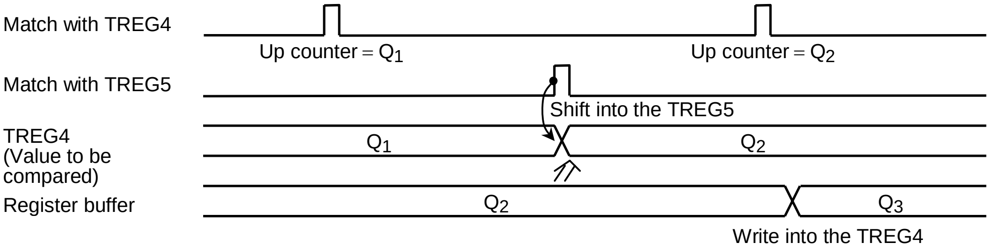


Figure 3.8.11 Operation of Register Buffer

Shows the block diagram of this mode.

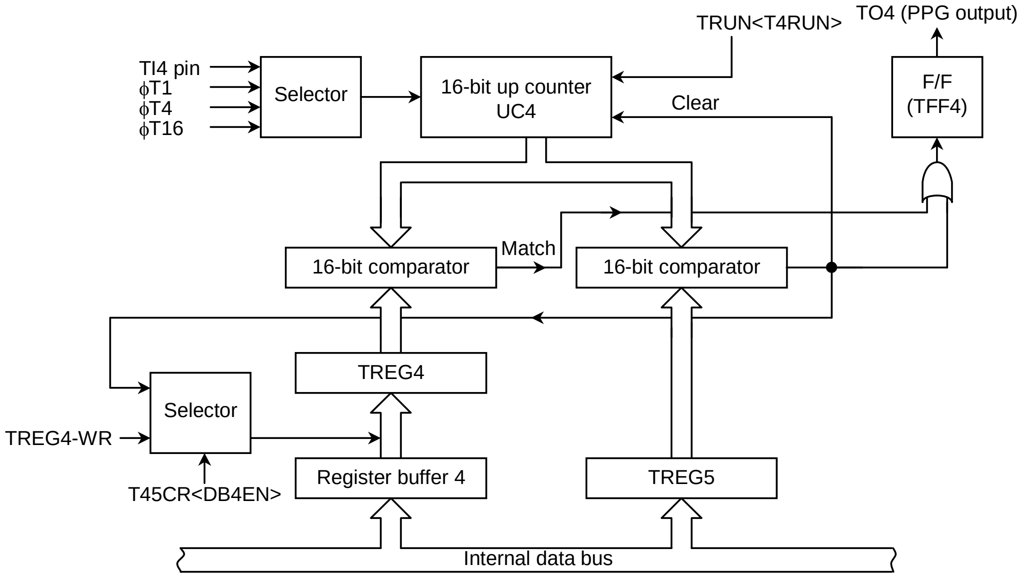


Figure 3.8.12 Block Diagram of 16-Bit PPG Mode

Setting 16-bit programmable pulse generation (PPG) output mode.

		7	6	5	4	3	2	1	0	
T45CR	←	0	X	X	X	X	X	–	0	Double Buffer of TRG4 disable.
TRUN	←	–	X	–	0	–	–	–	–	Stop timer 4.
TREG4	←	*	*	*	*	*	*	*	*	Set the duty (16 bits).
		*	*	*	*	*	*	*	*	
TREG5	←	*	*	*	*	*	*	*	*	Set the cycle (16 bits).
		*	*	*	*	*	*	*	*	
T45CR	←	0	X	X	X	X	X	–	1	Double Buffer of TREG4 enable. (Change the duty and cycle at the interrupt INTTR5)
T4FFCR	←	X	X	0	0	1	1	1	0	Set the mode to invert TFF4 at the match with TREG4/TREG5, and also clear the TFF4 to "0".
T4MOD	←	0	0	1	0	0	1	*	*	Select the internal clock for the input, and disable the capture function.
		(** = 01, 10, 11)								
P4CR	←	–	–	–	1	–	–	–	1	} Assign P44 as TO4.
P4FC	←	–	X	X	1	X	X	–	X	
TRUN	←	1	X	–	1	–	–	–	–	Start timer 4.

X: Don't care, –: No change

(4) Application examples of capture function

Used capture function, they can be applied in many ways, for example:

1. One-shot pulse output from external trigger pulse
2. Frequency measurement
3. Pulse width measurement
4. Time difference measurement

1. One-shot pulse output from external trigger pulse

Set to T4MOD<CAP12M1:0> = 01.

Set the up counter UC4 in free-running mode with the internal input clock, input the external trigger pulse from TI4 pin, and load the value of up counter into capture register CAP1 at the rise edge of the TI4 pin.

When the interrupt INT4 is generated at the rise edge of TI4 input, set the CAP1 value (c) plus a delay time (d) to TREG4 (= c + d), and set the above set value (c + d) plus a one-shot pulse width (p) to TREG5 (= c + d + p). When the interrupt INT4 occurs the T4FFCR<EQ5T4, EQ4T4>register should be set “11” and that the TFF4 inversion is enabled only when the up counter value matches TREG4 or TREG5. When interrupt INTTR5 occurs, this inversion will be disabled after one-shot pulse is output.

The (c), (d), and (p) correspond to c, d, and p in Figure 3.8.13.

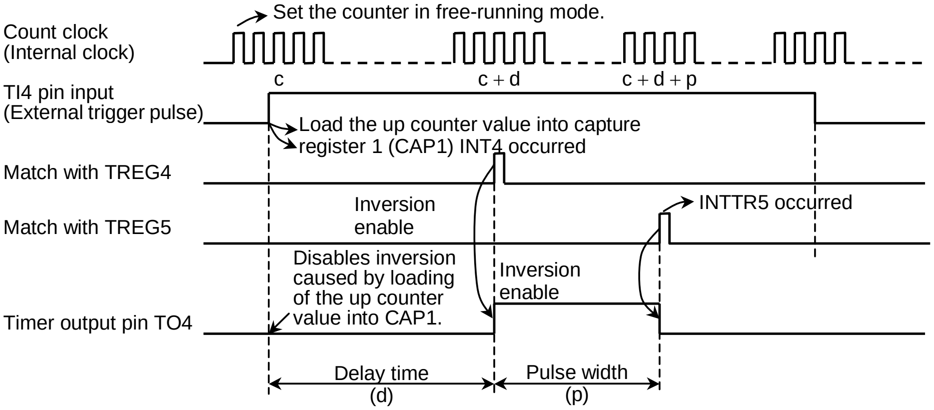


Figure 3.8.13 One-shot Pulse Output (with Delay)

Setting example: To output 2 ms one-shot pulse with 3 ms delay to the external trigger pulse to TI4 pin

* Clock Condition
Clock gear: 1 (fc)
Prescaler clock: f_{FPH}

Main setting

T4MOD	← – – 1 0 1 0 0 1	Keep counting (Free running). Count with φT1. Load the up counter value into CAP1 at the rise edge of TI4 pin input.
T4FFCR	← X X 0 0 0 0 1 0	Clear TFF4 to zero. Disable TFF4 inversion.
P4CR	← – – – 1 – – – 1	Select P44 as the TO4 pin.
P4FC	← – X X 1 X X – X	
INTE45	← – – – – 1 1 0 0	Enable INT4, and disable INTTR4 and INTTR5.
INTET54	← 1 0 0 0 1 0 0 0	
TRUN	← 1 X – 1 – – – –	Start timer 4.

Setting of INT4

TREG4	← CAP1 + 3 ms/φT1	
TREG5	← TREG4 + 2 ms/φT1	
T4FFCR	← X X – – 1 1 – –	Enable TFF4 inversion when the up counter value matches TREG4 or TREG5.
INTET54	← 1 1 0 0 – – – –	Enable INTTR5.

Setting of INTTR5

T4FFCR	← X X – – 0 0 – –	Disable TFF4 inversion when the up counter value matches TREG4 or TREG5.
INTET54	← 1 0 0 0 – – – –	Disable INTTR5.

X: Don't care, –: No change

When delay time is unnecessary, invert timer flip-flop TFF4 when the up counter value is loaded into capture register 1 (CAP1), and set the CAP1 value (c) plus the one-shot pulse width (p) to TREG5 when the interrupt INT4 occurs. The TFF4 inversion should be enabled when the up counter (UC4) value matches TREG5, and disabled when generating the interrupt INTTR5.

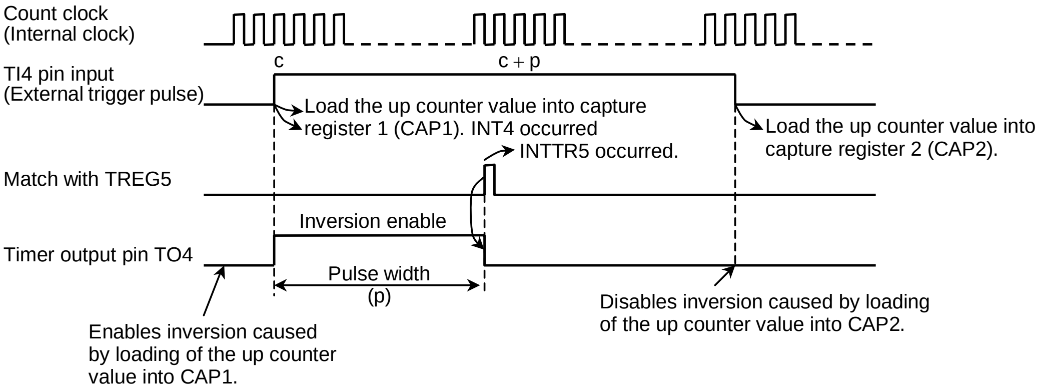


Figure 3.8.14 One-Shot Pulse Output (without Delay)

2. Frequency measurement

The frequency of the external clock can be measured in this mode. The clock is input through the TI4 pin, and its frequency is measured by the 8-bit timers (Timer 0 and Timer 1) and the 16-bit timer/event counter (Timer 4).

The TI4 pin input should be selected for the input clock of Timer 4. The value of the up counter is loaded into the capture register CAP1 at the rise edge of the timer flip-flop TFF1 of 8-bit timers (Timer 0 and Timer 1), and into CAP2 at its fall edge.

The frequency is calculated by the difference between the loaded values in CAP1 and CAP2 when the interrupt (INTT0 or INTT1) is generated by either 8-bit timer.

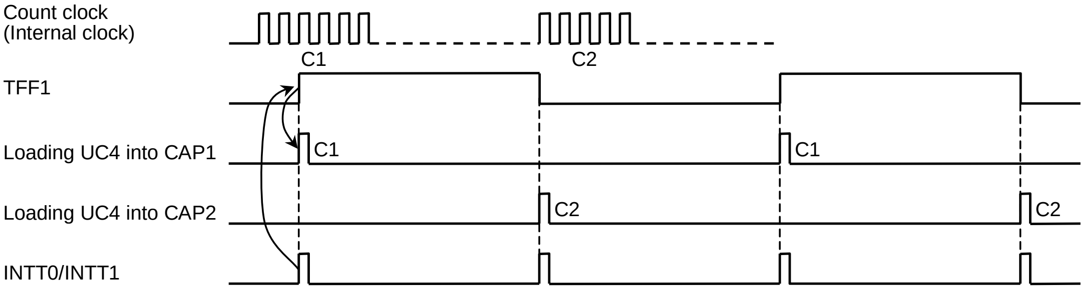


Figure 3.8.15 Frequency Measurement

For example, if the value for the level “1” width of TFF1 of the 8-bit timer is set to 0.5 s. and the difference between CAP1 and CAP2 is 100, the frequency will be 100 ÷ 0.5 [s] = 200 [Hz].

3. Pulse width measurement

This mode allows to measure the "H" level width of an external pulse. While keeping the 16-bit timer/event counter counting (free-running) with the internal clock input, the external pulse is input through the TI4 pin. Then the capture function is used to load the UC4 values into CAP1 and CAP2 at the rising edge and falling edge of the external trigger pulse respectively. The interrupt INT4 occurs at the falling edge of TI4.

The pulse width is obtained from the difference between the values of CAP1 and CAP2 and the internal clock cycle.

For example, if the internal clock is 0.8 microseconds and the difference between CAP1 and CAP2 is 100, the pulse width will be 100 × 0.8 μs = 80 μs.

Additionally, the pulse width which is over the UC4 maximum count time specified by the clock source can be measured by changing software.

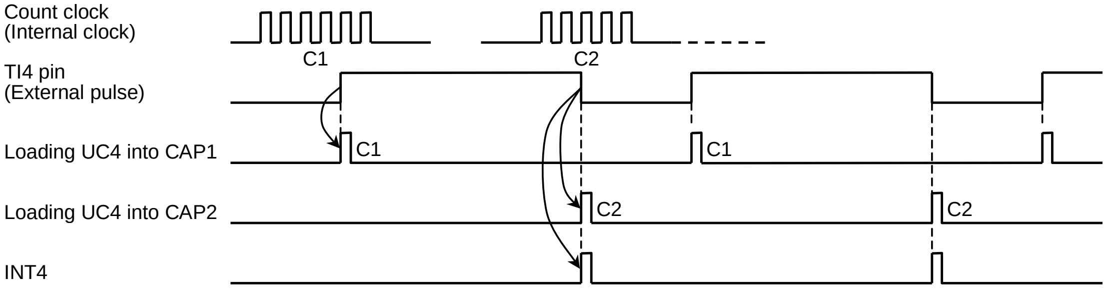


Figure 3.8.16 Pulse Width Measurement

Note: Only in this pulse width measuring mode (T4MOD<CAP12M1:0> = 10), external interrupt INT4 occurs at the falling edge of TI4 pin input. In other modes, it occurs at the rising edge.

The width of "L" level can be measured by multiplying the difference between the first C2 and the second C1 at the second INT4 interrupt and the internal clock cycle together. See Figure 3.8.17 Time Difference Measurement.

4. Time difference measurement

This mode is used to measure the difference in time between the rising edges of external pulses input through TI4 and TI5.

Keep the 16-bit timer/event counter (Timer 4) counting (free-running) with the internal clock, and load the UC4 value into CAP1 at the rising edge of the input pulse to TI4. Then the interrupt INT4 is generated.

Similarly, the UC4 value is loaded into CAP2 at the rising edge of the input pulse to TI5, generating the interrupt INT5.

The time difference between these pulses can be obtained from the difference between the time counts at which loading the up counter value into CAP1 and CAP2 was performed. (= (CAP2 − CAP1) × the internal clock cycle)

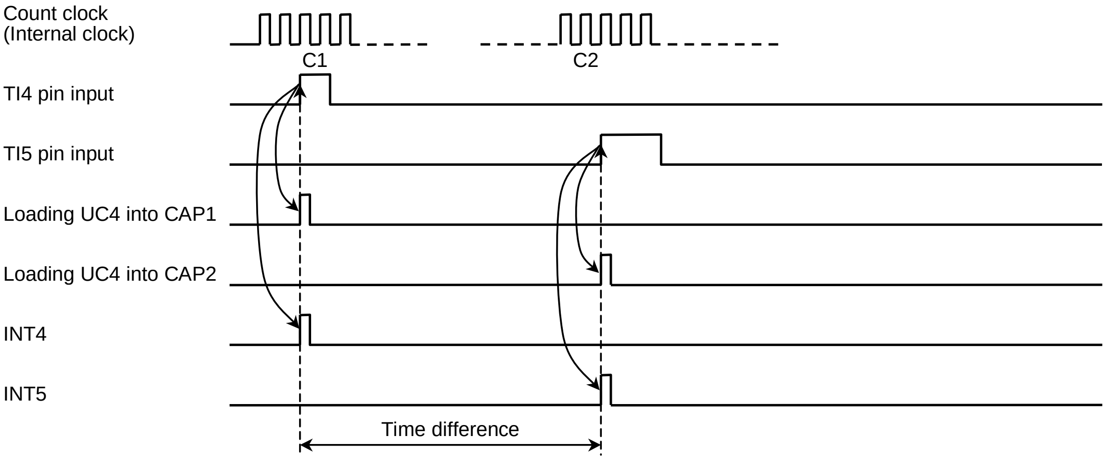


Figure 3.8.17 Time Difference Measurement

3.9 Serial Channel

TMP93CS32 contains 2 serial I/O channels for full duplex asynchronous transmission (UART) as well as for I/O extension.

The serial channel has the following operation modes.

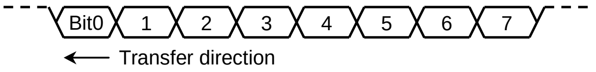
- I/O interface mode (channels 0 and 1) — Mode 0: To transmit and receive I/O data using the synchronizing signal SCLK for extending I/O.
- UART mode (channels 0 and 1)
  - Mode 1: 7-bit data
  - Mode 2: 8-bit data
  - Mode 3: 9-bit data

In mode 1 and mode 2, a parity bit can be added. Mode 3 has wake-up function for making the master controller start slave controllers in serial link.

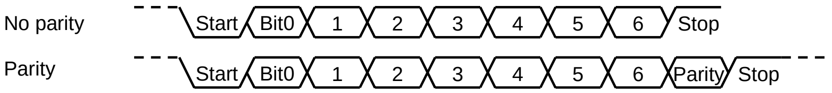
Figure 3.9.1 shows the data format (for one frame) in each mode.

Serial channels 0 and 1 can be used independently.

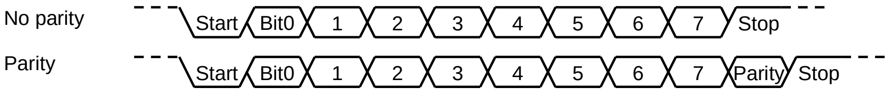
- Mode 0 (I/O interface mode)



- Mode 1 (7-bit UART mode)



- Mode 2 (8-bit UART mode)



- Mode 3 (9-bit UART mode)

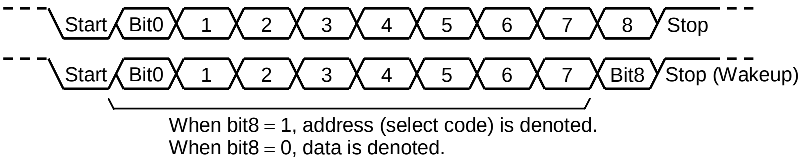


Figure 3.9.1 Data Formats

The serial channel has a buffer register for transmitting and receiving operations, in order to temporarily store transmitted or received data, so that transmitting and receiving operations can be done independently (Full duplex).

However, in I/O interface mode, SCLK (Serial clock) pin is used for both transmission and receiving, the channel becomes half-duplex.

The receiving data register is of a double buffer structure to prevent the occurrence of overrun error and provides one frame of margin before CPU reads the received data. The receiving data register stores the already received data while the buffer register receives the next frame data.

By using $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ (There is no $\overline{\text{RTS}}$ pin, so any 1 port must be controlled by software), it is possible to halt data send until the CPU finishes reading receive data every time a frame is received (Handshake function).

In the UART mode, a check function is added not to start the receiving operation by error start bits due to noise. The channel starts receiving data only when the start bit is detected to be normal at least twice in three samplings.

When the transmission buffer becomes empty and requests the CPU to send the next transmission data, or when data is stored in the receiving data register and the CPU is requested to read the data, INTTX or INTRX interrupt occurs. Besides, if an overrun error, parity error, or framing error occurs during receiving operation, flag SC0CR/SC1CR<OERR, PERR, FERR> will be set.

The serial channel 0/1 includes a special baud rate generator, which can set any baud rate by dividing the frequency of 4 clocks (ϕT0, ϕT2, ϕT8, and ϕT32) from the internal prescaler (shared by 8-bit/16-bit timer) by the value 1 to 16. In addition, serial channel 0/1 can operated by using external input clock (SCLK 0/1).

In I/O interface mode, it is possible to input synchronous signals as well as to transmit or receive data by external clock.

3.9.1 Control registers

The serial channel 0 is controlled by 3 control registers SC0CR, SC0MOD, and BR0CR. Transmitted and received data are stored in register SC0BUF.

Serial Channel 0 Mode Control Register

SC0MOD
(0052H)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE0	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	Undefined	0	0	0	0	0	0	0
Function	Transfer data bit8	Hand shake function 0: $\overline{\text{CTS0}}$ disable 1: $\overline{\text{CTS0}}$ enable	Receiving function 0: Receive disable 1: Receive enable	Wakeup function 0: Disable 1: Enable	Serial transmission mode 00: I/O interface mode 01: 7-bit UART 10: 8-bit UART 11: 9-bit UART		Serial transmission clock (UART) 00: TO2 trigger 01: Baud rate generator 0 10: Internal clock ϕ1 11: External clock (SCLK0 input)	

Serial transmission clock (UART)

00	Timer 2 match detect signal
01	Baud rate generator 0
10	Internal clock ϕ1 (f_{SYS})
11	External clock (SCLK0 input)

Serial transmission mode

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

Wakeup function
(Don’t care in the modes other than 9-bit UART)

0	Disable
1	Enable

Receiving function

0	Receive Disable
1	Receive Enable

Hand shake function ($\overline{\text{CTS0}}$ pin)

0	Disable (Always transferable)
1	Enable

Transmission data bit8

8-bit UART mode (Parity)	Stores transmission parity bit
9-bit UART mode	Stores transmission data bit8

Figure 3.9.2 Serial Channel 0 Related Register (1/7)

Serial Channel 0 Control Register

SC0CR
(0051H)

	7	6	5	4	3	2	1	0
Bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (cleared to Zero when read)			R/W	
After reset	Undefined	0	0	0	0	0	0	0
Function	Received data bit8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: Error			0: SCLK0 (↑) 1: SCLK0 (↓)	0: Baud rate generator 0 1: SCLK0 pin input
				Overrun	Parity	Framing		

Select I/O interface input clock

0	Baud rate generater 0 (Note 1)
1	SCLK0 pin input

Edge selection in SCLK0 pin input mode

0	Transmits and receives data at rise edge of SCLK0 (↑)
1	Transmits and receives data at fall edge of SCLK0 (↓)

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

Enable parity addition

0	Disable
1	Enable

Addition/check of even parity

0	Odd parity
1	Even parity

Receving data bit8

8-bit UART mode (parity)	Stores received parity bit
9-bit UART mode	Stores received data bit8

Note 1: To use baud rate generator, set TRUN<PRRUN> to “1”, putting the prescaler in RUN mode.

