TOSHIBA Original CMOS 16-Bit Microcontroller

TLCS-900/L1 Series

TMP91CP27

www.DataSheet4U.com

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 stats. However, the interrupts = (\overline{NMI}) , INTO, INTRTC), 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 5 clocks of f_{FPH}) with IDLE1 or STOP mode (IDLE2 is not applicable to this case). (In this case, an interupt request is kept on hold internally.)

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

CMOS 16-Bit Microcontrollers TMP91CP27U

1. Outline and Features

TMP91CP27 is a high-speed 16-bit microcontroller designed for the control of various mid- to large-scale equipment.

TMP91CP27U comes in a 64-pin flat package. Listed below are the features.

- (1) High-speed 16-bit CPU (900/L1 CPU)
 - Instruction mnemonics are upward-compatible with TLCS-90/900
 - 16 Mbytes of linear address space
 - General-purpose registers and register banks
 - 16-bit multiplication and division instructions; bit transfer and arithmetic instructions
 - Micro DMA: 4 channels (1.0 µs/2 bytes at 16 MHz)
- (2) Minimum instruction execution time: 148 ns (at 27 MHz)
- (3) Built-in RAM: 4 Kbytes Built-in ROM: 48 Kbytes
- (4) External memory expansion
 - Expandable up to 16 Mbytes (Shared program/data area)
 - Can simultaneously support 8-/16-bit width external data bus (Dynamic data bus sizing)
- (5) 8-bit timers: 6 channels
- (6) 16-bit timers: 1 channel
- (7) General-purpose serial interface: 2 channels
 - UART/Synchronous mode: 2 channels
 - IrDA Ver.1.0 (115.2 Kbps) mode selectable: 1 channel
- (8) Serial bus interface: 1 channel
 - I²C bus mode/clock synchronous mode selectable

030619EBP1

- The information contained herein is subject to change without notice.
- The information contained herein is subject to change without back. 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
- 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.

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.



www.DataSheet4U.com

Purchase of TOSHIBA I²C components conveys a license under the Philips I²C Patent Rights to use these components in an I²C system, provided that the system conforms to the I²C Standard Specification as defined by Philips.

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.

- (9) 10-bit AD converter (Sample hold circuit is inside): 4 channels
- (10) Watchdog timer
- (11) Timer for real time clock (RTC)
- (12) Chip select/wait controller: 4 blocks
- (13) Interrupts: 34 interrupts
 - 9 CPU interrupts: Software interrupt instruction and illegal instruction
 - 21 internal interrupts: 7 priority levels are selectable
 - 4 external interrupts: 7 priority levels are selectable (among 3 interrupts are selectable edge mode)
- (14) Input/output ports: 53 pins
- (15) Standby function
 - Three HALT modes: IDLE2 (Programmable), IDLE1 and STOP
- (16) Clock controller
 - Clock gear function: Select a high-frequency clock fc to fc/16
 - RTC (fs = 32.768 KHz)
- (17) Operating voltage
 - VCC = 2.7 V to 3.6 V (fc max = 27 MHz)
 - VCC = 1.8 V to 3.6 V (fc max = 10 MHz)
- (18) Package
 - P-LQFP64-1010-0.50D



www.DataSheet4U.com

(): Initial function after resert

Figure 1.1 TMP91CP27 Block Diagram

2. Pin Assignment and Pin Functions

The assignment of input/output pins for the TMP91CP27U, their names and functions are as follows:

2.1 Pin Assignment Diagram

Figure 2.1 shows the pin assignment of the TMP91CP27U.



Figure 2.1 Pin Assignment Diagram (64-pin LQFP)

2.2 Pin names and Functions

The names of the input/output pins and their functions are described below. Table 2.2.1 to Table 2.2.3 show pin names and functions.

	Pin Names	Number of Pins	I/O	Functions
P00 to P07		8	I/O	Port 0: I/O port that allows I/O to be selected at the bit level
	AD0 to AD7		I/O	Address data (Lower): 0 to 7 of address/data bus
	P10 to P17	8	I/O	Port1: I/O port that allows I/O to be selected at the bit level
	AD8 to AD15		I/O	Address data (Upper): 8 to 15 of address/data bus
	A8 to A15		Output	Address: 8 to 15 of address bus
	P20 to P25	6	I/O	Port 2: I/O port that allows I/O to be selected at the bit level
	A0 to A5		Output	Address: 0 to 5 of address bus
	A16 to A21		Output	Address: 16 to 21 of address bus
	P30	1	Output	Port 30: Output port
	RD		Output	Read: Strobe signal for reading external memory when read internal area also, output \overline{RD} by setting to P3 <p30> = 0, P3FC<p30f> = 1.</p30f></p30>
	P31	1	Output	Port 31: Output port
	WR		Output	Write: Strobe signal for writing data to pins AD0 to AD7
	P32	1	I/O	Port 32: I/O port (with pull-up resistor)
	HWR		Output	High write: Strobe signal for writing data to pins AD8 to AD15
	P40	1	I/O	Port 40: I/O port (with pull-up resistor)
	CS0		Output	Chip select 0: Outputs "0" when address is within specified address area.
	P41	1	I/O	Port41: I/O port (with pull-up resistor)
	CS1		Output	Chip select 1: Outputs "0" when address is within specified address area.
	P42	1	I/O	Port 42: I/O port (with pull-up resistor)
	CS2		Output	Chip select 2: Outputs "0" when address is within specified address area.
	P50 to P53	4	Input	Port 5: Input port
	AN0 to AN3		Input	Analog input: Analog input pins of the AD converter
	ADTRG		Input	AD trigger: Pin used to request AD start (Shared with P53).
	P60	1	I/O	Port 60: I/O port
	SCK		I/O	Serial bus interface clock I/O at SIO mode
	P61	1	I/O	Port 61: I/O port
www.DataSheet	\$0.com		Output	Serial bus interface send data at SIO mode
	SDA		I/O	Serial bus interface send/receive data at I ² C mode
				Open-drain output mode by programmable
	P62	1	I/O	Port 62: I/O port
	SI		Input	Serial bus interface receive data at SIO mode
	SCL		I/O	Serial bus interface clock I/O at I ² C mode
				Open-drain output mode by programmable
	P63	1	I/O	Port 63: I/O port (Schmitt input)
	INT0		Input	Interrupt request pin 0: Interrupt request pin with selectable level/rising/falling edge

Table 2.2.1	Pin Names and Functions	(1/3)
-------------	-------------------------	-------

		Number		
	Pin Names	of Pins	I/O	Functions
	P70	1	I/O	Port 70: I/O port
	TA0IN		Input	8-bit timer 0 input: Input pin of 8-bit timer TMRA0
	P71	1	I/O	Port 71: I/O port
	TA1OUT		Output	8-bit timer 1 output: Output pin of 8-bit timer TMRA0 or TMRA1
	P72	1	I/O	Port 72: I/O port
	TA3OUT		Output	8-bit timer 3 output: Output pin of 8-bit timer TMRA2 or TMRA3
	P73	1	I/O	Port 73: I/O port
	TA4IN		Input	8-bit timer 4 input: Input pin of 8-bit timer TMRA4
	P74	1	I/O	Port 74: I/O port
	TA5OUT		Output	8-bit timer 5 output: Output pin of 8-bit timer TMRA4 or TMRA5
	P80	1	I/O	Port 80: I/O port
	TB0IN0		Input	16-bit timer 0 Input 0: Input of count/capture trigger in 16-bit timer TMRB0
	INT5		Input	Interrupt request pin 5: Interrupt request pin with selectable rising/falling edge
	P81	1	I/O	Port 81: I/O port
	TB0IN1		Input	16-bit timer 0 input 1: Input of count/capture trigger in 16-bit timer TMRB0
	INT6		Input	Interrupt request pin 6: Interrupt request pin of rising edge
	P82	1	I/O	Port 82: I/O port
	TB0OUT0		Output	16-bit timer 0 output 0: Outpit pin of 16-bit timer TMRB0
	P83	1	I/O	Port 83: I/O port
	TB0OUT1		Output	16-bit timer 0 output 1: Output pin of 16-bit timer TMRB0
	P90	1	I/O	Port 90: I/O port
	TXD0		Output	Serial 0 send data: Open-drain output pin by programmable
	P91	1	I/O	Port 91: I/O port
	RXD0		Input	Serial 0 receive data
	P92	1	I/O	Port 92: I/O port
	SCLK0		I/O	Serial 0 clock I/O
	CTS0		Input	Serial 0 data send enable (Clear to send)
	P93	1	I/O	Port 93: I/O port
	TXD1		Output	Serial 1 send data: Open-drain output pin by programmable
	P94	1	I/O	Port 94: I/O port
	RXD1		Input	Serial 1 receive data
www.DataShee	4 095 om	1	I/O	Port 95: I/O port
	SCLK1		I/O	Serial 1 clock I/O
	CTS1		Input	Serial 1 data send enable (Clear to send)
	P96	1	I/O	Port 96: I/O port. Open-drain output pin
	XT1		Input	Low-frequency oscillator connection pin
	P97	1	I/O	Port 97: I/O port. Open-drain output pin
	XT2		Output	Low-frequency oscillator connection pin

Table 222	Pin Names and Functions	(2/3)
		(2,0)

Pin Names	Number of Pins	I/O	Functions
ALE	1	Output	Address latch enable (It can be set as prohibition of an output for noise reduction.)
NMI	1	Input	Non-maskable interrupt request pin: Interrupt request pin with programmable falling edge level or with both edge levels programmable (Schmitt input).
AM0, AM1	2	Input	Operation mode: Fixed to AM1 = "1" and AM0 = "1".
RESET	1	Input	Reset: Initialize LSI. (Schmitt input, with pull-up resistor)
AVCC	1		Pin used to both power supply pin for AD converter and standard power supply for AD converter (H).
AVSS	1		Pin used to both GND pin for AD converter (0 V) and standard power supply pin for AD converter (L).
X1/X2	2	I/O	High-frequency oscillator connection pin.
DVCC	1		Power supply pins (All DVCC pins should be connected with the power supply pin.)
DVSS	1		GND pins (All pins shuold be connected with GND (0 V).)

Table 2.2.3	Pin	Names	and	Functions	(3/3)
-------------	-----	-------	-----	-----------	-------

3. Operation

This following describes block by block the functions and operation of the TMP91CP27.

3.1 CPU

The TMP91CP27 incorporates a high-performance 16-bit CPU (The 900/L1-CPU). For CPU operation, see the "TLCS-900/L1 CPU".

The following describe the unique function of the CPU used in the TMP91CP27; these functions are not covered in the TLCS-900/L1 CPU section.

3.1.1 Reset

When resetting the TMP91CP27 microcontroller, ensure that the power supply voltage is within the operating voltage range, and that the internal high-frequency oscillator has stabilized. Then hold the $\overrightarrow{\text{RESET}}$ input to low level at least for 10 system clocks (80 µs at 4 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 fSYS is set to fc/32 (= $fc/16 \times 1/2$).

When the reset is accept, the CPU:

• Sets as follows the program counter (PC) in accordance with the reset vector stored at address FFFF00H to FFFF02H:

PC<7:0>	←	Value at FFFF00H address
PC<15:8>	←	Value at FFFF01H address
PC<23:16>	←	Value at FFFF02H address

- Sets the stack pointer (XSP) to 100H.
- Sets bits<IFF2:0> of the status register (SR) to 111 (Sets the interrupt level mask register to level 7).
- Sets the <MAX> bit of the status register (SR) to 1 (MAX mode).
- Clears bits<RFP2:0> of the status register (SR) to 000 (Sets the register bank to 0).

When reset is released, the CPU starts executing instructions in accordance with the program counter settings. CPU internal registers not mentioned above do not change when the reset is released.

When the reset is accepted, the CPU sets internal I/O, ports, and other pins as follows.

- Initializes the internal I/O registers.
- Sets the port pins, including the pins that also act as internal I/O, to general-purpose input or output port mode.
- Sets ALE pin to high impedance.
- Note: The CPU internal register (except to PC, SR, XSP in CPU) and internal RAM data do not change by resetting.

Figure 3.1.1 is a reset timing chart of the TMP91CP27.



3.2 Memory Map

Figure 3.2.1 is a memory map of the TMP91CP27.



Figure 3.2.1 Memory Map

3.3 Diversity of TMP91CW12A and TMP91CP27

TMP91CP27 is article cut function and pin from TMP91CW12A fundamentally.

Specification of function shows section 3.3.1 to 3.3.6. Wide difference of AC/DC characteristic is AC characteristics (Shown section 3.3.7). For the details, please refer to Chapter 4, "Electrical characteristics".

3.3.1 Cut Internal I/O

TMP91CP27 is micro controller cut 8-bit timer (TMRA6 to TMRA7), 16-bit timer (TMRB1) and clock doublers circuit (DFM) from TMP91CW12A.

Please don't access to special function register address of above internal I/O in TMP91CW12A.

Please refer to "Table of special function register".

3.3.2 Cut Port Function

TMP91CP27 be cut below port function from TMP91CW12A.

- Port 2: P27 (A23/A7) and P26 (A22/A6)
- Port 3: P37, P36 (R/\overline{W}), P35 (\overline{BUSAK}), P34 (\overline{BUSRQ}) and P33 (\overline{WAIT})
- Port 4: P43 (CS3)
- Port 5: P57 to P54 (AN7 to AN4)
- Port 6: P66, P65 and P64 (SCOUT)
- Port 7: P75 (TA7OUT)
- Port 8: P87 (TB1OUT1), P86 (TB1OUT0), P85 (TB1IN1/INT8) and P84 (TB1IN0/INT7)
- Port A: PA7 to PA4, PA3 to PA0 (INT4 to INT1)

3.3.3 Cut factor of Interrupt

TMP91CP27 be cut factor of interrupt by cut internal I/O and port function (Refer to Table 3.5.1). Please don't access to interrupt priority setting register for cut factor of interrupt. Please refer to "table of special function register".

3.3.4 Bus Release Function

TMP91CP27 don't include bus release function by cutting bus request pin (P34) and bus acknowledge pin (P35).

3.3.5 CS/WAIT Controller

When set TMP91CP27 to BxCS<BxW2:0> = "010" (1 + N) WAIT mode, it operates as 1 wait by cutting \overline{WAIT} pin.

And there is not $\overline{CS3}$ pin (P43), but when set MSAR3, MAMR3 and set B3CS<B3E> = "1", wait control is effective.

3.3.6 AD Converter

Analog input pin AN4 to AN7 be cut. Therefore please don't select cutting channel in ADMOD1<ADCH2:0>.

3.3.7 AC Characteristic

When accessing to external, AC characteristic don't guarantee at 2 V operation.

3.4 System Clock Function and Standby Control

TMP91CP27 contains (1) a clock gear, (2) stand-by controller and (3) noise-reduction circuit. It is used for low-power and low-noise systems.

The clock operating modes are as follows: (a) Single clock mode (X1 and X2 pins only), (b) Dual clock mode (X1, X2, XT1 and XT2 pins).

Figure 3.4.1 shows a transition figure.



Dual clock mode transition fiigure

Figure 3.4.1 Clock Operating Mode

Note: The clock frequency input from the X1 and X2 pins is called f_{OSCH} and the clock frequency input from the XT1 and XT2 pins is called fs. The clock frequency selected by SYSCR1<SYSCK> is www.DataSheet4U.com called f_{FPH}. The system clock f_{SYS} is defined as the divided clock of f_{FPH}, and one cycle of f_{SYS} is regret to as one state.

3.4.1 Block Diagram of System Clock



Figure 3.4.2 Block Diagram of System Clock

3.4.2 SFR

		7	6	5	4	3	2	1	0
SYSCR0	Bit symbol	XEN	XTEN	RXEN	RXTEN	RSYSCK	WUEF	PRCK1	PRCK0
(00E0H)	Read/Write				R/	W			
	After reset	1	0	1	0	0	0	0	0
	Function	High- frequency oscillator (fc) 0: Stop 1: Oscillation	Low- frequency oscillator (fs) 0: Stop 1: Oscillation	High- frequency oscillator (fc) after release of STOP mode 0: Stop 1: Oscillation	Low- frequency oscillator (fs) after release of STOP mode 0: Stop 1: Oscillation	Selects clock after release of STOP mode 0: fc 1: fs	Warm-up timer control 0 Write: Don't care 1 Write: Start warm-up 0 Read: End warm-up 1 Read: Do not end warm-up	Select preso 00: fFPH 01: Reserved 10: fc/16 11: Reserved	aler clock (Note 2)
		7	6	5	4	3	2	1	0
SYSCR1	Bit symbol					SYSCK	GEAR2	GEAR1	GEAR0
(00E1H)	Read/Write						R/	W	
	After reset	/				0	1	0	0
	Function					Select system clock 0: fc 1: fs	Select gear (fc) 000: fc 001: fc/2 010: fc/4 011: fc/8 100: fc/16 101: (Reserv 110: (Reserv 111: (Reserv	red) red) red)	n frequency
		7	6	5	4	3	2	1	0
SYSCR2	Bit symbol		-	WUPTM1	WUPTM0	HALTM1	HALTM0		DRVE
(00E2H) www.DataSheet4U.com	Read/Write		R/W	R/W	R/W	R/W	R/W		R/W
	After reset		0	1	0	1	1		0
	Function		Write to "0".	Select warr for oscillato 00: Reserved 01: 2 ⁸ /inputte 10: 2 ¹⁴ /inputt 11: 2 ¹⁶ /inputt	n-up time or d d frequency ed frequency ed frequency	HALT mode 00: Reserved 01: STOP mo 10: IDLE1 mo 11: IDLE2 mo	e J ode ode ode		Pin state control in STOP mode 0: I/O off 1: Remains the state before HALT

Note 1: SYSCR1
bit7:4>, SYSCR2
bit7,bit1> are read as undefined value.

Note 2: When using internal SBI, set prescaler clock select register SYSCR0<PRCK1:0> to fFPH.

Figure 3.4.3 SFR for System Clock

		7	6	5	4	3	2	1	0
EMCCR0	Bit symbol	PROTECT	-	-	-	ALEEN	EXTIN	DRVOSCH	DRVOSCL
(00E3H)	Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	1	0	0	0	1	1
	Function	Protect flag 0: OFF 1: ON	Write "0".	Write "1".	Write "0".	1: ALE output enable 0: ALE output disable	1: fc external clock	fc oscillator driver ability 1: NORMAL 0: WEAK	fs oscillator driver ability 1: NORMAL 0: WEAK
EMCCR1	Bit symbol								
Read/Write Protect OFF by writing "1FH". After reset Protect ON by writing except "1FH".									
						"1FH".			
	Function								

Figure 3.4.4 SFR for Noise Reduction

www.DataSheet4U.com

3.4.3 System Clock Controller

The system clock controller generates the system clock signal (f_{SYS}) for the CPU core and internal I/O. It contains two oscillation circuits and a clock gear circuit for high-frequency (fc) operation. The register SYSCR1<SYSCK> changes the system clock to either fc or fs, SYSCR0<XEN> and SYSCR0<XTEN> control enabling and disabling of each oscillator, and SYSCR1<GEAR2:0> sets the high-frequency clock gear to either 1, 2, 4, 8 or 16 (fc, fc/2, fc/4, fc/8 or fc/16). These functions can reduce the power consumption of the equipment in which the device is installed.

The combination of settings $\langle XEN \rangle = "1"$, $\langle XTEN \rangle = "0"$, $\langle SYSCK \rangle = "0"$ and $\langle GEAR2:0 \rangle = "100"$ will cause the system clock (fsys) to be set to fc/32 (fc/16 × 1/2) after a Reset.

For example, fSYS is set to 0.5 MHz when the 16 MHz oscillator connected to the X1 and X2 pins.

(1) Switching from NORMAL mode to SLOW mode

When the resonator is connected to the X1 and X2 pins, or to the XT1 and XT2 pins, the warm-up timer can be used to change the operation frequency after stable oscillation has been attained.

The warm-up time can be selected using SYSCR2<WUPTM1:0>.

This warm-up timer can be programmed to start and stop as shown in the following examples 1 and 2.

Table 3.4.1 shows the warm-up time.

- Note 1: When using an oscillator (other than a resonator) with stable oscillation, a warm-up timer is not needed.
- Note 2: The warm-up timer is operated by an oscillation clock. Hence, there may be some variation in warm-up time.
- Note 3: Note of using low-frequency oscillator
 - When connect low-frequency oscillator to ports 96 and 97, need below setting for cut consumption power.

(Case of resonators)

Set P9CR<P96C, P97C> = "11", P9<P96, P97> = "00"

(Case of oscillator)

Set P9CR<P96C, P97C> = "11", P9<P96, P97> = "10"

	Tal	ble 3.4.1	Warm-up	Times (whe	en changin	g clock)	
Timo							

Select Warm-up Time SYSCR2 <wuptm1:0></wuptm1:0>	Change to NORMAL (fc)	Change to SLOW (fs)	At f _{OSCH} = 16 MHz,
01 (2 ⁸ /frequency)	16 [μs]	7.8 [ms]	fs = 32.768 kHz
10 (2 ¹⁴ /frequency)	1.024 [ms]	500 [ms]	
11 (2 ¹⁶ /frequency)	4.096 [ms]	2000 [ms]	



SYSCR0	EQU	00E0H	
SYSCR1	EQU	00E1H	
SYSCR2	EQU	00E2H	
	LD	(SYSCR2), X–11– –X–B	; Sets warm-up time to 2 ¹⁶ /fs.
	SET	6, (SYSCR0)	; Enables low-frequency oscillation.
	SET	2, (SYSCR0)	; Clears and starts warm-up timer.
WUP:	BIT	2, (SYSCR0)	;] Detects stearing of warm we times
	JR	NZ, WUP	; J Delects stopping of warm-up timer.
	SET	3, (SYSCR1)	; Changes f _{SYS} from fc to fs.
	RES	7, (SYSCR0)	; Disables high-frequency oscillation



Example 2: Setting the clock Changing from low frequency (fs) to high frequency (fc).

SYSCR0	EQU	00E0H		
SYSCR1	EQU	00E1H		
SYSCR2	EQU	00E2H		
	LD	(SYSCR2), X-10X-B	;	Sets warm-up time to 2 ¹⁴ /fc.
	SET	7, (SYSCR0)	;	Enables high-frequency oscillation.
	SET	2, (SYSCR0)	;	Clears and starts warm-up timer.
WUP:	BIT	2, (SYSCR0)	;]	Detecto stopping of worm up timor
	JR	NZ, WUP	ر;	Detects stopping of warm-up timer.
	RES	3, (SYSCR1)	;	Changes f _{SYS} from fs to fc.
	RES	6, (SYSCR0)	;	Disables low-frequency oscillation.



X: Don't care, -: No change

(2) Clock gear controller

When the high-frequency clock fc is selected by setting SYSCR1<SYSCK> = "0", fFPH is set according to the contents of the clock gear select register SYSCR1<GEAR2:0> to either fc, fc/2, fc/4, fc/8 or fc/16. Using the clock gear to select a lower value of fFPH reduces power consumption.

Below show example of changing clock gear.

(Clock gear changing)

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

There is the 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 switching instruction by the clock gear after changing, input the dummy instruction as follows (instruction to execute the write cycle).

Example: SYSCR

1	EQU	00E1H								
	LD	(SYSCR1), XXXX0001B	;	Changes f _{SYS} to fc/4.						
	LD	(DUMMY), 00H	;	Dummy instruction						
	Instruction to be executed after clock gear has changed.									

3.4.4 Prescaler Clock Controller

For the internal I/O (TMRA01 to TMRA45, TMRB0, SIO0 to SIO1 and SBI) there is a prescaler which can divide the clock.

The ϕT , $\phi T0$ clock input to the prescaler is either the clock fFPH divided by 2 or the clock fc/16 divided by 4. The setting of the SYSCR0<PRCK1:0> register determines which clock signal is input.

When using internal SBI, set SYSCR0<PRCK1:0> to "00".

3.4.5 Noise Reduction Circuits

Noise reduction circuits are built in, allowing implementation of the following features.

- (1) Reduced drivability for high-frequency oscillator
- (2) Reduced drivability for low-frequency oscillator
- (3) Single drive for high-frequency oscillator
- (4) Output ALE pin disable
- (5) SFR protection of register contents

The above functions are performed by making the appropriate settings in the EMCCR0 to EMCCR1 registers.

(1) Reduced drivability for high-frequency oscillator

(Purpose)

Reduces noise and power for oscillator when a resonator is used.

(Block diagram)



(Setting method)

The drivability of the oscillator is reduced by writing "0" to EMCCR0<DRVOSCH> register. By reset, <DRVOSCH> is initialized to "1" and the oscillator starts oscillation by normal drivability when the power supply is on.

(2) Reduced drivability for low-frequency oscillator

(Purpose)

Reduces noise and power for oscillator when a resonator is used.

(Block diagram)



(Setting method)

The drivability of the oscillator is reduced by programming 0 to the EMCCR0<DRVOSCL> register. By Reset, <DRVOSCL> is initialized to "1".

(3) Single drive for high-frequency oscillator

(Purpose)

Not need twin-drive and protect mistake operation by inputted noise to X2 pin when the external oscillator is used.

(Block diagram)



www.DataSheet4U.com

(Setting method)

The oscillator is disabled by programming "1" to EMCCR0<EXTIN> register. X2 pin is always outputted "1".

By reset, <EXTIN> is initialized to "0".

(4) Output ALE pin disable

(Purpose)

Not need noise of clock property case of don't access external area is reduced.

(Block diagram)



(Setting method)

Output buffer of ALE pin is output disable by programming "0" to EMCCR0<ALEEN>. And ALE pin is high impedance.

By resetting, <ALEEN> is initialized to "0".

When you access to external area, you must program "1" to <ALEEN> before access.

(5) Runaway provision with SFR protection register

(Purpose)

Provision in runaway of program by noise mixing.

Write operation to specified SFR is prohibited so that provision program in runaway prevents that it is it in the state which is fetch impossibility by stopping of clock, memory control register (CS/WAIT controller) is changed.

```
Specified SFR list
```

1. CS/WAIT controller
B0CS, B1CS, B2CS, B3CS, BEXCS,
MSAR0, MSAR1, MSAR2, MSAR3,
MAMR0, MAMR1, MAMR2, MAMR3
2. Clock gear (write enable only EMCCR1)
SYSCR0, SYSCR1, SYSCR2, EMCCR0

(Block diagram)



(Setting method)

If writing except "1FH" code to EMCCR1 register, it become protect ON. By this operation, write operation to specified SFR is disabling.

If writing "1FH" to EMCCR1 register, it become protect OFF. State of protect can to confirm by reading EMCCR0<PROTECT>.

3.4.6 Standby Controller

(1) HALT modes

When the HALT instruction is executed, the operating mode switches to IDLE2, IDLE1 or STOP mode, depending on the contents of the SYSCR2<HALTM1:0> register.

The subsequent actions performed in each mode are as follows:

a. IDLE2: Only the CPU halts.

The internal I/O is available to select operation during IDLE2 mode by setting the following register.

Shows the registers of setting operation during IDLE2 mode.

Internal I/O	SFR
TMRA01	TA01RUN <i2ta01></i2ta01>
TMRA23	TA23RUN <i2ta23></i2ta23>
TMRA45	TA45RUN <i2ta45></i2ta45>
TMRB0	TB0RUN <i2tb0></i2tb0>
SIO0	SC0MOD1 <i2s0></i2s0>
SIO1	SC1MOD1 <i2s1></i2s1>
SBI	SBI0BR0 <i2sbi0></i2sbi0>
AD converter	ADMOD1 <i2ad></i2ad>
WDT	WDMOD <i2wdt></i2wdt>

Table 3.4.2 SFR Setting Operation during IDLE2 Mode

- b. IDLE1: Only the oscillator and the RTC (Real time clock) continue to operate.
- c. STOP: All internal circuits stop operating.

The operation of each of the different HALT modes is described in Table 3.4.3.

www.DataSheet/II.c	m	HALT Mode	IDLE2	IDLE1	STOP			
www.DataSheet+0.co	S	YSCR2 <haltm1:0></haltm1:0>	11	10	01			
		CPU	Stop					
		I/O port	Keep the state when the HALT executed.	instruction was	See Table 3.4.6, Table 3.4.7.			
		TMRA, TMRB	Available to select operation	St	ор			
	Block SIO, SBI		block					
		AD						
		WDT			_			
		RTC	Operate enable					
		Interrupt controller	Operate					

Table 3.4.3 I/O Operation during HALT Modes

(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 combination between the states of interrupt mask register <IFF2:0> and the HALT modes. The details for releasing the halt status are shown in Table 3.4.4.

• 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 interrupt mask register, the interrupt due to the source is processed after releasing the HALT mode, and CPU status 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 interrupt request level set before executing the interrupt request level set before executing the HALT mode regardless of the value of the interrupt request level set before executing the HALT instruction is less than the value of the interrupt request level set before executing the HALT mode is executed. In this case, interrupt processing, 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 = $(\overline{\text{NMI}}, \text{INT0}, \text{INTRTC})$ 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 5 clocks of f_{FPH}) with IDLE1 or STOP mode (IDLE2 is not applicable to this case). (In this case, an interrupt request is kept on hold internally.) If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficulty. The priority of this interrupt is compared with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.
- Releasing by resetting

Releasing all halt status is executed by resetting.

When the STOP mode is released by RESET, it is necessry enough resetting time (See Table 3.4.5) 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 settings contents are initialized. (Releasing due to interrupts keeps the state before the "HALT" instruction is executed.)

www.DataSheet4U.com

	ç	Status of Received Interrupt	Interrupt Enable (Interrupt level) ≥ (Interrupt mask)			Interrupt Disable (Interrupt level) < (Interrupt mask)			
HALT Mode			IDLE2	IDLE1	STOP	IDLE2	IDLE1	STOP	
		NMI	•	•	*1	_	-	-	
сe		INTWD	•	×	×	-	-	_	
ran		INT0 (Note1)	•	•	*1	0	0	° ^{*1}	
clea		INTRTC	♦	•	×	0	0	×	
teo	upt	INT5 to INT6	 ♦ (Note 2) 	×	×	×	×	×	
sta	iteri	INTTA0 to INTTA5	♦	×	×	×	×	×	
Halt	-	INTTB00, INTTB01, OF0	♦	×	×	×	×	×	
of ŀ		INTRX0 to INTRX1, INTTX0 to INTTX1	♦	×	×	×	×	×	
гce		INTSBI	♦	×	×	×	×	×	
Sou		INTAD	•	×	×	×	×	×	
		RESET	Initialize LSI						

Table 3.4.4	Source o	of Halt State	Clearance	and Halt	Clearance (Operation
10010 0.1.1	000.000	i i iait otato	orourarioo	ananan	olouraneo .	oporation

- •: After clearing the HALT mode, CPU starts interrupt processing.
- •: After clearing the HALT mode, CPU resumes executing starting from instruction following the HALT instruction. (Interrupt routine don't execute.)
- \times : It can not be used to release the HALT mode.
- -: The priority level (Interrupt request level) of non-maskable interrupts is fixed to 7, the highest priority level. There is not this combination type.
- *1: Releasing the HALT mode is executed after passing the warm-up time.
- Note 1: When the HALT mode is cleared by an INT0 interrupt of the level mode in the interrupt enabled status, hold high level until starting interrupt process. If low level was set before interrupt process is stared, interrupt process is not started correctly.