Note 2: As all error flags are cleared after reading, do not test only a single bit with a bit-testing instruction.

Figure 3.9.3 Serial Channel 0 Related Register (2/7)

Baud Rate Generator 0 Control Register

BR0CR
(0053H)

	7	6	5	4	3	2	1	0
Bit symbol	–		BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write	R/W		R/W					
After reset	0		0	0	0	0	0	0
Function	(Note) Always write “0”.		00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Setting of the divided frequency			

Setting of the divided frequency of baud rate generator 0

0000	16 divisions
0001 to 1111	1 division (Not divided) (Note 2) to 15 divisions

Selecting the input clock of baud rate generator

00	Internal clock ϕT0
01	Internal clock ϕT2
10	Internal clock ϕT8
11	Internal clock ϕT32

Note 1: To use baud rate generator, set TRUN<PRRUN> to “1”, putting the prescaler in RUN mode.

Note 2: “1 division” of baud rate generator can be used only UART mode. Do not set it in I/O interface mode.

Note 3: Bit6 of BR0CR is read as “1”.

Note 4: Don’t read from or write to BR0CR register during sending or receiving.

Serial Channel 0 Buffer Register

SC0BUF
(0050H)

	7	6	5	4	3	2	1	0
Bit symbol	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
	TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0
Read/Write	R (Receiving)/W (Transmission)							
After reset	Undefined							

Note: Read-modify-write instruction is prohibited for SC0BUF.

Figure 3.9.4 Serial Channel 0 Related Registers (3/7)

Serial Channel 1 Mode Control Register

SC1MOD
(0056H)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE1	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	Undefined	0	0	0	0	0	0	0
Function	Transfer data bit8	Hand shake 0: $\overline{\text{CTS1}}$ disable 1: $\overline{\text{CTS1}}$ enable	Receiving function 0: Receive disable 1: Receive enable	Wake up function 0: Disable 1: Enable	Serial transmission mode 00: I/O interface mode 01: 7-bit UART 10: 8-bit UART 11: 9-bit UART		Serial transmission clock (UART) 00: TO2 trigger 01: Baud rate generator 1 10: Internal clock ϕ1 11: External clock (SCLK1 input)	

Serial transmission clock (for UART)

00	Timer 2 match detect signal
01	Baud rate generator 1
10	Internal clock ϕ1 (f_{SYS})
11	External clock (SCLK1 input)

Serial transmission mode

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

Wake up function

(Don’t care in the modes other than 9-bit UART)

0	Disable
1	Enable

Receiving control

0	Receive disable
1	Receive enable

Hand shake function ($\overline{\text{CTS1}}$ pin) enable

0	Disable (always transferable)
1	Enable

Transmission data bit8

8-bit UART mode (parity)	Stores transmission parity bit
9-bit UART mode	Stores transmission data bit8

Figure 3.9.5 Serial Channel 1 Related Register (4/7)

Serial Channel 1 Control Register

SC1CR
(0055H)

	7	6	5	4	3	2	1	0
Bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (Cleared to zero when read.)			R/W	
After reset	Undefined	0	0	0	0	0	0	0
Function	Received data bit8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: Error			0: SCLK1 (↑) 1: SCLK1 (↓)	0: Baud rate generator 1 1: SCLK1 pin input
				Overrun	Parity	Framing		

Select I/O interface input clock

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

Edge selection in SCLK1 pin input mode

0	Transmits and receives data at rise edge of SCLK1 (↑)
1	Transmits and receives data at fall edge of SCLK1 (↓)

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

Enable parity addition

0	Disable
1	Enable

Addition/check of even parity

0	Odd parity
1	Even parity

Receiving data bit8

8-bit UART mode (parity)	Stores transmission parity bit
9-bit UART mode	Stores transmission data bit8

Note 1: To use baud rate generator, set TRUN<PRRUN> to “1”, putting the prescaler in RUN mode.

Note 2: As all error flags are cleared after reading, do not test only a single bit with a bit-testing instruction.

Figure 3.9.6 Serial Channel 1 Related Register (5/7)

Baud Rate Generator 1 Control Register

		7	6	5	4	3	2	1	0
BR1CR (0057H)	Bit symbol	–		BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
	Read/Write	R/W		R/W					
	After reset	0		0	0	0	0	0	0
	Function	(Note) Always write “0”.		00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Setting of the Divided frequency			

Setting of the divided frequency of baud rate generator 1

0000	16 divisions
0001 to 1111	1 division (Not divided) (Note 2) to 15 divisions

Selecting the input clock of baud rate generator

00	Internal clock ϕT0
01	Internal clock ϕT2
10	Internal clock ϕT8
11	Internal clock ϕT32

Note 1: To use baud rate generator, set TRUN<PRRUN>to “1”, putting the prescaler in RUN mode.

Note 2: “1 division” of baud rate generator can be used only UART mode. Do not set it in I/O interface mode.

Note 3: Bit6 of BR1CR is read as “1”.

Note 4: Don’t read from or write to BR1CR register during sending or receiving.

Serial Channel 1 Buffer Register

		7	6	5	4	3	2	1	0
SC1BUF (0054H)	Bit symbol	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0
		TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0
	Read/Write	R (Receiving)/W (Transmission)							
	After reset	Undefined							

Note: Read-modify-write instruction is prohibited for SC1BUF.

Figure 3.9.7 Serial Channel 1 Related Registers (6/7)

Port 6 Function Register

P6FC
(0016H)

	7	6	5	4	3	2	1	0
Bit symbol			P65F		P63F	P62F		P60F
Read/Write			W		W			W
After reset			0		0	0		0
Function			0: Port 1: SCLK1		0: Port 1: TXD1	0: Port 1: SCLK0		0: Port 1: TXD0

Setting TXD0 output of P60

0	Port
1	TXD0 output (Channel 0)

Setting SCLK0 output of P62

0	Port
1	SCLK0 output (Channel 0)

Setting TXD1 output of P63

0	Port
1	TXD1 output (Channel 1)

Setting SCLK1 output of P65

0	Port
1	SCLK1 output (Channel 1)

Note: Read-modify-write instruction is prohibited for P6FC.

Open Drain Enable Register

ODE
(0058H)

	7	6	5	4	3	2	1	0
Bit symbol					–	–	ODE63	ODE60
Read/Write					R/W			
After reset					0	0	0	0
Function					Always write “0”. (This bit is read as “0”.)		P63 0: CMOS 1: Open drain	P60 0: CMOS 1: Open drain

Setting P60 as open-drain output

0	CMOS output
1	Open-drain output

Setting P63 as open-drain output

0	CMOS output
1	Open-drain output

Note: Bit7 to 4 of ODE are read as “1”.

Figure 3.9.8 Serial Channel Related Registers (7/7)

3.9.2 Configuration

Figure 3.9.9 shows the block diagram of the serial channel 0.

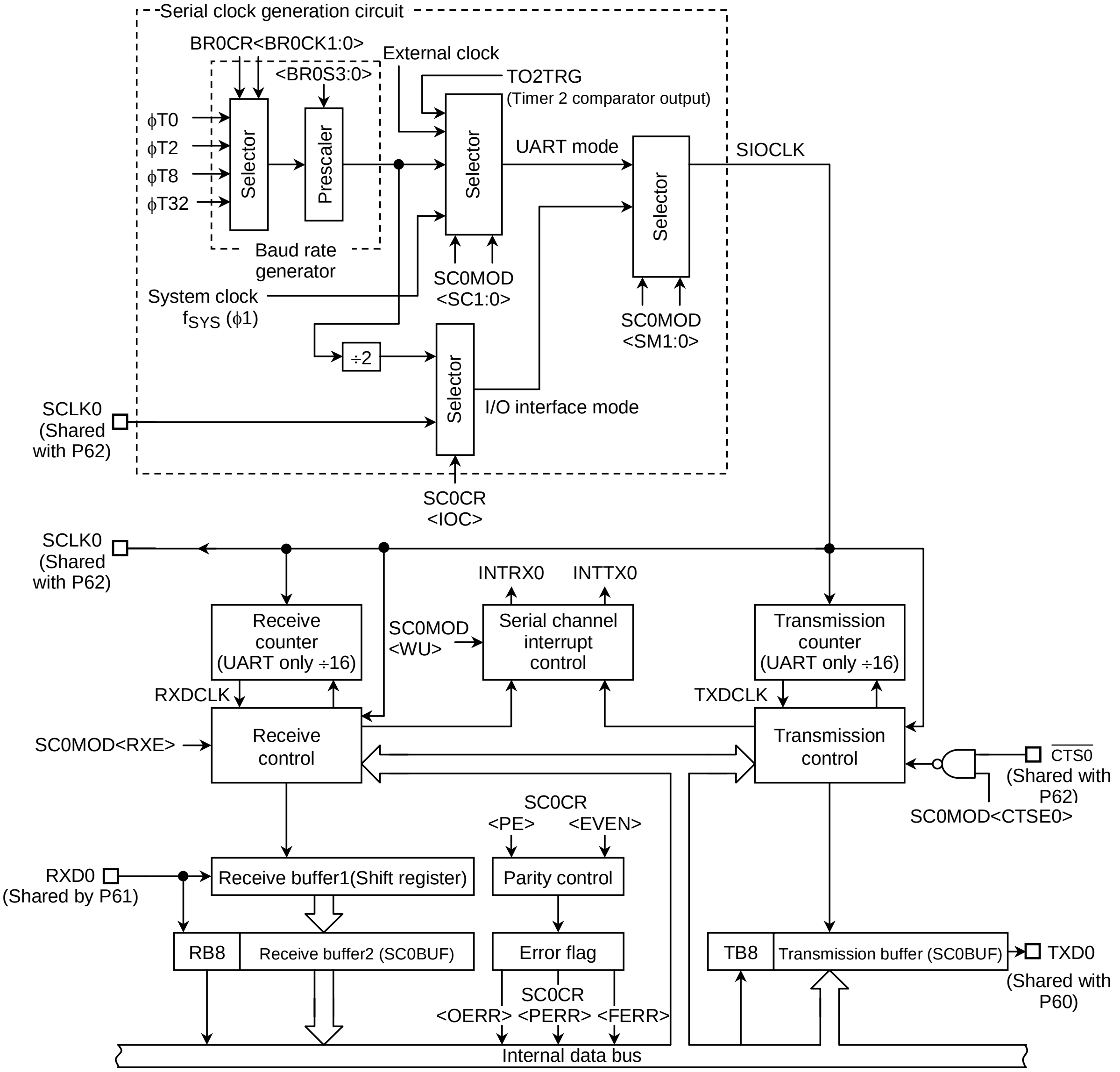


Figure 3.9.9 Block Diagram of the Serial Channel 0

Figure 3.9.10 shows the block diagram of the serial channel 1.

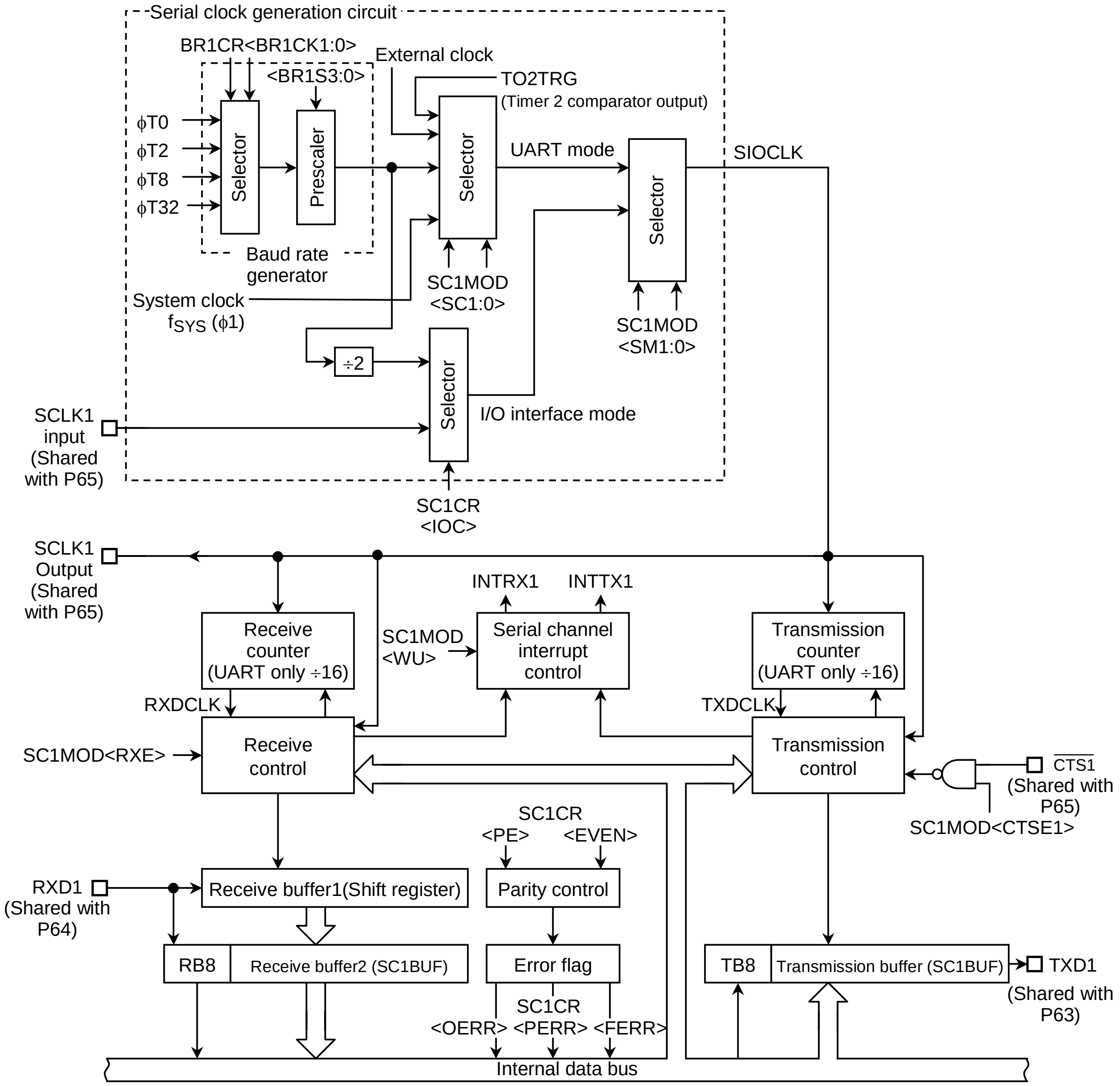


Figure 3.9.10 Block Diagram of the Serial Channel 1

Serial channel 0 and 1 can be used independently. All serial channels operate in the same manner, and thus only operation of serial channel 0 will be explained below.

1. Prescaler

There are 9-bit prescaler and prescaler clock selection registers to generate input clock for 8-bit timers 0, 1, 2, 3 and 16-bits timers 4, 5 and Serial interface 0, 1.

Figure 3.9.11 shows the block diagram. Table 3.9.1 shows prescaler clock resolution into the baud rate generator.

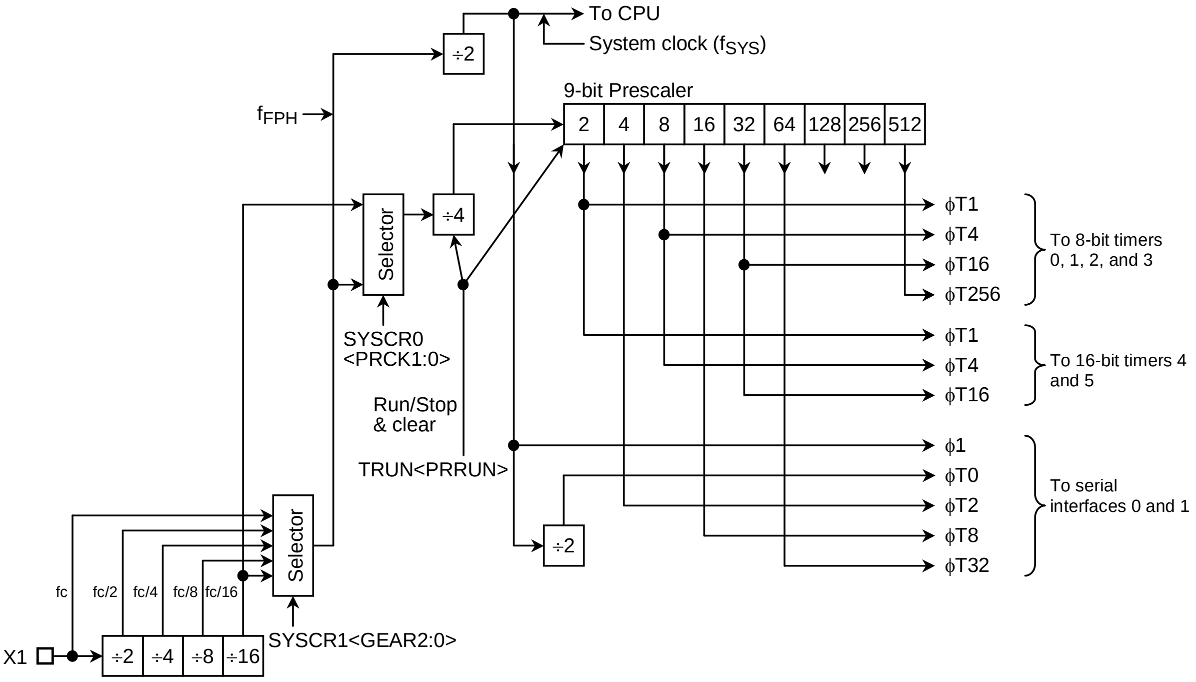


Figure 3.9.11 The Block Diagram of Prescaler

Table 3.9.1 Prescaler Clock Resolution to Baud Rate Generator

at fc = 20 MHz

Select Prescaler Clock <PRCK1:0>	Gear value <GEAR2:0>	Prescaler Output Clock Resolution			
		ϕT0	ϕT2	ϕT8	ϕT32
00 (f_{FPH})	000 (fc)	fc/2^{2} (0.2 μs)	fc/2^{4} (0.8 μs)	fc/2^{6} (3.2 μs)	fc/2^{8} (12.8 μs)
	001 (fc/2)	fc/2^{3} (0.4 μs)	fc/2^{5} (1.6 μs)	fc/2^{7} (6.4 μs)	fc/2^{9} (25.6 μs)
	010 (fc/4)	fc/2^{4} (0.8 μs)	fc/2^{6} (3.2 μs)	fc/2^{8} (12.8 μs)	fc/2^{10} (51.2 μs)
	011 (fc/8)	fc/2^{5} (1.6 μs)	fc/2^{7} (6.4 μs)	fc/2^{9} (25.6 μs)	fc/2^{11} (102.4 μs)
	100 (fc/16)	fc/2^{6} (3.2 μs)	fc/2^{8} (12.8 μs)	fc/2^{10} (51.2 μs)	fc/2^{12} (204.8 μs)
10 (fc/16 clock)	XXX	–	fc/2^{8} (12.8 μs)	fc/2^{10} (51.2 μs)	fc/2^{12} (204.8 μs)

XXX: Don't care –: Can not use

The clock selected among f_{FPH} clock and fc/16 clock is divided by 4 and input to this prescaler. This is selected by prescaler clock selection register SYSCR0<PRCK1:0>

Resetting sets <PRCK1:0> to “00” and selects the f_{FPH} clock input divided by 4.

Baud rate generator selects between 4 clock inputs ϕT0, ϕT2, ϕT8, and ϕT32 among the prescaler outputs.

The prescaler can be run or stopped by the timer operation control register TRUN<PRRUN>. Counting starts when <PRRUN> is set to “1”, while the prescaler is cleared to zero and stops operation when <PRRUN> is cleared to “0”.

When the IDLE1 mode (operates only oscillator) is used, clear TRUN<PRRUN> to “0” to reduce the power consumption of this prescaler before “HALT” instruction is executed.

2. Baud rate generator

Baud rate generator comprises a circuit that generates transmission and receiving clocks to determine the transfer rate of the serial channel.

The input clock to the baud rate generator, ϕT0, ϕT2, ϕT8, or ϕT32 is generated by the 9-bit prescaler which is shared by the timers. One of these input clocks is selected by the baud rate generator control register BR0CR<BR0CK1:0>.

The baud rate generator includes a 4-bit frequency divider, which divides frequency by 1 to 16 values to determine the transfer rate.

How to calculate a transfer rate when the baud rate generator is used is explained below.

- UART mode

$$\text{Baud rate} = \frac{\text{Input clock of baud rate generator}}{\text{Frequency divisor of baud rate generator}} \div 16$$

- I/O interface mode

$$\text{Baud rate} = \frac{\text{Input clock of baud rate generator}}{\text{Frequency divisor of baud rate generator}} \div 2$$

Accordingly, when source clock fc is 12.288 MHz, input clock is ϕT2 (fc/16), and frequency divisor is 5, the transfer rate in UART mode becomes as follows:

* Clock condition

Clock gear: 1 (fc)
Prescaler clock: f_{FPH}

$$\begin{aligned}\text{Baud rate} &= \frac{fc/16}{5} \div 16 \\ &= 12.288 \times 10^6 \div 16 \div 5 \div 16 = 9600 \text{ (bps)}\end{aligned}$$

The maximum baud rate of this baud rate generator is 307.2 kbps.

Table 3.9.2 shows an example of the transfer rate in UART mode.

Also with 8-bit timer 2, the serial channel can get a transfer rate. Table 3.9.3 shows an example of baud rate using timer 2.

Table 3.9.2 Selection of UART Transfer Rate (1) (when baud rate generator is used)

Unit (kbps)

fc [MHz]	Input clock / Frequency divisor	ϕT0 (4/fc)	ϕT2 (16/fc)	ϕT8 (64/fc)	ϕT32 (256/fc)
9.830400	1	153.600	38.400	9.600	2.400
↑	2	76.800	19.200	4.800	1.200
↑	4	38.400	9.600	2.400	0.600
↑	8	19.200	4.800	1.200	0.300
↑	16	9.600	2.400	0.600	0.150
12.288000	5	38.400	9.600	2.400	0.600
↑	10	19.200	4.800	1.200	0.300
14.745600	1	230.400	57.600	14.400	3.600
↑	3	76.800	19.200	4.800	1.200
↑	6	38.400	9.600	2.400	0.600
↑	12	19.200	4.800	1.200	0.300
17.2032	7	38.400	9.600	2.400	0.600
↑	14	19.200	4.800	1.200	0.300
19.6608	2	153.600	38.400	9.600	2.400
↑	4	76.800	19.200	4.800	1.200
↑	8	38.400	9.600	2.400	0.600
↑	16	19.200	4.800	1.200	0.300

Note 1: Transfer rate in I/O interface mode is 8 times faster than the values given in the above table.

Note 2: This table is calculated when fc/1 is selected as a clock gear, and the system clock as a prescaler clock.

Table 3.9.3 Selection of UART Transfer Rate (2)
(when timer 2 (input clock ϕT1) is used)

Unit (kbps)

fc / TREG2	19.6608 MHz	14.7456 MHz	12.288 MHz	12 MHz	9.8304 MHz	8 MHz	6.144 MHz
1H	153.6	115.2	96		76.8	62.5	48
2H	76.8	57.6	48		38.4	31.25	24
3H	51.2	38.4	32	31.25			16
4H	38.4	28.8	24		19.2		12
5H	30.72	23.04	19.2				9.6
8H	19.2	14.4	12		9.6		6
AH	15.36	11.52	9.6				4.8
10H	9.60	7.20	6		4.8		3
14H	7.68	5.76	4.8				2.4

How to calculate the transfer rate (when timer 2 is used):

$$\text{Transfer rate} = \frac{\text{The clock frequency selected by the register SYSCR0<PRCK1:0>}}{\text{TREG2} \times \underset{\uparrow}{8} \times 16}$$

(When Timer 2 (input clock ϕT1) is used.)

Note 1: Timer 2 match detect signal cannot be used as the transfer clock in I/O interface mode.

Note 2: This table is calculated when fc/1 is selected as a clock gear, and f_{FPH} as a prescaler clock.

3. Serial clock generation circuit

   This circuit generates the basic clock for transmitting and receiving data.

   - I/O interface mode

     When in SCLK output mode with the setting of SC0CR<IOC> = "0", the basic clock will be generated by dividing by 2 the output of the baud rate generator described before. When in SCLK input mode with the setting of SC0CR<IOC> = "1", the rising edge or falling edge will be detected according to the setting of SC0CR<SCLKS> register to generate the basic clock.

   - UART mode

     According to the setting of SC0MOD<SC1:0>, the above baud rate generator clock, internal clock $\phi1$ (Max 625 kbps at fc = 20 MHz), the match detect signal from timer 2, or external clock SCLK0 will be selected to generate the basic clock SIOCLK.

4. Receiving counter

   The receiving counter is a 4-bit binary counter used in UART mode and counts up by SIOCLK clock. 16 pulses of SIOCLK are used for receiving 1 bit of data, and the data bit is sampled three times at 7th, 8th, and 9th clock.

   With the three samples, the received data is evaluated by the rule of majority.

   For example, if the sampled data bit is "1", "0", and "1" at 7th, 8th, and 9th clock respectively, the received data is evaluated as "1". The sampled data "0", "0", and "1" is evaluated that the received data is "0".

5. Receiving control

   - I/O interface mode

     When in SCLK0 output mode with the setting of SC0CR<IOC> = "0", RXD0 signal will be sampled at the rising edge of shift clock which is output to SCLK0 pin.

     When in SCLK0 input mode with the setting SC0CR<IOC> = "1" RXD0 signal will be sampled at the rising edge or falling edge of SCLK0 input according to the setting of SC0CR<SCLKS> register.

   - UART mode

     The receiving control has a circuit for detecting the start bit by the rule of majority. When two or more "0" are detected during 3 samples, it is recognized as start bit and the receiving operation is started.

     Data being received are also evaluated by the rule of majority.

6. Receiving buffer

To prevent overrun error, the receiving buffer has a double buffer structure.

Received data are stored one bit by one bit in the receiving buffer 1 (shift register type). When 7 bits or 8 bits of data is stored in the receiving buffer 1, the stored data are transferred to the receiving buffer 2 (SC0BUF), generating an interrupt INTRX0. The CPU reads only receiving buffer 2 (SC0BUF). Even before the CPU reads the receiving buffer 2 (SC0BUF), the received data can be stored in the receiving buffer 1. However, unless the receiving buffer 2 (SC0BUF) is read before all bits of the next data are received by the receiving buffer 1, an overrun error occurs. If an overrun error occurs, the contents of the receiving buffer 1 will be lost, although the contents of the receiving buffer 2 and SC0CR<RB8> is still preserved.

The parity bit added in 8-bit UART mode and the most significant bit (MSB) in 9-bit UART mode are stored in SC0CR<RB8>.

When in 9-bit UART mode, the wake-up function of the slave controllers is enabled by setting SC0MOD<WU> to "1", and interrupt INTRX0 occurs only when SC0CR<RB8> is set to "1".

7. Transmission counter

Transmission counter is a 4-bit binary counter which is used in UART mode, counts by SIOCLK clock, and generating TXDCLK every 16 clock pulses.

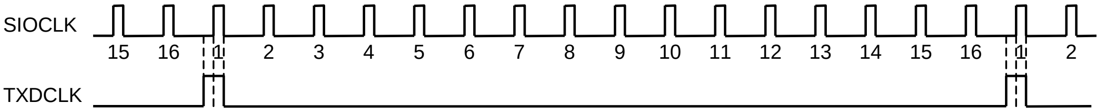


Figure 3.9.12 Generation of Transmission Clock

8. Transmission controller

- I/O interface mode

  In SCLK0 output mode with the setting of SC0CR<IOC> = "0", the data in the transmission buffer are output bit by bit to TXD0 pin at the rising edge of shift clock which is output from SCLK0 pin.

  In SCLK0 input mode with the setting of SC0CR<IOC> = "1", the data in the transmission buffer are output bit by bit to TXD0 pin at the rising edge or falling edge of SCLK0 input according to the setting of SC0CR<SCLKS> register.

- UART mode

  When transmission data are written in the transmission buffer sent from the CPU, transmission starts at the rising edge of the next TXDCLK, generating a transmission shift clock TXDSFT.

Handshake Function

The serial channels use the $\overline{\text{CTS0}}$ pin to transmit data in units of frames, thus preventing an overrun error. Use SC0MOD<CTSE0> to enable or disable the handshake function.

When $\overline{\text{CTS0}}$ goes high, data transmission is halted after the completion of the current transmission and is not restarted until $\overline{\text{CTS0}}$ returns to low. An INTTX0 interrupt is generated to request the CPU for the next data to transmit. When the CPU write 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 $\overline{\text{RTS}}$ function to high (in the receive interrupt routine) after data receive, and requests the transmitter to temporarily halt transmission.

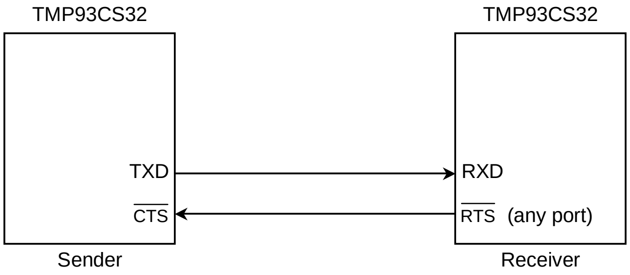
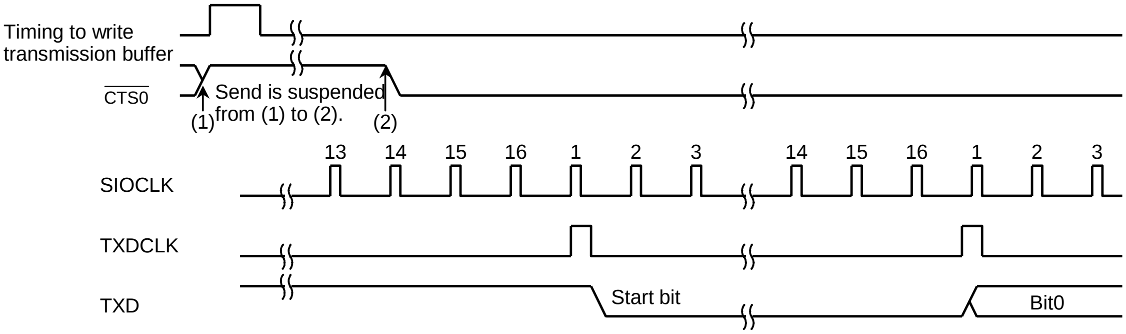


Figure 3.9.13 Handshake Function



Note 1: If the $\overline{\text{CTS}}$ signal rises during transmission, the next data is not sent after the completion of the current transmission.

Note 2: Transmission starts at the first TXDCLK clock fall after the $\overline{\text{CTS}}$ signal falls.

Figure 3.9.14 Timing of $\overline{\text{CTS}}$ (Clear to send)

9. Transmission buffer

   Transmission buffer (SC0BUF) shifts out and sends the transmission data written from the CPU from the least significant bit (LSB) in order. When all bits are shifted out, the transmission buffer becomes empty and generates INTTX0 interrupt.

10. Parity control circuit

    When serial channel control register SC0CR<PE>is set to "1", it is possible to transmit and receive data with parity. However, parity can be added only in 7-bit UART or 8-bit UART mode. With SC0CR<EVEN> register, even (odd) parity can be selected.

    For transmission, parity is automatically generated according to the data written in the transmission buffer SC0BUF, and data are transmitted after being stored in SC0BUF<TB7> when in 7-bit UART mode while in SC0MOD<TB8> when in 8-bit UART mode. <PE> and <EVEN> must be set before transmission data are written in the transmission buffer.

    For receiving, data are shifted in the receiving buffer 1, and parity is added after the data are transferred in the receiving buffer 2 (SC0BUF), and then compared with SC0BUF<RB7> when in 7-bit UART mode and with SC0MOD<RB8> when in 8-bit UART mode. If they are not equal, a parity error occurs, and SC0CR<PERR> flag is set.

11. Error flag

    Three error flags are provided to increase the reliability of receiving data.

    - Overrun error <OERR>

      If all bits of the next data are received in receiving buffer 1 while valid data are stored in receiving buffer 2 (SC0BUF), an overrun error will occur.

    - Parity error <PERR>

      The parity generated for the data shifted in receiving buffer 2 (SC0BUF) is compared with the parity bit received from RXD pin. If they are not equal, a parity error occurs.

    - Framing error <FERR>

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

12. Signal generation timing

1) In I/O Interface mode

Timing for send interrupt generation	SCLK0 output mode	Immediately after rise of last SCLK0 signal (See Figure 3.9.17.)
	SCLK0 input mode	Immediately after rise (Rising mode) or fall (Falling mode) of last SCLK0 signal (See Figure 3.9.18.)
Timing for receive interrupt generation	SCLK0 output mode	Immediately after final SCLK0 (When received data are transferred to receive buffer 2 (SC0BUF))(See Figure 3.9.19.)
	SCLK0 input mode	Immediately after final SCLK0 (When received data are transferred to receive buffer 2 (SC0BUF))(See Figure 3.9.20.)

2) In UART mode

Receive

Mode	9-Bit	8-Bit + Parity	8-Bit, 7-Bit + Parity, 7-Bit
Timing for interrupt generation	Center of last bit (Bit8)	Center of last bit(parity bit)	Center of stop bit
Timing for framing error generation	Center of stop bit	Center of stop bit	Center of stop bit
Timing for parity error generation	–	Center of last bit (parity bit)	Center of stop bit
Timing for overrun error generation	Center of last bit (Bit8)	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.

Send

Mode	9-Bit	8-Bit + Parity	8-Bit, 7-Bit + Parity, 7-Bit
Timing for interrupt generation	Immediately before stop bit sent	←	←

3.9.3 Operational description

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

This mode is used to increase the number of I/O pins of for transmitting or receiving data to or from the external shifter register.

This mode includes SCLK output mode to output synchronous clock SCLK0 and SCLK input mode to input external synchronous clock SCLK0.

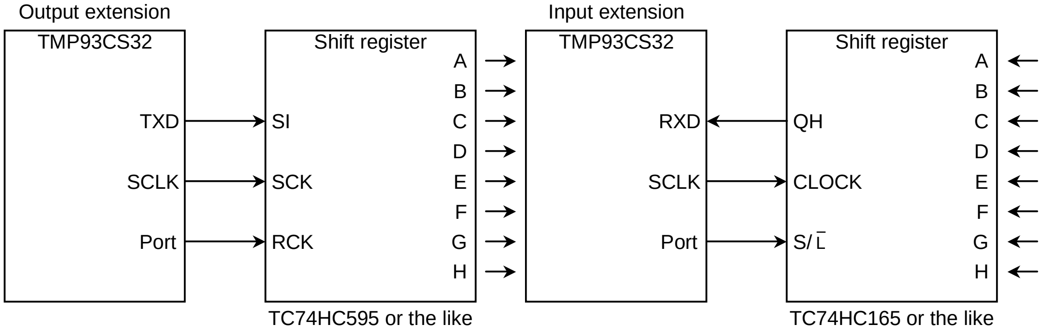


Figure 3.9.15 Example of SCLK Output Mode Connection

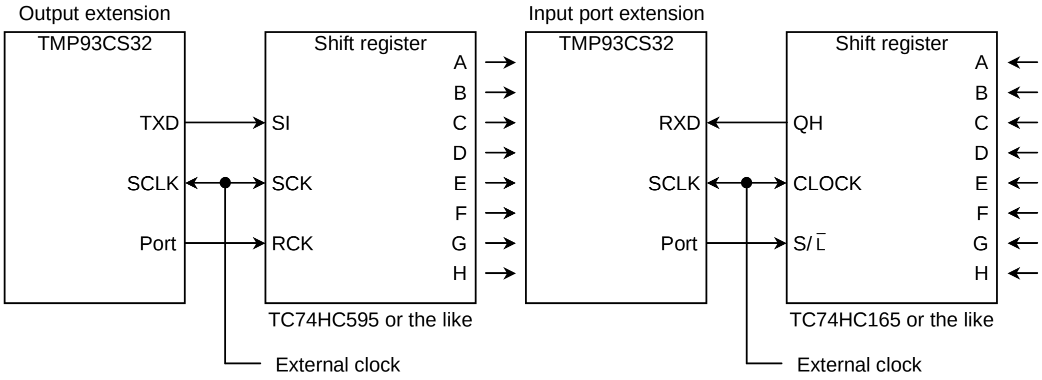


Figure 3.9.16 Example of SCLK Input Mode Connection

1. Transmission

In SCLK output mode, 8-bit data and synchronous clock are output from TXD0 pin and SCLK0 pin, respectively, each time the CPU writes data in the transmission buffer. When all data is output, INTES0<ITX0C> will be set to generate INTTX0 interrupt.

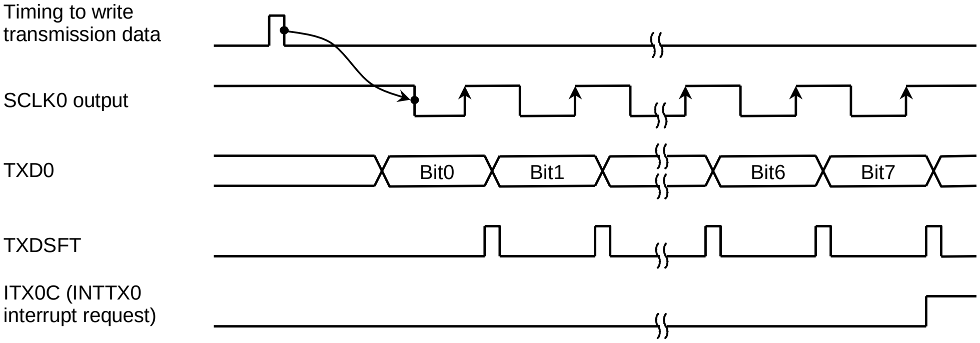


Figure 3.9.17 Transmitting Operation in I/O Interface Mode (SCLK0 output mode)

In SCLK output mode, 8-bit data are output from TXD0 pin when SCLK0 input becomes active while data are written in the transmission buffer by CPU.

When all data are output, INTES0<ITX0C> will be set to generate INTTX0 interrupt.

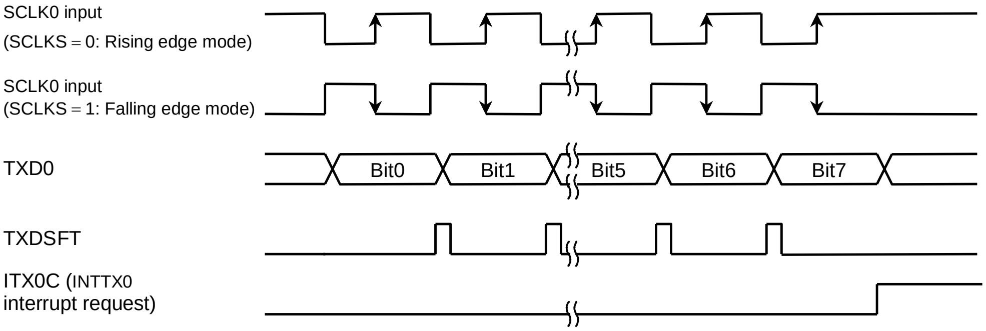


Figure 3.9.18 Transmitting Operation in I/O Interface Mode (SCLK0 input mode)

2. Receiving

In SCLK output mode, synchronous clock is outputted from SCLK0 pin and the data are shifted in the receiving buffer 1 whenever the receive interrupt flag INTES0<IRX0C> is cleared by reading the received data. When 8-bit data are received, the data will be transferred in the receiving buffer 2 (SC0BUF) at the timing shown below, and INTES0<IRX0C> will be set again to generate INTRX0 interrupt.

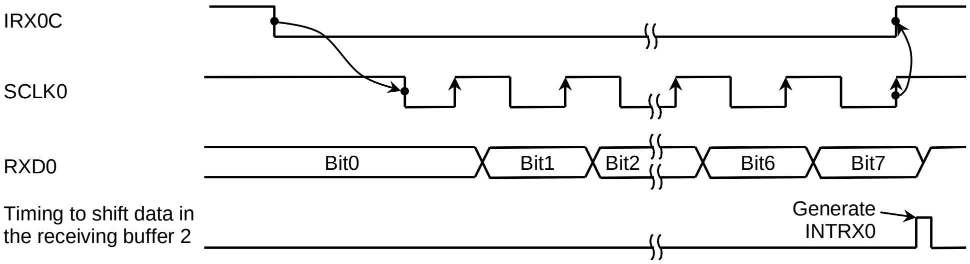


Figure 3.9.19 Receiving Operation in I/O Interface Mode (SCLK0 Output Mode)

In SCLK input mode, the data is shifted in the receiving buffer 1 when SCLK input becomes active while the receive interrupt flag INTES0<IRX0C> is cleared by reading the received data. When 8-bit data is received, the data will be shifted in the receiving buffer 2 (SC0BUF) at the timing shown below, and INTES0<IRX0C> will be set again to generate INTRX0 interrupt.

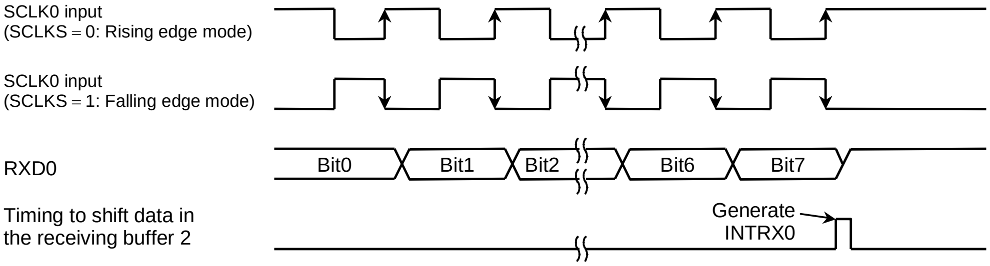


Figure 3.9.20 Receiving Operation in I/O Interface Mode (SCLK0 Input Mode)

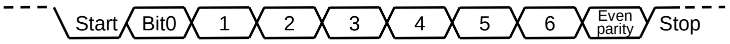
Note: For data receiving, the system must be placed in the receive enable state (SC0MOD<RXE> = “1”).

(2) Mode 1 (7-bit UART mode)

7-bit mode can be set by setting serial channel mode register SC0MOD<SM1:0> to “01”.

In this mode, a parity bit can be added, and the addition of a parity bit can be enabled or disabled by serial channel control register SC0CR<PE>, and even parity or odd parity is selected by SC0CR<EVEN> when <PE> is set to “1” (enable).

Setting example: When transmitting data with the following format, the control registers should be set as described below.



← Direction of transmission (transmission rate: 2400 bps at fc = 12.288 MHz)

* Clock condition

Clock gear: 1 (fc)
Prescaler clock: f_{FPH}

	7	6	5	4	3	2	1	0	
P6CR ←	X	X	–	–	–	–	–	1	Select P60 as the TXD pin.
P6FC ←	X	X	–	X	–	–	X	1	
SC0MOD ←	X	0	–	X	0	1	0	1	Set 7-bit UART mode.
SC0CR ←	X	1	1	X	X	X	0	0	Add an even parity.
BR0CR ←	0	X	1	0	0	1	0	1	Set transfer rate at 2400 bps.
TRUN ←	1	X	–	–	–	–	–	–	Start the prescaler for the baud rate generator.
INTES0 ←	1	1	0	0	–	–	–	–	Enable INTTX0 interrupt and set interrupt level 4.
SC0BUF ←	*	*	*	*	*	*	*	*	Set data for transmission.

X: Don’t care, –: No change

(3) Mode 2 (8-bit UART mode)

8-bit UART mode can be specified by setting SC0MOD<SM1:0> to “10”. In this mode, parity bit can be added, the addition of a parity bit is enabled or disabled by SC0CR<PE>, and even parity or odd parity is selected by SC0CR<EVEN> when <PE> is set to “1” (Enable).

Setting example: When receiving data with the following format, the control register should be set as described below.