Note 2: If using external interrupt INT5 to INT6 in IDLE2 mode, set 16-bit timer RUN register TB0RUN<I2TB0> to "1".



Example: Clearing halt state

An INTO interrupt clears the halt state when the device is in IDLE1 mode.

(3) Operation

a. IDLE2 mode

In IDLE2 mode only specific internal I/O operations, as designated by the IDLE2 setting register, can take place. Instruction execution by the CPU stops.

Figure 3.4.5 illustrates an example of the timing for clearance of the IDLE2 mode halt state by an interrupt.



Figure 3.4.5 Timing Chart for IDLE2 Mode Halt State Cleared by Interrupt

b. IDLE1 mode

In IDLE1 mode, only the internal oscillator and the RTC continue to operate. The system clock in the MCU stops.

In the halt state, the interrupt request is sampled asynchronously with the system clock; however, clearance of the Halt state (e.g., restart of operation) is synchronous with it.

Figure 3.4.6 illustrates the timing for clearance of the IDLE1 mode halt state by an interrupt.



Figure 3.4.6 Timing Chart for IDLE1 Mode Halt State Cleared by Interrupt

c. STOP mode

When STOP mode is selected, all internal circuits stop, including the internal oscillator. Pin status in STOP mode depends on the settings in the SYSCR2<DRVE> register. Table 3.4.6 and Table 3.4.7 summarizes the state of these pins in STOP mode.

After STOP mode has been cleared, system clock output starts when the warm-up time has elapsed, in order to allow oscillation to stabilize. After STOP mode has been cleared, either NORMAL mode or SLOW mode can be selected using the SYSCR0<RSYSCK> register. Therefore, <RSYSCK>, <RXEN> and <RXTEN> must be set. See the sample warm-up times in Table 3.4.5.

Figure 3.4.7 illustrates the timing for clearance of the STOP mode halt state by an interrupt.





 Table 3.4.5
 Sample Warm-up Times after Clearance of STOP Mode

		@ f _{OS}	_{SCH} = 16 MHz, fs = 32.768 kHz			
SYSCR0	SYSCR2 <wuptm1:0></wuptm1:0>					
<rsysck></rsysck>	01 (2 ⁸)	10 (2 ¹⁴)	11 (2 ¹⁶)			
0 (fc)	16 μs	1.024 ms	4.096 ms			
1 (fs)	7.8 ms	500 ms	2000 ms			

Example:

• The STOP mode is entered when the low-frequency operates, and high-frequency operates after releasing due to NMI.

Address SYSCR0 EQU 00E0H SYSCR1 EQU 00E1H 00E2H SYSCR2 EQU 8FFDH LD (SYSCR1), 08H ; $f_{SYS} = fs/2$ 9000H LD (SYSCR2), X-1001X1B ; Sets warm-up time to 2¹⁴/fc 9002H LD (SYSCR0), -11000 - - B ; Operates high frequency after released. HALT 9005H Clears and starts warm-up timer NMI pin input (High frequency) ≻ END NMI interrupt routine ÷ ĽD 9006H XX, XX RETI ł

-: No change

Note: When different modes are used before and after STOP mode as the above mentioned, there is possible to release the HALT mode without changing the operation mode by acceptance of the halt release interrupt request during execution of "HALT" instruction (during 6 state). In the system which accepts the interrupts during execution "HALT" instruction, set the same operation mode before and after the STOP mode.

						Input Buffer State					
				When the CPU is		In HAL	T mode	In HALT mode(STOP)			
	. .			Ope	rating	(IDLE2	/IDLE1)	<drv< td=""><td>/F>=1</td><td><dr\< td=""><td>/F>=0</td></dr\<></td></drv<>	/F>=1	<dr\< td=""><td>/F>=0</td></dr\<>	/F>=0
	Port Name	Input Function Name	During Reset	When Used as Function Pin	When Used as Input Port	When Used as Function Pin	When Used as Input Port	When Used as Function Pin	When Used as Input Port	When Used as Function Pin	When Used as Input Port
	P00-07	AD0-7		ON upon							
	P10-17	AD8-15	OFF	external read		OFF	OFF	OFF	OFF	OFF	
	P20-25	-			ON						
	P32(*1)	-									
	P40-42(*1)	-	ON	-		-	ON	-	ON	-	
	P50-52(*2)	-	OFF		ON upon		OFF		OFF		OFF
	P53(*2)	/ADTRG	_		port lead				-	ON	
	P60	SCK		ON -		ON		ON			
	P61	SDA			ON					OFF	
	P62	SI									
	1.02	SCL									
	P63	INT0								ON	ON
	P70	TAOIN								OFF	
	P71	-				-		-	- ON	-	
	P72	-					<u>ON</u> 				
	P73	TA4IN		ON		ON		ON		OFF	
	P74	-		-		-		-		-	
	P80	TB0UN0		ON		ON		ON ON			
		INT5	ON						ON	OFF	OFF
	P81	TB0IN1							_		
		INT6									
	P82	-									
	P83	-		-		-					
	P90	-									
www.DataSh	P91 eet4U.com	RXD0									
	P92	SCLKO		ON		ON		ON		OFF	
	Doo	/0150					-				
	P93	-		-		-	-	-		-	
	P94							ON			
	P95			ON		ON		ON		OFF	
		For oscillato	r OFF		OFF		OFF		OFF		
	P96	XT1 For port	. 011	OFF		OFF	011	OFF	011		
	P97	-	-1	-	ON	-	ON	-	ON	-	
	/NMI	-		-		-					
	/RESET	- ON		NC				ON		ON	
	AM0,AM1	-	1	ON	-	ON	-		-		-
	X1	_	1					OFF	1	OFF	1

Table 3.4.6	Input Buffer	State Table
-------------	--------------	-------------

ON: The buffer is always turned on. A current *1:Port having a pull-up/pull-down resistor. flows the input buffer if the input pin is not driven.

OFF :The buffer is always turned off.

*2:AIN input does not cause a current to flow through the buffer.

-: No applicable

				Output Buffer State									
					vvnen tn								
	Port	Outp	out Function	During	Oper	ating	(IDLE2	/IDLE1)	<dr< td=""><td>VE>=1</td><td><drv< td=""><td>'E>=0</td></drv<></td></dr<>	VE>=1	<drv< td=""><td>'E>=0</td></drv<>	'E>=0	
	Name		Name	During	When	When	When	When	When	When Used	When	When	
				Reset	Used as	Used as	Used as	Used as	Used as	as Output	Used as	Used as	
					Function	Output	Function	Output	Function	Port	Function	Output	
					Pin	Port	Pin	Port	Pin		Pin	Port	
	P00-07		AD0-7		ON upon		OFF		OFF				
	P10-17		AD8-15	055	write		011		OIT				
			A8-15	OFF									
	P20-25		A0-5										
	1 20-20		A16-21										
	P30		/RD	ON									
	P31		/WR	ON							OFF		
	P32(*1)		/HWR	-			ON						
			/CS0		ON				ON				
	P40-42(*1)		/CS1										
			/CS2										
	P60		SCK										
	P61		SDA	-									
	_		SO		- ON	- ON		<u></u>					
	P62		SCL										
	P63		-				-	ON	-	ON	-	055	
	P70		-										
	P71		TATOUT				ON		ON		OFF		
	P72		TA3OUT	OFF									
	P73		-		-		-		-		-		
	P74		1A5001		ON		ON		ON		OFF		
	P80		-		-		-		-		-		
	P81	т											
	P02	<u>і</u> т			ON		ON		ON		OFF		
	P00	1									011		
	P90		-		_				_		_		
	P92		SCLK0										
	P93				ON		ON		ON		OFF		
www.DataSh	eet40.com P94		-		-		-		-		-		
	P95		SCLK1		ON		ON		ON		OFF		
	P96		-	ON	-		-		-		-		
	D07	VTO	For oscillator	OFF	ON	OFF	ON	OFF	055	OFF	055		
	P97	T2 For port	ON	OFF	ON	OFF	ON	UFF	ON	UFF			
	ALE		-	OFF	ON		ON		ON		ON		
	X2		-	ON	ON	-	UN	-		-	ON	-	

91CP27-33

	• • •		~	-
Table 3.4.7	Output	buffer	State	lable

ON: The buffer is always turned on.

*1 : Port having a pull-up/pull-down resistor.

OFF: The buffer is always turned off.

-:No applicable

3.5 Interrupts

Interrupts are controlled by the CPU interrupt mask register SR<IFF2:0> and by the built-in interrupt controller.

The TMP91CP27 has a total of 34 interrupts divided into the following three types:

- Interrupts generated by CPU: 9 sources (Software interrupts, illegal instruction interrupt)
 - Internal interrupts: 21 sources
 - Interrupts on external pins ($\overline{\text{NMI}}$, INT0, INT5 and INT6): 4 sources

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

One of six (Variable) priority level can be assigned to each maskable interrupt.

The priority level of non-maskable interrupts are fixed at 7 as the highest level.

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

The CPU compares the priority level of the interrupt with the value of the CPU interrupt mask register <IFF2:0>. If the priority level of the interrupt is higher than the value of the interrupt mask register, the CPU accepts the interrupt.

The interrupt mask register <IFF2:0> value can be updated using the value of the EI instruction ("EI num" sets <IFF2:0> data to num).

For example, specifying "EI3" enables the maskable interrupts which priority level set in the interrupt controller is 3 or higher, and also non-maskable interrupts.

Operationally, the DI instruction ($\langle IFF2:0 \rangle = "7"$) is identical to the "EI7" instruction. DI instruction is used to disable maskable interrupts because of the priority level of maskable interrupts is 0 to 6. The EI instruction is vaild immediately after execution.

In addition to the above general-purpose interrupt processing mode, TLCS-900/L1 has a micro DMA interrupt processing mode as well. The CPU can transfer the data (1/2/4 bytes) automatically in micro DMA mode, therefore this mode is used for speed-up interrupt processing, such as transferring data to the internal or external peripheral I/O. Moreover, TMP91CP27 has software start function for micro DMA processing request by the software not by the hardware interrupt.

www.DataSheet4U.com

Figure 3.5.1 shows the overall interrupt processing flow.




3.5.1 General-Purpose Interrupt Processing

When the CPU accepts an interrupt, it usually performs the following sequence of operations. That is also the same as TLCS-900/L and TLCS-900/H.

(1) The CPU reads the interrupt vector from the interrupt controller.

If the same level interrupts occur simultaneously, the interrupt controller generates an interrupt vector in accordance with the default priority and clears the interrupt request.

(The default priority is already fixed for each interrupt: The smaller vector value has the higher priority level.)

- (2) The CPU pushes the value of program counter (PC) and status register (SR) onto the stack area (indicated by XSP).
- (3) The CPU sets the value which is the priority level of the accepted interrupt plus 1 (+1) to the interrupt mask register <IFF2:0>. However, if the priority level of the accepted interrupt is 7, the register's value is set to 7.
- (4) The CPU increases the interrupt nesting counter INTNEST by 1 (+1).
- (5) The CPU jumps to the address indicated by the data at address "FFFF00H + interrupt vector" and starts the interrupt processing routine. The above processing time is 18-states (2.25 μ s at 16 MHz) as the best case (16 bits

data bus width and 0 waits).

When the CPU complete the interrupt processing, use the RETI instruction to return to the main routine. RETI restores the contents of program counter (PC) and status register (SR) from the stack and decreases the interrupt nesting counter INTNEST by 1 (-1).

Non-maskable interrupts cannot be disabled by a user program. Maskable interrupts, however, can be enabled or disabled by a user program. A program can set the priority level for each interrupt source.

If an interrupt request which has a priority level equal to or greater than the value of the CPU interrupt mask register $\langle IFF2:0 \rangle$ comes out, the CPU accepts its interrupt. Then, the CPU interrupt mask register $\langle IFF2:0 \rangle$ is set to the value of the priority level for the accepted interrupt plus 1 (+1).

Therefore, if an interrupt is generated with a higher level than the current interrupt during its processing, the CPU accepts the later interrupt and goes to the nesting status of interrupt processing.

Moreover, if the CPU receives another interrupt request while performing the said (1) to (5) processing steps of the current interrupt, the latest interrupt request is sampled immediately after execution of the first instruction of the current interrupt processing routine. Specifying DI as the start instruction disables maskable interrupt nesting.

A Reset initializes the interrupt mask register ${\rm <IFF2:0>}$ to "7", disabling all maskable interrupts.

Figure 3.5.1 shows the TMP91CP27 interrupt vectors and micro DMA start vectors. The address FFFF00H to FFFFFFH (256 bytes) is assigned for the interrupt vector area.

	Default Priority	Туре	Interrupt Source and Source of Micro DMA Request	Vector Value (V)	Vector Reference Address	Micro DMA Start Vector
	1		"RESET" or SWI0 instruction	0000H	FFFF00H	-
	2		SWI1 instruction	0004H	FFFF04H	-
	3		INTUNDEF: Illegal Instruction or SWI2 instruction	0008H	FFFF08H	_
	4		SWI3 instruction	000CH	FFFF0CH	_
	5	Non mookohlo	SWI4 instruction	0010H	FFFF10H	_
	6	NUII-IIIASKADIe	SWI5 instruction	0014H	FFFF14H	-
	7		SWI6 instruction	0018H	FFFF18H	_
	8		SWI7 instruction	001CH	FFFF1CH	-
	9		NMI pin	0020H	FFFF20H	-
	10		INTWD: Watchdog timer	0024H	FFFF24H	-
	-		(Micro DMA)	-	-	-
	11		INT0 pin	0028H	FFFF28H	0AH
	12		Reserved	_	_	-
	13		Reserved	_	_	_
	14		Reserved	-	_	-
	15		Reserved	_	_	_
	16		INT5 pin	003CH	FFFF3CH	0FH
	17		INT6 pin	0040H	FFFF40H	10H
	18		Reserved	-	_	-
	19		Reserved	_	_	-
	20		INTTA0: 8-bit timer 0	004CH	FFFF4CH	13H
	21		INTTA1: 8-bit timer 1	0050H	FFFF50H	14H
	22		INTTA2: 8-bit timer 2	0054H	FFFF54H	15H
	23		INTTA3: 8-bit timer 3	0058H	FFFF58H	16H
	24		INTTA4: 8-bit timer 4	005CH	FFFF5CH	17H
	25		INTTA5: 8-bit timer 5	0060H	FFFF60H	18H
	26		Reserved	-	_	-
	27		Reserved	-	_	_
	28	Maskable	INTTB00: 16-bit timer 0 (TB0RG0)	006CH	FFFF6CH	1BH
	29		INTTB01: 16-bit timer 0 (TB0RG1)	0070H	FFFF70H	1CH
www.DataSheet4	U.com 30		Reserved	-	_	-
	31		Reserved	-	_	-
	32		INTTBOF0: 16-bit timer 0 (Over flow)	007CH	FFFF7CH	1FH
	33		Reserved	_	_	_
	34		INTRX0: Serial reception (Channel 0)	0084H	FFFF84H	21H
	35		INTTX0: Serial transmission (Channel 0)	0088H	FFFF88H	22H
	36		INTRX1: Serial reception (Channel 1)	008CH	FFFF8CH	23H
	37		INTTX1: Serial transmission (Channel 1)	0090H	FFFF90H	24H
	38		INTSBI: Serial bus interface interrupt	0094H	FFFF94H	25H
	39		INTRTC: Interrupt for RTC	0098H	FFFF98H	26H
	40		INTAD: AD conversion end	009CH	FFFF9CH	27H
	41		INTTC0: End of Micro DMA (Channel 0)	00A0H	FFFFA0H	
	42		INTTC1: End of Micro DMA (Channel 1)	00A4H	FFFFA4H	_
	43		INTTC2: End of Micro DMA (Channel 2)	00A8H	FFFFA8H	_
	44		INTTC3: End of Micro DMA (Channel 3)	00ACH	FFFFACH	—
			(Reserved)	00B0H	FFFFB0H	-
			: (Reserved)	: 00FCH	: FFFFFCH	:

Table 3.5.1 TMP91CP27 Interrupt Vectors and Micro DMA Start Vectors

3.5.2 Micro DMA

In addition to general-purpose interrupt processing, the TMP91CP27 supports a micro DMA function. Interrupt requests set by micro DMA perform micro DMA processing at the highest priority level among maskable interrupts, regardless of the priority level of the particular interrupt source. The micro DMA has 4 channels and is possible continuous transmission by specifying the say later burst mode.

Because the micro DMA function has been implemented with the cooperative operation of CPU, when CPU goes to a standby mode by HALT instruction, the requirement of micro DMA will be ignored (Pending).

(1) Micro DMA operation

When an interrupt request specified by the micro DMA start vector register is generated, the micro DMA triggers a micro DMA request to the CPU at interrupt priority highest level and starts processing the request in spite of any interrupt source's level. The micro DMA is ignored on $\langle IFF2:0 \rangle = "7"$.

The 4 micro DMA channels allow micro DMA processing to be set for up to 4 types of interrupts at any one time. When micro DMA is accepted, the interrupt request flip-flop assigned to that channel is cleared.

The data are automatically transferred once (1/2/4 bytes) from the transfer source address to the transfer destination address set in the control register, and the transfer counter is decreased by 1 (-1).

If the decreased result is "0", the micro DMA transfer end interrupt (INTTCn) passes from the CPU to the interrupt controller. In addition, the micro DMA start vector register DMAnV is cleared to 0, the next micro DMA is disabled and micro DMA processing completes.

If the decreased result is other than "0", the micro DMA processing completes if it isn't specified the say later burst mode. In this case, the micro DMA transfer end interrupt (INTTCn) aren't generated.

If an interrupt request is triggered for the interrupt source in use during the interval between the clearing of the micro DMA start vector and the next setting, general-purpose interrupt processing executes at the interrupt level set. Therefore, if only using the interrupt for starting the micro DMA (Not using the interrupts as a general-purpose interrupt: Level 1 to 6), first set the interrupts level to 0 (Interrupt requests disabled).

If using micro DMA and general-purpose interrupts together, first set the level of the interrupt used to start micro DMA processing lower than all the other interrupt levels. In this case, the cause of general interrupt is limited to the edge interrupt.

The priority of the micro DMA transfer end interrupt is defined by the interrupt level and the default priority as the same as the other maskable interrupt.

If a micro DMA request is set for more than one channel at the same time, the priority is not based on the interrupt priority level but on the channel number. The smaller channel number has the higher priority (Channel 0 (High) > channel 3 (Low)).

While the register for setting the transfer source/transfer destination addresses is a 32-bit control register, this register can only effectively output 24-bit addresses. Accordingly, micro DMA can access 16 Mbytes (The upper eight bits of the 32 bits are not valid).

Three micro DMA transfer modes are supported: 1-byte transfer, 2-byte transfer, and 4-byte transfer. After a transfer in any mode, the transfer source/destination addresses are increased, decreased, or remain unchanged.

This simplifies the transfer of data from I/O to memory, from memory to I/O, and from I/O to I/O. For details of the transfer modes, see 3.5.2 (4) "Detailed description of the transfer mode register: DMAM0 to DMAM3".

As the transfer counter is a 16-bit counter, micro DMA processing can be set for up to 65536 times per interrupt source. (The micro DMA processing count is maximized when the transfer counter initial value is set to 0000H.)

Micro DMA processing can be started by the 19 interrupts shown in the micro DMA start vectors of Figure 3.5.1 and by the micro DMA soft start, making a total of 20 interrupts.

Figure 3.5.2 shows the word transfer micro DMA cycle in transfer destination address INC mode (except for Counter mode, the same as for other modes).

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



Figure 3.5.2 Timing for Micro DMA Cycle (Word transfer)

 www.DataSheet4U.com
 States 1 to 3: Instruction fetch cycle (gets next address code).

 If 3 bytes and more instruction codes are inserted in the instruction queue buffer, this cycle becomes a dummy cycle.

 States 4 to 5:
 Micro DMA read cycles

 State 6:
 Dummy cycle (The address bus remains unchanged from state 5)

 States 7 to 8:
 Micro DMA write cycle

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

If the source address area is a 16-bit bus and the address starts from an odd number, it is increased by two states.

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

If the destination address area is a 16-bit bus and the address starts from an odd number, it is increased by two states.

(2) Soft start function

In addition to starting the micro DMA function by interrupts, TMP91CP27 includes a micro DMA software start function that starts micro DMA on the generation of the write cycle to the DMAR register.

Writing "1" to each bit of DMAR register causes micro DMA once. At the end of transfer, the corresponding bit of the DMAR register is automatically cleared to "0".

Only one-channel can be set once for micro DMA. (Do not write "1" to plural bits.)

When writing again "1" to the DMAR register, check whether the bit is "0" before writing "1".

When a burst is specified by DMAB register, data is continuously transferred until the value in the micro DMA transfer counter is "0" after start up of the micro DMA.

Symbol	Name	Address	7	6	5	4	3	2	1	0		
DMAR	DMA	89H	/	/	/	/		DMA request				
	software request register		/	/	/	/	DMAR3	DMAR2	DMAR1	DMAR0		
								R/	W			
			/		/	/	0	0	0	0		

(3) Transfer control registers

The transfer source address and the transfer destination address are set in the following registers in CPU. An instruction of the form "LDC cr,r" can be used to set these registers.





(DN	ЛАМ	0 to	DM	AM:	3)				
0	0		0		Mode Note: The upper three bits of data p these registers must always be	programmed to			
		Г				Execution time			
	ZZ: 0 = Byte transfer, 1 = Word transfer, 2 = 4-byte transfer, 3 = Reserved								
0	0	0	Z	Ζ	Transfer destination address INC mode I/O to memory	8 states (1000 ns)			
					$(DMADn+) \leftarrow (DMASn)$	@ byte/word transfer			
					DMACn ← DMACn – 1	12 states (1500 ns)			
					if DMACn = 0 then INTTC is generated	@4-byte transfer			
0	0	1	Ζ	Ζ	Transfer destination address DEC mode I/O to memory	8 states (1000 ns)			
					$(DMADn-) \leftarrow (DMASn)$	@ byte/word transfer			
					DMACn ← DMACn – 1	12 states (1500 ns)			
					if DMACn = 0 then INTTC is generated	@4-byte transfer			
0	1	0	Ζ	Ζ	Transfer source address INC mode Memory to I/O	8 states (1000 ns)			
					$(DMADn) \leftarrow (DMASn+)$	@ byte/word transfer			
					DMACn ← DMACn – 1	12 states (1500 ns)			
					if DMACn = 0 then INTTC is generated	@4-byte transfer			
0	1	1	Ζ	Ζ	Transfer source address DEC mode Memory to I/O	8 states (1000 ns)			
					$(DMADn) \leftarrow (DMASn-)$	@ byte/word transfer			
					DMACn ← DMACn – 1	12 states (1500 ns)			
					if DMACn = 0 then INTTC is generated	@4-byte transfer			
1	0	0	Ζ	Ζ	Address fixed mode I/O to I/O	8 states (1000 ns)			
					$(DMADn) \leftarrow (DMASn)$	@ byte/word transfer			
					DMACn ← DMACn – 1	12 states (1500 ns)			
					if DMACn = 0 then INTTC is generated	@4-byte transfer			
1	0	1	0	0	Counter mode · · for counting number of times interrupt is generated	5 states			
					$DMASn \leftarrow DMASn + 1$				
					DMACn ← DMACn – 1	(625 ns)			
					if DMACn = 0 then INTTC is generated				

(4) Detailed description of the transfer mode register: DMAM0 to DMAM3

Note 1: "n" is the corresponding micro DMA channels 0 to 3

DMADn+/DMASn+: Post increment (Increment register value after transfer)

DMADn-/DMASn-: Post decrement (Decrement register value after transfer)

www.DataSheet4U.com The I/Os in the table mean fixed address and the memory means increment (INC) or decrement (DEC) addresses.

Note 2: Execution time is under the condition of:

16-bit bus width/0 waits.

fc = 16MHz/selected high frequency mode ($fc \times 1$)

Note 3: Do not use an undefined code for the transfer mode register except for the defined codes listed in the above table.

3.5.3 Interrupt Controller Operation

The block diagram in Figure 3.5.3 shows the interrupt circuits. The left-hand side of the diagram shows the interrupt controller circuit. The right-hand side shows the CPU interrupt request signal circuit and the halt release circuit.

For each of the 25 interrupt channels there is an interrupt request flag (Consisting of a flip-flop), an interrupt priority setting register and a micro DMA start vector register. The interrupt request flag latches interrupt requests from the peripherals. The flag is cleared to zero in the following cases:

- When reset occurs
- When the CPU reads the channel vector after accepted its interrupt
- When executing an instruction that clears the interrupt (Program DMA start vector to INTCLR register)
- When the CPU receives a micro DMA request (When micro DMA is set.)
- When the micro DMA burst transfer is terminated

An interrupt priority can be set independently for each interrupt source by writing the priority to the interrupt priority setting register (e.g., INTEOAD or INTE56). 6 interrupt priorities levels (1 to 6) are provided. Setting an interrupt source's priority level to 0 (or 7) disables interrupt requests from that source. The priority of non-maskable interrupts ($\overline{\text{NMI}}$ pin interrupts and watchdog timer interrupts) is fixed at 7. If interrupt request with the same level are generated at the same time, the default priority (The interrupt with the lowest priority or, in other words, the interrupt with the lowest vector value) is used to determine which interrupt request is accepted first.

The 3rd and 7th bits of the interrupt priority setting register indicate the state of the interrupt request flag and thus whether an interrupt request for a given channel has occurred.

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> in the status register by the interrupt request signal with the priority value set: if the latter is higher, the interrupt is accepted. Then the CPU sets a value higher than the priority value by 1 (+1) in the CPU SR<IFF2:0>. Interrupt request where the priority value equals or is higher than the set value are accepted simultaneously during the previous interrupt routine.

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

The interrupt controller also has registers (4 channels) used to store the micro DMA start vector. Writing the start vector of the interrupt source for the micro DMA processing (See Table 3.5.1), enables the corresponding interrupt to be processed by micro DMA processing. The values must be set in the micro DMA parameter register (e.g., DMAS and DMAD) prior to the micro DMA processing.



Figure 3.5.3 Block Diagram of Interrupt Controller

Symbol	Name	Address	7	6	5	4	3	2	1	0
			INTAD				INTO			
	INTO &	0.04	IADC	IADM2	IADM1	IADM0	I0C	10M2	I0M1	I0M0
INTEUAD	enable	90h	R		R/W		R		R/W	•
			0	0	0	0	0	0	0	0
	INT5 &			IN	Т6			IN	T5	
		03h	16C	16M2	I6M1	16M0	I5C	I5M2	I5M1	I5M0
INTESO	enable	9311	R	R R/W			R		R/W	
			0	0	0	0	0	0	0	0
	INTTA0 & INTTA1 enable			INTTA1	(TMRA1)			INTTA0	(TMRA0)	
INTETA01		N0 &	ITA1C	ITA1M2	ITA1M1	ITA1M0	ITA0C	ITA0M2	ITA0M1	ITA0M0
			R		R/W		R		R/W	
			0	0	0	0	0	0	0	0
	INTTA2 & INTTA3 enable		INTTA3 (TMRA3)					INTTA2	(TMRA2)	
		OGh	ITA3C	ITA3M2	ITA3M1	ITA3M0	ITA2C	ITA2M2	ITA2M1	ITA2M0
		9011	R R/W			R	R/W			
			0	0	0	0	0	0	0	0
INTETA45				INTTA5	(TMRA5)		INTTA4 (TMRA4)			
	INTTA4 &	07h	ITA5C	ITA5M2	ITA5M1	ITA5M0	ITA4C	ITA4M2	ITA4M1	ITA4M0
	enable	5/11	R		R/W		R	R/W		
			0	0	0	0	0	0	0	0
	Interrupt requ	uest flag 🗲								

(1)	Interrupt level setting registers	
-----	-----------------------------------	--

		↓		
	lxxM2	lxxM1	lxxM0	Function (Write)
	0	0	0	Disable interrupt request
	0	0	1	Setting interrupt priority level to "1".
	0	1	0	Setting interrupt priority level to "2".
	0	1	1	Setting interrupt priority level to "3".
www.DataSheet4	LL com	0	0	Setting interrupt priority level to "4".
www.bataoneet-	1	0	1	Setting interrupt priority level to "5".
	1	1	0	Setting interrupt priority level to "6".
	1	1	1	Disable interrupt request

Symbol	Name	Address	7	6	5	4	3	2	1	0	
				INTTB01	(TMRB0)			INTTB00	(TMRB0)		
	Interrupt	0011	ITB01C	ITB01M2	ITB01M1	ITB01M0	ITB00C	ITB00M2	ITB00M1	ITB00M0	
INTETOU	TMRB0	990	R		R/W		R	R/W			
	-		0	0	0	0	0	0	0	0	
	Interrupt						IN	TTBOF0 (TM	IRB0 over flo	ow)	
	enable						ITF0C	ITF0M2	ITF0M1	ITF0M0	
INTERBUTV	TMRB0	эвн	R		R/W		R		R/W		
	(over flow)		0	0	0	0	0	0	0	0	
				INT	TX0			INT	RX0		
	Interrupt	0011	ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0	
INTESO enable Serial 0	900	R		R/W		R		R/W			
			0	0	0	0	0	0	0	0	
	Interrupt enable serial 1	t opu	INTTX1				INTRX1				
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0	
INTEST		900	R		R/W		R		R/W		
			0	0	0	0	0	0	0	0	
	Interrupt	0511	INTRTC				INT	SBI			
			IRTCC	IRTCM2	IRTCM1	IRTCM0	IS2C	IS2M2	IS2M1	IS2M0	
INTEGZRIC	SBI/RTC	960	R		R/W		R	R/W			
			0	0	0	0	0	0	0	0	
	INTTCO			INT	TC1		INTTCO				
	&	лон	ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0	
INTERCOT	INTTC1	AUL	R		R/W		R		R/W		
	enable		0	0	0	0	0	0	0	0	
	INTTC2			INT	тС3			INT	TC2		
INITETCOO	&	∧1⊔	ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0	
INTETC23	INTTC3	AIR	R		R/W		R		R/W		
	enable		0	0	0	0	0	0	0	0	
h cidl Leona	Interrupt request flag										

	\downarrow		
lxxM2	lxxM1	lxxM0	Function (Write)
0	0	0	Disable interrupt request
0	0	1	Setting interrupt priority level to "1".
0	1	0	Setting interrupt priority level to "2".
0	1	1	Setting interrupt priority level to "3".
1	0	0	Setting interrupt priority level to "4".
1	0	1	Setting interrupt priority level to "5".
1	1	0	Setting interrupt priority level to "6".
1	1	1	Disable interrupt request

Γ

Symbol	Name	Address	7	6	5	4	3	2	1	0
	Interrupt input mode control		-	-	-	-	-	I0EDGE	IOLE	NMIREE
						١	N			
		8CH (Prohibit RMW)	0	0	0	0	0	0	0	0
IIMC					Write "0".			INT0 edge 0: Rising 1: Falling	INT0 mode 0: Edge 1: Level	1: Operate even on rising/ falling edge of NMI
INT0 level enable 0 edge detect interrupt					.					