← Direction of transmission (transmission rate: 9600 bps at fc = 12.288 MHz)

* Clock condition — Clock gear: 1 (fc)
Prescaler clock: f_{FPH}

Main setting

		7	6	5	4	3	2	1	0	
P6CR	←	X	X	–	–	–	–	0	–	Select P61 (RXD) as the input pin.
SC0MOD	←	–	0	1	X	1	0	0	1	Enable receiving in 8-bit UART mode.
SC0CR	←	X	0	1	X	X	X	0	0	Add an odd parity.
BR0CR	←	0	X	0	1	0	1	0	1	Set transfer rate at 9600 bps.
TRUN	←	1	X	–	–	–	–	–	–	Start the prescaler for the baud rate generator.
INTES0	←	–	–	–	–	1	1	0	0	Enable INTRX0 interrupt and set interrupt level 4.

Interrupt processing

Acc ← SC0CR AND 00011100		Check for error.
if Acc ≠ 0 then ERROR		
Acc ← SC0BUF		Read the received data.

X: Don’t care, –: No change

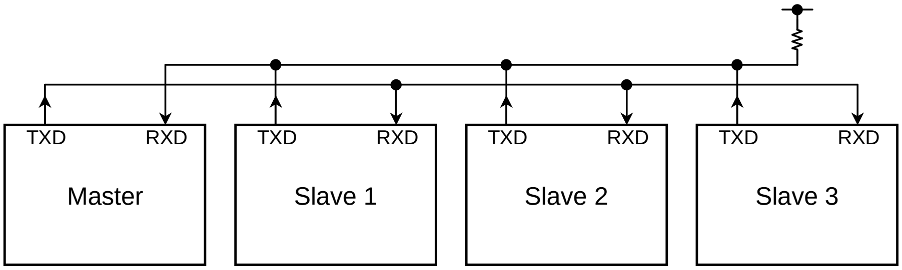
(4) Mode 3 (9-bit UART mode)

9-bit UART mode can be specified by setting SC0MOD<SM1:0> to “11”. In this mode, parity bit cannot be added.

For transmission, the MSB (9th bit) is written in SC0MOD<TB8>, while in receiving it is stored in SC0CR<RB8>. For writing and reading the buffer, the MSB is read or written first then SC0BUF.

Wake-up function

In 9-bit UART mode, the wake-up function of slave controllers is enabled by setting SC0MOD<WU> to “1”. The interrupt INTRX0 occurs only when <RB8> = 1.

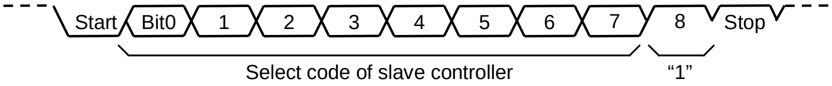


Note: TXD pin of the slave controllers must be in open-drain output mode.

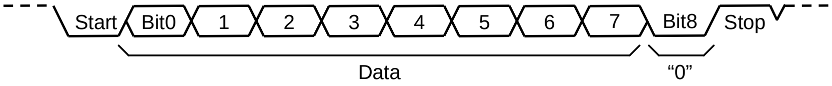
Figure 3.9.21 Serial Link Using Wake-up Function

Protocol

1. Select the 9-bit UART mode for the master and slave controllers.
2. Set SC0MOD<WU> bit of each slave controller to "1" to enable data receiving.
3. The master controller transmits one-frame data including the 8-bit select code for the slave controllers. The MSB (bit8) <TB8> is set to "1".



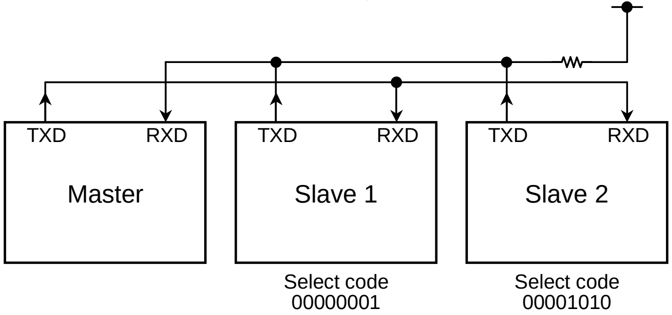
4. Each slave controller receives the above frame, and clears WU bit to "0" if the above select code matches its own select code.
5. The master controller transmits data to the specified slave controller whose SC0MOD<WU> bit is cleared to "0". The MSB (bit8) <TB8> is cleared to "0".



6. The other slave controllers (with the <WU> bit remaining at "1") ignore the receiving data because their MSBs (Bit8 or <RB8>) are set to "0" to disable the interrupt INTRX0.

   The slave controllers (<WU> = 0) can transmit data to the master controller, and it is possible to indicate the end of data receiving to the master controller by this transmission.

Setting example: To link two slave controllers serially with the master controller, and use the internal clock ϕ1 as the transfer clock.



Since serial channels 0 and 1 operate in exactly the same way, channel 0 is used for the purposes of explanation.

- Setting the master controller

```
Main
P6CR    ← X X – – – – 0 1   } Select P60 as TXD0 pin and P61 as RXD0 pin.
P6FC    ← X X – X – 0 X 1   }
INTES0  ← 1 1 0 0 1 1 0 1     Enable INTTX0 and set the interrupt level 4.
                              Enable INTRX0 and set the interrupt level 5.
SC0MOD  ← 1 0 1 0 1 1 1 0     Set ϕ1 as the transmission clock in 9-bit UART mode.
SC0BUF  ← 0 0 0 0 0 0 0 1     Set the select code for slave controller 1.
```

```
INTTX0 interrupt
SC0MOD  ← 0 – – – – – – –     Sets TB8 to“0”.
SC0BUF  ← * * * * * * * *     Set data for transmission.
```

- Setting the slave controller 1

```
Main
P6CR    ← X X – – – – 0 1   } Select P61 as RXD0 pin and P60 as TXD0 pin (open-drain
P6FC    ← X X – X – 0 X 1   } output).
ODE     ← X X X X X X – 1   }
INTES0  ← 1 1 0 1 1 1 1 0     Enable INTRX0 and INTTX0.
SC0MOD  ← 0 0 1 1 1 1 1 0     Set <WU> to “1” in the 9-bit UART transmission mode with
                              transfer clock ϕ1.
```

```
INTRX0 interrupt
Acc     ← SC0BUF
if Acc = Select code
Then
SC0MOD  ← – – – 0 – – – –     Clear <WU> to “0”.
```

3.10 Analog/Digital Converter

TMP93CS32 incorporate a high-speed, high-precision 10-bit analog/digital converter (AD converter) with 6-channel analog input.

Figure 3.10.1 is a block diagram of the AD converter. The 6-channel analog input pins (AN0 to AN5) are also used as input-only port P5 and can be also used as input ports.

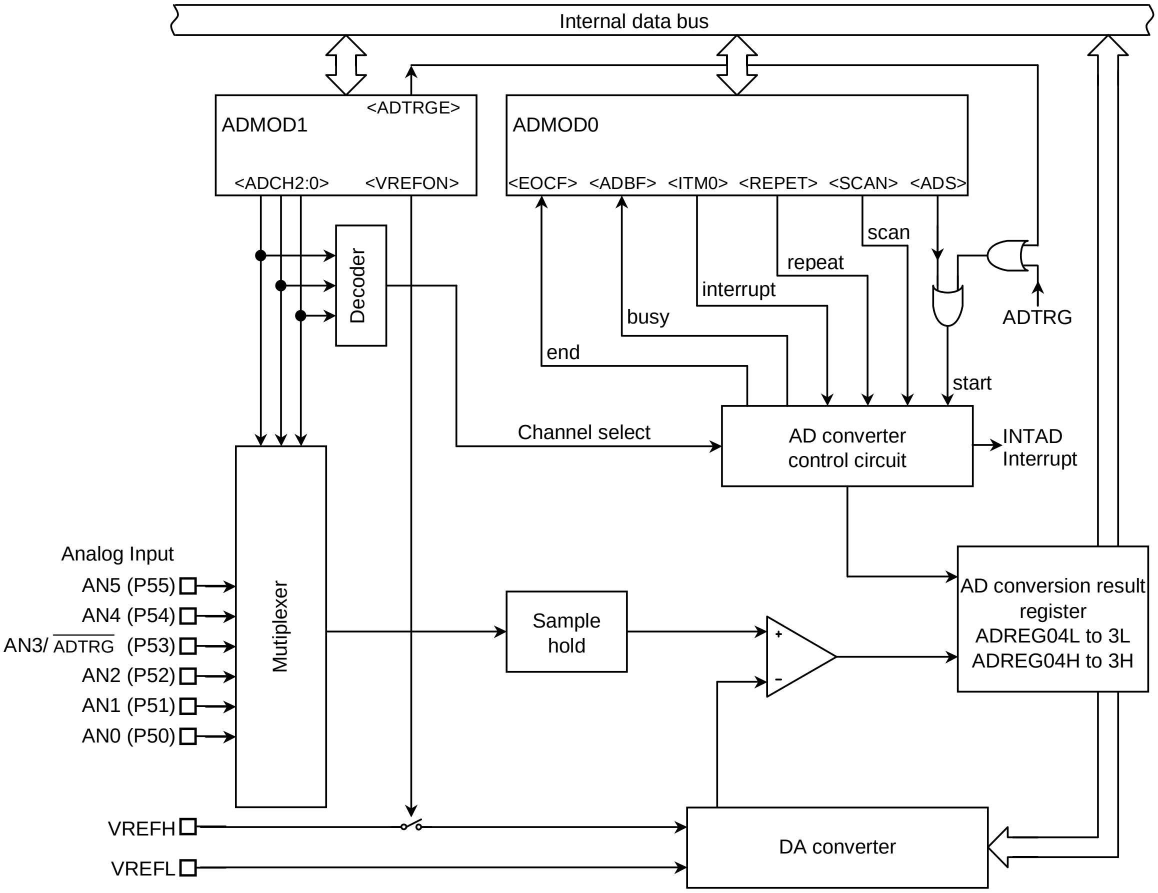


Figure 3.10.1 Block Diagram of AD Converter

Note 1: When the power supply current is reduced in IDLE2, IDLE1, STOP mode, there is possible to set a standby enabling the internal comparator due to a timing. Stop operation of AD converter before execution of “HALT” instruction.

Note 2: In regard to the lowest operation frequency.
The operation of AD converter is guaranteed with clock of $f_{FPH} \geq 4$ MHz.

3.10.1 Analog/Digital Converter Registers

AD converter is controlled by two AD mode control registers (ADMOD0 and ADMOD1). AD conversion result is stored in eight AD conversion result registers (ADREG04H/L, ADREG15H/L, ADREG2H/L, ADREG3H/L).

AD Mode Control Register 0

ADMOD0 (005EH)		7	6	5	4	3	2	1	0
	Bit symbol	EOCF	ADBF	–	–	ITM0	REPET	SCAN	ADS
	Read/Write	R		R/W					
	After reset	0	0	0	0	0	0	0	0
	Function	AD conversion end flag 0: Conversion in progress 1: Conversion end	AD conversion busy flag 0: Conversion is idle. 1: Conversion in progress	(Note) Always write “0”.	(Note) Always write “0”.	Specifies interrupts for fixed channel /repeat-conversion mode. 0: Every conversion 1: Every four conversions	Repeat mode specification 0: Single conversion mode 1: Repeat conversion mode	Scan mode specification 0: Fixed-channel mode 1: Channel scan mode	AD conversion start 0: Don’t care 1: Start conversion (Note)

AD conversion start

0	Don’t care
1	Start AD conversion

Note: Always read as “0”.

AD scan mode specification

0	AD conversion fixed-channel mode
1	AD conversion channel scan mode

AD repeat mode specification

0	AD single conversion mode
1	AD repeat conversion mode

AD conversion interrupt specification for fixed-channel/repeat-conversion mode

	Fixed channel/repeat conversion mode <SCAN> = “0”, <REPET> = “1”
0	Generates interrupt every conversion.
1	Generates interrupt every four conversions.

AD conversion busy flag

0	AD conversion is idle.
1	AD conversion in progress

AD conversion end flag

0	AD conversion in progress
1	AD conversion end

Figure 3.10.2 Register for AD Converter (1/4)

AD Mode Control Register 1

ADMOD1
(005FH)

	7	6	5	4	3	2	1	0
Bit symbol	VREFON				ADTRGE	ADCH2	ADCH1	ADCH0
Read/Write	R/W				R/W			
After reset	1				0	0	0	0
Function	String resistance 0: Off 1: On				External trigger start control 0: Disable 1: Enable	Analog input channel selection		

Analog input channel selection

<SCAN> <ADCH2:0>	0 (Channel fix)	1 (Channel scan)
000	AN0	AN0
001	AN1	AN0 → AN1
010	AN2	AN0 → AN1 → AN2
011 (Note)	AN3	AN0 → AN1 → AN2 → AN3
100	AN4	AN4
101	AN5	AN4 → AN5
110	Don't care	
111		

Conversion start control by external trigger ($\overline{\text{ADTRG}}$ pin input)

0	Disable
1	Enable

Analog reference voltage control

0	OFF
1	ON

Note: Set the <VREFON> bit to "1" before starting conversion (write "1" to ADMOD0<ADS>).

Note: As the AN3 and the $\overline{\text{ADTRG}}$ are the same pin, <ADCH2:0> = "011" can't be set when <ADTRGE> is set to 1 and $\overline{\text{ADTRG}}$ is used.

Figure 3.10.3 Register for AD Converter (2/4)

AD Conversion Result Register 0/4 Low

		7	6	5	4	3	2	1	0
ADREG04L (0060H)	Bit symbol	ADR01	ADR00						ADR0RF
	Read/Write	R							R
	After reset	Undefined							0
	Function	Stores lower 2 bits of AD conversion result.							Conversion result stored flag 1: Exist result

AD Conversion Result Register 0/4 High

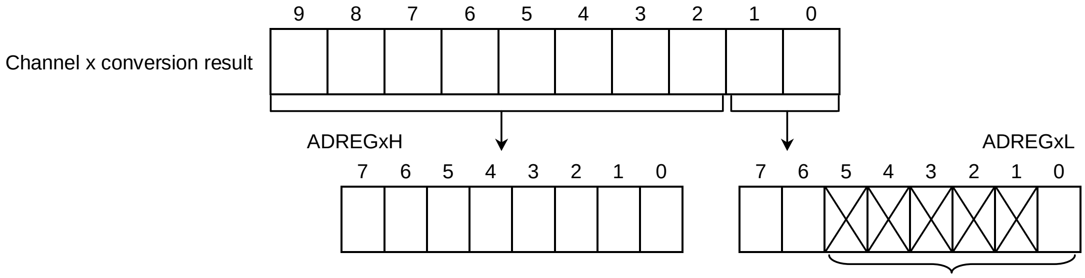
		7	6	5	4	3	2	1	0
ADREG04H (0061H)	Bit symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
	Read/Write	R							
	After reset	Undefined							
	Function	Stores upper 8 bits of AD conversion result.							

AD Conversion Result Register 1/5 Low

		7	6	5	4	3	2	1	0
ADREG15L (0062H)	Bit symbol	ADR11	ADR10						ADR1RF
	Read/Write	R							R
	After reset	Undefined							0
	Function	Stores lower 2 bits of AD conversion result.							Conversion result stored flag 1: Exist result

AD Conversion Result Register 1/5 High

		7	6	5	4	3	2	1	0
ADREG15H (0063H)	Bit symbol	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
	Read/Write	R							
	After reset	Undefined							
	Function	Stores upper 8 bits of AD conversion result.							



- Bits 5 to 1 are always read as “1”.
- Bit0 is conversion result stored flag bit <ADRxRF>.
  <ADRxRF> is set to “1” when the AD conversion result is stored.
  Reading either the ADREGxH or the ADREGxL registers clears <ADRxRF> to “0”.

Figure 3.10.4 Registers for AD Converter (3/4)

AD Conversion Result Register 2 Low

ADREG2L
(0064H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR21	ADR20						ADR2RF
Read/Write	R							R
After reset	Undefined							0
Function	Stores lower 2 bits of AD conversion result.							Conversion result stored flag 1: Exist result

AD Conversion Result Register 2 High

ADREG2H
(0065H)

	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	Stores upper 8 bits of AD conversion result.							

AD Conversion Result Register 3 Low

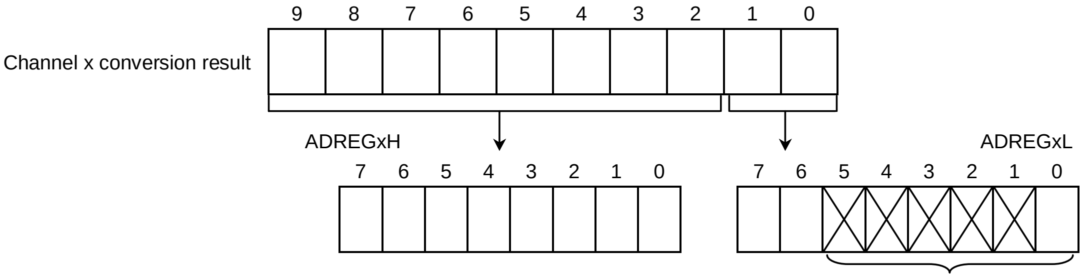
ADREG3L
(0066H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR31	ADR30						ADR3RF
Read/Write	R							R
After reset	Undefined							0
Function	Stores lower 2 bits of AD conversion result.							Conversion result stored flag 1: Exist result

AD Conversion Result Register 3 High

ADREG3H
(0067H)

	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	Stores upper 8 bits of AD conversion result.							



- Bits 5 to 1 are always read as “1”.
- Bit0 is conversion result stored flag bit <ADRxRF>. <ADRxRF> is set to “1” when the AD conversion result is stored. Reading either the ADREGxH or the ADREGxL registers clears <ADRxRF> to “0”.

Figure 3.10.5 Registers for AD Converter (4/4)

3.10.2 Operation

(1) Analog reference voltage

High analog reference voltage is applied to the VREFH pin, and low analog reference voltage is applied to the VREFL pin. The voltage between VREFH and VREFL is divided into 1024 increments using a string resistor. AD conversion is based on comparing the analog input voltage with these reference voltage increments.

To turn the switch between VREFH and VREFL off, program “0” to the ADMOD1<VREFON> bit.

To start AD conversion when the switch is off, first program “1” to <VREFON>. After that, wait at 3 μs long enough to get the stabilized oscillation, program “1” to ADMOD0<ADS>.

(2) Selecting analog input channels

The procedure for selecting analog input channels depends on the operating mode of the AD converter.

- When analog input channel is used to fix (ADMOD0<SCAN> = “0”)

  To set ADMOD1<ADCH2:0>, selecting one channel from analog input pins AN0 to AN5.

- When analog input channel is used to scan (ADMOD0<SCAN> = “1”)

  To set ADMOD1<ADCH2:0>, selecting one channel from 6 scan mode.

  Table 3.10.1 shows the analog input channel selection each operating mode.

  A reset initializes AD mode control register ADMOD1<ADCH2:0> to “000”, selecting pin AN0 for the AD converter input.

  The pins not used as analog input channels can be used as general-purpose input ports (P5).

Table 3.10.1 Analog Input Channel Selection

<ADCH2:0>	Fixed Channel <SCAN> = 0	Channel Scan <SCAN> = 1
000	AN0	AN0
001	AN1	AN0 → AN1
010	AN2	AN0 → AN1 → AN2
011	AN3	AN0 → AN1 → AN2 → AN3
100	AN4	AN4
101	AN5	AN4 → AN5
110	Don’t care	
111		

(3) Starting AD conversion

AD conversion starts when ADMOD0<ADS> to “1”, or ADMOD1<ADTRGE> is set to “1” and the falling edge is input through $\overline{\text{ADTRG}}$ pin.

When AD conversion starts, AD conversion busy flag ADMOD0<ADBF> is set to “1”, indicating AD conversion is in progress.

Writing “1” to <ADS> while conversion is in progress restarts the conversion. Check the conversion result stored flag ADREGxL<ADRxRF> to determine whether the AD conversion data are valid at this time.

Inputting the falling edge to the $\overline{\text{ADTRG}}$ pin while conversion is in progress is invalid.

(4) AD conversion modes and completion interrupt

Follow the four AD conversion modes are supported.

- Fixed channel single conversion mode
- Channel scan single conversion mode
- Fixed channel repeat conversion mode
- Channel scan repeat conversion mode

AD conversion mode can selected by setting AD mode control register ADMOD0<REPET, SCAN>.

When AD conversion ends, AD conversion completion interrupt INTAD request occurs. And the ADMOD0<EOCF> flag is set to “1” to indicate that AD conversion has completed.

1. Fixed channel single conversion mode

   Fixed channel single conversion mode can be specified by setting ADMOD0<REPET, SCAN> to “00”.

   In this mode, conversion of the specified single channel is executed once only. After conversion is completed, ADMOD0<EOCF> is set to “1”, ADMOD0<ADBF> is cleared to “0” and occurs INTAD interrupt request.

2. Channel scan single conversion mode

   Channel scan single conversion mode can be specified by setting ADMOD0<REPET, SCAN> to “01”.

   In this mode, conversion of the specified channel are executed once only. After conversion is completed, ADMOD0<EOCF> is set to “1”, ADMOD0<ADBF> is cleared to “0” and occurs INTAD interrupt request.

3. Fixed channel repeat conversion mode

   Fixed channel repeat conversion mode can be specified by setting ADMOD0<REPET, SCAN> to “10”.

   In this mode, conversion of the specified single channel is executed repeatedly. After conversion is completed, ADMOD0<EOCF> is set to “1”, ADMOD0<ADBF> remains “1”, not changed to “0”. The timing of INTAD interrupt request can selected by setting of ADMOD0<ITM0>.

   When <ITM0> is set to “0”, interrupt request occurs after every conversion.

   When <ITM0> is set to “1”, interrupt request occurs after every fourth conversion.

4. Channel scan repeat conversion mode

   Channel scan repeat conversion mode can be specified by setting ADMOD0<REPET, SCAN> to “11”.

   In this mode, specified channels are converted repeatedly. After every scan convert completion, ADMOD0<EOCF> is set to “1” and INTAD interrupt request occurs. ADMOD0<ADBF> remains “1”, not changed to “0”.

   To stop the repeat conversion mode (3. and 4. modes), program “0” to ADMOD0<REPET>. After the current conversion is completed repeat conversion mode is terminated, and ADMOD0<ADBF> is cleared to “0” .

   If the device enters the IDLE2, IDLE1 or STOP modes during AD conversion, the conversion halts immediately. After the HALT mode is released, AD conversion restarts from the beginning in repeat conversion mode (3. and 4. modes), it does not restart in single conversion mode (1. and 2. modes).

Table 3.10.2shows the relations between AD conversion modes and interrupt request.

Table 3.10.2 Relation between AD Conversion Modes and Interrupt Request

Mode	Interrupt Request Timing	ADMOD0		
		<ITM0>	<REPET>	<SCAN>
Fixed channel single conversion mode	After conversion	X	0	0
Channel scan single conversion mode	After conversion	X	0	1
Fixed channel repeat conversion mode (Every conversion)	After every conversion	0	1	0
Fixed channel repeat conversion mode (Every fourth conversion)	After every fourth conversion	1		
Channel scan repeat conversion mode	After every scan conversion	X	1	1

X: Don't care

(5) AD conversion time

140 states (14 μs at fc = 20 MHz) are required for AD conversion of one channel.

(6) Storing and reading the AD conversion result

AD conversion results are stored in AD conversion result registers high/low (ADREG04H/L to ADREG3H/L). These registers are read only.

In fixed channel repeat conversion mode, AD conversion results are stored in order from ADREG04H/L to ADREG3H/L. Except in this mode, AD conversion results for channel AN0 and AN4, AN1 and AN5, AN2, AN3 are stored severally ADREG04H/L, ADREG15H/L, ADREG2H/L, ADREG3H/L.

Table 3.10.3 shows correspondence between analog input channels and AD conversion result registers.

Table 3.10.3 Correspondence Between Analog Input Channels and AD Conversion Result Registers

Analog Input Channel (port 5)	AD Conversion Result Registers	
	Conversion Modes Except Right	Fixed Channel Repeat Conversion Mode (Every fourth conversion)
AN0	ADREG04H/L	ADREG04H/L ← ↓ ADREG15H/L ↓ ADREG2H/L ↓ ADREG3H/L
AN1	ADREG15H/L	
AN2	ADREG2H/L	
AN3	ADREG3H/L	
AN4	ADREG04H/L	
AN5	ADREG15H/L	

AD conversion result registers bit0 is AD conversion result stored flag <ADRxRF>. The flag shows that whether those registers are read or not. When AD conversion results are stored in those registers (ADREGxH or ADREGxL), this flag is set to "1". When each register is read, this flag is cleared to "0", and AD conversion end flag ADMOD0<EOCF> is also cleared to "0".

Setting example:

1. This example converts the analog input voltage at the AN3 pin. The INTAD interrupt routine writes the result to memory address 0800H.

Main routine setting:

		7	6	5	4	3	2	1	0	
INTE0AD	←	1	1	0	0	–	–	–	–	Enables INTAD and sets level 4.
ADMOD1	←	1	X	X	X	0	0	1	1	Sets analog input channel to AN3.
ADMOD0	←	X	X	0	0	0	0	0	1	Starts AD conversion in fixed channel single conversion mode.

Example of interrupt routine processing:

WA	← ADREG3	Reads ADREG3L and ADREG3H values and writes them to WA (16 bits).
WA	> > 6	Shifts right WA six times and zero-fills the upper bits.
(0800H)	← WA	Writes contents of WA to memory address 0800H.

2. This example repeatedly converts the analog input voltages at pins AN0 to AN2, using channel scan repeat conversion mode.

INTE0AD	←	1	0	0	0	–	–	–	–	Disables INTAD.
ADMOD1	←	1	X	X	X	0	0	1	0	Sets AN0 to AN2 as analog input channels.
ADMOD0	←	X	X	0	0	0	1	1	1	Starts AD conversion in channel scan repeat conversion mode.

X: Don’t care –: No change

3.11 Watchdog Timer (Runaway Detecting Timer), Warm-up Timer

TMP93CS32 contains a watchdog timer of Runaway detecting.

The watchdog timer (WDT) is used to return the CPU to the normal state when it detects that the CPU has started to malfunction (runaway) due to causes such as noise. When the watchdog timer detects a malfunction, it generates a non-maskable interrupt INTWD to notify the CPU of the malfunction.

Connecting the watchdog timer output to the reset pin internally forces a reset.

This watchdog timer consists of 7-stage and 15-stage binary counters.

These binary counters are also used as a warm-up timer for the internal oscillator stabilization. This is used for releasing the STOP.

3.11.1 Configuration

Figure 3.11.1 shows the block diagram of the watchdog timer (WDT).

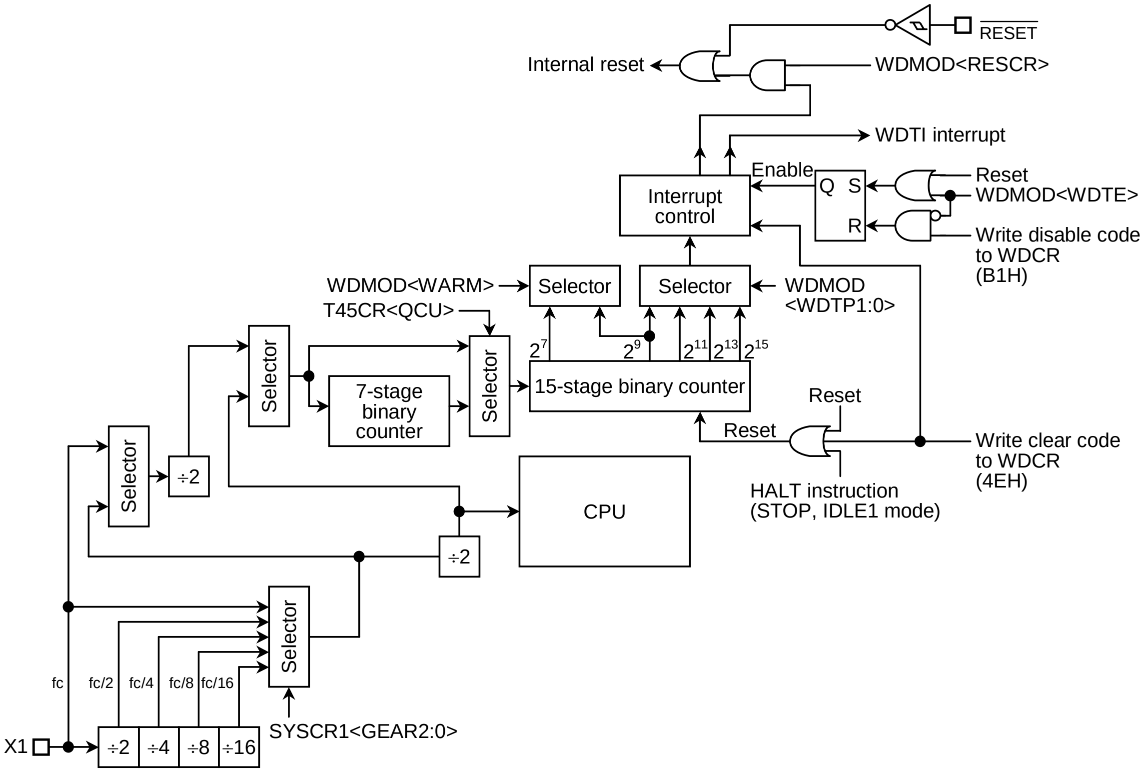


Figure 3.11.1 Block Diagram of Watchdog Timer/Warm-up Timer

The watchdog timer consists of 7-stage and 15-stage binary counters which use System clock (f_{SYS}) as the input clock. The 15-stage binary counter has $f_{SYS}/2^{15}$, $f_{SYS}/2^{17}$, $f_{SYS}/2^{19}$, and $f_{SYS}/2^{21}$ output. Selecting one of the outputs with the WDMOD<WDTP1:0> register generates a watchdog interrupt when an overflow occurs. The binary counter for the watchdog timer should be cleared to “0” with runaway detecting result software (instruction) before an interrupt occurs.

Example:

```
LDW     (WDMOD), B100H      ;   Disable.
LD      (WDCR), 4EH         ;   Write clear code.
SET     7, (WDMOD)          ;   Enable again.
```

The runaway detecting result can also be connected to the reset pin internally. In this case, the watchdog timer resets itself.

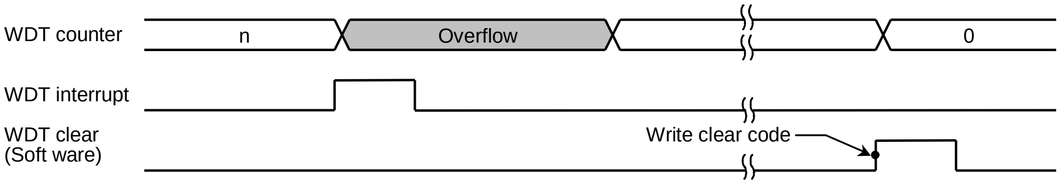


Figure 3.11.2 Normal Mode

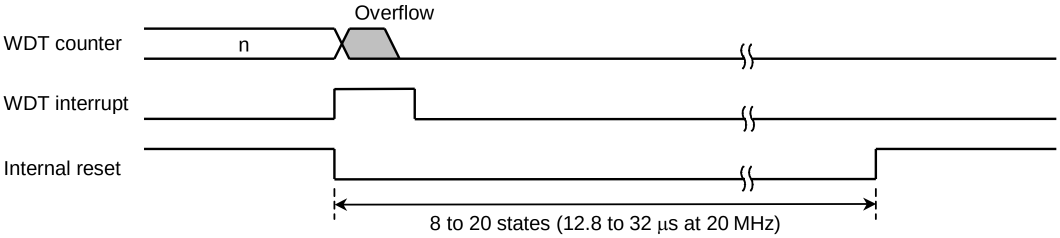


Figure 3.11.3 Reset Mode

For warm-up counter, 2^7 and 2^9 output of 15-stage binary counter can be selected using WDMOD<WARM> register. When a stable-external oscillator is used, shorter warm-up time is available using T45CR<QCU> register. When <QCU> = 1, counting value 2^7 is selected.

When the watchdog timer is in operation, this shorter warm-up time function cannot be available. This function can be available by setting <QCU> = 0.

3.11.2 Control registers

Watchdog timer WDT is controlled by two control registers WDMOD and WDCR.

(1) Watchdog timer mode register (WDMOD)

1. Setting the detecting time of watchdog timer <WDTP>

   This 2-bit register is used to set the watchdog timer interrupt time for detecting the runaway. This register is initialized to WDMOD<WDTP1:0> = 00 when reset.

   The defecting time of WDT is shown Table 3.11.1.

2. Watchdog timer enable/disable control register <WDTE>

   When reset, WDMOD<WDTE> is initialized to "1" enable the watchdog timer.

   To disable, it is necessary to set this bit to "0" and write the disable code (B1H) in the watchdog timer control register WDCR. This makes it difficult for the watchdog timer to be disabled by runaway.

   However, it is possible to return from the disable state to enable state by merely setting <WDTE> to "1".

3. Watchdog timer out reset connection <RESCR>

   This register is used to connect the output of the watchdog timer with RESET terminal, internally. Since WDMOD<RESCR> is initialized to 0 at reset, a reset by the watchdog timer will not be performed.

(2) Watchdog timer control register (WDCR)

This register is used to disable and clear of binary counter the watchdog timer function.

- Disable control

  By writing the disable code (B1H) in this WDCR register after clearing WDMOD<WDTE> to "0", the watchdog timer can be disabled.

```
┌ WDMOD    ← 0  –  –  –  –  –  X  X          Clear WDMOD<WDTE>to "0".
└ WDCR     ← 1  0  1  1  0  0  0  1          Write the disable code (B1H).
```

- Enable control

  Set WDMOD<WDTE> to "1".

- Watchdog timer clear control

  The binary counter can be cleared and resume counting by writing clear code (4EH) into the WDCR register.

```
WDCR       ← 0  1  0  0  1  1  1  0          Write the clear code (4EH).
             X: Don't care   –: No change
```

Watchdog Timer Mode Control Register

		7	6	5	4	3	2	1	0
WDMOD (005CH)	Bit symbol	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
	Read/Write	R/W							
	After reset	1	0	0	0	0	0	0	0
	Function	WDT control 0: Disable 1: Enable	Select the detecting time of WDT 00: $2^{15}/f_{SYS}$ 01: $2^{17}/f_{SYS}$ 10: $2^{19}/f_{SYS}$ 11: $2^{21}/f_{SYS}$ See Table 3.11.1.		Select the warm-up time.	Standby mode 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		1: Internally connects WDT out to the reset pin.	1: Drive the pin even in STOP mode.

Refer to section
3.3 “Standby function.”

Watchdog timer out control

0	Don’t care
1	Connects WDT out to a reset

Select the warm-up time at fc = 20 MHz

Gear value <GEAR2:0>	T45CR<QCU> = 0		T45CR<QCU> = 1
	<WARM> = 0	<WARM> = 1	<WARM> = X
000 (fc)	0.8192 ms	3.277 ms	6.4 µs
001 (fc/2)	1.638 ms	6.554 ms	12.8 µs
010 (fc/4)	3.277 ms	13.107 ms	25.6 µs
011 (fc/8)	6.554 ms	26.214 ms	51.2 µs
100 (fc/16)	13.107 ms	52.429 ms	102.4 µs

Watchdog timer enable/disable control

0	Disable
1	Enable

Note: When the watchdog timer is in operation, T45CR<QCU> is cleared to “0”.

Figure 3.11.4 Watchdog Timer Related Register (1/2)

Watchdog Timer Control Register

WDCR (005DH)		7	6	5	4	3	2	1	0
	Bit symbol	–							
	Read/Write	W							
	After reset	–							
	Function	B1H: WDT disable code 4EH: WDT clear code							

Disable/clear WDT

B1H	Disable code
4EH	Clear code
Others	Don’t set

Note: When the watchdog timer is in operation, T45CR<QCU> is cleared to “0”.

Figure 3.11.5 Watchdog Timer Related Register (2/2)

Table 3.11.1 Watchdog Timer Detecting Time

at fc = 20 MHz

Gear value <GEAR2:0>	Watchdog Timer Detecting Time			
	WDMOD<WDTP1:0>			
	00	01	10	11
000 (fc)	3.277 ms	13.107 ms	52.429 ms	209.715 ms
001 (fc/2)	6.554 ms	26.214 ms	104.858 ms	419.430 ms
010 (fc/4)	13.107 ms	52.429 ms	209.715 ms	838.861 ms
011 (fc/8)	26.214 ms	104.858 ms	419.430 ms	1.678 s
100 (fc/16)	52.429 ms	209.715 ms	838.861 ms	3.355 s

3.11.3 Operation

The watchdog timer generates interrupt INTWD after the detecting time set in the WDMOD<WDTP1:0>. The watchdog timer must be zero-cleared by software before an INTWD interrupt is generated. If the CPU malfunctions (Runaway) due to causes such as noise, but does not execute the instruction used to clear the binary counter, the binary counter overflows and an INTWD interrupt is generated. The CPU detects malfunction (Runaway) due to the INTWD Interrupt and it is possible to return to normal operation by an anti-mulfunction program. By connecting the watchdog timer out pin to peripheral devices' resets, a CPU malfunction can also be acknowledged to other devices.

The watchdog timer restarts operation immediately after resetting is released.

The watchdog timer stops its operation in the IDLE1 and STOP modes. In the RUN and IDLE2 modes, the watchdog timer is enabled.

However, the function can be disabled when entering the RUN, IDLE2 mode.

Example: 1. Clear the binary counter.

WDCR ← 0 1 0 0 1 1 1 0 Write clear code (4EH).

2. Set the watchdog timer detecting time to $2^{17}/f_{SYS}$.

WDMOD ← 1 0 1 – – – X X

3. Disable the watchdog timer.

WDMOD ← 0 – – – – – X X Clear WDTE to "0".

WDCR ← 1 0 1 1 0 0 0 1 Write disable code (B1H).

4. Set IDLE1 mode.

WDMOD ← 0 – – – 1 0 X X

WDCR ← 1 0 1 1 0 0 0 1 Disables WDT and sets IDLE1 mode.

Executes HALT command. Set the HALT mode.

5. Set the STOP mode (Warm-up time: $2^{16}/f_{SYS}$).

WDMOD ← – – – 1 0 1 X X Set the STOP mode.

Executes HALT command. Set the HALT mode.

X: Don't care –: No change

4. Electrical Characteristics

"X" used in an expression shows a cycle of clock f_{FPH}. If a clock gear or a low speed oscillator is selected, a value of "X" is different. The value as an example is gear = 1/fc (SYSCR1<GEAR2:0> = "000").

4.1 Absolute Maximum Ratings (TMP93CS32F)

Parameter	Symbol	Rating	Unit
Power supply voltage	V_{CC}	−0.5 to 6.5	V
Input voltage	V_{IN}	−0.5 to V_{CC} + 0.5	
Output current (per 1 pin) P7	I_{OL1}	20	mA
Output current (per 1 pin) except P7	I_{OL2}	2	
Output current (total)	ΣI_{OL}	120	
Output current (total)	ΣI_{OH}	−80	
Power dissipation (Ta = 85°C)	P_D	350	mW
Soldering temperature (10 s)	T_{SOLDER}	260	°C
Storage temperature	T_{STG}	−65 to 150	
Operating temperature	T_{OPR}	−40 to 85	

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

4.2 DC Characteristics

Ta = −40 to 85°C

Parameter		Symbol	Condition	Min	Typ. (Note)	Max	Unit
Power supply voltage $AV_{CC} = V_{CC}$ $AV_{SS} = V_{SS}$ = 0 V		V_{CC}	fc = 4 to 20 MHz	4.5		5.5	V
			fc = 4 to 12.5 MHz	2.7			
Input low voltage	AD0 to AD15	V_{IL}	V_{CC} ≥ 4.5 V	−0.3		0.8	
			V_{CC} < 4.5 V			0.6	
	Port 2 to 7 (Except P35)	V_{IL1}	V_{CC} = 2.7 to 5.5 V			0.3 V_{CC}	
	$\overline{\text{RESET}}$, $\overline{\text{NMI}}$, INT0	V_{IL2}				0.25 V_{CC}	
	$\overline{\text{EA}}$, AM8/$\overline{\text{AM16}}$	V_{IL3}				0.3	
	X1	V_{IL4}				0.2 V_{CC}	
Input high voltage	AD0 to AD15	V_{IH}	V_{CC} ≥ 4.5 V V_{CC} < 4.5 V	2.2		V_{CC} + 0.3	
				2.0			
	Port 2 to 7 (Except P35)	V_{IH1}	V_{CC} = 2.7 to 5.5 V	0.7 V_{CC}			
	$\overline{\text{RESET}}$, $\overline{\text{NMI}}$, INT0	V_{IH2}		0.75 V_{CC}			
	$\overline{\text{EA}}$, AM8/$\overline{\text{AM16}}$	V_{IH3}		V_{CC} − 0.3			
	X1	V_{IH4}		0.8 V_{CC}			
Output low voltage		V_{OL}	I_{OL} = 1.6 mA (V_{CC} = 2.7 to 5.5 V)			0.45	
Output low current (P7)		I_{OL7}	V_{OL} = 1.0 V (V_{CC} = 5 V ± 10%)	16			mA
			(V_{CC} = 3 V ± 10%)	7			
Output high voltage		V_{OH1}	I_{OH} = −400 μA (V_{CC} = 3 V ± 10%)	2.4			V
		V_{OH2}	I_{OH} = −400 μA (V_{CC} = 5 V ± 10%)	4.2			

Note: Typical values are for Ta = 25°C and V_{CC} = 5 V unless otherwise noted.

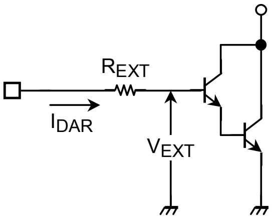
<table>
<tr><th>Parameter</th><th>Symbol</th><th colspan="2">Condition</th><th>Min</th><th>Typ. (Note1)</th><th>Max</th><th>Unit</th></tr>
<tr><td>Darlington drive current
(8 output pins max)</td><td>I_{DAR}
(Note2)</td><td colspan="2">$V_{EXT} = 1.5$ V
$R_{EXT} = 1.1$ kΩ
($V_{CC} = 5$ V ± 10% only)</td><td>−1.0</td><td></td><td>−3.5</td><td>mA</td></tr>
<tr><td>Input leakage current</td><td>I_{LI}</td><td colspan="2">$0.0 \le V_{IN} \le V_{CC}$</td><td></td><td>0.02</td><td>±5</td><td rowspan="2">μA</td></tr>
<tr><td>Output leakage current</td><td>I_{LO}</td><td colspan="2">$0.2 \le V_{IN} \le V_{CC} - 0.2$ V</td><td></td><td>0.05</td><td>±10</td></tr>
<tr><td>Power down voltage
(at STOP, RAM back up)</td><td>V_{STOP}</td><td colspan="2">$V_{IL2} = 0.2\ V_{CC}$,
$V_{IH2} = 0.8\ V_{CC}$</td><td>2.0</td><td></td><td>6.0</td><td>V</td></tr>
<tr><td rowspan="4">$\overline{\text{RESET}}$ pull-up resistor</td><td rowspan="4">R_{RST}</td><td colspan="2">$V_{CC} = 5.5$ V</td><td>45</td><td></td><td>130</td><td rowspan="4">kΩ</td></tr>
<tr><td colspan="2">$V_{CC} = 4.5$ V</td><td>50</td><td></td><td>160</td></tr>
<tr><td colspan="2">$V_{CC} = 3.3$ V</td><td>70</td><td></td><td>280</td></tr>
<tr><td colspan="2">$V_{CC} = 2.7$ V</td><td>90</td><td></td><td>400</td></tr>
<tr><td>Pin capacitance</td><td>C_{IO}</td><td colspan="2">fc = 1 MHz</td><td></td><td></td><td>10</td><td>pF</td></tr>
<tr><td>Schmitt width
$\overline{\text{RESET}}$, $\overline{\text{NMI}}$, INT0</td><td>V_{TH}</td><td colspan="2"></td><td>0.4</td><td>1.0</td><td></td><td>V</td></tr>
<tr><td rowspan="4">Programmable
pull-up resistor</td><td rowspan="4">R_{KH}</td><td colspan="2">$V_{CC} = 5.5$ V</td><td>45</td><td></td><td>130</td><td rowspan="4">kΩ</td></tr>
<tr><td colspan="2">$V_{CC} = 4.5$ V</td><td>50</td><td></td><td>160</td></tr>
<tr><td colspan="2">$V_{CC} = 3.3$ V</td><td>70</td><td></td><td>280</td></tr>
<tr><td colspan="2">$V_{CC} = 2.7$ V</td><td>90</td><td></td><td>400</td></tr>
<tr><td>NORMAL (Note3)</td><td rowspan="11">I_{CC}</td><td colspan="2" rowspan="4">$V_{CC} = 5$ V ± 10%
fc = 20 MHz</td><td></td><td>19</td><td>25</td><td rowspan="8">mA</td></tr>
<tr><td>RUN</td><td></td><td>17</td><td>25</td></tr>
<tr><td>IDLE2</td><td></td><td>10</td><td>15</td></tr>
<tr><td>IDLE1</td><td></td><td>3.5</td><td>5</td></tr>
<tr><td>NORMAL (Note3)</td><td colspan="2" rowspan="4">$V_{CC} = 3$ V ± 10%
fc = 12.5 MHz
(Typ.: $V_{CC} = 3.0$ V)</td><td></td><td>6.5</td><td>10</td></tr>
<tr><td>RUN</td><td></td><td>5.0</td><td>9</td></tr>
<tr><td>IDLE2</td><td></td><td>3.0</td><td>5</td></tr>
<tr><td>IDLE1</td><td></td><td>0.8</td><td>1.5</td></tr>
<tr><td rowspan="3">STOP</td><td>Ta ≤ 50°C</td><td rowspan="3">$V_{CC} = 2.7$ V
to 5.5 V</td><td></td><td rowspan="3">0.2</td><td>10</td><td rowspan="3">μA</td></tr>
<tr><td>Ta ≤ 70°C</td><td></td><td>20</td></tr>
<tr><td>Ta ≤ 85°C</td><td></td><td>50</td></tr>
</table>

Note 1: Typical values are for Ta = 25°C and $V_{CC} = 5$ V unless otherwise noted.