(2) External interrupt control

INT0 leve	el enable	
0	edge detect interrupt	<u> </u>
1	"H" level interrupt	
NMI risin	g edge enable	
0	Interrupt request generation at falling edge	
1	Interrupt request generation at rising/falling	
	edge	

(3) Interrupt request flag clear register

The interrupt request flag is cleared by writing the appropriate micro DMA start vector, as given in Table 3.5.1, to the register INTCLR.

For example, to clear the interrupt flag INTO, perform the following register operation after execution of the DI instruction.

INTCLR $\leftarrow 0AH$	Clears interrupt request flag INT0
-------------------------	------------------------------------

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTCLR	Interrupt clear control	88H (Prohibit RMW)			CLRV5	CLRV4	CLRV3	CLRV2	CLRV1	CLRV0
			/				V	V		
					0	0	0	0	0	0
							Interrup	t vector		

www.DataSheet4U.com

(4) Micro DMA start vector registers

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

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

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

Symbol	Name	Address	7	6	5	4	3	2	1	0	
							DMA0 st	art vector			
	DMA0 start	0 0 LL			DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0	
DIVIAUV	vector	001	/	/	R/W						
					0	0	0	0	0	0	
		81H	/	/			DMA1 st	art vector			
DMA1 DMA1V start vector	DMA1 start		/	/	DMA1V5	DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0	
	vector		/	/	R/W						
			/	/	0	0	0	0	0	0	
		0011	/	/	DMA2 start vector						
	DMA2 start		/	/	DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0	
DIVIAZV	vector	0211	/	/	R/W						
			/	/	0	0	0	0	0	0	
			/	/			DMA3 st	art vector			
	DMA3	0211			DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0	
DMA3V	vector	ö3H					R/	W			
					0	0	0	0	0	0	

(5) Micro DMA burst specification

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

	Symbol	Name	Address	7	6	5	4	3	2	1	0
		DMA		/	/		/	DMAR3	DMAR2	DMAR1	DMAR0
		software request register	89H					R/W	R/W	R/W	R/W
DIVIAR	DIVIAR			/	/	/	/	0	0	0	0
									1: DMA software request		
		DMA		/	/	/	/	DMAB3	DMAB2	DMAB1	DMAB0
		burst	011	/	/	/	/		R/	W	
	DIVIAD	request	оАП					0	0	0	0
www.DataShee	4U.com	register						1: DMA request on Burst Mode			

(6) Attention point

The instruction execution unit and the bus interface unit of this CPU operate independently. Therefore, immediately before an interrupt is generated, if the CPU fetches an instruction that clears the corresponding interrupt request flag, the CPU may execute the instruction that clears the interrupt request flag (*1) between accepting and reading the interrupt vector. In this case, the CPU reads the default vector 0008H and reads the interrupt vector address FFFF08H.

To avoid the above program, place instructions that clear interrupt request flags after a DI instruction. And in the case of setting an interrupt enable again by EI instruction after the execution of clearing instruction, execute EI instruction after clearing and more than 1 instructions (Example: "NOP" * 1 times). If placed EI instruction without waiting NOP instruction after execution of clearing instruction, interrupt will be enable before request flag is cleared.

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

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

INT0 level mode	In level mode INT0 is not an edge-triggered interrupt. Hence, in level mode the interrupt request flip-flop for INT0 does not function. The peripheral interrupt request passes through the S input of the flip-flop and becomes the Q output. If the interrupt input mode is changed from edge mode to level mode, the interrupt request flag is cleared automatically.						
	If the CPU enters the interrupt response sequence as a result of INT0 going from 0 to 1, INT0 must then be held at 1 until the interrupt response sequence has been completed. If INT0 is set to level mode so as to release a halt state, INT0 must be held at 1 from the time INT0 changes from 0 to 1 until the halt state is released. (Hence, it is necessary to ensure that input noise is not interpreted as a 0, causing INT0 to revert to 0 before the halt state has been released.) When the mode changes from level mode to edge mode, interrupt request flags which were set in level mode will not be cleared. Interrupt request flags must be cleared using the following sequence. DI						
	LD (IIMC), 00H ; Switches interrupt input mode from level mode to edge mode.						
	LD (INTCLR), 0AH; Clears interrupt request flag						
	NOP ; Wait EI instruction						
	El						
INTRX	The interrupt request flip-flop can only be cleared by a reset or by reading the serial channel receive buffer. It cannot be cleared by an instruction.						

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

INT0: Instructions which switch to Level Mode after an interrupt request has been generated in edge mode.

The pin input change from high to low after interrupt request has been generated

in Level Mode. (H \rightarrow L)

INTRX: Instruction which read the receive buffer

3.6 Port Function

The TMP91CP27 features 53 bit settings which relate to the various I/O ports. As well as general-purpose I/O port functionality, the port pins also have I/O functions which relate to the built-in CPU and internal I/Os. Table 3.6.1 lists the functions of each port pin. Table 3.6.2 to Table 3.6.3 lists I/O registers and their specifications.

	(R: \uparrow = with pull-up resistor)									
	Port Names	Pin Names	Number of Pins	Direction	R	Direction Setting Unit	Pin Names for Built-in Functions			
	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 P25	6	I/O	-	Bit	A16 to A21/A0 to A5			
	Port 3	P30	1	Output	-	Bit	RD			
		P31	1	Output	_	Bit	WR			
		P32	1	I/O	\uparrow	Bit	HWR			
	Port 4	P40	1	I/O	\rightarrow	Bit	CS0			
		P41	1	I/O	\uparrow	Bit	CS1			
		P42	1	I/O	\uparrow	Bit	CS2			
	Port 5	P50 to P53	4	Input	-	(Fixed)	AN0 to AN3, ADTRG (P53)			
	Port 6	P60	1	I/O	Ι	Bit	SCK			
		P61	1	I/O	-	Bit	SO/SDA			
		P62	1	I/O	-	Bit	SI/SCL			
		P63	1	I/O	-	Bit	INT0			
	Port 7	P70	1	I/O	-	Bit	TAOIN			
		P71	1	I/O	-	Bit	TA1OUT			
		P72	1	I/O	-	Bit	TA3OUT			
		P73	1	I/O	-	Bit	TA4IN			
		P74	1	I/O	-	Bit	TA5OUT			
	Port 8	P80	1	I/O	-	Bit	TB0IN0/INT5			
		P81	1	I/O	-	Bit	TB0IN1/INT6			
		P82	1	I/O	-	Bit	TB0OUT0			
		P83	1	I/O	-	Bit	TB0OUT1			
	Port 9	P90	1	I/O	-	Bit	TXD0			
www.DataSheet4	U.com	P91	1	I/O	-	Bit	RXD0			
		P92	1	I/O	-	Bit	SCLK0/ CTS0			
		P93	1	I/O	-	Bit	TXD1			
		P94	1	I/O	-	Bit	RXD1			
		P95	1	I/O	-	Bit	SCLK1/CTS1			
		P96	1	I/O	-	Bit	XT1			
		P97	1	I/O	-	Bit	XT2			

Table 3.6.1	Port Function
10010 0.0.1	

			3 (I/O Pagistar Satting Value			
	Ports	Pin Names	Specifications	I/O Regi	ster Settin	ig Values	
	1 0110		Cpeelineaderie	Pn	PnCR	PnFC	
	Port 0	P00 to P07	Input port	×	0		
			Output port	×	1	None	
			AD0 to AD7 bus	×	×		
	Port 1	P10 to P17	Input port	×	0	0	
			Output port	×	1	0	
			AD8 to AD15 bus	×	0	1	
			A8 to A15	×	1	1	
	Port 2	P20 to P25	Input port	×	0	0	
			Output port	×	1	0	
			A0 to A5 output	×	0	1	
			A16 to A21 output	×	1	1	
	Port 3	P30	Output port	×		0	
			RD output only when accessing an external area	1	None	1	
			Always RD output	0	1	1	
		P31	Output port	×		0	
			WR output only when accessing an external area	×	None	1	
		P32	Input port (without pull up)	0	0	0	
			Input port (with pull up)	1	0	0	
			Output port	×	1	0	
			HWR output	×	1	1	
	Port 4	P40 to P42	Input port (without pull up)	0	0	0	
			Input port (with pull up)	1	0	0	
			Output port	×	1	0	
		P40	CS0 output	×	1	1	
		P41	CS1 output	×	1	1	
		P42	CS2 output	×	1	1	
	Port 5	P50 to P53	Input port	×			
			AN0 to AN3 input (Note 1)	×	No	one	
		P53	ADTRG input (Note 2)	×			
www.DataSheet4U.com	Port 6	P60 to P63	Input port	×	0	0	
			Output port	×	1	0	
		P60	SCK input	×	0	0	
			SCK output	×	1	1	
		P61	SDA input	×	0	0	
			SDA output (Note 3)	×	1	1	
			SO output	×	1	1	
		P62	SI input	×	0	0	
			SCL input	×	0	0	
			SCL output (Note 3)	×	1	1	
		P63	INT0 input	×	0	1	

Table 3.6.2	I/O Port Setting I	ist (1/2)
10010 0.0.2		

X: Don't care

- Note 1: If use P50 to P53 as input channels of AD converter, channel selection set by using ADMODE1<ADCH2:0>.
- Note 2: If use P53 as ADTRG input, enabling external trigger is set by using ADMODE1<ADTRGE>.
- Note 3: If use P61 as open-drain output in SDA output and P62 as open-drain output in SCL output, please set ODE<ODE62:61>.

Porto	Din Nomoo	Specifications	I/O Register Setting Values				
FUILS	FILINALLES	Specifications	Pn	PnCR	PnFC		
Port 7	P70 to P74	Input port	×	0	0		
		Output port	×	1	0		
	P70	TA0IN input	×	0	None		
	P71	TA1OUT output	×	1	1		
	P72	TA3OUT output	×	1	1		
	P73	TA4IN input	×	0	None		
	P74	TA5OUT output	×	1	1		
Port 8	P80 to P83	Input port	×	0	0		
		Output port	×	1	0		
	P80	TB0IN0, INT5 input	×	0	1		
	P81	TB0IN1, INT6 input	×	0	1		
	P82	TB0OUT0 output	×	1	1		
	P83	TB0OUT1 output	×	1	1		
Port 9	P90 to P95	Input port	×	0	0		
		Output port	×	1	0		
	P90	TXD0 output	×	1	1		
	P91	RXD0 input	×	0	None		
	P92	SCLK0 input	×	0	0		
		SCLK0 output	×	1	1		
		CTS0 input	×	0	0		
	P93	TXD1 output	×	1	1		
	P94	RXD1 input	×	0	None		
	P95	SCLK1 input	×	0	0		
		SCLK1 output	×	1	1		
		CTS1 input	×	0	0		
	P96 to P97	Input port	×	0			
		Output port (Note 4)	×	1	None		
		XT1 to XT2 (Note 5)	×	0			

Table 3.6.3 I	O Port Setting	List (2/2)	
---------------	----------------	------------	--

X: Don't care

www.DataSheet4U.com Note 4: If use P96 to P97 as output port, it be set open-drain buffer.

Note 5: If use P96 to P97 as XT1 to XT2, enabling oscillation and so on are set by using SYSCR0 register.

By resetting, these port pin become general-purpose input port.

And it become input port except P96/XT1 and P97/XT2 that input pin of input and output is programmable.

When use port pin to internal function, need setting by program.

3.6.1 Port 0 (P00 to P07)

Port 0 is an 8-bit general-purpose I/O port. Each bit can be set individually for input or output using the control register POCR. Resetting, reset all bits of the control register POCR to "0" and sets port 0 to input mode.

In addition to functioning as a general-purpose I/O port, port 0 can also function as address data bus (AD0 to AD7).

When access external memory, port 0 function as address data bus (AD0 to AD7) and P0CR be cleared to "0".



Figure 3.6.1 Port 0

3.6.2 Port 1 (P10 to P17)

Port 1 is an 8-bit general-purpose I/O port. Each bit can be set individually for input or output using the control register P1CR and function register P1FC. Resetting reset all bits of output latch P1, the control register P1CR and function register P1FC to "0" and sets port 1 to input mode.

In addition to functioning as a general-purpose I/O port, port 1 can also function as address data bus (AD8 to AD15) and address bus (A8 to A15).



Figure 3.6.2 Port 1

				Port	0 Regis	ter			
		7	6	5	4	3	2	1	0
P0	Bit symbol	P07	P06	P05	P04	P03	P02	P01	P00
(0000H)	Read/Write				I	R/W			
	After reset		Data	a from extern	al port (Ou	tput latch regis	ter is und	lefined.)	
				Port 0 C	ontrol Re	egister			
		7	6	5	4	3	2	1	0
P0CR	Bit symbol	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
(0002H)	Read/Write					W			
	After reset	0	0	0	0	0	0	0	0
	Function	0: Input 1: 0	Output (Whei	n access to e	xternal, be	come AD7 to A	D0 and th	his register is cl	eared to "0".)
					•				
								→ Port I/O s	etting
								0 Ir	nput
								1 C	Output
								· · · · ·	
				Port	1 Reaist	ter			
		7	6	5	4	3	2	1	0
P1	Bit symbol	P17	P16	P15	P14	P13	 P12	P11	P10
(0001H)	Read/Write			1.10		R/W			
	After reset		Data f	from external	port (Outp	ut latch registe	er is clear	ed to "0".)	
	<u> </u>			Port 1 C	ontrol Re	eaister		,	
		7	6	5	4	3	2	1	0
P1CR	Bit symbol	P17C	P16C	P15C	P14C	P13C	 P12C	P11C	P10C
(0004H)	Read/Write				-	W	_	-	
	After reset	0	0	0	0	0	0	0	0
	Function			<<	Refer to co	olumn of P1FC	>>		
				Port 1 Fu	unction R	egister			
		7	6	5	4	3	2	1	0
P1FC	Bit symbol	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
(0005H)	Read/Write		•			W		•	
	After reset	0	0	0	0	0	0	0	0
ww.DataSheet4U.com	Function		P1FC/P1	CR = 00: Inp	ut, 01: Out	put, 10: AD15	to AD8, 1	1: A15 to A8	
					•				
	Note: Read	-modify-write	instructions	are					
	prohit	bited for regis	sters P0CR,	P1CR and					
	P1FC								
	-						> Por	rt1 function sett	ing
						P1FC	<p1xf></p1xf>	0	1
						P1CR <p1xc< td=""><td>></td><td>.</td><td></td></p1xc<>	>	.	
						0		Input Port	Address data b (AD15 to AD8
						1		Output Port	Address bus (A15 to A8)

- -.

Note: <P1XF>/<P1XC> is bit X of each register P1FC/P1CR.

Figure 3.6.3 Register for Ports 0 and 1

3.6.3 Port 2 (P20 to P25)

Port 2 is an 6-bit general-purpose I/O port. Each bit can be set individually for input or output using the control register P2CR and function register P2FC. Resetting, set all bits of output latch P2 to "1", and reset the control register P2CR and function register P2FC to "0" and sets port 2 to input mode.

In addition to functioning as a general-purpose I/O port, port 2 can also function as address bus (A0 to A5) and address bus (A16 to A21).





Figure 3.6.4 Port 2

				Port	2 Registe	er						
		7	6	5	4	3	2	1	0			
P2	Bit symbol	/	/	P25	P24	P23	P22	P21	P20			
(0006H)	Read/Write	/				R	Ŵ	•				
	After reset			Data from external port (Output latch register is set to "1".)								
				Port 2 C	Control Reg	gister						
		7	6	5	4	3	2	1	0			
P2CR	Bit symbol	/	/	P25C	C P24C P2		P22C	P21C	P20C			
(0008H)	Read/Write				•							
	After reset	/		0	0	0	0	0	0			
	Function				<<	Refer to colu	umn of P2F0	C>>				
	Port 2 Function Register											
Γ		7	6	5	4	3	2	1	0			
P2FC	Bit symbol	/		P25F	P24F	P23F	P22F	P21F	P20F			
(0009H)	Read/Write				W							
	After reset			0	0	0	0	0	0			
	Function			P2FC/P	2CR = 00: Ir	nput, 01: Out	put, 10: A5 t	5 to A0, 11: A21 to A16				
I	Note: Read prohit P2FC	-modify-write bited for reg	instructions	are and	F	P2FC P2CR <p2xc< td=""><td>→ Port 2 <p2xf></p2xf></td><td>function sett 0</td><td>ing1</td></p2xc<>	→ Port 2 <p2xf></p2xf>	function sett 0	ing1			
					0 Input Port			Address bus (A5 to A0)				

Output Port Note: <P2XF>/<P2XC> is bit X of each register P2FC/P2CR.

1

When set to address bus A21 to A16, set P2FC after set P2CR.

Address bus

(A21 to A16)

Figure 3.6.5 Register for Port 2

3.6.4 Port 3 (P30 to P32)

Port 3 is a 3-bit general-purpose I/O port (however P30 and P31 is only output port). Each bit can be set individually for input or output using the control register P3CR and function register P3FC. Resetting, all bits of output latch P3 is set to "1", and the control register P3CR (Bit0 and bit1 don't using) and function register P3FC are reset to "0". And P30 and P31 of port 3 output "High", and sets P32 to input mode with pull-up resister.

In addition to functioning as a general-purpose I/O port, port 3 can also function as the output for the CPU's control/status signal.

Case of P30 is defined as $\overline{\text{RD}}$ signal output mode (Case of <P30F> = "1"), when the output latch register <P30> clearing to "0", outputs the $\overline{\text{RD}}$ strobe (used for the pseudo static RAM) of the $\overline{\text{RD}}$ pin even when the internal addressed.

If the <P30 > remains "1", the $\overline{\text{RD}}$ strobe signal is output only when the external address area is accessed.



Figure 3.6.6 Port 3 (P30)



www.DataSheet4U.com

Figure 3.6.7 Port 3 (P31 and P32)

				Port	: 3 Registe	er				
		7	6	5	4	3	2	1	0	
P3	Bit symbol	/	/	/	/	/	P32	P31	P30)
(0007H)	Read/Write							R/W		
	After reset						Data from external port Note3	1	1	
	Function			,			Pull-up resistor 0: OFF 1: ON			
				Port 3 C	control Reg	gister				
		7	6	5	4	3	2	1	0	
P3CR	Bit symbol		\sim	\sim	· _	,	P32C		_	
(000AH)	Read/Write	\sim	\bigvee	\backslash	\backslash	\sim	W	\sim	\sim	
	After reset	\backslash	\bigvee	\backslash	\backslash	\sim	0	\sim	\sim	
							0: Input			_
								Innut/ Outpi	ıt settina	
								0 Input	it setting	
								1 Output		
				Port 3 Fi	unction Re	aister		- I		
	\sim	7	6	5	4	3	2	1	0	
P3FC	Bit symbol	_	\backslash	\sim	-	\sim	P32F	P31F	P30	F
(000BH)	Read/Write	W	\backslash	\backslash			1 021	W	1.001	·
	After reset	0					0	0	0	
	Function	Write "0".					0: Port	0: Port	0: Port	
							1: HWR	1: WR	1: RD	
								0 (RD) func	tion setti	ng
								30>		1
							<p30f< td=""><td>·>></td><td></td><td>"4" autaut</td></p30f<>	·>>		" 4 " autaut
							0		tout	
www.DataSheet4U.com								always	ipui	output during
							1	(Corres	pond	external
								to pseu	do	accesses.
									tion ootti	2
								231>		ing
							-P31F	0		1
							0	"0" outp	out "	'1" output
							1	WR is	only outp	out during
							P3	2 (HWR) fur	nction se	ttina
								32C>		
							<p32< td=""><td>× (</td><td></td><td>1</td></p32<>	× (1
							0	Input	port	Output port
							1	-		HWR output
	Note 1: Read	-modify-write	instrustions	are prohibite	d for registe	rs P3CR and	P3FC.			

- Note 2: When port 3 is used in Input mode, the P3 register controls the internal pull-up resistor. Read-modify-write instruction is prohibited in Input mode or I/O mode. Setting the internal pull-up resistor may be depend on the states of the input pin.
- Note 3: Output latch register is set to "1", and pull-up resistor is connected.

Figure 3.6.8 Register for Port 3

3.6.5 Port 4 (P40 to P42)

Port 4 is a 3-bit general-purpose I/O port. Each bit can be set individually for input or output using the control register P4CR and function register P4FC. Resetting, set P40 to P42 of output register to "1", the control register P4CR and function register P4FC reset to "0" and sets port 4 to input mode with pull-up resistor.

In addition to functioning as a general-purpose I/O port, port 4 can also function as chip select output signal ($\overline{\text{CS0}}$ to $\overline{\text{CS2}}$).



Figure 3.6.9 Port4

				Port	: 4 Registe	r					
		7	6	5	4	3	2	1	0		
P4	Bit symbol						P42	P41	P40		
	Read/Write					\square		R/W			
(000CH)	After reset						Data from				
	Function						Pull-up resistor				
			I	Port 4 C	Control Boc	nictor	0:	OFF 1:0	JN		
				F0114 C							
			6	5	4	$\overline{}$	2	1	0		
(000EH)	Bit symbol						P42C	P41C	P40C		
	Read/Write						-	W	-		
	After reset	\sim					0	0	0		
							0: 10	but 1:0	utput		
								1	Output		
				Port 4 Fu	unction Re	gister					
		7	6	Port 4 Fu 5	unction Re	gister 3	2	1	0		
P4FC	Bit symbol	7	6	Port 4 Fu	unction Re	gister 3	2 P42F	1 P41F	0 P40F		
P4FC (000FH)	Bit symbol Read/Write	7	6	Port 4 Fu	unction Re	gister	2 P42F	1 P41F W	0 P40F		
P4FC (000FH)	Bit symbol Read/Write After reset	7	6	Port 4 Fu	unction Re	gister 3	2 P42F 0	1 P41F W 0	0 P40F 0		
P4FC (000FH)	Bit symbol Read/Write After reset Function	7	6	Port 4 Fu	unction Re	gister 3	2 P42F 0 0	1 P41F W 0 : Port 1: CS	0 P40F 0		

- Note 1: Read-modify-write instructions are prohibited for registers, P4CR and P4FC.
- Note 2: When port 4 is used in Input mode, the P4 register controls the internal pull-up resistor. Read-modify-write instruction is prohibited in Input mode or I/O mode. Setting the internal pull-up resistor may be depend on the states of the input pin.
- Note 3: When output chip select signal (CSO to CS2), set bit of control register (P4CR) to "1" after set bit of function register (P4FC) to "1".
- Note 4: Output latch register is set to "1", and pull-up resistor is connected.

Figure 3.6.10 Register for Port 4

0

Port (P42) CS2 1

3.6.6 Port 5 (P50 to P53)

Port 5 is a 4-bit input port and can also be used as the analog input pin for the AD converter. P53 can also be used as AD trigger input pin for AD converter.



Figure 3.6.11 Port 5

	Port 5 Register									
		7	6	5	4	3	2	1	0	
P5	Bit symbol	/				P53	P52	P51	P50	
(000DH)	Read/Write	/				R				
www.DataSheet411.com	After reset						Data from e	external port		

Figure 3.6.12 Register for Port 5

Note: The input channel selection of AD converter and the permission of AD trigger input of P53 set by AD converter mode register ADMOD1.

3.6.7 Port 6 (P60 to P63)

Port 60 to P63 are 4-bit general-purpose I/O ports. Resetting, set to input port. All bits of output latch register P6 are set to "1".

In addition to functioning as a I/O port, port 6 can also function as input or output function of serial bus interface. This function enable each function by writing "1" to applicable bit of Port 6 function register P6FC.

Resetting, P6CR and P6FC reset to "0", all bit set input port.

(1) Port 60 (SCK)

In addition to functioning as an I/O port, port 60 can also function as clock SCK I/O port in SIO mode of serial bus interface.



Figure 3.6.13 Port 60

(2) Port 61 (SO/SDA)

In addition to functioning as an I/O port, port 61 can also function as data SDA I/O port in I^2C mode or data SO output pin in SIO mode of serial bus interface.



Figure 3.6.14 Port 61

(3) Port 62 (SI/SCL)

In addition to functioning as an I/O port, port 62 can also function as data receiving pin in SIO mode or clock SCL I/O pin in I²C bus mode of serial bus interface.



Figure 3.6.15 Port 62

(4) Port 63 (INT0)

In addition to functioning as an I/O port, port 63 can also function as INT0 input pin of external interrupt.



Figure 3.6.16 Port 63



Note: Read-modify-write instructions are prohibited for registers P6CR and P6FC.

Figure 3.6.17 Register for Port 6

3.6.8 Port 7 (P70 to P74)

Port 7 is a 5-bit general-purpose I/O port. Resetting, set to input port.

In addition to functioning as a I/O port, port 70 and 73 can also function as clock input pin TA0IN, TA4IN of 8-bit timer 0, 4 and port 71, 72, 74 can also function 8-bit timer output pin TA1OUT, TA3OUT, TA5OUT. This timer output function enable each function by writing "1" to applicable bit of Port 7 function register P7FC.

Resetting, P7CR and P7FC reset to "0", all bit set input port.



Figure 3.6.18 Port 7

Bit symbol Read/Write After reset Bit symbol Read/Write After reset		6	5 Port 7 C 5	4 P74 control Reg 4 P74C	3 P73 Data (Output lat gister 3	2 P72 R/W from externa ch register is 2	1 P71 al port s set to "1".) 1	0 P70		
Bit symbol Read/Write After reset Bit symbol Read/Write After reset		6	Port 7 C	P74 control Reg 4 P74C	P73 Data (Output lat gister 3	P72 R/W from externa ch register is 2	P71 al port s set to "1".)	P70		
Read/Write After reset Bit symbol Read/Write After reset	~ //	6	Port 7 C	control Reg 4 P74C	Data (Output lat gister 3	R/W from externa ch register is 2	al port s set to "1".) 1	0		
After reset	7	6	Port 7 C	ontrol Reg	Data (Output lat gister 3	from externa ch register is 2	al port s set to "1".) 1	0		
Bit symbol Read/Write After reset	7	6	Port 7 C	ontrol Reg 4 P74C	(Output lat gister 3	ch register is	s set to "1".)	0		
Bit symbol Read/Write After reset	7	6	Port 7 C	ontrol Reg 4	gister 3	2	1	0		
Bit symbol Read/Write After reset	7	6	5	4 P74C	3	2	1	0		
Bit symbol Read/Write After reset	\mathbb{N}	\mathbb{N}		P74C	5700					
Read/Write After reset			/	1740	P73C	P72C	P71C	P70C		
After reset				W						
				0	0	0	0	0		
Function				0: Input 1: Output						
							→ Por 0 1	t 7 I/O setting Input Output		
			Port 7 Fu	unction Re	gister 3	2	1	0		
			7 6	Port 7 Fu	Port 7 Function Re	Port 7 Function Register	Port 7 Function Register	Port 0 1 1 Port 7 Function Register 1		



Note 1: Read-Modify-Write instructions are prohibited for the registers P7CR and P7FC.

Note 2: P70/TA0IN and P73/TA4IN pin does not have a register changing Port/Function.

For example, when it is used as an input port, the input signal is inputted to 8-bit timer.

Figure 3.6.19 Register for Port 7

3.6.9 Port 8 (P80 to P83)

Port 8 is a 4-bit general-purpose I/O port. Resetting, set to input port. All bits of output latch register P8 are set to "1".

In addition to functioning as a I/O port, port 8 can also function as clock input of 16-bit timer, output of 16-bit timer F/F and input function of INT5 to INT6. This function enable each function by writing "1" to applicable bit of port 8 function register P8FC.

Resetting, P8CR and P8FC reset to "0", all bits set input port.