Note 2: I_{DAR} is guranteed for total of up to 8 ports.

Note 3: I_{CC} measurement conditions (NORMAL):
Only CPU is operational ; output pins are open and input pins are fixed.

(Reference) Definition of I_{DAR}



4.3 AC Characteristics

(1) $V_{CC} = 5\ V \pm 10\%$

No.	Parameter	Symbol	Variable		16 MHz		20 MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	Osc. period (= x)	t_{OSC}	50	31250	62.5		50		ns
2	CLK pulse width	t_{CLK}	2x – 40		85		60		ns
3	A0 to A23 valid → CLK hold	t_{AK}	0.5x – 20		11		5		ns
4	CLK valid → A0 to A23 hold	t_{KA}	1.5x – 70		24		5		ns
5	A0 to A15 valid → ALE fall	t_{AL}	0.5x – 15		16		10		ns
6	ALE fall → A0 to A15 hold	t_{LA}	0.5x – 20		11		5		ns
7	ALE high pulse width	t_{LL}	x – 40		23		10		ns
8	ALE fall → $\overline{\text{RD}}$ / $\overline{\text{WR}}$ fall	t_{LC}	0.5x – 25		6		0		ns
9	$\overline{\text{RD}}$ / $\overline{\text{WR}}$ rise → ALE rise	t_{CL}	0.5x – 20		11		5		ns
10	A0 to A15 valid → $\overline{\text{RD}}$ / $\overline{\text{WR}}$ fall	t_{ACL}	x – 25		38		25		ns
11	A0 to A23 valid → $\overline{\text{RD}}$ / $\overline{\text{WR}}$ fall	t_{ACH}	1.5x – 50		44		25		ns
12	$\overline{\text{RD}}$ / $\overline{\text{WR}}$ rise → A0 to A23 hold	t_{CA}	0.5x – 25		6		0		ns
13	A0 to A15 valid → D0 to D15 input	t_{ADL}		3.0x – 55		133		95	ns
14	A0 to A23 valid → D0 to D15 input	t_{ADH}		3.5x – 65		154		110	ns
15	$\overline{\text{RD}}$ fall → D0 to D15 input	t_{RD}		2.0x – 60		65		40	ns
16	$\overline{\text{RD}}$ low pulse width	t_{RR}	2.0x – 40		85		60		ns
17	$\overline{\text{RD}}$ rise → D0 to D15 hold	t_{HR}	0		0		0		ns
18	$\overline{\text{RD}}$ rise → A0 to A15 output	t_{RAE}	x – 15		48		35		ns
19	$\overline{\text{WR}}$ low pulse width	t_{WW}	2.0x – 40		85		60		ns
20	D0 to D15 valid → $\overline{\text{WR}}$ rise	t_{DW}	2.0x – 55		70		45		ns
21	$\overline{\text{WR}}$ rise → D0 to D15 hold	t_{WD}	0.5x – 15		16		10		ns
22	A0 to A23 valid → $\overline{\text{WAIT}}$ input ((1 + n) WAIT mode)	t_{AWH}		3.5x – 90		129		85	ns
23	A0 to A15 valid → $\overline{\text{WAIT}}$ input ((1 + n) WAIT mode)	t_{AWL}		3.0x – 80		108		70	ns
24	$\overline{\text{RD}}$ / $\overline{\text{WR}}$ fall → $\overline{\text{WAIT}}$ hold ((1 + n) WAIT mode)	t_{CW}	2.0x + 0		125		100		ns
25	A0 to A23 valid → Port input	t_{APH}		2.5x – 120		36		5	ns
26	A0 to A23 valid → Port hold	t_{APH2}	2.5x + 50		206		175		ns
27	$\overline{\text{WR}}$ rise → Port valid	t_{CP}		200		200		200	ns

AC measuring conditions

- Output level: High 2.2 V/Low 0.8 V, CL = 50 pF
  (However CL = 100 pF for AD0 to AD15, A0 to A23, ALE, $\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{HWR}}$, CLK)
- Input level: High 2.4 V/Low 0.45 V (AD0 to AD15)
  High 0.8 × V_{CC}/Low 0.2 × V_{CC} (Except for AD0 to AD15)

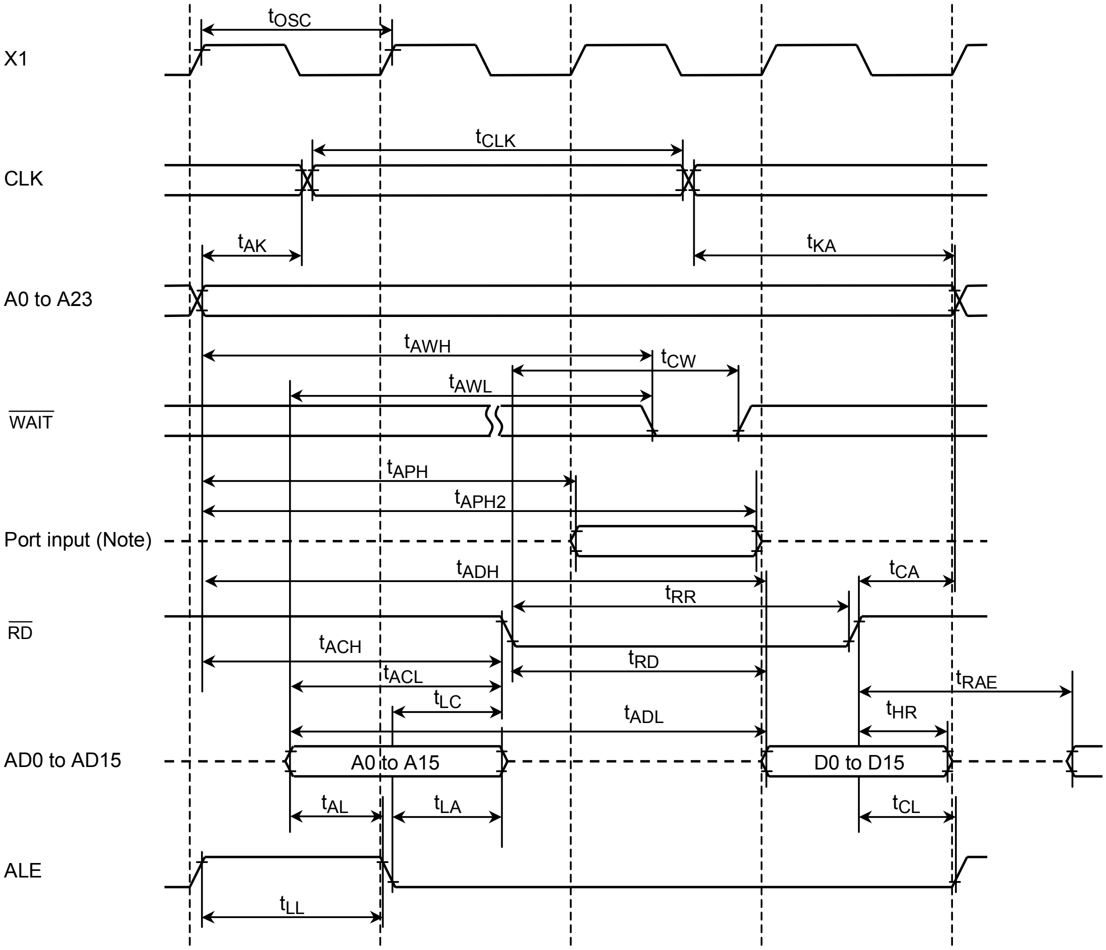
(2) $V_{CC} = 3\text{ V} \pm 10\%$

No.	Parameter	Symbol	Variable		12.5 MHz		Unit
			Min	Max	Min	Max	
1	Osc. period (= x)	t_{OSC}	80	31250	80		ns
2	CLK pulse width	t_{CLK}	2x – 40		120		ns
3	A0 to A23 valid → CLK hold	t_{AK}	0.5x – 30		10		ns
4	CLK valid → A0 to A23 hold	t_{KA}	1.5x – 80		40		ns
5	A0 to A15 valid → ALE fall	t_{AL}	0.5x – 35		5		ns
6	ALE fall → A0 to A15 hold	t_{LA}	0.5x – 35		5		ns
7	ALE high pulse width	t_{LL}	x – 60		20		ns
8	ALE fall → $\overline{RD}$ / $\overline{WR}$ fall	t_{LC}	0.5x – 35		5		ns
9	$\overline{RD}$ / $\overline{WR}$ rise → ALE rise	t_{CL}	0.5x – 40		0		ns
10	A0 to A15 valid → $\overline{RD}$ / $\overline{WR}$ fall	t_{ACL}	x – 50		30		ns
11	A0 to A23 valid → $\overline{RD}$ / $\overline{WR}$ fall	t_{ACH}	1.5x – 50		70		ns
12	$\overline{RD}$ / $\overline{WR}$ rise → A0 to A23 hold	t_{CA}	0.5x – 40		0		ns
13	A0 to A15 valid → D0 to D15 input	t_{ADL}		3.0x – 110		130	ns
14	A0 to A23 valid → D0 to D15 input	t_{ADH}		3.5x – 125		155	ns
15	$\overline{RD}$ fall → D0 to D15 input	t_{RD}		2.0x – 115		45	ns
16	$\overline{RD}$ low pulse width	t_{RR}	2.0x – 40		120		ns
17	$\overline{RD}$ rise → D0 to D15 hold	t_{HR}	0		0		ns
18	$\overline{RD}$ rise → A0 to A15 output	t_{RAE}	x – 25		55		ns
19	$\overline{WR}$ low pulse width	t_{WW}	2.0x – 40		120		ns
20	D0 to D15 valid → $\overline{WR}$ rise	t_{DW}	2.0x – 120		40		ns
21	$\overline{WR}$ rise → D0 to D15 Hold	t_{WD}	0.5x – 40		0		ns
22	A0 to A23 valid → $\overline{WAIT}$ input $\left(\begin{matrix}1+n\\ \text{WAIT mode}\end{matrix}\right)$	t_{AWH}		3.5x – 130		150	ns
23	A0 to A15 valid → $\overline{WAIT}$ input $\left(\begin{matrix}1+n\\ \text{WAIT mode}\end{matrix}\right)$	t_{AWL}		3.0x – 100		140	ns
24	$\overline{RD}$ / $\overline{WR}$ fall → $\overline{WAIT}$ hold $\left(\begin{matrix}1+n\\ \text{WAIT mode}\end{matrix}\right)$	t_{CW}	2.0x + 0		160		ns
25	A0 to A23 valid → Port input	t_{APH}		2.5x – 195		5	ns
26	A0 to A23 valid → Port hold	t_{APH2}	2.5x + 50		250		ns
27	$\overline{WR}$ rise → Port valid	t_{CP}		200		200	ns

AC measuring conditions

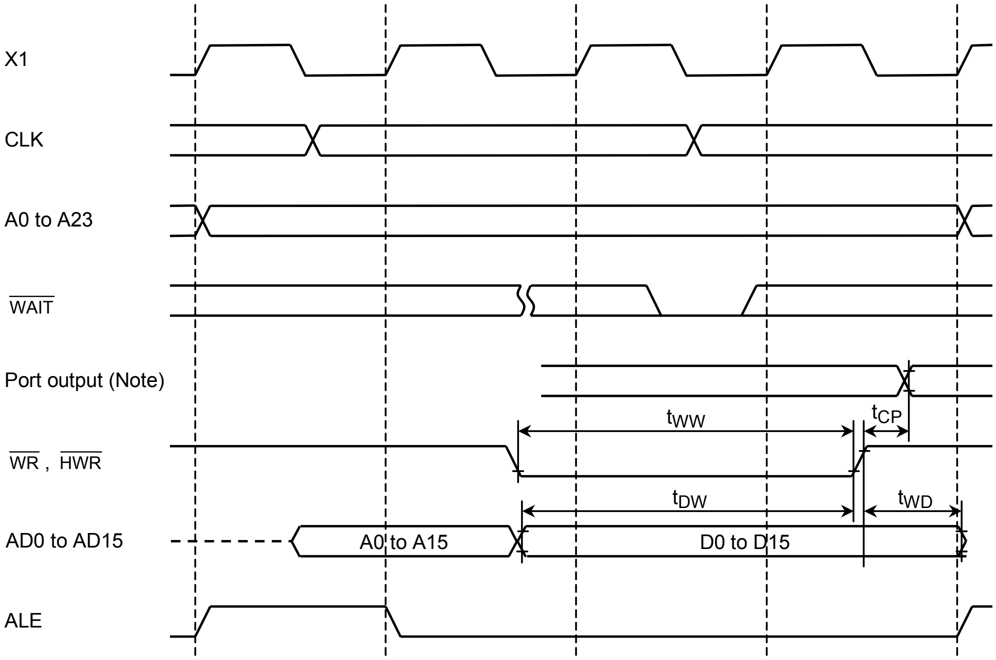
- Output level: High $0.7 \times V_{CC}$/Low $0.3 \times V_{CC}$, CL = 50 pF
- Input level: High $0.9 \times V_{CC}$/Low $0.1 \times V_{CC}$

(3) Read cycle



Note: Since the CPU accesses the internal area to read data from a port, the control signals of external pins such as $\overline{\text{RD}}$ and $\overline{\text{CS}}$ 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.

(4) Write cycle



Note: Since the CPU accesses the internal area to write data to a port, the control signals of external pins such as WR and 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.

4.4 Serial Channel Timing

(1) I/O interface mode

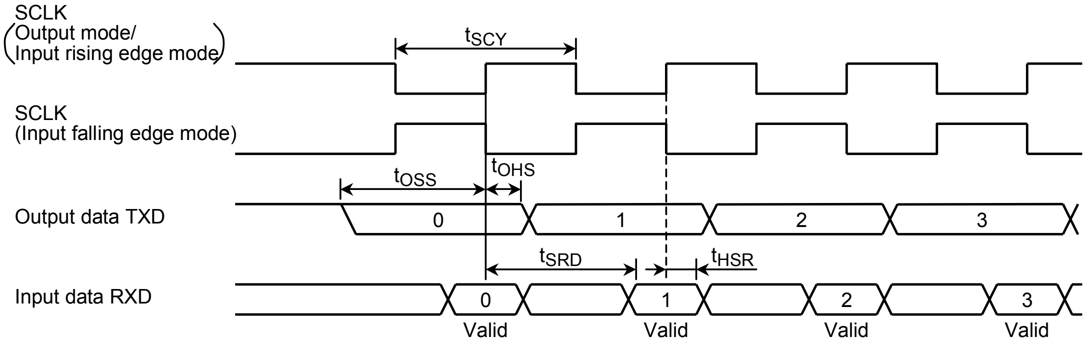
1. SCLK input mode

Parameter	Symbol	Variable		12.5 MHz		20 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK cycle	t_{SCY}	16x		1.28 μs		0.8 μs		ns
Output data → Rising/falling edge of SCLK	t_{OSS}	t_{SCY}/2 – 5x – 50		190		100		ns
SCLK rising/falling edge → Output data hold	t_{OHS}	5x – 100		300		150		ns
SCLK rising/falling edge → Input data hold	t_{HSR}	0		0		0		ns
SCLK rising/falling edge → Effective data input	t_{SRD}		t_{SCY} – 5x – 100		780		450	ns

Note: SCLK rising/falling timing; SCLK rising in the rising mode of SCLK,
SCLK falling in the falling mode of SCLK.

2. SCLK output mode

Parameter	Symbol	Variable		12.5 MHz		20 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK cycle (programmable)	t_{SCY}	16x	8192x	1.28 μs	655.36 μs	0.8 μs	409.6 μs	ns
Output data → SCLK rising edge	t_{OSS}	t_{SCY} – 2x – 150		970		550		ns
SCLK rising edge → Output data hold	t_{OHS}	2x – 80		80		20		ns
SCLK rising edge → Input data hold	t_{HSR}	0		0		0		ns
SCLK rising edge → Effective data input	t_{SRD}		t_{SCY} – 2x – 150		970		550	ns



(2) UART mode (SCLK0 and SCLK1 are external input)

Parameter	Symbol	Variable		12.5 MHz		20 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK cycle	t_{SCY}	4x + 20		340		220		ns
SCLK low level pulse width	t_{SCYL}	2x + 5		165		105		ns
SCLK high level pulse width	t_{SCYH}	2x + 5		165		105		ns

4.5 AD Conversion Characteristics

$AV_{CC} = V_{CC}$, $AV_{SS} = V_{SS}$

Parameter	Symbol	Power Supply	Min	Typ.	Max	Unit
Analog reference voltage (+)	V_{REFH}	$V_{CC} = 5\ V \pm 10\%$	$V_{CC} - 0.2$	V_{CC}	V_{CC}	V
		$V_{CC} = 3\ V \pm 10\%$	$V_{CC} - 0.2$	V_{CC}	V_{CC}	
Analog reference voltage (−)	V_{REFL}	$V_{CC} = 5\ V \pm 10\%$	V_{SS}	V_{SS}	$V_{SS} + 0.2$	
		$V_{CC} = 3\ V \pm 10\%$	V_{SS}	V_{SS}	$V_{SS} + 0.2$	
Analog input voltage range	V_{AIN}		V_{REFL}		V_{REFH}	
Analog current for analog reference voltage <VREFON> = 1	I_{REF} ($V_{REFL} = 0\ V$)	$V_{CC} = 5\ V \pm 10\%$		0.5	1.5	mA
		$V_{CC} = 3\ V \pm 10\%$		0.3	0.9	
<VREFON> = 0		$V_{CC} = 2.7$ to $5.5\ V$		0.02	5.0	μA
Error (except quantization errors)	−	$V_{CC} = 5\ V \pm 10\%$		±1.0	±3.0	LSB
		$V_{CC} = 3\ V \pm 10\%$		±1.0	±5.0	

Note 1: $1LSB = (V_{REFH} - V_{REFL})/2^{10}$ [V].

Note 2: The operation above is guaranteed for $f_{FPH} \geq 4$ MHz.

Note 3: The value I_{CC} includes the current which flows through the AVCC pin.

4.6 Event Counter Input Clock (External Input Clock: TI4, TI5, TI6, and TI7)

Parameter	Symbol	Variable		12.5 MHz		20 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Clock cycle	t_{VCK}	8X + 100		740		500		ns
Low level clock pulse width	t_{VCKL}	4X + 40		360		240		ns
High level clock pulse width	t_{VCKH}	4X + 40		360		240		ns

4.7 Interrupt and Capture Operation

(1) $\overline{NMI}$, INT0 interrupts

Parameter	Symbol	Variable		12.5 MHz		20 MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{NMI}$, INT0 low level pulse width	t_{INTAL}	4X		320		200		ns
$\overline{NMI}$, INT0 high level pulse width	t_{INTAH}	4X		320		200		ns

(2) INT4 to INT7 interrupts and capture

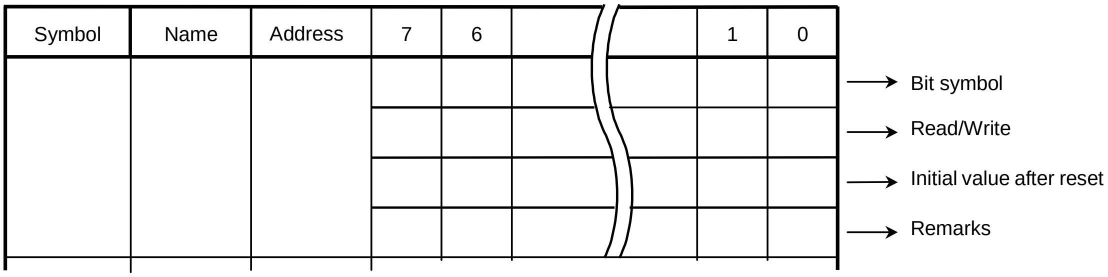
Parameter	Symbol	Variable		12.5 MHz		20 MHz		Unit
		Min	Max	Min	Max	Min	Max	
INT4 to INT7 low level pulse width	t_{INTBL}	4X + 100		420		300		ns
INT4 to INT7 high level pulse width	t_{INTBH}	4X + 100		420		300		ns

5. Table of Special Function Registers

The special function registers (SFRs) include the I/O ports and peripheral control registers allocated to the 128-bytes addresses from 000000H to 00007FH.

(1) I/O port
(2) I/O port control
(3) Clock control
(4) Interrupt control
(5) Bus width/wait control
(6) Timer control
(7) Serial channel control
(8) AD converter control
(9) Watchdog timer control

Configuration of the table



Note: "Prohibit RMW" in the table means that you cannot use RMW instructions on these registers.
Example: When setting only bit0 of register P0CR, "SET 0, (0002H)" cannot be used. The LD (transfer) instruction must be used to write all eight bits.

Table 5 I/O Register Address Map

Address	Name	Address	Name	Address	Name	Address	Name
000000H	P0	20H	TRUN	40H	TREG6L	60H	ADREG04L
1H	P1	21H	(Reserved)	41H	TREG6H	61H	ADREG04H
2H	P0CR	22H	TREG0	42H	TREG7L	62H	ADREG15L
3H	(Reserved)	23H	TREG1	43H	TREG7H	63H	ADREG15H
4H	P1CR	24H	T10MOD	44H	CAP3L	64H	ADREG2L
5H	P1FC	25H	TFFCR	45H	CAP3H	65H	ADREG2H
6H	P2	26H	TREG2	46H	CAP4L	66H	ADREG3L
7H	P3	27H	TREG3	47H	CAP4H	67H	ADREG3H
8H	P2CR	28H	T32MOD	48H	T5MOD	68H	WAITC0
9H	P2FC	29H	TRDC	49H	T5FFCR	69H	WAITC1
AH	P3CR	2AH	(Reserved)	4AH	(Reserved)	6AH	WAITC2
BH	P3FC	2BH		4BH		6BH	(Reserved)
CH	P4	2CH		4CH		6CH	(Reserved)
DH	P5	2DH		4DH		6DH	CKOCR
EH	P4CR	2EH		4EH		6EH	SYSCR0
FH	(Reserved)	2FH		4FH		6FH	SYSCR1
10H	P4FC	30H	TREG4L	50H	SC0BUF	70H	INTE0AD
11H	(Reserved)	31H	TREG4H	51H	SC0CR	71H	INTE45
12H	P6	32H	TREG5L	52H	SC0MOD	72H	INTE67
13H	P7	33H	TREG5H	53H	BR0CR	73H	INTET10
14H	P6CR	34H	CAP1L	54H	SC1BUF	74H	INTET32
15H	P7CR	35H	CAP1H	55H	SC1CR	75H	INTET54
16H	P6FC	36H	CAP2L	56H	SC1MOD	76H	INTET76
17H	(Reserved)	37H	CAP2H	57H	BR1CR	77H	INTEO54
18H		38H	T4MOD	58H	ODE	78H	INTES0
19H		39H	T4FFCR	59H	(Reserved)	79H	INTES1
1AH		3AH	T45CR	5AH		7AH	(Reserved)
1BH		3BH	(Reserved)	5BH		7BH	IIMC
1CH		3CH		5CH	WDMOD	7CH	DMA0V
1DH		3DH		5DH	WDCR	7DH	DMA1V
1EH		3EH		5EH	ADMOD0	7EH	DMA2V
1FH		3FH		5FH	ADMOD1	7FH	DMA3V

Note: Do not access to addresses which do not have register names allocated.

(1) I/O port

Symbol	Name	Address	7	6	5	4	3	2	1	0
P0	PORT0	00H	P07	P06	P05	P04	P03	P02	P01	P00
			R/W							
			Undefined							
			Input mode							
P1	PORT1	01H	P17	P16	P15	P14	P13	P12	P11	P10
			R/W							
			0	0	0	0	0	0	0	0
			Input mode							
P2	PORT2	06H (Prohibit RMW*)	P27	P26	P25	P24	P23	P22	P21	P20
			R/W							
			1	1	1	1	1	1	1	1
			Input mode							
P3	PORT3	07H (Prohibit RMW*)			P35	–	–	P32	P31	P30 (Note)
					R/W					
					1	–	–	1	1	1
					Input mode	(Note) Always write "1".		Input mode	Output mode	
P4	PORT4	0CH	P47	P46	P45	P44	P43	P42	P41	–
			R/W							
			1	1	1	1	1	1	1	–
			Input mode							(Note) Always write "1".
P5	PORT5	0DH			P55	P54	P53	P52	P51	P50
					R					
					Undefined					
					Input mode					
P6	PORT6	12H (Prohibit RMW*)	–	–	P65	P64	P63	P62	P61	P60
			R/W							
			–	–	1	1	1	1	1	1
			(Note) Always write "1".		Input mode					
P7	PORT7	13H	–	–	–	–	–	–	P71	P70
			R/W							
			–	–	–	–	–	–	1	1
			(Note) Always write "1".						Input mode	

Note: When P30 pin is defined as $\overline{\text{RD}}$ signal output mode (P3FC<P30F> = 1), clearing the output latch register P30 to "0" outputs the $\overline{\text{RD}}$ strobe from P30 pin for PSRAM, even when the internal address is accessed. If the output latch register P30 remains "1", the $\overline{\text{RD}}$ strobe is output only when the external address is accessed.

Read/Write

R/W: Either read or write is possible.
R: Only read is possible.
W: Only write is possible.
Prohibit RMW: Prohibit Read Modify Write. (Prohibit RES/SET/TSET/CHG/STCF/ANDCF/ORCF/XORCF instruction.)
Prohibit RMW*: Read-modify-write is prohibited when controlling the PU resistor.

(2) I/O port control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P0CR	PORT0 control	02H (Prohibit RMW)	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
			W							
			0	0	0	0	0	0	0	0
			0: IN 1: OUT (When external access, set as AD7 to AD0 and cleared to “0”.)							
P1CR	PORT1 control	04H (Prohibit RMW)	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
			W							
			0	0	0	0	0	0	0	0
			<<Refer to the “P1FC”>>							
P1FC	PORT1 function	05H (Prohibit RMW)	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
			W							
			0	0	0	0	0	0	0	0
			P1FC/P1CR = 00: IN, 01: OUT, 10: AD15 to AD8, 11: A15 to A8							
P2CR	PORT2 control	08H (Prohibit RMW)	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
			W							
			0	0	0	0	0	0	0	0
			<<Refer to the “P2FC”>>							
P2FC	PORT2 function	09H (Prohibit RMW)	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
			W							
			0	0	0	0	0	0	0	0
			P2FC/P2CR = 00: IN, 01: OUT, 10: A7 to A0, 11: A23 to A16							
P3CR	PORT3 control	0AH (Prohibit RMW)			P35C	–	–	P32C		
					W					
					0	–	–	0		
					0: IN 1: OUT	(Note) Always write “1”.		0: IN 1: OUT		
P3FC	PORT3 function	0BH (Prohibit RMW)		–		–	–	P32F	P31F	P30F
				W		W				
				–		–	–	0	0	0
				(Note) Always write “0”.		(Note) Always write “0”.		0: Port 1: $\overline{HWR}$	0: Port 1: $\overline{WR}$	0: Port 1: $\overline{RD}$
P4CR	PORT4 control	0EH (Prohibit RMW)	P47C	P46C	P45C	P44C	P43C	P42C	P41C	–
			W							
			0	0	0	0	0	0	0	–
			0: IN 1: OUT							(Note) Always write “1”.
P4FC	PORT4 function	10H (Prohibit RMW)	P47F			P44F			P41F	
			W			W			W	
			0			0			0	
			0: Port 1: TO6			0: Port 1: TO4			0: Port 1: TO3	

I/O port control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P6CR	PORT6 control	14H (Prohibit RMW)	–	–	P65C	P64C	P63C	P62C	P61C	P60C
			W							
			–	–	0	0	0	0	0	0
			(Note) Always write “1”.		0: IN 1: OUT					
P7CR	PORT7 control	15H (Prohibit RMW)	–	–	–	–	–	–	P71C	P70C
			W							
			–	–	–	–	–	–	0	0
			(Note) Always write “1”.						0: IN 1: OUT	
P6FC	PORT6 function	16H (Prohibit RMW)			P65F		P63F	P62F		P60F
					W		W			W
					0		0	0		0
					0: Port 1: SCLK1		0: Port 1: TXD1	0: Port 1: SCLK0		0: Port 1: TXD0

(3) Clock control

Symbol	Name	Address	7	6	5	4	3	2	1	0
CKOCR	Clock output control register	006DH							ALEEN	CLKEN
									R/W	
									0	0
									ALE pin control 0: HZ output 1 1: ALE output	CLK pin control 0: HZ output 1: CLK output
SYSCR0	System clock control register 0	006EH	–	–	–	–	–	–	PRCK1	PRCK0
			R/W							
			1	0	1	0	0	0	0	0
			(Note) Always write "1".	(Note) Always write "0".	(Note) Always write "1".	(Note) Always write "0".	(Note) Always write "0".	(Note) Always write "0".	Select prescaler clock 00: f_{FPH} 01: (Reserved) 10: fc/16 11: (Reserved)	
SYSCR1	System clock control register 1	006FH					–	GEAR2	GEAR1	GEAR0
							R/W			
							0	1	0	0
							(Note) Always write "0".	Select gear value of high frequency (fc) 000: fc/1 001: fc/2 010: fc/4 011: fc/8 100: fc/16 101: (Reserved) 110: (Reserved) 111: (Reserved)		

(4) Interrupt control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE0AD	INT0/AD enable register	70H	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
		(Prohibit PMW)	R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE45	INT4/5 enable register	71H	INT5				INT4			
			I5C	I5M2	I5M1	I5M0	I4C	I4M2	I4M1	I4M0
		(Prohibit PMW)	R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTE67	INT6/7 enable register	72H	INT7				INT6			
			I7C	I7M2	I7M1	I7M0	I6C	I6M2	I6M1	I6M0
		(Prohibit PMW)	R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTET10	INTT1/0 enable register	73H	INTT1 (Timer1)				INTT0 (Timer0)			
			IT1C	IT1M2	IT1M1	IT1M0	IT0C	IT0M2	IT0M1	IT0M0
		(Prohibit PMW)	R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTET32	INTT3/2 enable register	74H	INTT3 (Timer3)				INTT2 (Timer2)			
			IT3C	IT3M2	IT3M1	IT3M0	IT2C	IT2M2	IT2M1	IT2M0
		(Prohibit PMW)	R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTET54	INTT5/4 enable register	75H	INTTR5 (TREG5)				INTTR4 (TREG4)			
			IT5C	IT5M2	IT5M1	IT5M0	IT4C	IT4M2	IT4M1	IT4M0
		(Prohibit PMW)	R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTET76	INTT7/6 enable register	76H	INTTR7 (TREG7)				INTTR6 (TREG6)			
			IT7C	IT7M2	IT7M1	IT7M0	IT6C	IT6M2	IT6M1	IT6M0
		(Prohibit PMW)	R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTEO54	INTTO5/4 enable register	77H	INTTO5				INTTO4			
			ITO5C	ITO5M2	ITO5M1	ITO5M0	ITO4C	ITO4M2	ITO4M1	ITO4M0
		(Prohibit PMW)	R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTES0	INTRX0/ TX0 enable register	78H	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
		(Prohibit PMW)	R/W	W			R/W	W		
			0	0	0	0	0	0	0	0
INTES1	INTRX1/ TX1 enable register	79H	INTTX1				INTTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
		(Prohibit PMW)	R/W	W			R/W	W		
			0	0	0	0	0	0	0	0

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	Prohibits interrupt request.

IxxC	Function (Read)	Function (Write)
0	Indicates no interrupt request.	Clears interrupt request flag.
1	Indicates interrupt request.	- - - - - Don’t care - - - - -

Interrupt control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMA0V	Micro DMA 0 request vector	7CH (Prohibit RMW)				DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
						W				
						0	0	0	0	0
						Micro DMA0 start vector				
DMA1V	Micro DMA 1 request vector	7DH (Prohibit RMW)				DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
						W				
						0	0	0	0	0
						Micro DMA1 start vector				
DMA2V	Micro DMA 2 request vector	7EH (Prohibit RMW)				DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
						W				
						0	0	0	0	0
						Micro DMA2 start vector				
DMA3V	Micro DMA 3 request vector	7FH (Prohibit RMW)				DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
						W				
						0	0	0	0	0
						Micro DMA3 start vector				
IIMC	Interrupt input mode control	7BH (Prohibit RMW)			–			I0IE	I0LE	NMIREE
					W			W		
					0			0	0	0
					(Note) Always write “0”.			1: INT0 input enable	0: INT0 edge mode 1: INT0 level mode	1: Operation even at $\overline{\text{NMI}}$ rising edge

(5) Bus-width/Wait control

Symbol	Name	Address	7	6	5	4	3	2	1	0
WAITC0	Block 0 wait control register	68H (Prohibit RMW)				B0BUS	B0W1	B0W0	B0C1	B0C0
						W				
						0	0	0	0	0
						0: 16-bit bus 1: 8-bit bus	00: 2 waits 01: 1 wait 10: (1 + n) waits 11: 0 waits		00: 7F00H to 7FFFH 01: 400000H to 10: 800000H to 11: C00000H to	
WAITC1	Block 1 wait control register	69H (Prohibit RMW)				B1BUS	B1W1	B1W0	B1C1	B1C0
						W				
						0	0	0	0	0
						0: 16-bit bus 1: 8-bit bus	00: 2 waits 01: 1 wait 10: (1 + n) waits 11: 0 waits		00: 880H to 7FFFH 01: 400000H to 10: 800000H to 11: C00000H to	
WAITC2	Block 2 wait control register	6AH (Prohibit RMW)				B2BUS	B2W1	B2W0	B2C1	B2C0
						W				
						0	0	0	1	1
						0: 16-bit bus 1: 8-bit bus	00: 2 waits 01: 1 wait 10: (1 + n) waits 11: 0 waits		00: 8000H to 01: 400000H to 10: 800000H to 11: C00000H to	

(6) Timer control (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
TRUN	Timer run control register	20H	PRRUN		T5RUN	T4RUN	T3RUN	T2RUN	T1RUN	T0RUN
			R/W		R/W					
			0		0	0	0	0	0	0
			Prescaler & timer run/stop control 0: Stop & clear 1: Run (count up)							
TREG0	8-bit timer register 0	22H (Prohibit RMW)	–							
			W							
			Undefined							
TREG1	8-bit timer register 1	23H (Prohibit RMW)	–							
			W							
			Undefined							
T10 MOD	8-bit timers 0 and 1 source CLK & MODE control register	24H	T10M1	T10M0			T1CLK1	T1CLK0	T0CLK1	T0CLK0
			R/W				R/W			
			0	0			0	0	0	0
			00: 8-bit timer 01: 16-bit timer 10: – 11: –				00: TO0TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: Don’t set 01: ϕT1 10: ϕT4 11: ϕT16	
TFFCR	8-bit timer flip-flop control register	25H	TFF3C1	TFF3C0	TFF3IE	TFF3IS	TFF1C1	TFF1C0	TFF1IE	TFF1IS
			W		R/W		W		R/W	
			1	1	0	0	1	1	0	0
			00: Invert TFF3 01: Set TFF3 10: Clear TFF3 11: Don’t care		1: TFF3 invert enable	TFF3 inversion source 0: Timer 2 1: Timer 3	00: Invert TFF1 01: Set TFF1 10: Clear TFF1 11: Don’t care		1: TFF1 invert enable	TFF1 inversion source 0: Timer 0 1: Timer 1
TREG2	8-bit timer register 2	26H (Prohibit RMW)	–							
			W							
			Undefined							
TREG3	8-bit timer register 3	27H (Prohibit RMW)	–							
			W							
			Undefined							
T32 MOD	8-bit timers 2 and 3 source CLK & MODE control register	28H	T32M1	T32M0	PWM21	PWM20	T3CLK1	T3CLK0	T2CLK1	T2CLK0
			R/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 PWM 10: 2^7 – 1 cycle 11: 2^8 – 1		00: TO2TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: Don’t set 01: ϕT1 10: ϕT4 11: ϕT16	
TRDC	Timer register double buffer control register	29H							TR2DE	–
									R/W	
									0	0
									0: Double buffer disable 1: Double buffer enable	(Note) Always write “0”.

Timer control (2/3)

<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">TREG4L</td><td rowspan="3">16-bit timer register 4 low</td><td rowspan="3">30H
(Prohibit 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">TREG4H</td><td rowspan="3">16-bit timer register 4 high</td><td rowspan="3">31H
(Prohibit 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">TREG5L</td><td rowspan="3">16-bit timer register 5 low</td><td rowspan="3">32H
(Prohibit 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">TREG5H</td><td rowspan="3">16-bit timer register 5 high</td><td rowspan="3">33H
(Prohibit 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">CAP1L</td><td rowspan="3">Capture register 1 low</td><td rowspan="3">34H</td><td colspan="8">–</td></tr>
<tr><td colspan="8">R</td></tr>
<tr><td colspan="8">Undefined</td></tr>
<tr><td rowspan="3">CAP1H</td><td rowspan="3">Capture register 1 high</td><td rowspan="3">35H</td><td colspan="8">–</td></tr>
<tr><td colspan="8">R</td></tr>
<tr><td colspan="8">Undefined</td></tr>
<tr><td rowspan="3">CAP2L</td><td rowspan="3">Capture register 2 low</td><td rowspan="3">36H</td><td colspan="8">–</td></tr>
<tr><td colspan="8">R</td></tr>
<tr><td colspan="8">Undefined</td></tr>
<tr><td rowspan="3">CAP2H</td><td rowspan="3">Capture register 2 high</td><td rowspan="3">37H</td><td colspan="8">–</td></tr>
<tr><td colspan="8">R</td></tr>
<tr><td colspan="8">Undefined</td></tr>
<tr><td rowspan="4">T4MOD</td><td rowspan="4">16-bit timer 4 source CLK & MODE control register</td><td rowspan="4">38H</td><td></td><td></td><td>CAP1IN</td><td>CAP12M1</td><td>CAP12M0</td><td>CLE</td><td>T4CLK1</td><td>T4CLK0</td></tr>
<tr><td></td><td></td><td>W</td><td colspan="5">R/W</td></tr>
<tr><td></td><td></td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td></td><td></td><td>0: Software capture
1: Don’t care</td><td colspan="2">Capture timing
00: Disable
01: TI4↑ TI5↑
10: TI4↑ TI4↓
11: TFF1↑ TFF1↓</td><td>1: UC4 clear enable</td><td colspan="2">Source clock
00: TI4
01: φT1
10: φT4
11: φT16</td></tr>
<tr><td rowspan="5">T4FFCR</td><td rowspan="5">16-bit timer 4 flip-flop control register</td><td rowspan="5">39H</td><td></td><td></td><td>CAP2T4</td><td>CAP1T4</td><td>EQ5T4</td><td>EQ4T4</td><td>TFF4C1</td><td>TFF4C0</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>1</td><td>1</td></tr>
<tr><td rowspan="2"></td><td rowspan="2"></td><td colspan="4">TFF4 invert trigger
0: Trigger disable
1: Trigger enable</td><td colspan="2" rowspan="2">TFF4 control
00: Invert TFF4
01: Set TFF4
10: Clear TFF4
11: Don’t care</td></tr>
<tr><td>Invert when the UC value is loaded to CAP2.</td><td>Invert when the UC value is loaded to CAP1.</td><td>Invert when the UC matches TREG5.</td><td>Invert when the UC matches TREG4.</td></tr>
</table>

Timer control (3/3)

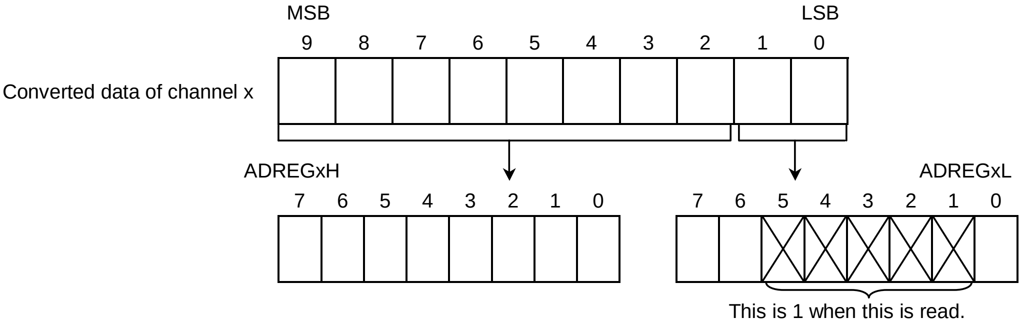
Symbol	Name	Address	7	6	5	4	3	2	1	0
T45CR	T4, T5 control register	3AH	QCU						DB6EN	DB4EN
			R/W						R/W	
			0						0	0
			Watchdog/ Warm-up timer control						Double buffer 0: Disable 1: Enable	
									Double buffer of TREG6	Double buffer of TREG4
TREG6L	16-bit timer register 6 low	40H (Prohibit RMW)	–							
			W							
			Undefined							
TREG6H	16-bit timer register 6 high	41H (Prohibit RMW)	–							
			W							
			Undefined							
TREG7L	16-bit timer register 7 low	42H (Prohibit RMW)	–							
			W							
			Undefined							
TREG7H	16-bit timer register 7 high	43H (Prohibit RMW)	–							
			W							
			Undefined							
CAP3L	Capture register 3 low	44H	–							
			R							
			Undefined							
CAP3H	Capture register 3 high	45H	–							
			R							
			Undefined							
CAP4L	Capture register 4 low	46H	–							
			R							
			Undefined							
CAP4H	Capture register 4 high	47H	–							
			R							
			Undefined							
T5MOD	16-bit timer 5 source CLK & mode control register	48H			CAP3IN	CAP34M1	CAP34M0	CLE	T5CLK1	T5CLK0
					W	R/W				
					1	0	0	0	0	0
					0: Software capture 1: Don't care	Capture timing 00: Disable 01: TI6↑ TI7↑ 10: TI6↑ TI6↓ 11: TFF1↑ TFF1↓		1: UC5 clear enable	Source clock 00: TI6 01: φT1 10: φT4 11: φT16	
T5FFCR	16-bit timer 5 flip-flop control register	49H			CAP4T6	CAP3T6	EQ7T6	EQ6T6	TFF6C1	TFF6C0
					R/W				W	
					0	0	0	0	1	1
					TFF6 invert trigger 0: Trigger disable 1: Trigger enable				TFF6 control 00: Invert TFF6 01: Set TFF6 10: Clear TFF6 11: Don't care	
					Invert when the UC value is loaded to CAP4.	Invert when the UC value is loaded to CAP3.	Invert when the UC matches TREG7.	Invert when the UC matches TREG6.		