(1) P80 to P83




P8 (0018H) 7 6 5 4 3 2 1 Bit symbol Read/Write P83 P83 P82 P81 After reset R/W Data from external port (Output latch register is set to " Port 8 Control Register P8CR (001AH) 7 6 5 4 3 2 1 P8CR (001AH) 8 983C P82C P81C W After reset 0 0 0 0 0 After reset 0 0 0 0 0 D 0 0 0 0 0 0	1 0 P81 P8(nal port is set to "1".) 1 0 81C P80(0 0 Output 0	1 P81 W external port gister is set to 1 P81C W	2 P82 Data from e put latch reg 2 P82C	3 P83 (Out	4	5	6	7	Bit symbol
P8 (0018H) Bit symbol P83 P82 P81 Read/Write R/W R/W Data from external port (Output latch register is set to " Port 8 Control Register P8CR (001AH) T 6 5 4 3 2 1 Bit symbol T 6 5 4 3 2 1 P8CR (001AH) Bit symbol T 6 5 4 3 2 1 P8CR (001AH) Bit symbol O 0 0 0 0 0 P8CR (001AH) Bit symbol O 0 0 0 0 0 P8CR (001AH) Bit symbol O 0 0 0 0 0 P8CR (001AH) Bit symbol O 0 0 0 0 P8CR (001AH) Bit symbol D D 0 0 0 0	P81 P8(nal port	P81 R/W external port gister is set to 1 P81C W	P82 R Data from e put latch reg 2 P82C	P83 (Out			//	//	Bit symbol
Read/Write RW After reset Data from external port (Output latch register is set to " Port 8 Control Register P8CR (001AH) 7 6 5 4 3 2 1 P8CR (001AH) 8it symbol P83C P82C P81C Read/Write W 0 0 0 After reset 0 0 0 0	1 0 81C P800 0 0 0 0 0 0	R/W external port gister is set to 1 P81C W	R Data from e put latch reg 2 P82C	(Out			\backslash		Pood/M/rito
After reset Data from external port (Output latch register is set to " Port 8 Control Register P8CR (001AH) 7 6 5 4 3 2 1 Bit symbol Read/Write 7 6 5 4 3 2 1 V V V V V V V After reset 0 0 0 0	1 0 81C P800 0 0 0 0 0 0	external port gister is set to 1 P81C W	Data from e put latch reg 2 P82C	(Out		$\overline{}$			Reau/White
Port 8 Control Register P8CR (001AH) 7 6 5 4 3 2 1 P8CR (001AH) 8it symbol 983C P82C P81C W After reset 0 0 0 0 0 0	is set to "1".) 1 0 81C P800 0 0 0 0 Output 0	gister is set tr 1 P81C W	put latch reg 2 P82C	(Out gister					After reset
Port 8 Control Register P8CR (001AH) 7 6 5 4 3 2 1 Bit symbol P83C P83C P82C P81C Read/Write W W 0 0 0 After reset 0 0 0 0 0	1 0 81C P800 0 0 Dutput	1 P81C W	2 P82C	gister					
P8CR (001AH) 7 6 5 4 3 2 1 Bit symbol Read/Write P83C P82C P81C W After reset 0 0 0 0 0: Input 1: Output 1: Output 1: Output	1 0 81C P800 0 0 Dutput	1 P81C W	2 P82C		control Reg	Port 8 C			
P8CR (001AH) Bit symbol P83C P82C P81C Read/Write W W After reset 0 0 0: Input 1: Output	81C P80 0 0 Dutput	P81C	P82C	3	4	5	6	7	
After reset 0 0 0 0 After reset 0 0 0 0 0 Rept 8	0 0 Dutput	<u>W</u>		P83C					Bit symbol
After reset 0 0 0 0 0: Input 1: Output	0 0 Dutput	-	١						Read/Write
0: Input 1: Output	Dutput	0	0	0					After reset
Port 9	· · · · · · · · · · · · · · · · · · ·	1: Output	0: Input						
Port 8 Function Register	1 Output	1		gister	unction Re	Port 8 Fu			
	1 0	1	2	3	4	5	6	7	
						•		~	
Bit symbol P83F P82F P81F	81F P80	POIL	P82F	P83F		\checkmark			Bit symbol
P8FC Bit symbol P83F P82F P81F (001BH) Read/Write W W W	81F P80 W W	W	P82F W	P83F W			//		Bit symbol Read/Write
Bit symbol P83F P82F P81F Read/Write W W W After reset 0 0 0	81F P80 W W 0 0	0	P82F W 0	P83F W 0			///		Bit symbol Read/Write After reset
Bit symbol P8FC (001BH) P83F P82F P81F After reset 0 0 0 0 Function 0: Port 0: Port 0: Port 0: Port	81F P80 W W 0 0 ort 0: Port	0 0: Port	P82F W 0 0: Port	P83F W 0 0: Port					Bit symbol Read/Write After reset Function
P8FC (001BH) Bit symbol P83F P82F P81F After reset 0 0 0	81F P80 W W 0 0	0	P82F W 0	P83F W 0			///		Bit symbol Read/Write After reset
Bit symbol P83F P82F P81F Read/Write W W W After reset 0 0 0 Function 0: Port 0: Port 0: Port 0: Port I: TB0OUT1 1: TB0OUT0 1: TB0IN1 INT6 input	81F P80 W W 0 0 vrt 0: Port 30IN1 1: TB0IN F6 input INT5 in	0: Port 1: TB0IN1 INT6 input	P82F W 0 0: Port 1: TB0OUT0	P83F W 0 0: Port 1: TB0OUT1					Bit symbol Read/Write After reset Function
Bit symbol P83F P82F P81F Read/Write W W W W After reset 0 0 0 0 Function 0: Port 0: Port 0: Port 0: Port 0: Port 0: TB0/UT1 W W W W W W W W	81F P80 W W 0 0 ort 0: Port 30IN1 1: TB0IN T6 input INT5 input 2 TB0OUT0 out	P81F W 0 0: Port 1: TB0IN1 INT6 input	P82F W 0 0: Port 1: TB0OUT0	P83F W 0 0: Port 1: TB0OUT1					Bit symbol Read/Write After reset Function
Bit symbol P83F P82F P81F Read/Write W W W W After reset 0 0 0 0 Function 0: Port 0: Port 0: Port 0: Port 1: TB00UT1 NT6 input P82 TB00U P82 TB00U P87C P82 TB00U	81F P80 W W 0 0 ort 0: Port 30IN1 1: TB0IN T6 input INT5 i 2 TB00UT0 out 3FC <p82f></p82f>	P81F W 0 0: Port 1: TB0IN1 INT6 input ► P82 TB0C P8FC<<	P82F W 0 0: Port 1: TB0OUT0	P83F W 0 0: Port 1: TB0OUT1					Bit symbol Read/Write After reset Function

Note: Read-modify-write instructions are prohibited for registers P8CR and P8FC.

Figure 3.6.21 Register for Port 8

3.6.10 Port 9 (P90 to P97)

• Ports 90 to 95

Ports 90 to 95 are a 6-bit general-purpose I/O port. Resetting, set to input port. All bits of output latch register are set to "1".

In addition to functioning as a I/O port, port 90 to 95 can also function as I/O of SIO0, SIO1. This function enable each function by writing "1" to applicable bit of port 9 function register P9FC.

Resetting, P9CR and P9FC reset to "0", all bits set input port.

• Ports 96 to 97

Ports 96 to 97 are a 2-bit general-purpose I/O port. Case of output port, this is open drain output. Resetting, output latch register and control register set to "1", and set to "High-Z" (High impedance).

In addition to functioning as a I/O port, ports 96 to 97 can also function as low-frequency oscilator connection pin (XT1 and XT2) during using low speed clock function. Therefore, dual clock function can use by setting of system clock control registers SYSCR0 and SYSCR1.

(1) Ports 90 and 93 (TXD0 and TXD1)

In addition to functioning as a I/O port, Ports 90 and 93 can also function as TXD output pin of serial channel.

And P90 and P93 have a programmable open-drain function which can be controlled by the ODE<ODE90, 93> register.



Figure 3.6.22 Ports 90 and 93

(2) Ports 91 and 94 (RXD0 and RXD1)

In addition to functioning as a I/O port, ports 91 and 94 can also function as RXD input pin of serial channel.



Figure 3.6.23 Ports 91 and 94

(3) Ports 92 and 95 (CTS0 /SCLK0, CTS1 /SCLK1)

In addition to functioning as a I/O port, ports 92 and 95 can also function as $\overline{\text{CTS}}$ input pin or SCLK I/O pin of serial channel.



Figure 3.6.24 Port 92, 95

(4) Ports 96 (XT1) and 97 (XT2)

In addition to functioning as a I/O port, ports 96 and 97 can also function as low frequency oscillator connection pins.





				Port	9 Registe	ers				
		7	6	5	4	3	2	1	0	
P9	Bit symbol	P97	P96	P95	P94	P93	P92	P91	P90	
(0019H)	Read/Write				R	/W	•			
	After reset	1	1			Data from	external port			
					(Ou	tput latch re	gister is set to	"1".)		
				Port 9 (Control Re	gister				
		7	6	5	4	3	2	1	0	
P9CR	Bit symbol	P97C	P96C	P95C	P94C	P93C	P92C	P91C	P90C	
(001CH)	Read/Write				\	W				
	After reset	1	1	0	0	0	0	0	0	
					0: Input	1: Outpu	t		1	
	B	<u>t </u>			•	•				
					I			\rightarrow Port9 I/C) setting	
								1	Outout	
								I	Oulpul	
	Note: Ports	96 and 97 ar	e open-drain	output pins.						
				Port 9 F	unction Re	egister				
DOFO		7	6	5	4	3	2	1	0	
(001DH)	Bit symbol			P95F		P93F	P92F		P90F	
(001011)	Read/Write	\backslash		W		W	W		W	
	After reset			0		0	0	/	0	
	Function			0: Port		0: Port	0: Port		0: Port	
				1: SCLK1		1: TXD1	1: SCLK0		1: TXD0	
				output		<u> </u>	output			
								• P90 TXD0	output setting	g
								P9FC <p9< td=""><td>90F></td><td></td></p9<>	90F>	
								P9CR <p9< td=""><td>90C></td><td></td></p9<>	90C>	
							L	P92 SCLK	0 output setti	ng
vw.DataSheet4U.com								P9FC <p9< td=""><td>)2F></td><td></td></p9<>)2F>	
								P9CR <p9< td=""><td>92C></td><td></td></p9<>	92C>	
								• P93 TXD1	output setting	g
								P9FC <ps< td=""><td>)3F></td><td></td></ps<>)3F>	
								P90R <p< td=""><td>350></td><td></td></p<>	350>	
									1 outout cotti	na
							,			ng
								P9CR <p9< td=""><td>95C></td><td></td></p9<>	95C>	
	Note 1: Rea	d-modify-write	ainstructions	are prohibite	ad for the rec	nistore POCE	and POFC			
	Note 2 [·] Whe	n set TXD pi	n to open-dr	ain output, w	rite "1" to bit	0 of ODF re	aister (for TXI	00 pin), or b	it1 (for TXD1	pin).
	P91/	RXD0 and P	94/RXD1 pin	does not hav	ve a register	changing P	ort/Function.	20 p), er o		P
	For e	example, whe	en it is also u	sed as an inc	out port, the i	nput signal i	s inputted to S	SIO as serial	receiving dat	a.
	Note 3: Low	frequency os	cillation circu	ıit '					0	
	То с	onnect a low	frequency re	sonator to po	orts 96 and 9	97, it is nece	ssary to set a	following pr	ocedure to re	duce
	the c	consumption (power supply	<i>.</i>						
	(Cas	e of resonato	or connection)						
	P9C	R <p96c, p97<="" td=""><td>7C> = "11", F</td><td>9<p96:97> =</p96:97></td><td>= "00"</td><td></td><td></td><td></td><td></td><td></td></p96c,>	7C> = "11", F	9 <p96:97> =</p96:97>	= "00"					

Figure 3.6.26 Register for Port 9

(Case of oscillator connection)

P9CR<P96C, P97C> = "11", P9<P96:97> = "10"

3.7 Chip select/Wait Controller

On the TM91CP27, four user-specifiable address areas (CS0 to CS3) can be set. The data bus width and the number of waits can be set independently for each address area (CS0 to CS3 and others).

The pins $\overline{\text{CS0}}$ to $\overline{\text{CS2}}$ (which can also function as port pins P40 to P42) are the respective output pins for the CS0 to CS2 areas. When the CPU specifies an address in one of these areas, the corresponding CS0 to CS2 pin outputs the chip select signal for the specified address area (in ROM or SRAM). However, in order for the chip select signal to be output, the port 4 control register P4CR and function register P4FC must be set.

The areas CS0 to CS3 are defined by the values in the memory start address registers MSAR0 to MSAR3 and the memory address mask registers MAMR0 to MAMR3.

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

3.7.1 Specifying an Address Area

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

At each bus cycle, a compare operation is performed to determine if the address on the specified a location in the CS0 to CS3 area. If the result of the comparison is a match, this indicates an access to the corresponding CS area. In this case, the $\overline{\text{CS0}}$ to $\overline{\text{CS2}}$ pin outputs the chip select signal and the bus cycle operates in accordance with the settings in chip select/wait control register B0CS to B3CS. (See section 3.7.2, "Chip Select/Wait Control Registers".)

(1) Memory start address registers

Figure 3.7.1 shows the memory start address registers. The memory start address registers MSAR0 to MSAR3 set the start addresses for the CS0 to CS3 areas. Set the upper 8 bits (A23 to A16) of the start address in <S23:S16>. The lower 16 bits of the start address (A15 to A0) are permanently set to 0. Accordingly, the start address can only be set in 64-Kbyte increments, starting from 000000H. shows the relationship between the start address and the start address register value.

		7	6	5	4	3	2	1	0			
MSAR0 /MSAR1	Bit symbol	S23	S22	S21	S20	S19	S18	S17	S16			
(00C8H)/ (00CAH)	Read/Write	R/W										
MSAR2 /MSAR3	After reset	1	1	1	1	1	1	1	1			
(00CCH) (00CEH)	Function			Determ	ines A23 to /	A16 of start a	address					
	l				•							

Memory Start Address Register (CS0 to CS3 area)

→ Sets start addresses for areas CS0 to CS3



		Start address	Start address register value (MSAR0 to MSAR3)
Address 000000H	64 Khytes		00H
	04 Rbytes	<────010000H	01H
		< 020000H	02H
		<──── 030000H	03Н
		< 040000H	04H
		< 050000H	05H
		• 060000H	06Н
		:	:
FFFFFFH		✓ FF0000H	FFH

www.DataSheet4U.com

Figure 3.7.2 Start Address and Start Address Register Value

(2) Memory address mask registers

Figure 3.7.3 shows the memory address mask registers. Memory address mask registers MAMR0 to MAMR3 are used to set the size of the CS0 to CS3 areas by specifying a mask for each bit of the start address set in memory start address registers MSAR0 to MSAR3. The compare operation used to determine if an address is in the CS0 to CS3 areas is only performed for bus address bits corresponding to bits set to "0" in these registers. Also, the address bits that can be masked by MAMR0 to MAMR3 differ between CS0 to CS3 areas. Accordingly, the size that can be each area is different.

-			Memor	y Audiess	inger ive				
		7	6	5	4	3	2	1	0
MAMR0	Bit symbol	V20	V19	V18	V17	V16	V15	V14 to 9	V8
(00C9H)	Read/Write				R/	W			
	After reset	1	1	1	1	1	1	1	1
Function Sets area size of CS0. 0: Used for address compare									

Memory Address Mask Register (CS0 Area)

Range of possible settings for CS0 area size: 256 bytes to 2 Mbytes

Memory Address Mask Register (CS1 Area)

				•					
	/	7	6	5	4	3	2	1	0
MAMR1	Bit symbol	V21	V20	V19	V18	V17	V16	V15 to 9	V8
(00CBH)	Read/Write	e R/W							
	After reset	1	1	1	1	1	1	1	1
	Function		Se	ets area size	of CS1. 0:	Used for ad	dress compa	ire	

Range of possible settings for CS1 area size: 256 bytes to 4 Mbytes.

			Memory A	ddress Ma	ask Regist	er (CS2 aı	nd CS3 Ar	ea)	
		7	6	5	4	3	2	1	0
MAMR2 /MAMR3	Bit symbol	V22	V21	V20	V19	V18	V17	V16	V15
(00CDH) (00CFH)	Read/Write				R/	W			
	After reset	1	1	1	1	1	1	1	1
	Function Sets area size of CS2 and CS3. 0: Used for address compare								

www.DataSheet4U.com

Range of possible settings for CS2 and CS3 area sizes: 32 Kbytes to 8 Mbytes.

Figure 3.7.3 Memory Address Mask Register

(3) Setting memory start address and address area

Figure 3.7.4 show an example of specifying a 64-Kbyte address area starting from 010000H using the CS0 areas.

Set "01H" in memory start address registers MSAR0<S23:16> (Corresponding to the upper 8 bits of the start address). Next, calculate the difference between the start address and the anticipated end address (01FFFFH). Bits 20 to 8 of the result correspond to the mask value to be set for the CS0 area. Setting this value in memory address mask register MAMR0<V20:8> sets the area size. This example sets "07H" in MAMR0 to specify a 64-Kbyte area.





After a reset, MSAR0 to MSAR3 and MAMR0 to MAMR3 are set to "FFH". B0CS<B0E>, B1CS<B1E> and B3CS<B3E> are reset to "0". Therefore this is disabling the CS0, CS1 and CS3 areas. However, set as B2CS<B2M> to "0" and B2CS<B2E> to "1", CS2 is enabled from 002000H to FF3FFFH in TMP91CP27. Also, the bus width and number of waits specified in BEXCS are used for accessing addresses outside the specified CS0 to CS3 area. (See section 3.7.2, "Chip Select/Wait Control Registers".)

(4) Address area size specification

Table 3.7.1 shows the relationship between CS area and area size. " Δ " indicates areas that cannot be set by memory start address register and address mask register combinations. When setting an area size using a combination indicated by " Δ ", set the start address in the desired steps starting from 000000H.

If the CS2 area is set to 16 Mbytes or if two or more areas overlap, the smaller CS area number has the higher priority.

Example: To set the area size for CS0 to 128 Kbytes:

a. Valid start addresses



b. Invalid start addresses

000000H	\sum	64 Kbytes)	
010000H	Ĵ	128 Kbytes	ł	This is not an integer multiple of the desired area size setting. Hence, none of these
030000H	Ĵ	128 Kbytes		addresses can be set as the start address.
050000H	-)	

	Size (Byte) CS Area	256	512	32 K	64 K	128 K	256 K	512 K	1 M	2 M	4 M	8 M
	CS0	0	0	0	0	Δ	Δ	Δ	Δ	Δ		
	CS1	0	0		0	Δ	Δ	Δ	Δ	Δ	Δ	
www.DataSheet/II	CS2			0	0	Δ	Δ	Δ	Δ	Δ	Δ	Δ
www.bataoneet+0.	CS3			0	0	Δ	Δ	Δ	Δ	Δ	Δ	Δ

Table 3.7.1 Valid Area Sizes for Each CS Area

Note: Δ indicates areas that cannot be set by memory start address register and memory address mask register combinations.

3.7.2 Chip Select/Wait Control Registers

Figure 3.7.5 lists the chip select/wait control registers.

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

			-						
		7	6	5	4	3	2	1	0
BOCS	Bit symbol	B0E	/	B0OM1	B0OM0	B0BUS	B0W2	B0W1	B0W0
(00C0H)	Read/Write	W	/			۱. ۱	N		
	After reset	0	/	0	0	0	0	0	0
	Function	0: Disable		Chip select	output	Data bus	Number of	waits	1
		1: Enable		waveform s	election	width	000: 2 waits	100: Re	served
				00: For ROM	/SRAM	0: 16 bits	001: 1 wait	101: 3 w	vaits
				01: Don't care	е	1 [·] 8 bits	010: (1 + N) v	waits 110:4 w	vaits
				10: Don't care	e		011: 0 waits	111: 8 w	vaits
			~	11: Don't care	e				
B1CS	Bit symbol	B1E		B1OM1	B1OM0	B1BUS	B1W2	B1W1	B1W0
(00C1H)	Read/Write	W				N	V		
	After reset	0		0	0	0	0	0	0
	Function	0: Disable		Chip select	output	Data bus	Number of	waits	
		1: Enable		waveform s	election	width	000: 2 waits	100: Re	served
				00: For ROM	/SRAM	0: 16 bits	001: 1 wait	101: 3 w	vaits
				01: Don't care	Э	1: 8 bits	010: (1 + N) v	waits 110:4 w	vaits
				10: Don't care	9		011: 0 waits	111: 8 w	vaits
				11: Don't care	9				
B2CS	Bit symbol	B2E	B2M	B2OM1	B2OM0	B2BUS	B2W2	B2W1	B2W0
(00C2H)	Read/Write				\ \	N			-
	After reset	1	0	0	0	0	0	0	0
	Function	0: Disable	CS2 area	Chip select	output	Data bus	Number of	waits	
		1: Enable	selection	waveform s	election	width	000: 2 waits	100: Re	served
			0: 16-Mbyte	00: For ROM	/SRAM	0: 16 bits	001: 1 wait	101: 3 w	vaits
			area	10: Don't care	9	1: 8 bits	010: (1 + N) V	vaits 110:4 w	vaits
			1: CS area	11: Don't care	5		011. 0 Walts	111.0 %	vaits
B3CS	Bit symbol	B3E	/	B3OM1	B3OM0	B3BUS	B3W2	B3\W1	B3\W/0
(00C3H)	Read/Write	W		DOOMIT	Doomo	DODOO	N	DOWN	Domo
, , , , , , , , , , , , , , , , , , ,		0		0	0		0	0	0
	Function	0 Dischla		Chin coloct	outout	Data bua	Number of		0
www.DataSheet4U.com	Function			waveform s	election	width			aanvad
		1: Enable		00: For ROM	/SRAM	0: 16 hito	000. 2 waits	100. Re 101: 3 w	vaits
				01: Don't care	9		010: (1 + N)	vaits 110:4 w	vaits
				10: Don't care	9	1:8 Dits	011: 0 waits	111: 8 w	vaits
				11: Don't care	Э				
BEXCS	Bit symbol	/	/	/		BEXBUS	BEXW2	BEXW1	BEXW0
(00C7H)	Read/Write	\sim	\sim	\sim	\sim		V	N	
	After reset	\sim	\sim	\sim	\sim	0	0	0	0
	Function					Data bus	Number of	Waits	
						width	000: 2 waits	100: Re	served
						0: 16 bits	001: 1 wait	101: 3 w	vaits
						1.8 bits	010: (1 + N) v	waits 110:4 w	vaits
							011: 0 waits	111: 8 w	vaits
	Master enable	bit 🗸		Lq				•	
Г		S aroc	—	Chip select ou	r Itput waveform	ı	Number	of address are	a waite
ł			——	selection		л	(See section	i 3.7.2, "(3) Wa	it Control.")
l	1 Enable CS area		00 For RO	M/SRAM		→ Data bus width selection			
	CS2 area selection		01 Don't ca	are		0 16.bi	t data hue		
Ι	0 16-Mhvte	area		10 Don't ca	are		1 0 6:4	data bua	
ł		addroca area		11 Don't ca	are		I Ö-DIT	uata dus	
	 Specified 	audress area	a '			-			

Chin	Select/Wait	Control	Registe
	Select/Wall	COLLED	REGISTE

Note: Read-modify-write instructions are prohibited for registers B0CS, B1CS, B2CS, B3CS and BEXCS.

Figure 3.7.5 Chip Select/Wait Control Register

(1) Master enable bits

Bit7 (<B0E>, <B1E>, <B2E> or <B3E>) of a chip select/wait control register is the master bit that is used to enable or disable settings for the corresponding address area. Writing "1" to this bit enables the settings. Reset disables (Sets to "0") <B0E>, <B1E> and <B3E>, and enables (sets to "1") <B2E>. This enables area CS2 only.

(2) Data bus width selection

Bit3 (<B0BUS>, <B1BUS>, <B2BUS>, <B3BUS> or <BEXBUS>) of a chip select/wait control register specifies the width of the data bus. This bit should be set to "0" when memory is to be accessed using a 16-bit data bus and to "1" when an 8-bit data bus is to be used.

This process of changing the data bus width according to the address being accessed is known as "dynamic bus sizing". For details of this bus operation see Table 3.7.2.

	Operand Data	Operand Start	Memory Data		CPU	Data
	Bus Width	Address	Bus Width	CF 0 Address	D15 to D8	D7 to D0
	8 bits	2n + 0	8 bits	2n + 0	XXXXX	b7 to b0
		(Even number)	16 bits	2n + 0	XXXXX	b7 to b0
		2n + 1	8 bits	2n + 1	XXXXX	b7 to b0
		(Odd number)	16 bits	2n + 1	b7 to b0	XXXXX
	16 bits	2n + 0	8 bits	2n + 0	XXXXX	b7 to b0
		(Even number)		2n + 1	xxxxx	b15 to b8
			16 bits	2n + 0	b15 to b8	b7 to b0
		2n + 1	8 bits	2n + 1	XXXXX	b7 to b0
		(Odd number)		2n + 2	xxxxx	b15 to b8
			16 bits	2n + 1	b7 to b0	XXXXX
				2n + 2	XXXXX	b15 to b8
	32 bits	2n + 0	8 bits	2n + 0	XXXXX	b7 to b0
		(Even number)		2n + 1	XXXXX	b15 to b8
				2n + 2	xxxxx	b23 to b16
www.DataSheet4U.co	m			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	8 bits	2n + 1	XXXXX	b7 to b0
		(Odd number)		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

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

(3) Wait control

Bits 0 to 2 (<B0W0:2>, <B1W0:2>, <B2W0:2>, <B3W0:2>, <BEXW0:2>) of a chip select/wait control register specify the number of waits that are to be inserted when the corresponding memory area is accessed.

The following types of wait operation can be specified using these bits. Bit settings other than those listed in the table should not be made.

<bxw2:0></bxw2:0>	Number of Waits	Wait Operation
000	2	Inserts a wait of 2 states.
001	1	Inserts a wait of 1 state.
010	(1 + N)	Same operation with 1 wait because of nothing WAIT pin.
011	0	Ends the bus cycle without a wait.
100	Reserved	Invalid setting
101	3	Inserts a wait of 3 states.
110	4	Inserts a wait of 4 states.
111	8	Inserts a wait of 8 states.

Table 3.7.3	Wait Operation	Settino
14010 0.1.0	man operation	oottinig

Note: A Reset sets these bits to "000" (2 waits).

(4) Bus width and wait control for an area other than CS0 to CS3

The chip select/wait control register BEXCS controls the bus width and number of waits when memory locations that are not in one of the four user-specified address areas (CS0 to CS3) are accessed. The BEXCS register settings are always enabled for areas other than CS0 to CS3.

(5) Selecting 16-Mbyte area/specified address area

Setting B2CS<B2M> (Bit6 of the chip select/wait control register for CS2) to "0" designates the 16-Mbyte areas 002000H to FF3FFFH as the CS2 area. Setting B2CS<B2M> to "1" designates the address area specified by the start address register MSAR2 and the address mask register MAMR2 as CS2 (e.g., if B2CS<B2M> = 1, CS2 is specified in the same manner as CS0, CS1 and CS3 are).

A Reset clears this bit to "0", specifying CS2 as a 16-Mbytes address area.

(6) Procedure for setting chip select/wait control

When using the chip select/wait control function, set the registers in the following order:

- a. Set the Memory Start Address Registers MSAR0 to MSAR3. Set the start addresses for CS0 to CS3.
- b. Set the Memory Address Mask Registers MAMR0 to MAMR3. Set the sizes of CS0 to CS3.
- c. Set the chip select/wait control registers B0CS to B3CS. Set the chip select output waveform, data bus width, number of waits and master enable/disable status for CS0 to CS3 areas.

The $\overline{\text{CS0}}$ to $\overline{\text{CS2}}$ pins can also function as pins P40 to P42. To output a chip select signal using one of these pins, set the corresponding bit in the port 4 function register P4FC and port 4 control register P4CR to "1".

If a CS0 to CS3 address is specified which is actually an internal I/O, RAM and ROM area address, the CPU accesses the internal address area and no chip select signal is output on any of the $\overline{\text{CS0}}$ to $\overline{\text{CS2}}$ pins.

Example:

In this example CS0 is set to be the 64-Kbyte area 010000H to 01FFFFH. The bus width is set to 16 bits and the number of waits is cleared to 0.

MSAR0 = 01H	Start address: 010000H
MAMR0 = 07H	Address area: 64 Kbytes
BOCS = 83H	ROM/SRAM, 16-bit data bus, zero waits, CS0 area settings enabled $% \mathcal{C}$

3.7.3 Connecting External Memory

Figure 3.7.6 shows an example of how to connect external memory to the TMP91CP27. In this example the ROM is connected using a 16-bit bus. The RAM and I/O are connected using an 8-bit bus.



Figure 3.7.6 Example of External Memory Connection

(ROM uses 16-bit bus, RAM and I/O uses 8-bit bus)

A reset clears all bits of the port 4 control register P4CR and the port 4 function register P4FC to "0" and disables output of the CS signal. To output the CS signal, the appropriate bit must be set P4CR to "1" after set P4FC to "1".

3.8 8-Bit Timers (TMRA)

The TMP91CP27 features 6 channels (TMRA0 to TMRA5) built-in 8-bit timers. These timers are paired into 3 modules: TMRA01, TMRA23 and TMRA45. Each module consists of 2 channels and can operate in any of the following 4 operating modes.

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

Figure 3.8.1 to Figure 3.8.3 show block diagrams for TMRA01, TMRA23 and TMRA45.

Each channel consists of an 8-bit up counter, an 8-bit comparator and an 8-bit timer register. In addition, a timer flip-flop and a prescaler are provided for each pair of channels.

The operation mode and timer flip-flops are controlled by 5 bytes registers (SFRs: Special function registers).

Each of the three modules (TMRA01, TMRA23 and TMRA45) can be operated independently. All modules operate in the same manner; hence only the operation of TMRA01 is explained here.

Specifica	Module	TMRA01	TMRA23	TMRA45
External	Input pin for external clock	TA0IN (Shared with P70)	None	TA4IN (Shared with P73)
pins	Output pin for timer flip-flop	TA1OUT (Shared with P71)	TA3OUT (Shared with P72)	TA5OUT (Shared with P74)
	Timer RUN register	TA01RUN (0100H)	TA23RUN (0108H)	TA45RUN (0110H)
SFR	Timer register	TA0REG (0102H) TA1REG (0103H)	TA2REG (010AH) TA3REG (010BH)	TA4REG (0112H) TA5REG (0113H)
(Address)	Timer mode register	TA01MOD (0104H)	TA23MOD (010CH)	TA45MOD (0114H)
(, (301000))	Timer flop-flop control register	TA1FFCR (0105H)	TA3FFCR (010DH)	TA5FFCR (0115H)

Table 3.8.1 Registers and Pins for Each Module

www.DataSheet4U.com

3.8.1 Block Diagram



Figure 3.8.1 Block Diagram of TMRA01



Figure 3.8.2 Block Diagram of TMRA23



Figure 3.8.3 Block Diagram of TMRA45

3.8.2 Operation of Each Circuit

(1) Prescalers

A 9-bit prescaler generates the input clock to TMRA01.

The ϕ T0 as the input clock to prescaler is a clock divided by 4, which selected using the prescaler clock selection register SYSCR0<PRCK1:0>.

The prescaler's operation can be controlled using TA01RUN<TA0PRUN> in the timer control register. Setting <TA0PRUN> to "1" starts the count; setting <TA0PRUN> to "0" clears the prescaler to zero and stops operation. Table 3.8.2 shows the various prescaler output clock resolutions.

					@fc = 16IMI	HZ, fS = 32.768 KHZ
System Clock	Prescaler Clock	Clock Gear		Prescaler Output	ut Clock Resoluti	on
Selection <sysck></sysck>	Selection <prck1:0></prck1:0>	Value <gear2:0></gear2:0>	φT1	φ T 4	φT16	φT256
1 (fs)		XXX	fs/2 ³ (244 μs)	fs/2 ⁵ (977 μs)	fs/27 (3.9 ms)	fs/2 ¹¹ (62.5 ms)
		000 (fc)	fc/2 ³ (0.5 μs)	fc/2 ⁵ (2.0 μs)	fc/2 ⁷ (8.0 μs)	fc/2 ¹¹ (128 μs)
	00	001 (fc/2)	fc/2 ⁴ (1.0 μs)	fc/2 ⁶ (4.0 μs)	fc/2 ⁸ (16 μs)	fc/2 ¹² (256 μs)
	(f _{FPH})	010 (fc/4)	fc/2 ⁵ (2.0 μs)	fc/2 ⁷ (8.0 μs)	fc/2 ⁹ (32 μs)	fc/2 ¹³ (512 μs)
0 (fc)		011 (fc/8)	fc/2 ⁶ (4.0 μs)	fc/2 ⁸ (16 μs)	fc/2 ¹⁰ (64 μs)	fc/2 ¹⁴ (1024 μs)
		100 (fc/16)	fc/2 ⁷ (8.0 μs)	fc/2 ⁹ (32 μs)	fc/2 ¹¹ (128 μs)	fc/2 ¹⁵ (2048 μs)
	10 (fc/16 clocks)	XXX	fc/2 ⁷ (8.0 μs)	fc/2 ⁹ (32 μs)	fc/2 ¹¹ (128 μs)	fc/2 ¹⁵ (2048 μs)

Table 3.8.2 Prescaler Output Clock Resolution

xxx: Don't care

(2) Up counters (UC0 and UC1)

These are 8-bit binary counters which count up the input clock pulses for the clock specified by TA01MOD.

The input clock for UC0 is selectable and can be either the external clock input via the TA0IN pin or one of the three internal clocks ϕ T1, ϕ T4, and ϕ T16. The clock setting is specified by the value set in TA01MOD<TA0CLK1:0>.

The input clock for UC1 depends on the operation mode. In 16-bit timer mode, the overflow output from UC0 is used as the input clock. In any mode other than 16-bit timer mode, the input clock is selectable and can either be one of the internal clocks ϕ T1, ϕ T16, and ϕ T256, or the comparator output (The match detection signal) from TMRA0 by setting TA01MOD<TA1CLK1:0>.

For each interval timer the timer operation control register bits TA01RUN<TA0RUN> and TA01RUN<TA1RUN> can be used to stop and clear the up counters and to control their count. A reset clears both up counters, stopping the timers.

www.DataSheet4U.com

(3) Timer registers (TA0REG and TA1REG)

These are 8-bit registers that can be used to set a time interval. When the value set in the timer register TAOREG or TA1REG matches the value in the corresponding up counter, the comparator match detect signal goes Active. If the value set in the timer register is 00H, the signal goes Active when the up counter overflows.

The TAOREG are double buffer structure, each of which makes a pair with register buffer.

The setting of the bit TA01RUN<TA0RDE> determines whether TA0REG's double buffer structure is enabled or disabled. It is disabled if $\langle TA0RDE \rangle = "0"$ and enabled if $\langle TA0RDE \rangle = "1"$.

When the double buffer is enabled, data is transferred from the register buffer to the timer register when a $2^n - 1$ overflow occurs in PWM mode, or at the start of the PPG cycle in PPG mode. Hence the double buffer cannot be used in timer mode.

A Reset initializes <TAORDE> to "0", disabling the double buffer. To use the double buffer, write data to the timer register, set <TAORDE> to "1", and write the following data to the register buffer. Figure 3.8.4 shows the configuration of TAOREG.



Figure 3.8.4 Configuration of Timer Register 0 (TA0REG)

Note: The same memory address is allocated to the timer register and the register buffer when write data to TAOREG. When <TAORDE> = 0, the same value is written to the register buffer and the timer register; when <TAORDE> = 1, only the register buffer is written to.

The address of each timer register is as follows.

TA0REG: 000102H	TA1REG: 000103H
TA2REG: 00010AH	TA3REG: 00010BH
TA4REG: 000112H	TA5REG: 000113H

All these registers are write only and cannot be read.

(4) Comparator (CP0 and CP1)

The comparator compares the value in an up counter with the value set in a timer register. If they match, the up counter is cleared to zero and an interrupt signal (INTTA0 or INTTA1) is generated. If timer flip-flop inversion is enabled, the timer flip-flop is inverted at the same time.

(5) Timer flip-flop (TA1FF)

The timer flip-flop (TA1FF) is a flip-flop inverted by the match detects signal (8-bit comparator output) of each interval timer.

Whether inversion is enabled or disabled is determined by the setting of the bit TA1FFCR<TAFF1IE> in the Timer Flip-Flop Control Register.

A Reset clears the value of TA1FF to "0".

Programming "01" or "10" to TA1FFCR<TAFF1C1:0> sets TA1FF to 1 or 0. Programming "00" to these bits inverts the value of TA1FF (This is known as software inversion).

The TA1FF signal is output via the TA1OUT pin (Concurrent with P71). When this pin is used as the timer output, the timer flip-flop should be set beforehand using the Port 7 relation registers P7CR and P7FC.

3.8.3 SFR

				TMRA0	1 Run Reg	gister					
		7	6	5	4	3	2	1	0		
TA01RUN Bi	Bit symbol	TAORDE				I2TA01	TA01PRUN	TA1RUN	TAORUN		
(0100H)	Read/Write	R/W		/			R/	W			
	After reset	0		/		0	0	0	0		
	Function	Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	8-bit timer r 0: Stop and 1: Run (Cou	imer run/stop control p and clear n (Count up)			
		TAOREG d	ouble buffer o able	control				→ Timer ru	In/stop control Stop and clear		
							1074.04.				

I2TA01: Operation in IDLE2 mode TA01PRUN: Run prescaler TA1RUN: Run TMRA1 TA0RUN: Run TMRA0

Note: The values of bits 4, 5 and 6 of TA01RUN are undefined when read.

TMRA23 Run Register



I2TA23: Operation in IDLE2 mode TA23PRUN: Run prescaler TA3RUN: Run TMRA3 TA2RUN: Run TMRA2

Note: The values of bits 4, 5 and 6 of TA23RUN are undefined when read.

Figure 3.8.5 Register for TMRA

_							JISIEI			
		7		6	5	4	3	2	1	0
TA45RUN	Bit symbol	TA4R	DE		/	/	I2TA45	TA45PRUN	TA5RUN	TA4RUN
(0110H)	Read/Write	R/W	/					R/	W	
	After reset	0				/	0	0	0	0
	Function	Double					IDLE2	8-bit timer r	un/stop cont	rol
		buffer					0: Stop	0: Stop and	0: Stop and clear	
		0: Disable					1: Operate	1: Run (Count up)		
		1: Enal	ole							
		↓								
		TA4RE	G dou	uble buffer o	control				 Count op 	perate
		0 Disable							0 5	Stop and clear
		1	Enab	ble					1 F	Run (Count up
			2.100					1274 45.		

TMRA45 Run Register

I2TA45: Operation in IDLE2 mode TA45PRUN: Run prescaler TA5RUN: Run TMRA5 TA4RUN: Run TMRA4

Note: The values of bits 4, 5 and 6 of TA45RUN are undefined when read.

Figure 3.8.6 Register for TMRA

	7	6	5	4	3	2	1	0
Bit symbol	TA01M1	TA01M0	PWM01	PWM00	TA1CLk	TA1CLK0	TA0CLK	TA0CLK0
Read/Write				R/	W	·		
After reset	0	0	0	0	0	0	0	0
Function	Operation n	node	PWM cycle		Source of	lock for TMRA1	Source clo	ock for TMRA0
	00: 8-bit tim	er mode	00: Reserve	ed	00: TA0	ſRG	00: TA0IN	l pin
	01: 16-bit tir	ner mode	01: 2 ⁶ – 1		01:		01: φT1	
	10: 8-bit PP	G mode	10: 2 [′] – 1		10:	i	10:	
	11: 8-bit PW	/M mode	11: 2 [°] – 1		11:	6	11: φT16	
						•		•
							•	
					TMRA0 s	ource clock sele	ection	
					00	TA0IN (External	input)	
					01 (pT1 (Prescaler)		
					10 0	∮T4 (Prescaler)		
					11 (∮T16 (Prescaler)	
					TMRA1 s	ource clock sele	ection	
					-	ГА01MOD <ta01m1:0> ≠ (</ta01m1:0>	TA 01 <t< td=""><td>01MOD A01M1:0> = 01</td></t<>	01MOD A01M1:0> = 01
					00 00	Comparator output from TMRA	0 Ov 0 TN	erflow output fro
					01 0	∳T1	(16	bit timer mode)
					10 0	φ T 16		
					11 (þT256		
			l		PWM cvc	le selection		
					00	Reserved		
					01 ($(2^6 - 1) \times \text{Clock}$	source	
					10 ($(2^7 - 1) \times \text{Clock}$	source	
					11 ($(2^8-1) \times \text{Clock}$	source	
	Bit symbol Read/Write After reset Function	7 Bit symbol TA01M1 Read/Write 0 After reset 0 Function Operation n 00: 8-bit tim 01: 16-bit tim 10: 8-bit PP 11: 8-bit PW	7 6 Bit symbol TA01M1 TA01M0 Read/Write 0 0 After reset 0 0 Function Operation mode 00: 8-bit timer mode 01: 16-bit timer mode 10: 8-bit PPG mode 11: 8-bit PWM mode	7 6 5 Bit symbol TA01M1 TA01M0 PWM01 Read/Write After reset 0 0 0 Function Operation mode 00: 8-bit timer mode 10: 8-bit PPG mode 11: 8-bit PWM mode PWM cycle 00: Reserve 01: 2 ⁶ - 1 10: 2 ⁷ - 1 11: 2 ⁸ - 1	7 6 5 4 Bit symbol TA01M1 TA01M0 PWM01 PWM00 Read/Write R/ R/ R/ After reset 0 0 0 0 Function Operation mode PWM cycle 00: Reserved 01: 2 ⁶ - 1 10: 8-bit timer mode 10: 2 ⁷ - 1 10: 2 ⁷ - 1 11: 2 ⁸ - 1	7 6 5 4 3 Bit symbol TA01M1 TA01M0 PWM01 PWM00 TA1CLK Read/Write R/W After reset 0 0 0 0 Function Operation mode 00: Reserved 00: TA01 00: TA01 00: TA01 01: 16-bit timer mode 01: 2 ⁶ - 1 01: 471 10: 471 10: 471 10: 8-bit PPG mode 11: 2 ⁸ - 1 10: 4716 11: 4725	7 6 5 4 3 2 Bit symbol TA01M1 TA01M0 PWM01 PWM00 TA1CLK1 TA1CLK0 Read/Write R/W R/W R/W R/W R/W After reset 0 0 0 0 0 0 Function Operation mode 00: 8-bit timer mode 00: 2 ⁶ - 1 00: TA0TRG 01: \$T1 00: TA0TRG 01: 16-bit timer mode 10: 2 ⁶ - 1 10: \$T1 10: \$T16 11: \$T256 11: \$T256 I1: 8-bit PVM mode 11: 2 ⁸ - 1 11: \$T0256 11: \$T00 (External 0) 14 \$T16 (Prescaler) 01 \$T11 10: \$T401N (External 0) 11 \$T41 (Prescaler) 11 \$T41 (Prescaler) 10 \$T41 (Prescaler) 11 \$T16 (Prescaler) 10 \$T41 (Prescaler) 11 \$T16 (Prescaler) 11 \$T16 (Prescaler) 11 10 \$T16 (Prescaler) 11 \$T16 (Prescaler) 11 11 \$T16 (Prescaler) 11 \$T16 (Prescaler) 11	7 6 5 4 3 2 1 Bit symbol TA01M1 TA01M0 PWM01 PWM00 TA1CLK1 TA1CLK0 TA0CLK1 ReadWrite RW ReadWrite RW ReadWrite RW After reset 0 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1<

TMRA01 Mode Register

www.DataSheet4U.com

Figure 3.8.7 Register for TMRA

Two 8-bit timers

8-bit PWM (TMRA0) + 8-bit timer (TMRA1)

16-bit timer

8-bit PPG

00

01

10

11

		7	6	5	4	3	2	1	0
TA23MOD	Bit symbol	TA23M1	TA23M0	PWM21	PWM20	TA3CLK ²	1 TA3CLK0	TA2CLI	<1 TA2CLK0
(010CH)	Read/Write			•	R	/W	21TA3CLK01TA3CLK01TA3CLK01TA3CLK01TA3CLK01TA3CLK01TA3CLK01TMRA3511: ϕ 1 ϕ 00: ϕ 611: ϕ 1 ϕ 0not setT1(Prescaler)T4 (Prescaler)T16 (Prescaler)00TA23M1:0> \neq 01Comparator outputrom TMRA2T1T16T256Is eselectionReserved2 ⁶ - 1) × Clock source2 ⁸ - 1) × Clock source2 ⁸ - 1) × Clock source0000001000 <td>•</td> <td></td>	•	
	After reset	0	0	0	0	0	0	0	0
	Function	Operation n	node	PWM cycle		Source cl	ock for TMRA3	Source c	lock for TMRA2
		00: 8-bit tim	ier mode	00: Reserve	ed	00: TA2T	RG	00: Res	erved
		01: 16-bit ti	mer mode	01: 2 ⁶ – 1		01: φT1		01:	
		10: 8-bit PP	'G mode	10: 2′ – 1		10:		10:	
		11: 8-bit PV	VM mode	11: 2° – 1		11:	5	11: ¢T16	3
				L	•		•		_ _
						THE			
							OUTCE CLOCK SEL	ection	
							T1 (Prescaler)		
						01 φ 10 φ	T/ (Prescaler)		
						10 φ	T16 (Prescaler)	
						Ψ		/	
					\square	TMRA3 sc	urce clock sele	ection	
						T.		T	A23MOD
							IAZSIVII.U> ≠ (DI <	$\frac{1}{1} \frac{1}{2} \frac{1}$
						fr	om TMRA2		MRA2
						01 φ [*]	Т1	(1	16-bit timer mode)
						10 ¢	Г16		
						11 φ ⁻	T256		
					\longrightarrow	PWM cycle	e selection		
						00 R		001150	———————————————————————————————————————
						01 (2	$(2^{7} - 1) \times \text{Clock}$	source	
						10 (2	$(2 - 1) \times \text{Clock}$	source	
						11 (2	(1) × CIOCK	source	
						TMRA23 c	peration mode	e selectior	า
						00 T	wo 8-bit timers		
www.DataSheet4U.com						01 1	6-bit timer		
						10 8-	bit PPG		
						11 8	bit PWM (TMF	RA2) + 8-I	bit timer (TMRA3)

TMRA23 Mode Register

Figure 3.8.8 Register for TMRA

		_		-		-			
		(6	5	4	3	2	1	0
TA45MOD	Bit symbol	TA45M1	TA45M0	PWM41	PWM40	TA5CLK1	TA5CLK0	TA4CLk	TA4CLK0
(0114H)	Read/Write		1		R	/W			
	After reset	0	0	0	0	0	0	0	0
	Function	Operation n	node	PWM cycle		Source cloc	k for TMRA5	Source cle	ock for TMRA4
		00: 8-bit tim	ier mode	00: Reserve	ed	00: TA4TR	G	00: TA4I	N pin
		01: 16-bit ti	mer mode	01: 2 ⁶ –1		01:		01:	
		10: 8-bit PP	'G mode	10: 2 ⁷ –1		10:		10:	
		11: 8-bit PV	VM mode	11: 2 ⁸ –1		11:		11:	
							•		
							•		•
					_	TMRA4 sou	irce clock sele	ection	
						00 TA	4IN (External	l input)	
						01 φT	1 (Prescaler)		
						10 φT	4 (Prescaler)		
						11 φT	16 (Prescaler)	
						TMRA5 sou	irce clock sel	ection	
						TA <t< td=""><td>45MOD A45M1:0> ≠ 0</td><td>01 <</td><td>445MOD FA45M1:0> = 01</td></t<>	45MOD A45M1:0> ≠ 0	01 <	445MOD FA45M1:0> = 01
						00 Co fro	mparator out m TMRA4	put O TI	verflow output fro MRA4
						01 φT	1	(1	6-bit timer mode)
						10 φT	16		
						11 φT	256		
						PWM cycle	selection		
					·	00 Re	served		
						01 (26	– 1) × Clock	source	
						10 (2 ⁷	- 1) × Clock	source	
						11 (2 ⁸	– 1) × Clock	source	
						`			
							peration mode	e selection	l
							a o-un nuers		
ww.DataSheet4U.com						01 16	-hit timer	,	
ww.DataSheet4U.com						01 16	-bit timer	•	

TMRA45 Mode Register

Figure 3.8.9 Register for TMRA



Note: The values of bits 4, 5, 6 and 7 of TA1FFCR are undefined when read.

Figure 3.8.10 Register for TMRA



Note: The values of bits 4, 5, 6 and 7 of TA3FFCR are undefined when read.

Figure 3.8.11 Register for 8-Bit Timer



Note: The values of bits 4, 5, 6 and 7 of TA5FFCR are undefined when read.

Figure 3.8.12 Register for TMRA

3.8.4 Operation in Each Mode

(1) 8-bit timer mode

Both TMRA0 and TMRA1 can be used independently as 8-bit interval timers.

When set function and counter data, be stopped operation of TMRA0 and TMRA1 registers beforehand.

a. Generating interrupts at a fixed interval (using TMRA1)

To generate interrupts at constant intervals using TMRA1 (INTTA1), first stop TMRA1 then set the operation mode, input clock and a cycle to TA01MOD and TA1REG register, respectively. Then, enable the interrupt INTTA1 and start TMRA1 counting.

Example: To generate an INTTA1 interrupt every 20 μs at fc = 16 MHz, set each register as follows:

```
* Clock state
                 System clock: High frequency (fc)
                 Clock gear: 1(fc)
                 Prescaler clock: fFPH
                  MSB
                                              LSB
                  7
                     6 5 4 3 2 1 0
TA01RUN
                     - X X - - 0 -
                                                     Stop TMRA1 and clear it to 0.
TA01MOD
              \leftarrow 0 \quad 0 \quad X \quad X \quad 0 \quad 1 \quad X \quad X
                                                     Select 8-bit timer mode and select \phiT1 (0.5 \mus at fc =
                                                     16 MHz) as the input clock.
TA1REG
                 0 0 1 0 1 0 0 0
                                                     Set TA1REG to 20 \mu s \div \phi T1 = 40 = 28 H
INTETA01
              ← X 1 0 1
                                                     Enable INTTA1 and set it to level 5.
                               _
TA01RUN
              \leftarrow - \ X \ X \ X \ - \ 1 \ 1 \ -
                                                     Start TMRA1 counting.
```

X: Don't care, -: No change

Select the input clock using Table 3.8.2.

Note: The input clocks for TMRA0 and TMRA1 are different from as follows. TMRA0: TA0IN input, φT1, φT4 or φT16 TMRA1: Comparator output from TMRA0, φT1, φT16 and φT256

b. Generating a 50% duty ratio square wave pulse

The state of the timer flip-flop (TA1FF) is inverted at constant intervals and its status output via the timer output pin (TA1OUT).

Example: To output a 3.0 µs square wave pulse from the TA1OUT pin at fc = 16 MHz, use the following procedure to make the appropriate register settings. This example uses TMRA1; however, either TMRA0 or TMRA1 may be used.



Figure 3.8.13 Square Wave Output Timing Chart (50% Duty)

c. Making TMRA1 count up on the match signal from the TMRA0 comparator

Select 8-bit timer mode and set the comparator output from TMRA0 to be the input clock to TMRA1.



Figure 3.8.14 TMRA1 Count Up on Signal from TMRA0

(2) 16-bit timer mode

A 16-bit interval timer is configured by pairing the two 8-bit timers TMRA0 and TMRA1.

To make a 16-bit interval timer in which TMRA0 and TMRA1 are cascaded together, set TA01MOD<TA01M1:0> to "01".

In 16-bit timer mode, the overflow output from TMRA0 is used as the input clock for TMRA1, regardless of the value set in TA01MOD<TA1CLK1:0>. Table 3.8.2 shows the relationship between the timer (Interrupt) cycle and the input clock selection.

Timer interrupt cycle set lower 8 bits to TAOREG and set upper 8 bits to TA1REG. Please keep setting TA0REG first because setting data for TA0REG inhibit its compare function and setting data for TA1REG permit it.

Setting example: To generate an INTTA1 interrupt every 0.5 s at fc = 16 MHz, set the timer registers TA0REG and TA1REG as follows:

* Clock state System clock: High frequency (fc) Clock gear: 1 (fc) Prescaler clock: f_{FPH}

www.DataSheet4U.com

If ϕ T16 (8.0 µs at 16 MHz) is used as the input clock for counting, set the following value in the registers:

 $0.5 \text{ s/8.0 } \mu \text{s} = 62500 = F424\text{H};$

i.e. set TA1REG to F4H and TA0REG to 24H.

The comparator match signal is output from TMRA0 each time the up counter UC0 matches TA0REG, though the up counter UC0 is not be cleared and also INTTA0 is not generated.

In the case of the TMRA1 comparator, the match detect signal is output on each comparator pulse on which the values in the up counter UC1 and TA1REG match. When the match detect signal is output simultaneously from both the comparators TMRA0 and TMRA1, the up counters UC0 and UC1 are cleared to 0 and the interrupt INTTA1 is generated. Also, if inversion is enabled, the value of the timer flip-flop TA1FF is inverted.

Example: When TA1REG = 04H and TA0REG = 80H



Figure 3.8.15 Timer Output by 16-Bit Timer Mode

(3) 8-bit PPG (Programmable pulse generation) output mode

Square wave pulses can be generated at any frequency and duty ratio by TMRA0. The output pulses may be active low or active high. In this mode TMRA1 cannot be used.

TMRA0 outputs pulses on the TA1OUT pin (shared with P71).



Figure 3.8.16 8-Bit PPG Output Waveforms

In this mode, a programmable square wave is generated by inverting the timer output each time the 8-bit up counter (UC0) matches the value in one of the timer registers TAOREG or TA1REG.

The value set in TA0REG must be smaller than the value set in TA1REG.

Although the up counter for TMRA1 (UC1) is not used in this mode, TA01RUN<TA1RUN> should be set to "1", so that UC1 is set for counting.

Figure 3.8.17 shows a block diagram representing this mode.



Figure 3.8.17 Block Diagram of 8-Bit PPG Output Mode

If the TAOREG double buffer is enabled in this mode, the value of the register buffer will be shifted into TAOREG each time TA1REG matches UCO.

Use of the double buffer facilitates the handling of low-duty waves (when duty is varied).



Figure 3.8.18 Operation of Register Buffer



Example: To generate 1/4-duty 50-kHz pulses (at fc = 16 MHz):



Calculate the value that should be set in the timer register.

To obtain a frequency of 50 kHz, the pulse cycle t should be: $t = 1/50 \text{ kHz} = 20 \text{ }\mu\text{s}$

 $\phi T1 = 0.5 \ \mu s$ (at 16 MHz);

 $20 \ \mu s / 0.5 \ \mu s = 40$

Therefore set TA1REG to 40 (28H)

The duty is to be set to 1/4: $t\times 1/4$ = 20 $\mu s\times 1/4$ = 5 μs

```
5\ \mu s/0.5\ \mu s=10
```

Therefore, set TAOREG = 10 = 0AH.

	MSB	LSB
	7 6 5 4 3 2 7	1 0
TA01RUN	$\leftarrow 0 X X X - 0 0$	0 0 Stop TMRA0 and TMRA1 and clear it to "0".
TA01MOD	\leftarrow 1 0 X X X X (0 1 Set the 8-bit PPG mode, and select ϕ T1 as input clock.
TA0REG	$\leftarrow 0 0 0 0 1 0 2$	1 0 Write 0AH
TA1REG	$\leftarrow 0 0 1 0 1 0 0$	0 0 Write 28H
TA1FFCR	$\leftarrow X X X X 0 1$	1 X Set TA1FF, enabling inversion.
		→ Writing "10" provides negative logic pulse.
P7CR	$\leftarrow X X X '$	1 - 1 Sot D71 on the TA10UIT pin
P7FC	$\leftarrow X X X - X - Y$	1 X J
TA01RUN	← 1 X X X - 1 *	1 1 Start TMRA0 and TMRA1 counting and enable double buffer.

X: Don't care, -: No change
(4) 8-bit PWM output mode

This mode is only valid for TMRA0. In this mode, a PWM pulse with the maximum resolution of 8 bits can be output.

When TMRA0 is used the PWM pulse is output on the TA1OUT pin (which is also used as P71). TMRA1 can also be used as an 8-bit timer.

The timer output is inverted when the up counter (UC0) matches the value set in the timer register TA0REG or when $2^n - 1$ counter overflow occurs (n = 6, 7 or 8 as specified by TA01MOD<PWM01:00>). The up counter UC0 is cleared when $2^n - 1$ counter overflow occurs.

The following conditions must be satisfied before this PWM mode can be used.

Value set in TAOREG < Value set for $2^{n} - 1$ counter overflow



Figure 3.8.19 8-Bit PWM Output Wave Form

Figure 3.8.20 shows a block diagram representing this mode.



Figure 3.8.20 Block Diagram of 8-Bit PWM Output Mode

In this mode, the value of the register buffer will be shifted into TA0REG if $2^n - 1$ overflow is detected when the TA0REG double buffer is enabled.

Use of the double buffer facilitates the handling of low duty ratio waves.



Figure 3.8.21 Operation of Register Buffer

Example: To output the following PWM waves on the TA1OUT pin at fc = 16 MHz:



* Clock state System clock: High frequency (fc) Clock gear: 1 (fc) Prescaler clock: f_{FPH}

To achieve a 63.5- μ s PWM cycle by setting ϕ T1 to 0.5 μ s (at fc = 16 MHz):

 $63.55 \ \mu s/0.5 \ \mu s = 127 = 2^n - 1$

Therefore n should be set to 7.

Since the low-level period is 36.0 μ s when ϕ T1 = 0.5 μ s,

set the following value for TAOREG:

 $36.0 \ \mu s/0.5 \ \mu s = 72 = 48 H$

www.DataSheet4U.com

	MS	В						L	SB	
		7	6	5	4	3	2	1	0	
TA01RUN	←	_	Х	Х	Х	_	_	_	0	Stop TMRA0 and clear it to 0.
TA01MOD	←	1	1	1	0	-	-	0	1	Select 8-bit PWM mode (cycle: $2^7 - 1$) and select $\phi T1$ as the input clock.
TA0REG	←	0	1	0	0	1	0	0	0	Write 48H.
TA1FFCR	←	Х	Х	Х	Х	1	0	1	Х	Clear TA1FF to 0; enable the inversion.
P7CR P7FC TA01RUN	$\begin{array}{c} \downarrow \\ \downarrow \\ \downarrow \end{array}$	X X 1	X X X	X X X	- - X	- X -	- - 1	1 1 -	- X } 1	Set P71 to function as the TA1OUT pin. Start TMRA0 counting, and enable the double buffer.

X: Don't care, -: No change

	@ fc = 16 MHz, fs = 32.768 kHz													
Select System	Select Prescaler	Gear Value	PWM Cycle											
Clock <sysck></sysck>	Clock			2 ⁶ – 1			2 ⁷ – 1		2 ⁸ – 1					
	<prck1:0></prck1:0>	<gear2:0></gear2:0>	φT1	φT4	φT16	φT1	φT4	φT16	φT1	φT4	φT16			
1 (fs)		XXX	15.4 ms	61.5 ms	246 ms	31.0 ms	124 ms	496 ms	62.3 ms	249 ms	996 ms			
		000 (fc)	31.5 μs	126 μs	504 μs	63.5 μs	254 μs	1016 µs	127.5 μs	510 μs	2040 μs			
	00 (f _{FPH})	001 (fc/2)	63.0 μs	252 μs	1008 μs	127 μs	508 μs	2032 μs	255 μs	1020 μs	4080 µs			
		010 (fc/4)	126 μs	504 μs	2016 µs	254 μs	1016 μs	4064 μs	510 μs	2040 μs	8160 μs			
0 (fc)		011 (fc/8)	252 μs	1008 μs	4032 μs	508 μs	2032 μs	8128 μs	1020 μs	4080 μs	16.32 ms			
		100 (fc/16)	504 μs	2016 µs	8064 μs	1016 μs	4064 μs	16.256 ms	2040 μs	8160 μs	32.64 ms			
	10 (fc/16 clock)	XXX	504 μs	2016 μs	8064 μs	1016 μs	4064 μs	16.256 ms	2040 μs	8160 μs	32.64 ms			

Table 3.8.3 PWM Cycle

XXX: Don't care

(5) Settings for each mode

Table 3.8.4 shows the SFR settings for each mode.

	Register Name			TA1FFCR			
	<bit symbol=""></bit>	<ta01m1:0></ta01m1:0>	<pwm01:00></pwm01:00>	<ta1clk1:0></ta1clk1:0>	<ta0clk1:0></ta0clk1:0>	<taff1is></taff1is>	
	Function	Timer Mode	PWM Cycle	Upper Timer Input Clock	Lower Timer Input Clock	Timer F/F Invert Signal Select	
	8-bit timer × 2 channels	00	_	Lower timer match, φT1, φT16, φT256 (00, 01, 10, 11)	External clock, φT1, φT4, φT16 (00, 01, 10, 11)	0: Lower timer output 1: Upper timer output	
	16-bit timer mode	01	-	_	External clock, φT1, φT4, φT16 (00, 01, 10, 11)	_	
	8-bit PPG × 1 channel	10	_	_	External clock, φT1, φT4, φT16 (00, 01, 10, 11)	_	
www.DataSheet	8-bit PWM × 1 channel 4U.com	11	$2^{6} - 1, 2^{7} - 1, 2^{8} - 1$ (01, 10, 11)	_	External clock, φT1, φT4, φT16 (00, 01, 10, 11)	_	
	8-bit timer × 1 channel	11	_	φT1, φT16, φT256 (01, 10, 11)	_	Output disabled	

Table 3.8.4	Timer Mode	Setting	Registers

-: Don't care

3.9 16-Bit Timer/Event Counters (TMRB)

The TMP91CP27 contains one multifunctional 16-bit timer/event counter (TMRB0) which have the following operation modes:

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable square wave pulse generation output mode (PPG: Variable duty cycle with variable period)

Can be used following operation modes by capture function:

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

Figure 3.9.1 show block diagram of TMRB0. Timer/event counter consists of a 16-bit up counter, two 16-bit timer registers (One of them with a double-buffer structure), two 16-bit capture registers, two comparators, a capture input controller, a timer flip-flop and a control circuit.

Timer/Event counter is controlled by 11-byte control register (SFR).

Spec	Channel	TMRB0				
External pin	External clock/ capture trigger input pin	TB0IN0 (Shared with P80) TB0IN1 (Shared with P81)				
	Timer flip-flop output pin	TB0OUT0 (Shared with P82)				
		TB0OUT1 (Shared with P83)				
	Timer RUN register	TB0RUN (0180H)				
	Timer mode register	TB0MOD (0182H)				
	Timer flip-flop control register	TB0FFCR (0183H)				
		TB0RG0L (0188H)				
055	Timor registor	TB0RG0H (0189H)				
SFR name		TB0RG1L (018AH)				
(//dd/000)		TB0RG1H (018BH)				
		TB0CP0L (018CH)				
	Conturo register	TB0CP0H (018DH)				
	Capture register	TB0CP1L (018EH)				
		TB0CP1H (018FH)				

Table 3.9.1 Registers and Pins for TMRB0

3.9.1 Block Diagram of TMRB0



Figure 3.9.1 Block Diagram of TMRB0

3.9.2 Operation of Each Circuit

(1) Prescaler

The 5-bit prescaler generates the source clock for TMRB0. The prescaler clock (ϕ T0) is divided clock (divided by 4) from selected clock by the register SYSCR0<PRCK1:0> of clock gear.

This prescaler can be started or stopped using TBORUN<TBOPRUN>. Counting starts when <TBOPRUN> is set to 1; the prescaler is cleared to zero and stops operation when <TBOPRUN> is cleared to 0.

Table 3.9.2 show prescaler output clock resolution.