(7) Serial channel control

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC0BUF	Serial channel 0 buffer register	50H (Prohibit RMW)	RB7 TB7	RB6 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 RB1	RB0 TB0
			R (Receiving)/W (Transmission)							
			Undefined							
SC0CR	Serial channel 0 control register	51H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Cleared to 0 by reading)			R/W	
			undefined	0	0	0	0	0	0	0
			Receiving data bit8	Parity 0: Odd 1: Even	1: Parity enable	1: Error			0: SCLK0 ↑ 1: SCLK0 ↓	1: Input SCLK0 pin
						Overrun	Parity	Framing		
SC0MOD	Serial channel 0 mode control register	52H	TB8	CTSE0	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			Undefined	0	0	0	0	0	0	0
			Transmisson data bit8	1: CTS0 enable	1: Receive enable	1: Wake up enable	00: I/O interface 01: UART 7-bit 10: UART 8-bit 11: UART 9-bit		00: TO2 trigger 01: Baud rate generator 0 10: Internal clock ϕ1 11: External clock SCLK0	
BR0CR	Baud rate 0 control register	53H	–		BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
			R/W		R/W					
			0		0	0	0	0	0	0
			Fix at ”0”		00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Set frequency divisor 0000: 16 divisions 0001 to 1111 } 1 to 15 divisions			
SC1BUF	Serial channel 1 buffer register	54H (Prohibit RMW)	RB7 TB7	RB6 TB6	RB5 TB5	RB4 TB4	RB3 TB3	RB2 TB2	RB1 RB1	RB0 TB0
			R (Receiving)/W (Transmission)							
			Undefined							
SC1CR	Serial channel 1 control register	55H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Cleared to 0 by reading.)			R/W	
			Undefined	0	0	0	0	0	0	0
			Receiving data bit8	Parity 0: Odd 1: Even	1: Parity enable	1: Error			0: SCLK1 ↑ 1: SCLK1 ↓	1: Input SCLK1 pin
						Overrun	Parity	Framing		
SC1MOD	Serial channel 1 mode control register	56H	TB8	CTSE1	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			undefined	0	0	0	0	0	0	0
			Transmisson data bit8	1: CTS1 enable	1: Receive enable	1: Wake up enable	00: I/O interface 01: UART 7-bit 10: UART 8-bit 11: UART 9-bit		00: TO2 trigger 01: Baud rate generator 1 10: Internal clock ϕ1 11: External clock SCLK1	
BR1CR	Baud rate 1 control register	57H	–		BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
			R/W		R/W					
			0		0	0	0	0	0	0
			Fix at ”0”		00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Set frequency divisor 0000: 16 divisions 0001 to 1111 } 1 to 15 divisions			
ODE	Serial open drain enable	58H					–	–	ODE63	ODE60
							R/W			
							0	0	0	0
							(Note) Always write “0”.		1: P63 open drain	1: P60 open drain

(8) AD converter control

Symbol	Name	Address	7	6	5	4	3	2	1	0
ADMOD0	AD mode control register 0	5EH	EOCF	ADBF	–	–	ITM0	REPET	SCAN	ADS
			R		R/W					
			0	0	0	0	0	0	0	0
			1: End	1: Busy	(Note) Always write “0”.		0: Every conversion 1: Every four conversion	0: Single 1: Repeat	0: Fixed-channel 1: Scan	1: Start (Note) Always read “0”.
ADMOD1	AD mode control register 1	5FH	VREFON				ADTRGE	ADCH2	ADCH1	ADCH0
			R/W				R/W			
			1				0	0	0	0
			0: OFF 1: ON				External trigger start control 0: Disable 1: Enable	Analog input channel selection		
*1) ADREG04L	AD conversion result register 0/4 low	60H	ADR01	ADR00						ADR0RF
			R							R
			Undefined							0
			Stores lower two bits of AD conversion result							Conversion result stored flag
ADREG04H	AD conversion result register 0/4 high	61H	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
			R							
			Undefined							
			Stores upper eight bits of AD conversion result							
*1) ADREG15L	AD conversion result register 1/5 low	62H	ADR11	ADR10						ADR1RF
			R							R
			Undefined							0
			Stores lower two bits of AD conversion result							Conversion result stored flag
ADREG15H	AD conversion result register 1/5 high	63H	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
			R							
			Undefined							
			Stores upper eight bits of AD conversion result							
*1) ADREG2L	AD conversion result register 2 low	64H	ADR21	ADR20						ADR2RF
			R							R
			Undefined							0
			Stores lower two bits of AD conversion result							Conversion result stored flag
ADREG2H	AD conversion result register 2 high	65H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
			R							
			Undefined							
			Stores upper eight bits of AD conversion result							
*1) ADREG3L	AD conversion result register 3 low	66H	ADR31	ADR30						ADR3RF
			R							R
			Undefined							0
			Stores lower two bits of AD conversion result							Conversion result stored flag
ADREG3H	AD conversion result register 3 high	67H	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
			R							
			Undefined							
			Stores upper eight bits of AD conversion result							



This is 1 when this is read.

*1: Data to be stored in AD Conversion Result Reg Low are the lower 2 bits of the conversion result. The contents of the 5 to 1 bits of this register are always read as “1”.
Bit0 conversion result stored flag bit <ADRxRF>
<ADRxRF> is set to “1” when the AD conversion result is stored.
Reading either the ADREGxH or the ADREGxL registers clears <ADRxRF> to “0”.

(9) Watchdog timer

Symbol	Name	Address	7	6	5	4	3	2	1	0
WDMOD	Watchdog timer mode control register	5CH	WDTE	WDTP1	WDTP0	WARM	HALTM1	HALTM0	RESCR	DRVE
			R/W							
			1	0	0	0	0	0	0	0
			1: WDT enable	00: $2^{15}/f_{SYS}$ 01: $2^{17}/f_{SYS}$ 10: $2^{19}/f_{SYS}$ 11: $2^{21}/f_{SYS}$		Warm-up time 0: 2^{14}/Inputted frequency 1: 2^{16}/Inputted frequency	HALT mode 00: RUN mode 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		1: Connect internally WDT out to reset pin	1: Drive the pin in STOP mode
WDCR	Watchdog timer control register	5DH	–							
			W							
			–							
			B1H: WDT disable code 4EH: WDT clear code							

6. Port Section Equivalent Circuit Diagram

- Reading the circuit diagram

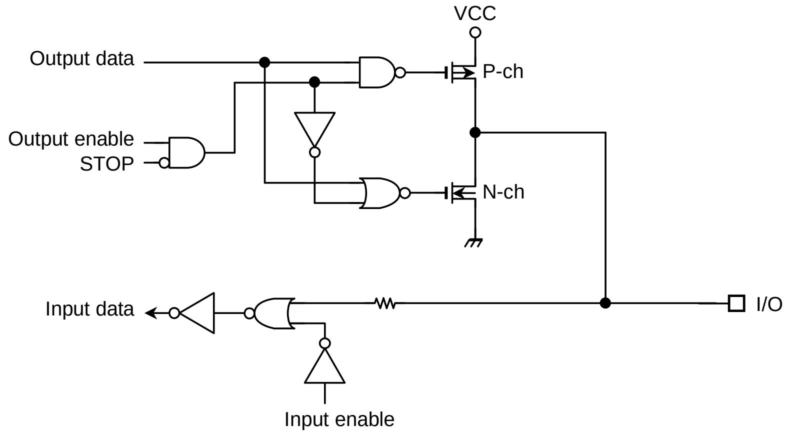
Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.

The dedicated signal is described below.

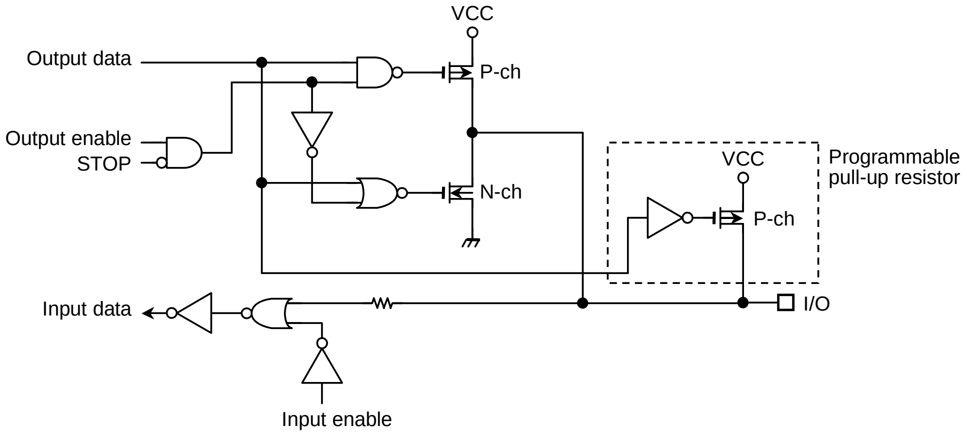
STOP: This signal becomes active "1" when the HALT mode setting register is set to the STOP mode (WDMOD<HALTM1:0> = 0, 1) and the CPU executes the HALT instruction. When the drive enable bit WDMOD<DRVE> is set to "1", however, STOP remains at "0".

- The input protection resistance ranges from several tens of ohms to several hundreds of ohms.

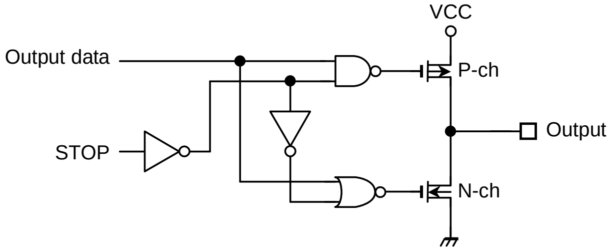
■ P0 (AD0 to AD7), P1 (AD8 to AD15/A8 to A15), P4, and P7



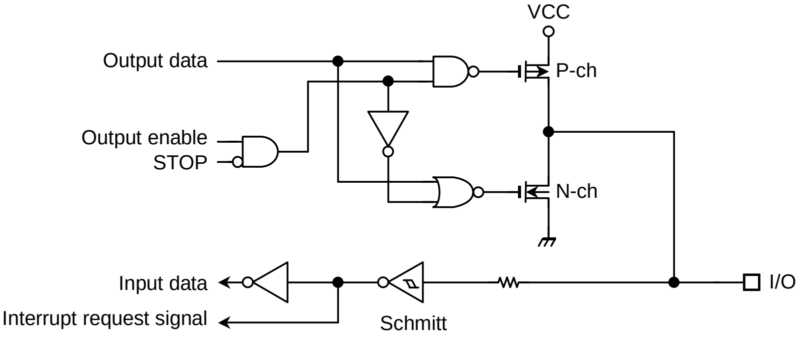
■ P2 (A16 to A23/A0 to A7), P32, P61, P62, P64, and P65



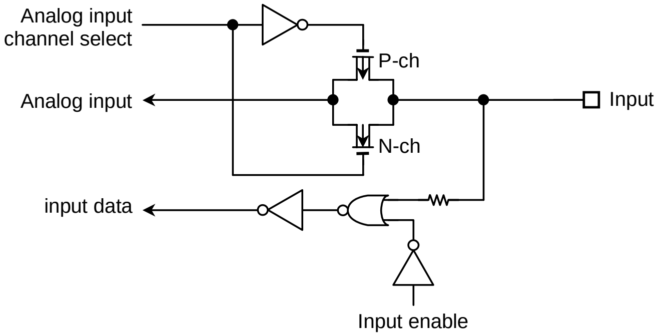
- P30 ($\overline{\text{RD}}$) and P31 ($\overline{\text{WR}}$)



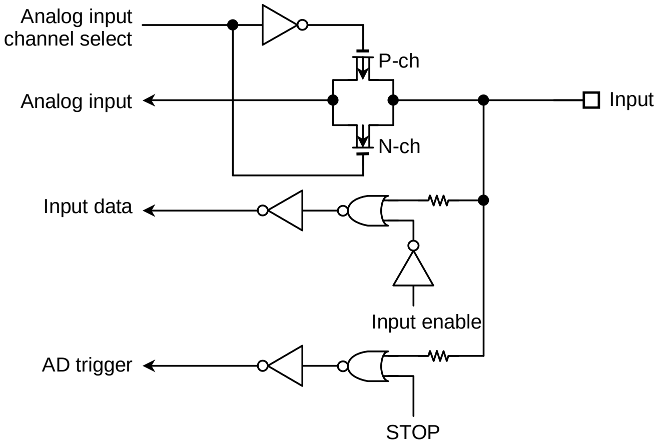
- P35 (INT0)



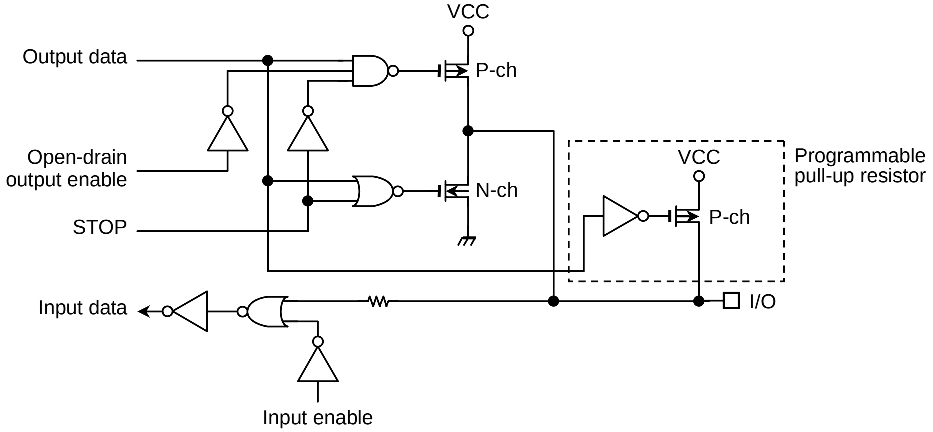
- P50 to P52 (AN0 to AN2), P54 (AN4), and P55 (AN5)



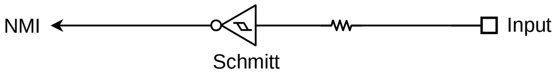
- P53 (AN3/ $\overline{\text{ADTRG}}$)



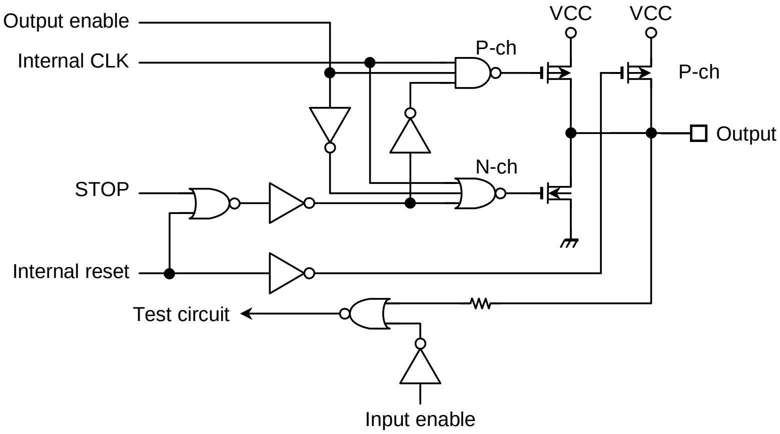
- P60 (TXD0) and P63 (TXD1)



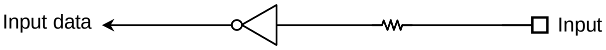
- $\overline{\text{NMI}}$



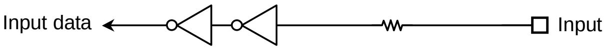
- CLK



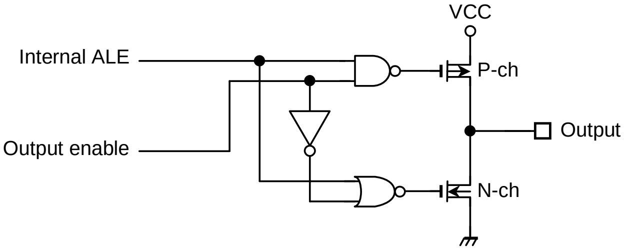
- $\overline{\text{EA}}$



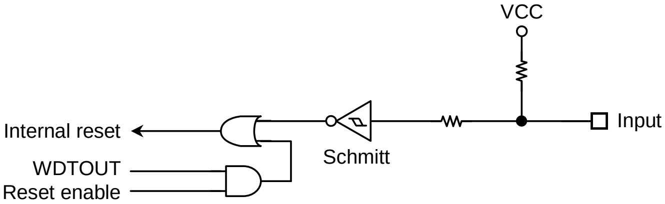
- AM8/$\overline{\text{AM16}}$



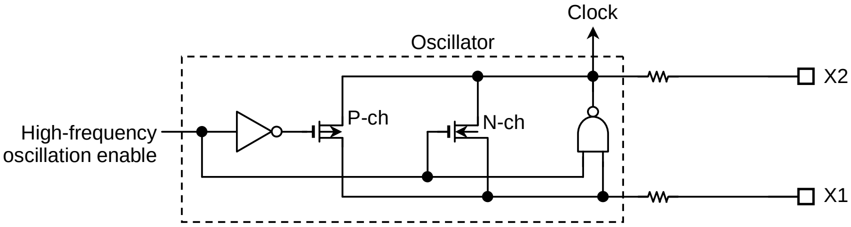
- ALE



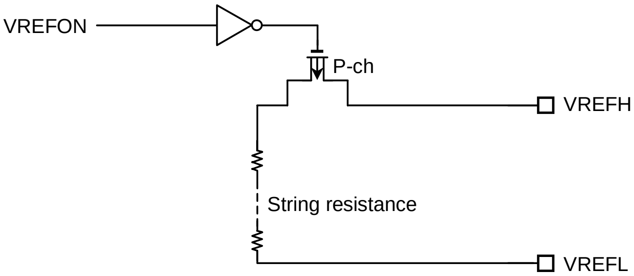
- $\overline{\text{RESET}}$



- X1 and X2



- VREFH and VREFL



7. Points of Note and Restriction

(1) Notation

1. Explanation of a built-in I/O register: register symbol <bit symbol>
   e.g.) TRUN<T0RUN> … Bit T0RUN of Register TRUN
2. Read, modify and write instruction

   An instruction in which the CPU executes following by one instruction.

   1. CPU reads data of the memory.
   2. CPU modifies the data.
   3. CPU writes the data to the same memory.

   ex1) SET 3, (TRUN) … Set bit3 of TRUN
   ex2) INC 1, (100H) … Increment the data of 100H

   - A sample read, modify and write instructions using the TLCS-900

     Exchange

EX	(mem), R		

     Arithmetic operation

ADD	(mem), R/#	ADC	(mem), R/#
SUB	(mem), R/#	SBC	(mem), R/#
INC	#3, (mem)	DEC	#3, (mem)

     Logical operation

AND	(mem), R/#	OR	(mem), R/#
XOR	(mem), R/#		

     Bit manipulation

STCF	#3/A, (mem)	SET	#3, (mem)
RES	#3, (mem)	TSET	#3, (mem)
CHG	#3, (mem)		

     Rotate and shift

RLC	(mem)	RRC	(mem)
RL	(mem)	RR	(mem)
SLA	(mem)	SRA	(mem)
SLL	(mem)	SRL	(mem)
RLD	(mem)	RRD	(mem)

3. fc, f_{FPH}, f_{SYS}, 1 state

   The clock frequency input from pins X1 and X2 is called fc. The clock frequency selected by SYSCR1<GEAR2:0> is called system clock f_{FPH}, and the clock frequency given by f_{FPH} divided by 2 is called f_{SYS}. One cycle of f_{SYS} is called 1 state.

(2) Care points

1. $\overline{\text{EA}}$, AM8/$\overline{\text{AM16}}$ pin

   Fix these pins V_{CC} unless changing voltage.

2. HALT mode (IDLE1)

   When IDLE1 mode (oscillator operation only) is used, clear TRUN<PRRUN> to "0" to stop prescaler before "HALT" instruction is executed.

3. Warm-up counter

   The warm-up counter operates when STOP mode is released even if the system is using an external oscillator. As a result, it takes warm-up time from inputting the releasing request to outputting the system clock.

4. Programmable pull-up resistor

   The programmable pull-up resistors can be turned ON/OFF by the program when the ports are used as input ports. When the ports are used as outputs, they can not be selected ON/OFF by the program.

   The data registers (e.g. P6 register ...) are used for the pull-up resistors ON/OFF. Consequently, Read-modify-write instructions are prohibited.

5. Watchdog timer

   The watchdog timer starts operation immediately after the reset is released. When the watchdog timer is not used, disable it.

6. AD converter

   The string register between VREFH and VREFL pins can be cut by a program to reduce power consumption. When the Standby mode is used, disable the resistor using the program before the "HALT" instruction is executed.

7. CPU (Micro DMA)

   Only the "LDC cr, r", "LDC r, cr" instructions can be used to access the control registers in the CPU like the transfer source address register (DMASn).

8. POP SR instruction

   Please execute POP SR instruction during DI condition.

9. Releasing the HALT mode by requesting an interruption

   Usually, interrupts can release all halts status. However, the interrupts = ($\overline{\text{NMI}}$, INT0) which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 3 clocks of f_{FPH}) with IDLE1 or STOP mode (IDLE2/RUN are 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.

8. TMP93XX32 Different Points

Item	TMP93CS32	TMP93PW32
Built-in ROM	64-Kbyte Mask ROM (FF0000H to FFFFFFH)	128-Kbyte OTP (FE0000H to FFFFFFH)
Built-in RAM	2-Kbyte (80H to 87FH)	4-Kbyte (80H to 107FH)
CS1 mapping area (WAITC1<B1C1:0> = 00)	880H to 7FFFH	1080H to 7FFFH
CS2 mapping area (WAITC2<B2C1:0> = 11)	C00000H to FEFFFFH	C00000H to FDFFFFH

9. Package Dimensions

P-QFP64-1414-0.80A

Unit: mm