@fc = 16 MHz, fs	= 32.768 KHz									
System Clock	Prescaler Clock	Clock Gear	Prescaler Output Clock Resolution							
<sysck></sysck>	<prck1:0></prck1:0>	<gear2:0></gear2:0>	φ T 1	φ T 4	φT16					
1 (fs)		XXX	fs/2 ³ (244 μs)	fs/2 ⁵ (977 μs)	fs/27 (3.9 ms)					
		000 (fc)	fc/2 ³ (0.5 μs)	fc/2 ⁵ (2.0 μs)	fc/2 ⁷ (8.0 μs)					
	00	001 (fc/2)	fc/2 ⁴ (1.0 μs)	fc/2 ⁶ (4.0 μs)	fc/2 ⁸ (16.0 μs)					
	(f _{FPH})	010 (fc/4)	fc/2 ⁵ (2.0 μs)	fc/2 ⁷ (8.0 μs)	fc/2 ⁹ (32.0 μs)					
0 (fc)		011 (fc/8)	fc/2 ⁶ (4.0 μs)	fc/2 ⁸ (16.0 μs)	fc/2 ¹⁰ (64.0 μs)					
		100 (fc/16)	fc/2 ⁷ (8.0 μs)	fc/2 ⁹ (32.0 μs)	fc/2 ¹¹ (128 μs)					
	10 (Note) (fc/16 clock)	XXX	fc/2 ⁷ (8.0 μs)	fc/2 ⁹ (32.0 μs)	fc/2 ¹¹ (128 μs)					

XXX: Don't care

(2) Up counter (UC0)

UC0 is a 16-bit binary counter which counts up according to input from the clock specified by TB0MOD<TB0CLK1:0> register.

As the input clock, one of the prescaler internal clocks $\phi T1$, $\phi T4$ and $\phi T16$ or an external clock from TB0IN0 pin can be selected. Counting or stopping and clearing of the counter is controlled by timer operation control register TB0RUN<TB0RUN>.

When clearing is enabled, the up counter UC0 will be cleared to zero each time its value matches the value in the timer register TB0RG1H/L. Clearing can be enabled or disabled using TB0MOD<TB0CLE>.

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

A timer overflow interrupt (INTTBOF0) is generated when UC0 overflow occurs.

(3) Timer registers (TB0RG0H/L and TB0RG1H/L)

These two 16-bit registers are used to set the interval time. When the value in the up counter UC0 matches set value of timer register, the comparator match detect signal will be active.

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

The TB0RG0 timer register has a double-buffer structure, which is paired with register buffer 0. The timer control register TB0RUN<TB0RDE> control whether the double buffer structure should be enabled or disabled: it is disabled when $\langle TB0RDE \rangle = 0$, and enabled when $\langle TB0RDE \rangle = 1$.

When the double buffer is enabled, data is transferred from the register buffer to the timer register when the values in the up counter (UC0) and the timer register TB0RG1 match.

After a Reset, TB0RG0 and TB0RG1 are undefined. To use the 16-bit timer after reset, data should be written beforehand.

When reset, <TB0RDE> is initialized to 0, whereby the double buffer is disabled. To use the double buffer, write data to the timer register, set <TB0RDE> to 1, then write following data to the register buffer.

TB0RG0 and the register buffer are allocated to the same memory address 0188H/0189H. When $\langle TB0RDE \rangle = 0$, same value will be written to both the timer register and register buffer. When $\langle TB0RDE \rangle = 1$, the value is written into only the register buffer.

Therefore, when write initial value to timer register, set register buffer to disable.

The addresses of the timer registers are as follows:



www.DataSheet4U.com

The TB0RG0 to TB0RG1 are write-only registers and thus cannot be read.

(4) Capture registers (TB0CP0H/L, TB0CP1H/L)

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

Data in the capture register should be read all 16 bits. For example, using 2-byte data load instruction or using 1-byte date load instruction twice for lower 8 bits and upper 8 bits in order.

The addresses of the capture registers are as follows:



The TB0CP0 to TB0CP1 are read-only registers and thus cannot be read.

(5) Capture, external interrupt control

This circuit controls the timing to latch the value of up counter UC0 into TB0CP0, TB0CP1 and control generation of external interrupt. The latch timing of capture register and selection of edge for external interrupt is set in TB0M0D<TB0CPM1:0>.

The edge of external interrupt INT6 is fixed to rising edge.

Besides, the value of up counter can be loaded into a capture registers by software. Whenever 0 is written to TB0MOD<TB0CP0I>, the current value in the up counter is loaded into capture register TB0CP0. It is necessary to keep the prescaler in run mode (e.g., TB0RUN<TB0PRUN> must be held at a value of 1).

Note: As described above, whenever 0 is written to TB0MOD<TB0CP0I>, the current value in the up counter is loaded into capture register TB0CP0. However, note that the current value in the up counter is also loaded into capture register TB0CP0 when 1 is written to TB0MOD<TB0CP0I> while this bit is holding 0.



(6) Comparators (CP0 and CP1)

CP0 and CP1 are 16-bit comparators which compare the value in the up counter UC0 value with the value set of TB0RG0 or TB0RG1 respectively, in order to detect a match. If a match is detected, the comparators generate an interrupt (INTTB00 or INTTB01 respectively).

(7) Timer flip-flops (TB0FF0 and TB0FF1)

These flip-flops are inverted by the match detect signals from the comparators and the latch signals to the capture registers. Inversion can be enabled and disabled for each element using TB0FFCR<TB0C1T1, TB0C0T1, TB0E1T1, TB0E0T1>.

After a reset, the value of TB0FF0 and TB0FF1 is undefined. If "00" is written to TB0FFCR<TB0FF0C1:0> or <TB0FF1C1:0>, TB0FF0 or TB0FF1 will be inverted. If "01" is written to the flip-flops control registers, the value of TB0FF0 and TB0FF1 will be set to "1". If "10" is written to the flip-flops control registers, the value of TB0FF0 and TB0FF0 and TB0FF1 will be cleared to "0".

The values of TB0FF0 and TB0FF1 can be output to the timer output pins TB0OUT0 (which is shared with P82), TB0OUT1 (which is shared with P83). Timer output should be specified by using the port 8 function register P8FC and port 8 control register P8CR.

3.9.3 SFR

TBORUN	
(0180H)	

			TMRB0 F	RUN Regist	ter				
/	7	6	5	4	3	2	1	0	
Bit symbol	TB0RDE	-			I2TB0	TB0PRUN		TB0RUN	
Read/Write	R/W	R/W			R/W	R/W		R/W	
After reset	0	0			0	0		0	
Function	Double buffer 0: Disable	Write "0".			IDLE2 0: Stop 1: Operation	16-bit timer run/stop control 0: Stop and clear 1: Run (Count up)			

Count operation

0	Stop and clear
1	Count

I2TB0:Operation of IDLE2 modeTB0PRUN:Operation of prescalerTB0RUN:Operation of TMRB0

Note: The values of bits 1, 4 and 5 of TB0RUN are undefined when read.

Figure 3.9.2 Register for TMRB

www.DataSheet4U.com

91CP27-116

	7	6		5	4	3	2		1	0
Bit symbol	TB0CT1	TB0ET1	TB0	CP0I	TB0CPM1	TB0CPM0	TB0CL	E	TB0CLK1	TB0CLK0
Read/Write	R/W		V	/*		R/W	/			
After reset	0	0		1	0	0	0		0	0
Function	TB0FF1 Inv trigger 0: Trigger d 1: Trigger e	vertion lisable mable	Softv captu contr	vare ure ol	Capture tin 00: Disable INT5 is r 01: TB0IN0	ning ising edge ↑ TB0IN1 ↑	Up cour control 0: Clear disab	nter 7 s le 0	TMRB0 sou select 00: TB0IN0 01:	urce clock) pin input
	Invert	Invert			INT5 is r	ising edge	1: Clear	· 1	Ι0: φT4	
	when	when	0: Sof	tware	10: TB0IN0	10: TB0IN0 ↑ TB0IN0 ↓			Ι1:	
	capture to	match	cap 1: Uno	ture defined	INT5 is f	alling edge ⊤↑				
	register 1	timer			TA10UT	⊤↓				
	0	register 1			INT5 is r	ising edge				
		[]			
					→ TMRB0 s	source closk				
					00 E	External input	clock (T	BOINO	pin input)	
					01 ¢	σT1			,	
					10 ¢	oT4				
					11 ¢	T16				
					→ Clear of	up counter 0	(UC0)			
					0 [Disable clear o	of up cou	unter		
					1 (Clear by matc	h with Tl	B0RG1	1	
			L		Capture/	Interrupt timir	ng			
						Capture	e control		11	NT5 control
					00 0	Capture disab	le		INT5 gene	rate by rising T
					01 T	B0CP0 by ris	sing TB0	IN0		
					1	B0CP1 by ris	sing TB0	IN1		
					10 1	BOCP0 by ris	Sing TBO		INT5 gene	rate by falling
						BUCPI by la	lling ты	JIINU		
					11 1					
					11 T	B0CP0 by ris	sing TA1 Iling TA1		INT5 gene	rate by rising T
					11 T	B0CP0 by ris	sing TA1 Iling TA1	OUT IOUT	INT5 gene	rate by rising T
				L,	11 1	BOCP0 by ris BOCP1 by fa capture Capture value	lling TA1		INT5 gene	rate by rising T
	Bit symbol Read/Write After reset Function	7 Bit symbol TB0CT1 Read/Write R/ After reset 0 Function TB0FF1 Invest Urigger 0: Trigger de 1: Trigger e Invert when capture to capture register 1	7 6 Bit symbol TB0CT1 TB0ET1 Read/Write R/W After reset 0 0 Function TB0FF1 Invertion trigger 0: Trigger disable 1: Trigger enable Invert Invert Invert when when capture to match capture UC0 with register 1 timer	7 6 Bit symbol TB0CT1 TB0ET1 TB0 Read/Write R/W W After reset 0 0 Function TB0FF1 Invertion Software 0: Trigger disable 1: Trigger enable Invert Invert Invert when 0: Software capture to match capture capture UC0 with 1: Und register 1 timer register 1	7 6 5 Bit symbol TBOCT1 TBOET1 TBOCPOI Read/Write R/W W* After reset 0 0 1 Function TBOFF1 Invertion Software 0: Trigger enable control 1: Trigger enable Invert Invert Invert when capture 020 with 1: capture UC0 with 1: register 1 timer register 1	7 6 5 4 Bit symbol TB0CT1 TB0ET1 TB0CP0I TB0CPM1 Read/Write R/W W* After reset 0 0 1 0 Function TB0FF1 Invertion trigger Software capture Capture tir control 0: Disable 0: Trigger disable 1: Trigger enable INT5 is r 01: TB0IN0 Invert Invert Invert Invert INT5 is r when when capture 0: Software 10: TB0IN0 INT5 is r Capture UC0 with 1: Undefined 11: TA10UT register 1 timer register 1 INT5 is r 00 E 00 E 01 d 10 d 11 TA10UT INT5 is r	7 6 5 4 3 Bit symbol TB0CT1 TB0CT1 TB0CP0I TB0CPMI TB0CPM0 Read/Write R/W W* After reset 0 0 1 0 0 Function TB0FF1 Invertion Software Capture timing 00: Disable 01: Tb0IN0 ↑ 1NT5 is rising edge 1: Trigger enable Invert Invert Invert INT5 is rising edge 01: TB0IN0 ↑ TB0IN0 ↓ Invert Invert Invert Invert 10: TB0IN0 ↑ TB0IN0 ↓ Invert Invert Invert 10: TB0IN0 ↑ 10: TB0IN0 ↓ INT5 is rising edge 10: TB0IN0 ↑ 10: TB0IN0 ↓ 10: TB0IN0 ↓ INT5 is rising edge 11: Undefined 11: TA10UT ↓ 1NT5 is rising edge 10: TMRB0 source closk 00 External input 01 \$T1 10: \$\phiT1 10 \$\phiT4 11 \$\phiT16 Capture/Interrupt timir Capture/Interrupt timir Capture/Interrupt timir 00: Capture disab 01 TB0CP0 by ris	7 6 5 4 3 2 Bit symbol TB0CT1 TB0ET1 TB0CP01 TB0CPM1 TB0CPM0 TB0CPM0 Read/Write R/W W* R/M After reset 0 0 1 0 0 0 Function TB0FF1 Invertion Software Capture timing Up cour 0: Trigger disable 0: Trigger enable INT5 is rising edge 0: Clear 1: Trigger enable Invert Invert Invert 0: Software capture to capture 0: Software 0: Software 0: TB0IN0 ↑ TB0IN1 ↑ INT5 is rising edge 1: Clear 1: TA1OUT ↑ enab register 1 timer 1: Undefined 1: TA1OUT ↓ INT5 is rising edge 1: TMRB0 source closk 00 External input clock (T 01< φT4	7 6 5 4 3 2 Bit symbol TBOCT1 TBOET1 TBOCP01 TBOCPM0 TBOCLE Read/Write R/W W* R/W After reset 0 0 1 0 0 0 Function TBOFF1 Invertion trigger Software 0: Trigger disable Capture timing 0: Disable Up counter control Capture timing 0: Disable Up counter control Software 0: Clear Capture timing 0: Disable 0: Clear 0: Clear Invert Invert Invert or Software capture 0: Software capture INT5 is rising edge 10: TBOIN0 ↑ TBOIN0 ↓ 1: Clear 1: Clear Invert When when 0: Software capture INT5 is rising edge 1: Clear 1: Clear Invert Invert UCO with register 1 1: Undefined INT5 is rising edge 1: Clear 1: Clear INT5 is rising edge INT6 is rising edge III TAIOUT ↓ INT6 is rising edge 1: Clear 1: Clear INT6 is rising edge III of TA IIII of TA IIII of TA 1: Clear 1: Clear INT5 is rising edge IIII of TA IIIII of TA IIIII of TA 1: Or Clear 1: Clear IIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIIII	7 6 5 4 3 2 1 Bit symbol TB0CT1 TB0ET1 TB0CP01 TB0CPM1 TB0CPM0 TB0CLE TB0CLK1 Read/Write R/W W* R/W R/W After reset 0 0 1 0 0 0 0 Function TB0FF1 Invertion trigger enable Software capture control Capture timing control Up counter ocntrol Select 00: TB0IN0 1 1: Clear 0: Clear 0: TB0IN0 0: \$elect 0: TB0IN0 1: \$enable 0: TB0IN0 1: \$enable 0: TB0IN0 1: \$enable 0: \$elect 0: \$elect 0: TB0IN0 1: \$enable 0: \$elect 0: TB0IN0 1: \$enable 0: \$elect 0: \$elect

TMRB0 Mode Register

Note: Whenever programming "0" to TB0MOD<TB0CP0I> bit, present value of up counter is received to capture register TB0CP0. But, write "1" to TB0MOD<TB0CP0I> in condition of written "0" to TB0MOD<TB0CP0I> bit, present value of up counter is received to capture register TB0CP0. Therefore you must to regard.

Figure 3.9.3 Register for TMRB

		7	6		5		4	4	3	3		2	1	0			
TB0FFCR	Bit symbol	TB0FF1C1	TB0FF1C0	Т	B0C1T1		TB00	C0T1	TB0E	1T1	TB0	E0T1	TB0FF0C1	TB0FF0C0			
(0183H)	Read/Write	W	/*			•		R/W				W*					
	After reset	1	1		0		(0	C)	0		1	1			
Read-Modify -write instruction is prohibited	Function	TB0FF1 cor 00: Invert 01: Set 10: Clear	TB0FF1 control 00: Invert 01: Set 10: Clear			OFF1 control TE Invert 0: Set 1:			TB0FF0 inversior 0: Trigger disable 1: Trigger enable			r				TB0FF0 Co 00: Invert 01: Set 10: Clear	ntrol
		11: Don't ca	are		ert wher UC	n I	Invert the UC	when C	Invert when the UC		Invert the U	when C	11: Don't ca	re			
		* Always read as "11".			value is value is value is value is value is value is value in to lo		value i loadeo TB0CI	is d in to P0.	matches with TB0RG1.		match with TB0RG0.		* Always read as "11".				
-]									
								Timer	Flip-Fl	op (Tf	30FF0) contr	ol				
							ſ	00	Inve	rt to T	BOFFC) (Softv	vare inversio	n).			
								01	Set ⁻	TB0FF	⁻ 0 to "	1".					
								10	Clea	r TB0	FF0 to	"0".					
								11	Don'	t care							
						L	\rightarrow	Invers TB0R	sion trig G0	ger of	TB0F	F0 wh	en the UC m	atch with			
								0	Trigg	ger dis	able (Disable	e inversion)				
								1	Trigg	ger en	able (E	Enable	inversion)				
							→ _	Invers TB0R	sion trig G1	ger of	TB0F	F0 wh	en the UC m	atch with			
							_	0	Trigg	ger dis	able (Disable	e inversion)				
								1	Trigg	ger en	able (E	Enable	inversion)				
							\rightarrow	Invers loade	sion trig d in to	ger of TB0Cl	TB0F P0	F0 Wh	ien the UC va	alue is			
								0	Trigg	ger dis	able (Disable	e inversion)				
								1	Trigg	ger en	able (E	Enable	inversion)				
				L			\rightarrow	Invers loade	sion trig d in to	ger of TB0Cl	TB0F P1	F0 Wh	ien the UC va	alue is			
www.DataSheet4U.com								0	Trigg	ger dis	able (Disable	e inversion)				
								1	Trigg	ger en	able (B	Enable	inversion)				

TMRB0 Flip-Flop Control Register

Figure 3.9.4 Register for TMRB

3.9.4 Operation in Each Mode

(1) 16-bit interval timer mode

Generating interrupts at fixed intervals In this example,the interval time is set the timer register TB0RG1 to generate the interrupt INTTB01.

		7	6	5	4	3	2	1	0	
-		,	0		т 	0	~		0	
IBORUN	\leftarrow	0	0	Х	Х	-	0	Х	0	Stop IMRB0.
INTETB0	←	Х	1	0	0	Х	0	0	0	Enable INTTB01 and set interrupt level 4. Disable INTTB00.
TB0FFCR	\leftarrow	1	1	0	0	0	0	1	1	Disable the trigger.
TB0MOD	\leftarrow	0	0	1	0	0	1	*	*	Select source clock and
						(** =	= 01,	10,	11)	Disable the capture function.
TB0RG1	\leftarrow	*	*	*	*	*	*	*	*	Set the interval time (16 bits).
		*	*	*	*	*	*	*	*	
TB0RUN	←	0	0	Х	Х	-	1	Х	1	Start TMRB0.

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 (TB0IN0 pin input) as the input clock.

Up counter counting up by rising edge of TB0IN0 pin input. And execution software capture and reading capture value enable reading count value.

	TBORUN P8CR	← ↓	7 0 X	6 0 X	5 X X	4 X X	3 - -	2 0 -	1 X -	0 0 0	J	Stop TMRB0.
	P8FC	←	х	х	х	х	_	_	_	1	}	Set P80 to TB0IN0 input mode.
	INTETB0	←	Х	1	0	0	Х	0	0	0		Enable INTTB01 and set interrupt level 4. Disable INTTB00.
www.DataSheet4U.com	TB0FFCR	←	1	1	0	0	0	0	1	1		Disable trigger.
	TB0MOD	←	0	0	1	0	0	1	0	0		Set input clock to TB0IN0 pin input.
	TB0RG1	\leftarrow	*	*	*	*	*	*	*	*		Set number of count. (16 bits)
			*	*	*	*	*	*	*	*		
	TB0RUN	←	0	0	Х	Х	-	1	Х	1		Start TMRB0.

X: Don't care, -: No change

When used as an event counter, set the prescaler to "RUN". (TB0RUN<TB0PRUN> = "1") (3) 16-bit programmable pulse generation (PPG) output mode

Square wave pulses can be generated at any frequency and duty ratio. The output pulse may be either low active or high active.

The PPG mode is obtained by inversion of the timer flip-flop TB0FF0 that is to be enabled by the match of the up counter UC0 with timer register TB0RG0 or TB0RG1 and to be output to TB0OUT0. In this mode, the following conditions must be satisfied.

(Set value of TB0RG0) < (Set value of TB0RG1)



Figure 3.9.5 Programmable Pulse Generation (PPG) Output Waveforms

When the TB0RG0 double buffer is enabled in this mode, the value of register buffer 0 will be shifted into TB0RG0 at match with TB0RG1. This feature makes easy the handling of low-duty waves.



Figure 3.9.6 Operation of Register Buffer

The following block diagram illustrates this mode.





The following example shows how to set 16-bit PPG output mode:

_		7	6	5	4	3	2	1	0	
TBORUN	←	0	0	Х	Х	-	0	Х	0	Disable the TB0RG0 double buffer and stop TMRB0.
TB0RG0	←	*	*	*	*	*	*	*	*	Set the duty ratio (16 bits).
TB0RG1	←	*	*	*	*	*	*	*	*	Set the frequency (16 bits).
TBORUN	←	1	0	Х	х	-	0	Х	0	Enable the TB0RG0 double buffer. (The duty and frequency are changed on an INTTB01 interrupt.)
TB0FFCR	←	Х	Х	0	0	1	1	1	0	Set the mode to invert TB0FF0 at the match with TB0RG0/TB0RG1. Clear TB0FF0 to 0.
TB0MOD	←	0	0	1	0	0 (** =	1 = 01,	* 10,	* 11)	Select the source clock and disable the capture function
P8CR P8FC TB0RUN	$\begin{array}{c} \leftarrow \\ \leftarrow \\ \leftarrow \end{array}$	X X 1	X X 0	X X X	X X X	- - -	1 1 1	- - X	- - 1	<pre>Set P82 to function as TB0OUT0. Start TMRB0.</pre>

X: Don't care, -: No change

la ahanna

(4) Capture function examples

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

- a. One-shot pulse output from external trigger pulse
- b. For frequency measurement
- c. For pulse width measurement
- d. For time difference measurement
- a. One-shot pulse output from external trigger pulse

Set the up counter UC0 in free-running mode with the internal input clock, input the external trigger pulse from TB0IN0 pin, and load the value of up-counter into capture register TB0CP0 at the rise edge of the TB0IN0 pin.

When the interrupt INT5 is generated at the rise edge of TB0IN0 input, set the TB0CP0 value (c) plus a delay time (d) to TB0RG0 (= c + d), and set the above set value (c + d) plus a one-shot width (p) to TB0RG1 (= c + d + p). And, set "11" to timer flip-flop control register TB0FFCR<TB0E1T1, TB0E0T1>. Set to trigger enable for be inverted timer flip-flop TB0FF0 by UC0 matching with TB0RG0 and with TB0RG1. When interrupt INTTB01 occurs, this inversion will be disabled after one-shot pulse is output.

The (c), (d) and (p) correspond to c, d and p figure 3.9.8.



Figure 3.9.8 One-shot Pulse Output (with delay)



						* Cl	ock	state)		
						ſ	Syst	em d	clock	: High f	requency (fc)
							Cloc	k ge	ar: 1	(fc)	
Main potting						Ĺ	Pres	cale	r cloo	ck: f _{FPH}	1
Main setting							_			>	Set free running
										\rightarrow	Count using ϕ T1
Гтвомор	←	х	х	1	0	1	0	0	1	,	
			,,	•	Ļ	ب					Load the up counter value into TB0CP0 at the rising edge of TB0IN0 pin input.
TB0FFCR	\leftarrow	Х	Х	0	0	0	0	1	0		
				L							Clear TB0FF0 to 0.
											Disable inversion of TB0FF0.
P8CR	\leftarrow	Х	Х	Х	Х	-	1	-	0	}	Set P82 to function as the TB0OUT0 pin.
P8FC	←	Х	Х	Х	Х	-	1	-	1	J	Set P80 to TB0IN0 input mode.
INTE56	←	Х	_	_	_	Х	1	0	0	}	Enable INT5. Disable INTTB00 and INTTB01.
INTETB0	\leftarrow	Х	0	0	0	Х	0	0	0	J	
	←		0	Х	Х	-	1	Х	1		Start TMRB0.
X: Don't care	e, –:	No c	chan	ge							
Setting in INT	Г5										
Г 					_						
TBORGO	←	TB	0CP	0+3	3 ms	s/φT1					
TB0RG1	~	IB	ORG	i0 + 2	2 ms	5/φ I ΄	1				
TBOFFCR	\leftarrow	Х	Х	-	-	1	1 لــــ	-	-		
										\rightarrow	Enable Inversion of TBOFF0 when the up counter value
	,	v	1	0	0	v					
X: Don't car	<u></u>	No	ı char		0	~	_	_	_		
A. Don't car	с, –.	NO C	Jiai	iye							
Setting in INT	гтво	1									
Гтвоеесе	,	v	v			0	0				
	 	~	~	_		0 L		_	_		Disable inversion of TROFFO when the up counter value
							·				match with value of TB0RG0 or TB0RG1
	←	х	0	0	0	х	_	_	_		Disable INTTB01
		~	0	U	U	~					
X: Don't care	e, -:	No c	han	ge							

When delay time is unnecessary, invert timer flip-flop TB0FF0 when up-counter value is loaded into capture register (TB0CP0), and set the TB0CP0 value (c) plus the one-shot pulse width (p) to TB0RG1 when the interrupt INT5 occurs. The TB0FF0 inversion should be enable when the up counter (UC0) value matches TB0RG1, and disabled when generating the interrupt INTTB01.



Figure 3.9.9 One-shot Pulse Output of External Trigger Pulse (without delay)

b. Frequency measurement

The frequency of the external clock can be measured in this mode. The clock is input through the TB0IN0 pin, and its frequency is measured by the 8-bit timers TMRA01 and the 16-bit timer/event counter (TMRB0). (TMRA01 is used to setting of measurement time by inversion TA1FF.)

The TB0IN0 pin input should be for the input clock of TMRB0. Set to TB0MOD <TB0CPM1:0> = "11". The value of the up counter (UC0) is loaded into the capture register TB0CP0 at the rise edge of the timer flip-flop TA1FF of 8-bit timers (TMRA01), and into TB0CP1 at its fall edge.

The frequency is calculated by difference between the loaded values in TB0CP0 and TB0CP1 when the interrupt (INTTA0 or INTTA1) is generates by either 8-bit timer.



Figure 3.9.10 Frequency Measurement

For example, if the value for the level 1 width of TA1FF of the 8-bit timer is set to 0.5 s and the difference between the values in TB0CP0 and TB0CP1 is 100, the frequency is $100 \div 0.5 \text{ s} = 200 \text{ Hz}$.

c. Pulse width measurement

This mode allows to measure the high-level width of an external pulse. While keeping the 16-bit timer/event counter counting (Free running) with the internal clock input, external pulse is input through the TB0IN0 pin. Then the capture function is used to load the UC0 values into TB0CP0 and TB0CP1 at the rising edge and falling edge of the external trigger pulse respectively. The interrupt INT5 occurs at the falling edge of TB0IN0.

The pulse width is obtained from the difference between the values of TB0CP0 and TB0CP1 and the internal clock cycle.

For example, if the internal clock is 0.8 μ s and the difference between TB0CP0 and TB0CP1 is 100, the pulse width will be $100 \times 0.8 \ \mu$ s = 80 μ s.

Additionally, the pulse width which is over the UC0 maximum count time specified by the clock source, can be measured by changing software.



Figure 3.9.11 Pulse Width Measurement

Note: Pulse width measure by setting "10" to TB0MOD<TB0CPM1:0>. The external interrupt INT5 is generated in timing of falling edge of TB0IN0 input. In other modes, it is generated in timing of rising edge of TB0IN0 input.

The width of low-level can be measured from the difference between the first C2 and the second C1 at the second INT5 interrupt.

d. Measurement of difference time

.

This mode is used to measure the difference in time between the rising edges of external pulses input through TB0IN0 and TB0IN1.

Keep the 16-bit timer/event counter (TMRB0) counting (Free running) with the internal clock, and load the UC0 value into TB0CP0 at the rising edge of the input pulse to TB0IN0. Then the interrupt INT5 is generated.

Similarly, the UC0 value is loaded into TB0CP1 at the rising edge of the input pulse to TB0IN1, generating the interrupt INT6.

The time difference between these pulses can be obtained by multiplying the value subtracted TB0CP0 from TB0CP1 and the internal clock cycle together at which loading the up counter value into TB0CP0 and TB0CP1 has been done.



Figure 3.9.12 Measurement of Difference Time

3.10 Serial Channels

TMP91CP27 includes 2 serial I/O channels. Each channel is called SIO0 and SIO1. For both channels either UART mode (Asynchronous transmission) or I/O interface mode (Synchronous transmission) can be selected.

• I/O interface mode	Mode 0:	For transmitting and receiving I/O data using the
		synchronizing signal SCLK for extending I/O.
	_Mode 1:	7-bit data
• UART mode	Mode 2:	8-bit data
	L_Mode 3:	9-bit data

In mode 1 and mode 2 a parity bit can be added. Mode 3 has a wakeup function for making the master controller start slave controllers via a serial link (A multi-controller system).

Figure 3.10.2 and Figure 3.10.3 are block diagrams for each channel. Each channel is structured in prescaler, serial clock generation circuit, receiving buffer and control circuit, transfer buffer and control circuit.

Serial channels 0 and 1 can be used independently.

Both channels operate in the same fashion except for the following points; hence only the operation of channel 0 is explained below.

	SIO0	SIO1
Pin name	TXD0 (P90) RXD0 (P91) CTS0 /SCLK0 (P92)	TXD1 (P93) RXD1 (P94) CTS1 /SCLK1 (P95)
IrDA mode	Yes	No

Table 3.10.1 Differences Between Channels 0 to 1



Figure 3.10.1 Data Format

www.DataSheet









Figure 3.10.3 Block Diagram of SIO1

3.10.2 Operation of Each Circuit

(1) Prescaler

There is a 6-bit prescaler for generating a clock to SIO0. The clock selected using SYSCR0<PRCK1:0> is divided by 4 and input to the prescaler as ϕ T0. The prescaler can be run only case of selecting the baud rate generator as the serial transfer clock.

Table 3.10.2 shows prescaler clock resolution into the baud rate generator.

Select System Clock	Select Prescaler Clock	Clock Gear Value	Prescaler Input Clock Resolution				
<sysck></sysck>	<prck1:0></prck1:0>	<gear2:0></gear2:0>	φ Τ0	φT2	φ Τ8	φT32	
1 (fs)		XXX	fs/2 ²	fs/24	fs/2 ⁶	fs/2 ⁸	
		000 (fc)	fc/2 ²	fc/2 ⁴	fc/2 ⁶	fc/2 ⁸	
	00	001 (fc/2)	fc/2 ³	fc/2 ⁵	fc/27	fc/2 ⁹	
	(f _{FPH})	010 (fc/4)	fc/2 ⁴	fc/2 ⁶	fc/2 ⁸	fc/2 ¹⁰	
0 (fc)		011 (fc/8)	fc/2⁵	fc/27	fc/2 ⁹	fc/2 ¹¹	
		100 (fc/16)	fc/2 ⁶	fc/28	fc/2 ¹⁰	fc/2 ¹²	
	10 (fc/16 Clock)	ххх	-	fc/2 ⁸	fc/2 ¹⁰	fc/2 ¹²	

Table 3.10.2 Prescaler Clock Resolution to Baud Rate Generato	Resolution to Baud Rate Generator
---	-----------------------------------

XXX: Don't care, -: Can not be used

The serial interface baud rate generator selects between 4 clock inputs: $\phi T0$, $\phi T2$, $\phi T8$, and $\phi T32$ among the prescaler outputs.

(2) Baud rate generator

The baud rate generator is a circuit that generates transmission and receiving clocks that determine the transfer rate of the serial channels.

The input clock to the baud rate generator, $\phi T0$, $\phi T2$, $\phi T8$ or $\phi T32$, is generated by the 6-bit prescaler which is shared by the timers. One of these input clocks is selected using the BR0CR<BR0CK1:0> field in the baud rate generator control register.

The baud rate generator includes a frequency divider, which divides the frequency by 1 or N + (16 - K)/16 to 16 values, determining the transfer rate.

The transfer rate is determined by the settings of BR0CR<BR0ADDE, BR0S3:0> and BR0ADD<BR0K3:0>.

- In UART mode
- (1) When BROCR < BROADDE > = 0

The settings BR0ADD<BR0K3:0> are ignored. The baud rate generator divides the selected prescaler clock by N, which is set in BR0CR<BR0S3:0>. (N = 1, 2, 3 \cdots 16)

(2) When BROCR < BROADDE > = 1

The N + (16 - K)/16 division function is enabled. The baud rate generator divides the selected prescaler clock by N + (16 - K)/16 using the value of N set in BR0CR<BR0S3:0> and the value of K set in BR0ADD<BR0K3:0>. (N = 2, 3 \cdots 15, K = 1, 2, 3 \cdots 15)

Note: If N = 1 or N = 16, the N + (16 - K)/16 division function is disabled. Clear BR0CR<BR0ADDE> register to "0".

• In I/O interface mode

The N + (16 - K)/16 division function is not available in I/O Interface Mode. Set BR0CR<BR0ADDE> to 0 before dividing by N.

The method for calculating the transfer rate when the baud rate generator is used is explained below.

In UART mode

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

In I/O interface mode

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

• Integer divider (N divider)

For example, when the source clock frequency (fc) = 12.288 MHz, the input clock frequency = ϕ T2 (fc/16), the frequency divider N (BR0CR<BR0S3:0>) = 5, and BR0CR<BR0ADDE> = 0, the baud rate in UART Mode is as follows:

```
* Clock state

\begin{cases}
System clock: & High frequency (fc) \\
Clock gear: & 1 (fc) \\
Prescaler clock: f_{FPH}
\end{cases}

Baud rate = \frac{fc/16}{5} \div 16

= 12.288 \times 10^6 \div 16 \div 5 \div 16 = 9600 (bps)
```

Note: The N + (16 - K)/16 division function is disabled and setting BR0ADD<BR0K3:0> is invalid.

• N + (16 – K)/16 divider (UART mode only)

Accordingly, when the source clock frequency (fc) = 4.8 MHz, the input clock frequency = ϕ T0, the frequency divider N (BR0CR<BR0S3:0>) = 7, K (BR0ADD<BR0K3:0>) = 3, and BR0CR<BR0ADDE> = 1, the baud rate in UART Mode is as follows:

```
* Clock state

System clock: High frequency (fc)

Clock gear: 1 (fc)

Prescaler clock: f_{FPH}

Baud rate = \frac{fc/4}{7 + (16 - 3)/16} \div 16

= 4.8 \times 10^6 \div 4 \div (7 + 13/16) \div 16 = 9600 (bps)
```

www.DataSheet4U.com

Table 3.10.3, Table 3.10.4 show examples of UART mode transfer rates.

Additionally, the external clock input is available in the serial clock. (Serial channels 0 and 1). The method for calculating the baud rate is explained below:

In UART mode

Baud rate = external clock input frequency \div 16 It is necessary to satisfy (external clock input cycle) \ge 4/fc

In I/O interface mode

Baud rate = External clock input frequency It is necessary to satisfy (external clock input cycle) \ge 16/fc

(w)	(when baud rate generator is used and BR0CR <br0adde> = 0.) Unit (kbps)</br0adde>								
fc [MHz]	Input Clock Divider N (Set to BR0CR <br0s3:0>)</br0s3:0>	φ Τ Ο	φT2	φT8	φT32				
9.830400	2	76.800	19.200	4.800	1.200				
\uparrow	4	38.400	9.600	2.400	0.600				
↑	8	19.200	4.800	1.200	0.300				
↑	0	9.600	2.400	0.600	0.150				
12.288000	5	38.400	9.600	2.400	0.600				
\uparrow	A	19.200	4.800	1.200	0.300				
14.745600	2	115.200							
\uparrow	3	76.800	19.200	4.800	1.200				
\uparrow	6	38.400	9.600	2.400	0.600				
\uparrow	С	19.200	4.800	1.200	0.300				

Table 3.10.3	Transfer Rate Selection
(when baud rate generator	is used and BR0CR <br0adde> = 0)</br0adde>

Note 1: Transfer rates in I/O interface mode are eight times faster than the values given above.

Note 2: The values in this table are calculated for when fc is selected as the system clock, fc/1 is selected as the clock gear and f_{FPH} is selected as the clock for prescaler.

Table 3.10.4 UART Baud Rate Selection

(when TMRA0 with input clock ϕ T1 is used and trigger output of TMRA0 is used.)

					Offic (kups)
fc	12.288	12	9.8304	8	6.144
TAOREG	MHz	MHz	MHz	MHz	MHz
1H	96		76.8	62.5	48
2H	48		38.4	31.25	24
3H	32	31.25			16
4H	24		19.2		12
5H	19.2				9.6
8H	12		9.6		6
AH	9.6				4.8
10H	6		4.8		3
14H	4.8				2.4

www.DataSheet4U.com

Method for calculating the transfer rate (when TMRA0 is used):

Transfer rate = $\frac{\text{Clock frequency determined by SYSCR0<PRCK1:0>}}{\text{TA0REG} \times \frac{8}{4} \times 16}$

 \Box (When TMRA0 with input clock ϕ T1 is used)

- Note 1: The TMRA0 match detect signal cannot be used as the transfer clock in I/O interface mode.
- Note 2: The values in this table are calculated for when fc is selected as the system clock, fc/1 is selected as the clock gear, and f_{FPH} is selected as the clock for prescaler.

(3) Serial clock generation circuit

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

In I/O interface mode

In SCLK output mode with the setting SCOCR < IOC > = 0, the basic clock is generated by dividing the output of the baud rate generator by 2, as described previously.

In SCLK input mode with the setting SCOCR < IOC > = 1, the rising edge or falling edge will be detected according to the setting of the SCOCR < SCLKS > register to generate the basic clock.

• In UART mode

The SC0MOD0<SC1:0> setting determines whether the baud rate generator clocks, the internal system clock f_{SYS}, the trigger output signal from timer TMRA0 or the external clock (SCLK0) is used to generate the basic clock SIOCLK.

(4) Receiving counter

The receiving counter is a 4-bit binary counter used in UART mode that counts up the pulses of the SIOCLK clock. It takes 16 SIOCLK pulses to receive 1 bit of data; each data bit is sampled three times – on the 7th, 8th and 9th clock cycles.

The value of the data bit is determined from these three samples using the majority rule.

For example, if the data bit is sampled respectively as 1, 0 and 1 on 7th, 8th and 9th clock cycles, the received data bit is taken to be 1. A data bit sampled as 0, 0 and 1 are taken to be 0.

(5) Receiving control

• In I/O interface mode

In SCLK output mode with the setting SCOCR < IOC > = 0, the RXD0 signal is sampled on the rising edge of the shift clock which is output on the SCLK0 pin.

In SCLK input mode with the setting SC0CR<IOC> = 1, the RXD0 signal is sampled on the rising or falling edge of the SCLK input, according to the SC0CR<SCLKS> setting.

• In UART mode

The receiving control block has a circuit that detects a start bit using the majority rule. Received bits are sampled three times; when two or more out of three samples are 0, the bit is recognized as the start bit and the receiving operation commences.

The values of the data bits that are received are also determined using the majority rule.

(6) The receiving buffers

To prevent overrun errors, the receiving buffers are arranged in a double-buffer structure.

Received data is stored one bit at a time in receiving buffer 1 (which is a shift register). When 7 or 8 bits of data have been stored in receiving buffer 1, the stored data is transferred to receiving buffer 2 (SC0BUF); this causes an INTRX0 interrupt to be generated.

The CPU only reads receiving buffer 2 (SC0BUF). Even before the CPU reads receiving buffer 2 (SC0BUF), the received data can be stored in receiving buffer 1. However, unless receiving buffer 2 (SC0BUF) is read before all bits of the next data are received by receiving buffer 1, an overrun error occurs. If an overrun error occurs, the contents of receiving buffer 1 will be lost, although the contents of receiving buffer 2 and SC0CR<RB8> will be preserved.

SCOCR < RB8 > is used to store either the parity bit – added in 8-bit UART mode – or the most significant bit (MSB) – in 9-bit UART mode.

In 9-bit UART mode the wake-up function for the slave controller is enabled by setting SC0MOD0<WU> to 1; in this mode INTRX0 interrupts occur only when the value of SC0CR<RB8> is 1.

(7) Transmission counter

The transmission counter is a 4-bit binary counter that is used in UART mode and which, like the receiving counter, counts the SIOCLK clock pulses; a TXDCLK pulse is generated every 16 SIOCLK clock pulses.



Figure 3.10.4 Generation of Transmission Clock

www.DataSheet4U.com

- (8) Transmission controller
 - In I/O interface mode

In SCLK output mode with the setting SCOCR<IOC> = 0, the data in the transmission buffer is output one bit at a time to the TXD0 pin on the rising edge of the shift clock which is output on the SCLK0 pin.

In SCLK input mode with the setting SCOCR < IOC > = 1, the data in the transmission buffer is output one bit at a time on the TXD0 pin on the rising or falling edge of the SCLK0 input, according to the SCOCR < SCLKS > setting.

• In UART mode

When transmission data sent from the CPU is written to the transmission buffer, transmission starts on the rising edge of the next TXDCLK, generating a transmission shift clock TXDSFT.

Handshake function

Serial channels 0 and 1 each has a $\overline{\text{CTS}}$ pin. Use of this pin allows data can be sent in units of one data format; thus, overrun errors can be avoided. The handshake functions is enabled or disabled by the SC0MOD0<CTSE> setting.

When the CTSO pin condition is High level, after completed the current data send, data transmission is halted until the CTSO pin condition is low again. However, the INTTXO Interrupt is generated, it requests the next send data to the CPU. The next data is written in the transmission buffer and data sending is halted.

Though there is no $\overline{\text{RTS}}$ pin, a handshake function can be easily configured by setting any port assigned to be the $\overline{\text{RTS}}$ function. The $\overline{\text{RTS}}$ should be output "High" to request send data halt after data receive is completed.



Figure 3.10.5 Hand Shake Function



- Note 1: If the CTS signal goes high during transmission, no more data will be sent after completion of the current transmission.
- Note 2: Transmission starts on the first falling edge of the TXDCLK clock after the CTS signal has fallen.

Figure 3.10.6 CTS (Clear to send) Timing

(9) Transmission buffer

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

(10) Parity control circuit

When SCOCR<PE> in the serial channel control register is set to 1, it is possible to transmit and receive data with parity. However, parity can be added only in 7-bit UART mode or 8-bit UART mode. The SCOCR<EVEN> field in the serial channel control register allows either even or odd parity to be selected.

In the case of transmission, parity is automatically generated when data is written to the transmission buffer SCOBUF. The data is transmitted after the parity bit has been stored in SCOBUF<TB7> in 7-bit UART mode or in SCOMODO<TB8> in 8-bit UART mode. SCOCR<PE> and SCOCR<EVEN> must be set before the transmission data is written to the transmission buffer.

In the case of receiving, data is shifted into receiving buffer 1, and the parity is added after the data has been transferred to receiving buffer 2 (SC0BUF), and then compared with SC0BUF<RB7> in 7-bit UART mode or with SC0CR<RB8> in 8-bit UART mode. If they are not equal, a parity error is generated and the SC0CR<PERR> flag is set.

(11) Error flags

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

1. Overrun error<OERR>

If all the bits of the next data item have been received in receiving buffer 1 while valid data still remains stored in receiving buffer 2 (SC0BUF), an overrun error is generated.

The below is a recommended flow when the overrun error is generated. (INTRX interrupt routine)

- 1) Read receiving buffer
- 2) Read error flag
- 3) If $\langle OERR \rangle = 1$
 - then
 - a. Set to disable receiving (write 0 to SC0MOD0<RXE>)
 - b. Wait to terminate current frame
 - c. Read receiving buffer
 - d. Read error flag
 - e. Set to enable receiving (write 1 to SC0MOD0<RXE>)
 - f. Request to transmit again
- 4) Other

2. Parity error<PERR>

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

3. Framing error<FERR>

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

(12) Timing generation

a. In UART mode

Receiving

Mode	9 bits	8 bits + Parity	8 bits, 7 bits + Parity, 7 bits
Interrupt generation timing	Center of last bit (Bit8)	Center of last bit (Parity bit)	Center of stop bit
Framing error generation timing	Center of stop bit	Center of stop bit	Center of stop bit
Parity error generation timing	_	Center of last bit (Parity bit)	Center of last bit (Parity bit)
Overrun error generation timing	Center of last bit (Bit8)	Center of last bit (Parity bit)	Center of stop bit

Note: In 9 bits and 8 bits + Parity modes, 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.

Transmitting

Mode	9 bits	8 bits + Parity	8 bits, 7 bits + Parity, 7 bits
Interrupt timing	Just before stop bit is	Just before stop bit is	Just before stop bit is
	transmitted	transmitted	transmitted

b. I/O interface

Transmission interrupt timing	SCLK output mode	Immediately after rise of last SCLK signal. (See Figure 3.10.19.)			
	SCLK input mode	Immediately after rise of last SCLK signal rising mode, or immediately after fall in falling mode. (See Figure 3.10.20.)			
Receiving interrupt timing	SCLK output mode	Timing used to transfer received to data receive buffer 2 (SC0BUF) (e.g., immediately after last SCLK). (See Figure 3.10.21.)			
	SCLK input mode	Timing used to transfer received data to receive buffer 2 (SC0BUF) (e.g., immediately after last SCLK). (See Figure 3.10.22.)			

3.10	.3 SFR									
		7	6	5	4	3	2	1	0	
SC0MOD0 (0202H)	Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0	
	Read/Write				R/V	V				
	After reset	0	0	0	0	0	0	0	0	
	Function	Transfer data bit8	Hand shake	Receive	Wakeup	Serial tran	smission	Serial trans	mission clock	
			function	control	function	mode		(UART)		
			control 0: CTS disable 1: CTS	0: Receive disable 1: Receive enable	0: Disable 1: Enable	00: I/O inte	0: I/O interface mode 1: 7-bit UART mode 0: 8-bit UART mode		00: TMRA0 trigger 01: Baud rate generator 10: Internal clock f _{SYS}	
						01: 7-bit U				
						10: 8-bit U				
						11: 9-bit UART mode		11: External clcok		
			enable					(SCLK0 in	put)	
						Serial	Serial transmission clock source (UAR ⁻)			
						00	Timer TMRA0 trigger output signal			
					01 E	01 Baud rate generator				
						10 I	10 Internal clock f _{SYS} 11 External clock (SCLK0 input)			
						11 E				
						Note:	te: The clock selection for the I/O interface mode is controlled by the			
						0 a vial	serial control register (SCUCR).			
							00 I/O Interface mode			
						01		7	'-bit mode	
						10	UART mo	de 8	3-bit mode	
						11		g)-bit mode	
						→ Wake	Wakeup function			
						ę	- 9-bit UART	(Other modes	
						0 Ir	nterrupt gene when data is r	rated eceived	Don't care	
							nterrupt gene	rated 8 = 1		
						-> Recei	Receiving function			
www.DataSheet4U.com						0 F) Receive disabled			
						1 F	Receive enabled			
			L			- Hands	andshake function (CTS pin)			
						0	Disabled (Alwa	ays transfer	able)	
						1 6	Enabled			
						-> Trans	mission data t	oit8		

Figure 3.10.7 Serial Mode Control Register 0 (for SIO0 and SC0MOD0)



Figure 3.10.8 Serial Mode Control Register 0 (for SIO1 and SC1MOD0)
		7	6	5	4	3	2	1	0
SC0CR	Bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
(0201H)	Read/Write	R	R/	W	R (Clea	ared to 0 whe	n read)	R/	Ŵ
	After reset	Undefined	0	0	0	0	0	0	0
	Function	Received data bit8	Parity 0: Odd 1: Even	Parity addition 0: Disable		1: Error			0:Baud rate generator 1:SCLK0
					Overrun	Parity	Framing		pin input
						$ \begin{array}{c} \downarrow \\ \downarrow $	terface input Baud rate ge SCLK0 pin in e selection fo Transmits an data on rising Transmits an data on falling ning error flag rrun error flag rrun error flag run error flag gy addition en Disabled Enabled parity additio Odd parity Even parity	clock selectionerator put r SCLK0 pin d receivers g edge of SCI d receivers g edge SCLK G Cle h ables n/check	(I/O mode) (I/O m

www.DataSheet4U.com

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

Figure 3.10.9 Serial Control Register (for SIO0 and SC0CR)



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

www.DataSheet4U.com

Figure 3.10.10 Serial Control Register (for SIO1 and SC1CR)

		7	6	ł	5	4	4	3	2	1	0
BR0CR	Bit symbol	-	BR0ADDE	BRC	OCK1	BRC	OCK0	BR0S3	BR0S2	BR0S1	BR0S0
(0203H)	Read/Write						R/	W			-
	After reset	0	0	(0	(0	0	0	0	0
	Function	Always	+(16 – K)/16	00: ¢	т0						
		write "0".	division	01: ¢	T2						
			0: Disable	10: ¢	T8			Set	ting of the d	ivided freque	ncy
			1: Enable	11:¢	132						
	Г										
	+ (16 - K)/16 (divisions en	ahla	Sett	tina the	↓ input	clock	of baud rate	generator		
			abic	000	T T	mput			gonorator		
				00	Inter	nal clo	ck oT	0			
				01	Inter	nal clo	CK of Z	2			
				10	Inter	nai cio	CK φI	5			
				11	Inter		υς φι	52			
									r	1	
		7	6		5		4	3	2	1	0
BR0ADD	Bit symbol					-		BR0K3	BR0K2	BR0K1	BR0K0
(0204H)	Read/Write					-	/		R/	W	
	After reset		\rightarrow					0	0	0	0
	Function										
								S	Sets frequer	icy divisor "K	"
								(0	divided by N	+ (16 – K)/16	6)
	Sets b	baud rate de	nerator freque	ncv d	ivisor s	etina	←				
			BR0CR <b< td=""><td>R0AD</td><td>DE> =</td><td>1</td><td>BR00</td><td>CR<br0add< td=""><td>E> = 0</td><td></td><td></td></br0add<></td></b<>	R0AD	DE> =	1	BR00	CR <br0add< td=""><td>E> = 0</td><td></td><td></td></br0add<>	E> = 0		
/ww.DataSheet4U.com	В	R0CR	0000 (N = 16)	00	10 (N =	2)	0001	(N = 1) (UAR	T only)		
		BR0S3:0>	or		to			to			
	BR0ADD		0001 (N = 1)	111	1 (N = ⁻	15)		1111 (N = 15	5)		
	<br0k3:0></br0k3:0>			<u> </u>	<u></u>			0000 (N = 16	5)		
	000	U 7 1)	Disable		Disable	;					
	t_0 Disable					by		Divided by N	1		
	1111 (K = 15))/10					
		,		1							
	Note: Who	en N + (16 -	K)/16 divisior	n funct	tion is u	used o	or not				
	N		LIART Mor	1e	1/0	Mode					

Set BR0CR<BR0ADDE> to 1 after setting K (K = 1 to 15) to BR0ADD<BR0K3:0> when + (16 - K)/16 division function is used.

Figure 3.10.11 Baud Rate Generator Control (for SIO0, BR0CR and BR0ADD)

×

×

2 to 15

1, 16

0

×

		7	6		5	4	3	2	1	0
BR1CR	Bit symbol	-	BR1ADDE	BR	1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
(020BH)	Read/Write					F	2/W			
	After reset	0	0		0	0	0	0	0	0
	Function	Always	+(16 – K)/16	00: (φ Τ0					
		write "0".	division	01: _φ T2						
			0: Disable	10: 0	 фТ8			Divided frequ	ioncy sottin	a
			1: Enable	11:¢	T32			Divided lieq	acticy settin	9
		l				1				
	_									
	\downarrow					Ļ				
	+ (16 – K)/16 (divisions ena	ble	Inp	ut clock	selection f	or baud rate g	enerator		
Г	0 Disabled			00	Inter	mal clock _{\$} T	0			
F	1 Enabled			01	Inter	nal clock _{\$} T	2			
	·		•	10	Inter	nal clock _{\$} T	8			
				11	Inter	nal clock _{\$} T	32			
		7	6		5	4	3	2	1	0
BR1ADD	Bit symbol						BR1K3	BR1K2	BR1K1	BR1K0
(020CH)	Read/Write			/	<u> </u>			R	/W	
	After reset			/			0	0	0	0
	Function									
								Set frequer	ncy divisor k	
							(1	Jivided by N	+ (16 – K)/	16)
	Raur	l rate cenera	tor frequency	/ divie	or setti	na 🖌	L			
	Bauc	uto gonela	BR1CI						0	
Num DataChaot411 ann		BR1CE					0001 (N - 1		v)	
ww.DataSneet4U.com		<br1s3:0< td=""><td>) 0000 (N =</td><td>16)</td><td>0010</td><td>) (N = 2)</td><td></td><td></td><td>y)</td><td></td></br1s3:0<>) 0000 (N =	16)	0010) (N = 2)			y)	
			Or	1)	1111		1111	(N = 15)		
	<pre>BRIADD <br1k3:0></br1k3:0></pre>	\sim	0001 (N =	= 1)	1111	(11 = 15)	0000	(N = 16)		
	00	00	Disable	e	Di	sable		. /		
	0001 0	(K = 1)	Disabil	-	ייים	ded by				
	t	0	Disable	e	N + (1	6 – K)/16	Divid	ed by N		
	1111 (K = 15)			(-	, -				
	0001 (ta 1111 (l	(K = 1) o K = 15)	Disable	e	Divi N + (1	ded by 6 – K)/16	Divid	ed by N		

Note: When N + (16 - K)/16 division function is used or not

N	UART Mode	I/O Mode
2 to 15	0	×
1, 16	×	×

Set BR1CR<BR1ADDE> to 1 after setting K (K = 1 to 15) to BR1ADD<BR1K3:0> when + (16 - K)/16 division function is used.

Figure 3.10.12 Baud Rate Generator Control (for SIO1, BR1CR and BR1ADD)



Note: Prohibit read modify write for SC0BUF.

Figure 3.10.13 Serial Transmission/Receiving Buffer Register (for SIO0 and SC0BUF)

		7	6	5	4	3	2	1	0
SC0MOD1	Bit symbol	I2S0	FDPX0			/		/	/
(0205H)	Read/Write	R/W	R/W			/	/	/	
	After reset	0	0			/	/	/	
	Function	IDLE2	Duplex						
		0: Stop	0: Half						
		1: Run	1: Full						





Note: Prohibit read modify write for SC1BUF.

Figure 3.10.15 Serial Transmission/Receiving Buffer Register (for SIO1 and SC1BUF) www.DataSheet4U.com

SC1 (02

		7	6	5	4	3	2	1	0
IMOD1	Bit symbol	I2S1	FDPX1	/	/	/	/		
20DH)	Read/Write	R/W	R/W	/	/	/	/		
	After reset	0	0	/	/	/	/		
	Function	IDLE2	Duplex						
		0: Stop	0: Half						
		1: Run	1: Full						

Figure 3.10.16 Serial Mode Control Register1 (for SIO1 and SC1MOD1)

3.10.4 Operation of Each Mode

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

This mode allows an increase in the number of I/O pins available for transmitting data to or receiving data from an external shift register.

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



Figure 3.10.17 Example of SCLK Output Mode Connection



Figure 3.10.18 Example of SCLK Input Mode Connection

a. Transmission

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





In SCLK Input Mode, 8-bit data is output from the TXD0 pin when the SCLK0 input becomes active after the data has been written to the transmission buffer by the CPU. When all data is outputted, INTESO<ITX0C> will be set to generate INTTX0 interrupt.



www.DataSheet4U.com Figure 3.10.20 Transmission Operation in I/O Interface Mode (SCLK0 input mode)

b. Receiving

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

The outputting for the first SCLK0 starts by setting SC0MOD0<RXE> to 1.





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



www.DataSheet4U.com Figure 3.10.22 Receiving Operation in I/O Interface Mode (SCLK0 input mode)

Note: If receiving, set to the receive enable state (SC0MOD0<RXE> = 1) in both SCLK input mode and output mode.

c. Transmission and receiving (Full duplex mode)

When the full duplex mode is used, set the level of receive interrupt to "0" and set enable the interrupt level (1 to 6) to the Transfer interrupts. In the transfer interrupt program, the receiving operation should be done like the below example before setting the next transfer data.

Example: Channel 0, SCLK output

Baud rate = 9600 bpsfc = 14.7456 MHz

* Clock state System clock: High frequency (fs) Clock gear: 1 (fc) Prescaler clock: f_{FPH}

Main routine

	7	6	5	4	3	2	1	0		
INTES0	0	0	0	1	0	0	0	0		Set transmission interrupt level, and disable receiving interrupt.
P9CR	-	-	-	-	-	1	0	1	۱	
P9FC	_	_	-	-	-	1	_	1	ſ	Set to P90 (TXD0) and P92 (SCLK0).
SC0MOD0	0	0	0	0	0	0	0	0		Set to I/O interface mode.
SC0MOD1	1	1	0	0	0	0	0	0		Set to full duplex mode.
SC0CR	0	0	0	0	0	0	0	0		SCLK out, transmit on negative edge, receive on positive edge
BR0CR	0	0	0	1	0	1	1	0		Set to 9600 bps.
SC0MOD0	0	0	1	0	0	0	0	0		Enable receiving.
SC0BUF	*	*	*	*	*	*	*	*		Set the transfer data.
INTTX0 ir	nter	rup	t ro	utin	е					

11117.01	noi	ıup		uun	0				
Acc←SC0B	UF								Read the receiving data.
SC0BUF	*	*	*	*	*	*	*	*	Set the next transfer data.
X: Don't	care	э, -	-: N	o cl	han	ge			

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

7-bit UART mode is selected by setting serial channel mode register SC0MOD0<SM1:0> to 01.

In this mode, a parity bit can be added. Use of a parity bit is enabled or disabled by the setting of the serial channel control register SCOCR<PE> bit; whether even parity or odd parity will be used is determined by the SCOCR<EVEN> setting when SCOCR<PE> is set to 1 (Enabled).

Setting example: When transmitting data of the following format, the control registers should be set as described below. This explanation applies to channel 0.



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

8-bit UART mode is selected by setting SC0MOD0<SM1:0> to 10. In this mode, a parity bit can be added (use of a parity bit is enabled or disabled by the setting of SC0CR<PE>); whether even parity or odd parity will be used is determined by the SC0CR<EVEN> setting when SC0CR<PE> is set to 1 (Enabled).

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



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

9-bit UART mode is selected by setting SC0MOD0<SM1:0> to 11. In this mode parity bit cannot be added.

In the case of transmission the MSB (9th bit) is written to SC0MOD0<TB8>. In the case of receiving it is stored in SC0CR<RB8>. When the buffer is written and read, the MSB is read or written first, before the rest of the SC0BUF data.

Wakeup function

In 9-bit UART mode, the wakeup function for slave controllers is enabled by setting SC0MOD0<WU> to 1. The interrupt INTRX0 occurs only when $\langle RB8 \rangle = 1$.



Note: The TXD pin of each slave controller must be in open-drain output mode.

Figure 3.10.23 Serial Link Using Wakeup Function

Protocol

- a. Select 9-bit UART mode on the master and slave controllers.
- b. Set the SC0MOD0<WU> bit on each slave controller to 1 to enable data receiving.
- c. 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".



- d. Each slave controller receives the above frame. If it matches with own select code, clears<WU> bit to "0".
- e. The master controller transmits data to the specified slave controller whose SC0MOD0<WU> bit is cleared to "0". The MSB (Bit8)<TB8> is cleared to "0".



f. The other slave controllers (whose<WU> bits remain at 1) ignore the received data because their MSB (Bit8 or <RB8>) are set to "0", disabling INTRX0 interrupts. The slave controller (<WU> bit = "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.



Example: To link two slave controllers serially with the master controller using the system clock fSYS as the transfer clock.

www.DataSheet4U.com

91CP27-156

3.10.5 Support for IrDA

SIO0 includes support for the IrDA 1.0 infrared data communication specification. Figure 3.10.24 shows the block diagram.



Figure 3.10.24 Block Diagram of IrDA

(1) Modulation of transmission data

When the transmission data is 0, output "H" level with either 3/16 or 1/16 times for width of baud rate (Selectable in software). When data is "1", modem output "L" level.



Figure 3.10.25 Example of Modulation of Transmission Data

www.DataSheet4U.com

(2) Modulation of receiving data

When the receive data has the effective high level pulse width (Software selectable), the modem outputs "0" to SIO0. Otherwise modem outputs "1" to SIO0. Receive pulse logic is selectable by SIRCR<RXSEL>.



Figure 3.10.26 Example of Modulation of Receiving Data

(3) Data format

Format of transmission/receiving must set to data length 8 bits, without parity bit, 1-bit of stop bit.

Any other settings don't guarantee the normal operation.

(4) SFR

Figure 3.10.27 shows the control register SIRCR. If change setting this register, must set it after set operation of transmission/receiving to disable (Both <TXEN> and <RXEN> of this register should be cleared to 0).

Any changing for this register during transmission or receiving operation doesn't guarantee the normal operation.

The following example describes how to set this register:

1)	SIO setting ↓	; Set SIO side.
2)	LD (SIRCR), 07H	; Set receiving effect pulse width to 8/16.
3)	LD (SIRCR), 37H	; Enable transmission/receiving by setting <txen>, <rxen> bit to "1".</rxen></txen>
	\checkmark	
4)	Start of transmission/receiving	; The modem operates as follows:
		 SIO0 starts transmitting.

• IR receiver starts receiving.

(5) Notes

 Making baud rate when using IrDA In baud rate during using IrDA, must set "01" to SC0MOD0<SC1:0> in SIO by using baud rate generator. TA0TRG, fSYS, SCLK0 input of except for it can not using.

2. Output pulse width and baud rate generator during transmission IrDA As the IrDA 1.0 physical layer specification, the data transfer speed and infra-red pulse width is specified.

Transfer Rate	Modulation	Transfer Rate Tolerance (% of Rate)	Minimum of Pulse Width	Typical of Pulse Width 3/16	Maximum of Pulse Width
2.4 kbps	RZI	±0.87	1.41 μs	78.13 μs	88.55 μs
9.6 kbps	RZI	±0.87	1.41 μs	19.53 μs	22.13 μs
19.2 kbps	RZI	<u>±</u> 0.87	1.41 μs	9.77 μs	11.07 μs
38.4 kbps	RZI	±0.87	1.41 μs	4.88 μs	5.96 μs
57.6 kbps	RZI	±0.87	1.41 μs	3.26 μs	4.34 μs
115.2 kbps	RZI	±0.87	1.41 μs	1.63 μs	2.23 μs

Table 3.10.5	Specification	of Transfer	Rate and	Pulse Widt	h
--------------	---------------	-------------	----------	------------	---

The infra-red pulse width is specified either baud rate T \times 3/16 or 1.6 µs (1.6 µs is equal to T \times 3/16 pulse width when baud rate is 115.2 kbps).

The TMP91CP27 has function that selectable the pulse width in during transmitting to either 3/16 or 1/16. But T \times 1/16 pulse width can be selected when the baud rate is equal or less than 38.4 kbps only. When 57.6 kbps and 115.2 kbps, the output pulse width should not be set to T \times 1/16.

As the same reason, + (16 - K)/16 division function in the baud rate generator of SIO0 cannot be used to generate 115.2 kbps baud rate.

Also when the 38.4 kbps and 1/16 pulse width, + (16 - K)/16 division function can not be used. Table 3.10.6 shows baud rate and pulse width for (16 - K)/16 division function.

www.DataSheet4U.com

Pulse Width			Baud	Rate		
	115.2 kbps	57.6 kbps	38.4 kbps	19.2 kbps	9.6 kbps	2.4 kbps
T × 3/16	×	0	0	0	0	0
T × 1/16	-	_	×	0	0	0
1 × 1110			~			

Table 3.10.6 Baud Rate and Pulse Width For (16 - K)/16 Division Function

 \odot : Can be used (16 – K)/16 division function.

 \times : Cannot be used (16 – K)/16 division function.

-: Cannot be set to $T \times 1/16$ pulse width.

		7	6	5	4	3	2	1	0	
SIRCR	Bit symbol	PLSEL	RXSEL	TXEN	RXEN	SIRWD3	SIRWD2	SIRWD1	SIRWD0	
(0207H)	Read/Write				R/W					
	After reset	0	0	0	0	0	0	0	0	
	Function	Selection	Receiving	Transmission	Receiving	Select receiving effective pulse width.				
		transmission	data logic	operation	operation	Set effective	e pulse width	tor equal or	more than	
		pulse width	0: "H" pulse	0: Disable	0: Disable	$2x \times (Value)$	+ 1).			
		0: 3/16	1: "L" pulse	1: Enable	1: Enable	Can be set:	1 to14			
		1: 1/16				Cannot be	set: 0, 15			
			J							
					Sele	Select receiving effective pulse width Formula:				
					> Rec	► Receiving effective pulse width ≥ 2x × (Setting value +				
					X = 1	x = 1/f _{FPH}				
					00	0000 Cannot be set.				
					00	01 is effect	tive.	or more than	14x + 100 ns	
					(·		00 100	
					111	10 Pulse w is effect	idth of equal live.	or more than	30x + 100 ns	
					11	11 Cannot	be set.			
					→ Ena	ble of receivi	ng operation	l		
					0	Disable	receiving op	eration		
					- 1	(Receiv	ed input is ig	nored.)		
						Enable	receiving op	eration.		
			L		> Ena	ble of transm	nission opera	tion		
					0	Disable	transmission	n operation		
						Enable	uansmission	operation		
		L			> Sele	ct transmiss	ion pulse wid	ith		
					1	Pulse w	/iath of 3/16			
						ruise w				

Note: If baud rate is late and secure of pulse width of IrDA 1.0 standard (Min 1.6 μs) is enable, reducing infrared raywww.DataSheet4U.comlighting time and consumption power by setting this bit to "1"are enable.

Figure 3.10.27 IrDA Control Register

3.11 Serial Bus Interface (SBI)

The TMP91CP27 has a 1-channel serial bus interface which employs a clocked-synchronous 8-bit SIO mode and an I²C bus mode (Multi muster).

The serial bus interface is connected to an external device through P61 (SDA) and P62 (SCL) in the I²C bus mode; and through P60 (SCK), P61 (SO), and P62 (SI) in the clocked-synchronous 8-bit SIO mode.

Each pin is specified as follows.

	ODE <ode62, 61=""></ode62,>	P6CR <p62c:60c></p62c:60c>	P6FC <p62f:60f></p62f:60f>
I ² C bus mode	11	11X	11X
Clock synchronous	XX	011	V11
8-bit SIO mode		010	

X: Don't care

3.11.1 Configuration



Figure 3.11.1 Serial Bus Interface (SBI)

3.11.2 Control

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

- Serial bus interface control register 1 (SBI0CR1)
- Serial bus interface control register 2 (SBI0CR2)
- Serial bus interface data buffer register (SBI0DBR)
- I²C bus address register (I2C0AR)
- Serial bus interface status register (SBI0SR)
- Serial bus interface baud rate register 0 (SBI0BR0)
- Serial bus interface baud rate register 1 (SBI0BR1)

The above registers differ depending on a mode to be used.

Refer to Section, "3.11.4 I²C Bus Mode Control Register" and "3.11.7 Clocked Synchronous 8-Bit SIO Mode Control."

3.11.3 Data Format in I²C Bus Mode

Data format in $I^{2}C$ bus mode is shown Figure 3.11.2.

(a) Addressing format



(b) Addressing format (With restart)



(c) Free data format (Transfer-format transfer from master device to slave device.)



Figure 3.11.2 Data Format in I²C Bus Mode

3.11.4 I²C Bus Mode Control Register

The following registers are used to control and monitor the operation status when using the serial bus interface (SBI) in the I²C bus mode.



Note 1: Set the <BC2:0> to "000" before switching to a clock-synchronous 8-bit SIO mode.

Note 2: For the frequency of the SCL line clock, see section 3.11.5, (3) "Serial clock".

Note 3: After reset, default value of <SCK0> is cleared to "0". Also, default value of <SWRMON> is set to "1".

Figure 3.11.3 Register for I²C Bus Mode

	/	7	6	5	4	3	2	1	0	
SBI0CR2	Bit symbol	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0	
(0243H)	Read/Write		V	V		W (Note 1)		W (Note 1)		
	After reset	0	0	0	1	0	0	0	0	
Read- modify-write instruction is prohibited.	Function Master/ slave selection		/ Transmitter/ Start/st receiver condition on selection genera		Cancel INTSBI interrupt request	Serial bus operation r selection (I 00: Port m 01: SIO m 10: I ² C but	Serial bus interface operation mode selection (Note 2) 00: Port mode 01: SIO mode 10: I ² C bus mode		set control nd "01" in an internal is	
-						11: (Rese	ved)			
						Serial bus 00 Port r 01 Clock 10 I ² C bu 11 (Rese INTSBI into 0 - 1 Cance	interface ope node (Serial I ed synchrono is mode erved) errupt reques el interrupt re	rating mode s pus interface pus 8-bit SIO t t	selection (Note output disable mode	e 2) ed)
						Start/stop	condition gen	eration		
						0 Gene	rates the stop	condition		
					,			ection		
						1 Trans	mitter			
						Master/slav	ve selection			
					,	0 Slave				
						1 Maste	er			
						L				

Note 1: Reading this register function as SBI0SR register. www.DataSheet4U.com

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

Switch a mode between I²C bus mode and clock-synchronous 8-bit SIO mode after confirming that input signals via port are high level.

Figure 3.11.4 Register for I²C Bus Mode

	\sim	7	6	5	4	3	2	1	0
SDIOSD	Bit symbol	, MST	TRY	BB	PIN	۵ ۵۱			
(0243H)	Boad/M/rito	10101			FIN	R	743	ADU	
· · · · · · · · · · · · · · · · · · ·	After reset	0	0	0	1	0	0	0	0
	Function	Master/	Transmitter/	I ² C bus	INTSBI	Arbitration	Slave	GENERAL	Last
		slave	receiver	status	interrupt	lost	address	CALL	received bit
- .		status	status	monitor	request	detection	match	detection	monitor
Read- modify-write		selection	selection		monitor	monitor	detection	monitor	0: 0
instruction is		monitor	monitor			0: –	monitor	0: Undetected	1:1
prohibited.						1: Detected	0: Undetected	1: Detected	
							1: Detected		
						🖵	Last receive	d bit monitor	
							0 Last re	ceived bit wa	as 0.
							1 Last re	ceived bit wa	as 1.
						$ \longrightarrow$	GENERAL	CALL detecti	on monitor
							0 Undete	ected	
							1 GENE	RAL CALL de	etected
						$\square \longrightarrow$	Slave addre	ss match de	tection monitor
							0 Undete	ected	
							1 Slave a	address mate detected	ch or GENERAL
						>	Arbitration lo	ost detection	monitor
							0 –		
							1 Arbitra	tion lost	
						>	INTSBI inter	rrupt request	monitor
							0 Interru	pt requested	
							1 Interru	pt canceled	
							I ² C bus state	us monitor	
							0 Free		
							1 Busy		
						\longrightarrow	Transmitter/	receiver stat	us monitor
WWW DataShoot411.com							0 Receiv	er	
www.DataSneet4U.Com							1 Transn	nitter	
						>	Master/slave	e status mon	itor
							0 Slave		
							1 Master		

Serial Bus Interface Status Register

Note: Writing in this register functions as SBI0CR2.

Figure 3.11.5 Register for I²C Bus Mode



www.DataSheet4U.com

Note 2: SBI0DBR can't be read the written data. Therefore read-modify-write insturction (e.g., "BIT" instruction) is prohibitted.

Note 3: Written data in SBI0DBR is cleared by INTSBI signal.

I²C Bus Address Register

-					<u> </u>							
		7	6	5	4	3	2	1	0			
I2C0AR (0242H)	Bit symbol	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS			
	Read/Write		W									
Read- modify-write instruction is prohibited.	After reset	0	0	0	0	0	0	0	0			
	Function	Slave address selection for when device is operating as slave device										
									•			

Address recognition mode specification

0 Slave address recognition

1 Non slave address recognition



3.11.5 Control in I²C Bus Mode

(1) Acknowledge mode specification

Set the SBI0CR1<ACK> to 1 for operation in the acknowledge mode. The TMP91CP27 generates an additional clock pulse for an acknowledge signal when operating in master mode. In the transmitter mode during the clock pulse cycle, the SDA pin is released in order to receive the acknowledge signal from the receiver. In the receiver mode during the clock pulse cycle, the SDA pin is set to the low in order to generate the acknowledge signal.

Clear the <ACK> to 0 for operation in the non-acknowledge mode, the TMP91CP27 does not generate a clock pulse for the acknowledge signal when operating in the master mode.

(2) Select number of transfer bits

The SBI0CR1<BC2:0> is used to select a number of bits for next transmission and receiving data.

Since the $\langle BC2:0 \rangle$ is cleared to 000 as a start condition, a slave address and direction bit transmission are executed in 8 bits. Other than these, the $\langle BC2:0 \rangle$ retains a specified value.

(3) Serial clock

a. Clock source

The SBI0CR1<SCK2:0> is used to select a maximum transfer frequency outputted on the SCL pin in master mode.



Note 1: f_{SBI} shows f_{FPH}.

Note 2: It's prohibit to use fc/16 prescaler clock (SYSCR0<PRCK1:0> = "10") when using SBI block. (I^2C bus and clocked synchronous)





b. Clock synchronization

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

The TMP91CP27 has a clock synchronization function for normal data transfer even when more than one master exists on the bus.

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



Figure 3.11.8 Clock Synchronization

As master A pulls down the internal SCL output to the low level at point "a", the SCL line of the bus becomes the low level. After detecting this situation, Master B resets a counter of high-level width of an own clock pulse and sets the internal SCL output to the low level.

Master A finishes counting low-level width of an own clock pulse at point "b" and sets the internal SCL output to the high level. Since master B holds the SCL line of the bus at the low level, master A wait for counting high-level width of an own clock pulse. After master B finishes counting low-level width of an own clock pulse at point "c" and master A detects the SCL line of the bus at the high level, and starts counting high level of an own clock pulse. The clock pulse on the bus is determined by the master device with the shortest high-level width and the master device with the longest low-level width from among those master devices connected to the bus.

(4) Slave address and address recognition mode specification

When the TMP91CP27 is used as a slave device, set the slave address <SA6:0> and <ALS> to the I2C0AR. Clear the <ALS> to "0" for the address recognition mode.

(5) Master/slave selection

Set the SBI0CR2<MST> to "1" for operating the TMP91CP27 as a master device. Clear the SBI0CR2<MST> to "0" for operation as a slave device. The <MST> is cleared to "0" by the hardware after a stop condition on the bus is detected or arbitration is lost.

(6) Transmitter/receiver selection

Set the SBI0CR2<TRX> to "1" for operating the TMP91CP27 as a transmitter. Clear the <TRX> to "0" for operation as a receiver. In slave mode, when transfer data in addressing format, when received slave address is same value with setting value to I2C0AR, or GENERAL CALL is received (All 8-bit data are "0" after a start condition), the <TRX> is set to "1" by the hardware if the direction bit (R/\overline{W}) sent from the master device is "1", and <TRX> is cleared to "0" by the hardware if the bit is "0". In the Master Mode, after an Acknowledge signal is returned from the slave device, the <TRX> is cleared to "0" by the hardware if a transmitted direction bit is "1", and is set to "1" by the hardware if a transmitted direction bit is "1", and is set to "1" by the hardware if a transmitted direction bit is "1", and is set to "1" by the hardware if it is "0". When an acknowledge signal is not returned, the current condition is maintained.

The $\langle TRX \rangle$ is cleared to "0" by the hardware after a stop condition on the bus is detected or arbitration is lost.

(7) Start/stop condition generation

When the SBI0SR<BB> is "0", slave address and direction bit which are set to SBI0DBR are output on a bus after generating a start condition by writing "1" to the SBI0CR2<MST, TRX, BB, PIN>. It is necessary to set transmitted data to the data buffer register (SBI0DBR) and set "1" to <ACK> beforehand.





www.DataSheet4U.com

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



Figure 3.11.10 Generation of Stop Condition

The state of the bus can be ascertained by reading the contents of SBI0SR<BB>. SBI0SR<BB> will be set to 1 if a start condition has been detected on the bus, and will be cleared to 0 if a stop condition has been detected (Bus free status).

And about generation of stop condition in master mode, there are some limitation points. Please refer to section 3.11.6, (4) "Stop condition generation".

(8) Interrupt service requests and interrupt cancellation

When a serial bus interface interrupt request (INTSBI) occurs, the SBI0CR2<PIN> is cleared to "0". During the time that the SBI0CR2<PIN> is "0", the SCL line is pulled down to the low level.

The <PIN> is cleared to "0" when a 1 word of data is transmitted or received. Either writing/reading data to/from SBI0DBR sets the <PIN> to "1".

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

In the address recognition mode ($\langle ALS \rangle = 0$), $\langle PIN \rangle$ is cleared to "0" when the received slave address is the same as the value set at the I2COAR or when a GENERAL CALL is received (All 8-bit data are "0" after a start condition). Although SBI0CR2 $\langle PIN \rangle$ can be set to "1" by the program, the $\langle PIN \rangle$ is not clear it to "0" when it is written "0".

(9) Serial bus interface operation mode selection

SBI0CR2<SBIM1:0> is used to specify the serial bus interface operation mode. Set SBI0CR2<SBIM1:0> to "10" when the device is to be used in I²C bus mode after confirming pin condition of serial bus interface to "H".

Switch a mode to port after confirming a bus is free.

(10) Arbitration lost detection monitor

Since more than one master device can exist simultaneously on the bus in I^2C bus mode, a bus arbitration procedure has been implemented in order to guarantee the integrity of transferred data.

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

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





www.DataSheet4U

The TMP91CP27 compares the levels on the bus's SDA line with those of the internal SDA output on the rising edge of the SCL line. If the levels do not match, arbitration is lost and SBI0SR<AL> is set to "1".

When SBI0SR<AL> is set to "1", SBI0SR<MST, TRX> are cleared to "0" and the mode is switched to slave receiver mode. Thus, clock output is stopped in data transfer after setting <AL> = "1".

SBI0SR<AL> is cleared to "0" when data is written to or read from SBI0DBR or when data is written to SBI0CR2.





(11) Slave address match detection monitor

www.DataSheet4U.com

SBI0SR<AAS> is set to "1" in slave mode, in address recognition mode (e.g., when I2C0AR<ALS> = "0"), when a GENERAL CALL is received, or when a slave address matches the value set in I2C0AR. When I2C0AR<ALS> = "1", SBI0SR<AAS> is set to "1" after the first word of data has been received. SBI0SR<AAS> is cleared to "0" when data is written to or read from the data buffer register SBI0DBR.

(12) General call detection monitor

SBI0SR<AD0> is set to "1" in slave mode, when a GENERAL CALL is received (All 8-bit received data is "0" after a start condition). SBI0SR<AD0> is cleared to "0" when a start condition or stop condition is detected on the bus.

(13) Last received bit monitor

The SDA line value stored at the rising edge of the SCL line is set to the SBI0SR<LRB>. In the acknowledge mode, immediately after an INTSBI interrupt request is generated, an acknowledge signal is read by reading the contents of the SBI0SR<LRB>.

(14) Software reset function

The software reset function is used to initialize the SBI circuit, when SBI is rocked by external noises, etc.

An internal reset signal pulse can be generated by setting SBI0CR2<SWRST1:0> to "10" and "01". This initializes the SBI circuit internally. All command (except SBI0CR2<SBIM1:0>) registers and status registers are initialized as well.

 ${\rm SBI0CR1}{<}{\rm SWRMON}{>}$ is automatically set to "1" after the SBI circuit has been initialized.

(15) Serial bus interface data buffer register (SBI0DBR)

The received data can to read and transmission data can to write by reading or writing SBI0DBR.

In the master mode, after the start condition is generated the slave address and the direction bit are set in this register.

(16) I²C bus address register (I2C0AR)

I2C0AR<SA6:0> is used to set the slave address when the TMP91CP27 functions as a slave device.

The slave address outputted from the master device is recognized by setting the I2C0AR<ALS> to "0". The data format is the addressing format. When the slave address is not recognized at the $\langle ALS \rangle = "1"$, the data format is the free data format.

(17) Baud rate register (SBI0BR1)

Write "1" to SBI0BR1<P4EN> before operation commences.

(18) Setting register for IDLE2 mode operation (SBI0BR0)

SBI0BR0<I2SBI0> is the register setting operation/stop during IDLE2 mode. Therefore, setting <I2SBI0> is necessary before the HALT instruction is executed.

3.11.6 Data Transfer In I²C Bus Mode

(1) Device initialization

Set the SBI0BR1<P4EN>, SBI0CR1<ACK, SCK2:0>, set SBI0BR1 to "1" and clear bits 7 to 5 and 3 in the SBI0CR1 to "0".

Set a slave address $\langle SA6:0 \rangle$ and the $\langle ALS \rangle = "0"$ when an addressing format) to the I2C0AR.

For specifying the default setting to a slave receiver mode, clear "0" to the <MST, TRX, BB> and set "1" to the <PIN>, "10" to the <SBIM1:0>.

(2) Start condition and slave address generation

a. Master mode

In the master mode, the start condition and the slave address are generated as follows.

Check a bus free status (when $\langle BB \rangle = "0"$).

Set the SBI0CR1<ACK> to "1" (Acknowledge mode) and specify a slave address and a direction bit to be transmitted to the SBI0DBR.

When SBI0CR2<BB> = "0", the start condition are generated by writing "1111" to SBI0CR2<MST, TRX, BB, PIN>. Subsequently to the start condition, nine clocks are output from the SCL pin. While eight clocks are output, the slave address and the direction bit which are set to the SBI0DBR. At the 9th clock, the SDA line is released and the acknowledge signal is received from the slave device.

An INTS2 interrupt request occurs at the falling edge of the 9th clock. The <PIN> is cleared to "0". In the master mode, the SCL pin is pulled down to the low level while <PIN> is "0". When an interrupt request occurs, the <TRX> is changed according to the direction bit only when an acknowledge signal is returned from the slave device.

b. Slave mode

In the slave mode, the start condition and the slave address are received.

After the start condition is received from the master device, while eight clocks are output from the SCL pin, the slave address and the direction bit that are output from the master device are received.

When a GENERAL CALL or the same address as the slave address set in I2C0AR is received, the SDA line is pulled down to the low level at the 9th clock, and the acknowledge signal is output.

An INTSBI interrupt request generate on the falling edge of the 9th clock. The <PIN> is cleared to "0". In slave mode the SCL line is pulled down to the low level while the <PIN> = "0".





(3) 1-word data transfer

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

a. If <MST> = "1" (Master mode)

Check the <TRX> and determine whether the mode is a transmitter or receiver. When the <TRX> = "1" (Transmitter mode)

• Check the <LRB>. When <LRB> is "1", a receiver does not request data. Implement the process to generate a stop condition (Refer to 3.11.6 (4)) and terminate data transfer.

When the <LRB> is "0", the receiver requests new data. When the next transmitted data is 8 bits, write the transmitted data to SBI0DBR. When the next transmitted data is other than 8 bits, set the <BC2:0> <ACK> and write the transmitted data to SBI0DBR. After written the data, <PIN> becomes "1", a serial clock pulse is generated for transferring a new 1 word of data from the SCL pin, and then the 1-word data is transmitted. After the data is transmitted, an INTSBI interrupt request generates. The <PIN> becomes "0" and the SCL line is pulled down to the low level. If the data to be transferred is more than one word in length, repeat the procedure from the <LRB> checking above.



Figure 3.11.14 Example in Which <BC2:0> = "000" and <ACK> = "1" (Transmitter mode)

When the <TRX> is "0" (Receiver mode)

When the next transmitted data is other than 8 bits, set <BC2:0> <ACK> and read the received data from SBI0DBR to release the SCL line (Data which is read immediately after a slave address is sent is undefined). After the data is read, <PIN> becomes "1". Serial clock pulse for transferring new 1 word of data is defined SCL and outputs "L" level from SDA pin with acknowledge timing.

An INTSBI interrupt request then generates and the <PIN> becomes "0", Then the TMP91CP27 pulls down the SCL pin to the low level. The TMP91CP27 outputs a clock pulse for 1-word of data transfer and the acknowledge signal each time that received data is read from the SBI0DBR.



Figure 3.11.15 Example of When <BC2:0> = "000" and <ACK> = "1" (Receiver mode)

In order to terminate the transmission of data to a transmitter, clear <ACK> to "0" before reading data which is 1-word before the last data to be received. The last data word does not generate a clock pulse as the acknowledge signal. After the data has been transmitted and an interrupt request has been generated, set <BC2:0> to "001" and read the data. The TMP91CP27 generates a clock pulse for a 1-bit data transfer. Since the master device is a receiver, the SDA line on the bus remains high. The transmitter receives the high signal as an ACK signal. The receiver indicates to the transmitter that data transfer is complete.

After the one data bit has been received and an interrupt request been generated, the TMP91CP27 generates a stop condition (See Section 3.11.6 (4)) and terminates data transfer.





WW

b. If $\langle MST \rangle = 0$ (Slave mode)

In the slave mode the TMP91CP27 operates either in normal slave mode or in slave mode after losing arbitration.

In the slave mode, an INTSBI interrupt request occurs when the TMP91CP27 receives a slave address or a GENERAL CALL from the master device, or when a GENERAL CALL is received and data transfer is complete, or after matching received address. In the master mode, the TMP91CP27 operates in a slave mode if it losing arbitration. An INTSBI interrupt request generate when a word data transfer terminates after losing arbitration. When an INTSBI interrupt request generate the <PIN> is cleared to "0" and the SCL pin is pulled down to the low level. Either reading/writing from/to the SBI0DBR or setting the <PIN> to "1" will release the SCL pin after taking tLOW time.

Check the SBI0SR<AL>, <TRX>, <AAS>, and <AD0> and implements processes according to conditions listed in the next table.

	<trx></trx>	<al> <aas:< th=""><th><ad0></ad0></th><th>Conditions</th><th>Process</th></aas:<></al>		<ad0></ad0>	Conditions	Process	
	1	1	1	1	0	The TMP91CP27 loses arbitration when transmitting a slave address, and receives a slave address for which the value of the direction bit sent from another master is "1".	Set the number of bits of single word to <bc2:0>, and write the transmit data to SBI0DBR</bc2:0>
		0	1	0	In slave receiver mode, the TMP91CP27 receives a slave address for which the value of the direction bit sent from the master is "1".		
			0	0	In slave transmitter mode, transmission of data of single word is terminated.	Check the <lrb> setting. If <lrb> is set to "1", set <pin> to "1" since the receiver win no request the data which follows. Then, clear <trx> to "0" to release the bus. If <lrb> is cleared to "0" of and write the transmitted data to SBI0DBR since the receiver requests next data.</lrb></trx></pin></lrb></lrb>	
v.DataSheet4	J.com	1	1	1/0	The TMP91CP27 loses arbitration when transmitting a slave address, and receives a slave address or GENERAL CALL for which the value of the direction bit sent from another master is "0".	Read the SBI0DBR for setting the <pin> to "1" (reading dummy data) or set the <pin> to "1".</pin></pin>	
			0	0	The TMP91CP27 loses arbitration when transmitting a slave address or data, and terminates word data transfer.		
		0	1	1/0	In slave receiver mode the TMP91CP27 receives a slave address or GENERAL CALL for which the value of the direction bit sent from the master is "0".		
			0	1/0	In slave receiver mode the TMP91CP27 terminates receiving word data.	Set <bc2:0> to the number of bits in a word and read the received data from SBI0DBR.</bc2:0>	

Table 3.11.1 Operation in the Slave Mode

(4) Stop condition generation

When SBI0SR<BB> = 1, the sequence for generating a stop condition start by writing "1" to SBI0CR2<MST, TRX, PIN> and "0" to SBI0CR2<BB>. Do not modify the contents of SBI0CR2<MST, TRX, PIN, BB> until a stop condition has been generated on the bus. When the bus's SCL line has been pulled Low by another device, the TMP91CP27 generates a stop condition when the other device has released the SCL line and SDA pin rising.

When SBI0CR2<MST, TRX, PIN> are written "1" and <BB> is written "0" (Generate stop condition in master mode), <BB> changes to "0" by internal SCL changes to "1", without waiting stop condition. To check whether SCL and SDA pin are "1" by sensing their ports is needed to detect bus free condition.



Figure 3.11.17 Stop Condition Generation (Single master)



Figure 3.11.18 Stop Condition Generation (Multi master)

(5) Restart

Restart is used during data transfer between a master device and a slave device to change the data transfer direction.

The following description explains how to restart when the TMP91CP27 is in master mode.

Clear SBI0CR2<MST, TRX, BB> to 0 and set SBI0CR2<PIN> to 1 to release the bus. The SDA line remains high and the SCL pin is released. Since a stop condition has not been generated on the bus, other devices assume the bus to be in busy state.

And confirm SCL pin, that SCL pin is released and become bus-free state by SBI0SR $\langle BB \rangle = "0"$ or signal level "1" of SCL pin in port mode. Check the $\langle LRB \rangle$ until it becomes 1 to check that the SCL line on a bus is not pulled down to the low level by other devices. After confirming that the bus remains in a free state, generate a start condition using the procedure described in 3.11.6 (2).

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



Figure 3.11.19 Timing Chart for Generate Restart
3.11.7 Clocked Synchronous 8-Bit SIO Mode Control

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

	-			Ochai Da			cgister i			
		7	6	5	4	3	2	1	0	
SBI0CR1	Bit symbol	SIOS	SIOINH	SIOM1	SIOM		SCK2	SCK1	SCK0	
(0240H)	Read/Write		V	V				W		
	After reset	0	0	0	0		0	0	0	
Read- modify-write instruction is prohibited.	Function	Transfer start/stop 0: Stop 1: Start	Continue/ abort transfer 0: Continue transfer 1: Abort transfer	Transfer mode 00: Transmiss 01: (Reserved 10: Transmiss receiving r 11: Receiving	e selection ion mode i) ion/ mode mode		Selection of serial clock frequency			
					Serial	clock selection $<$ SC	CK2:0>	\downarrow		
					000 001 010 011 100 101 110 111 → Transf 00		$ \begin{array}{c} $		ic) 1 (pin) f <u>c</u> [Hz]) X Pin)	
					01	(Reserved)				
					10	8-bit transmission/r	eceiving mode)		
					11	8-bit receiving mod	e			
					→ Contin	nue/Abort transfer				
					0	Continue transfer				
					1	Abort transfer (Auto	omatically clea	red after transf	er aborted)	
www.DataSheet4U.com					→ Tranet	fer start/stop				
					0	Stopped				
Note: S	set the tranfer m	ode and the	serial clock a	after setting	1	1 Started				
<	SIOS> to "0" and	d <sioinh></sioinh>	to "1".							

Serial	Bus	Interface	Control	Register 1
oonai	Duo	monuoo	001101	rtogiotor i

Corial	D	Interfees	Data	D #	Decister
Senar	BUS	Intenace	пана	BIIIIEE	Redister
COLICI	Pao	IIII COLICOU	Pau	Panor	I COGIOCOI

	/	7	6	5	4	3	2	1	0
SBI0DBR	Bit symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
(0241H) Dood	Read/Write				R (Receiver)	/W (Transfer)			
Read- modify-write	After reset				Unde	fined			

instruction is prohibited.

Figure 3.11.20	Register for	SIO Mode
----------------	--------------	----------

			Condin			n rogiotor	2		
	/	7	6	5	4	3	2	1	0
SBI0CR2	Bit symbol			/	/	SBIM1	SBIM0	_	_
(0243H)	Read/Write					V	V	W	W
	After reset					0	0	0	0
Read- modify- write instruction is prohibited.	Function					Serial bus in operation m selection 00: Port mo 01: SIO mo 10: I ² C bus 11: (Reserv	nterface iode de de mode ied)	Write	∋ "O".
						\downarrow			

Serial Bus Interface Control Register 2

Serial bus interface operation mode selection

00	Port	m	ode	(serial	bus	inte	erfa	ace	outp	out	disable	ed)

01 Clocked-synchronous 8-bit SIO mode

10	I ² C bus mode
11	(Reserved)

Note 1: Set the SBI0CR1<BC2:0> to "000" before switching to a clocked-synchronous 8-bit SIO mode. Note 2: Please always write "00" to SBICR2<1:0>.

			Ocha				gister			
		7	6	5	4	3	3	2	1	0
SBI0SR	Bit symbol					SI	OF	SEF		
(0243H)	Read/Write						F	2		
	After reset					0	C	0		
	Function					Serial transf opera status monit	l fer ation s tor	Shift operation status monitor		
www.DataSheet411.com							Sł (hift operation Shift o Shift o	n status moni peration termi peration in pro	tor nated ogress
www.bataoneet+o.com							→ <u>Se</u>	rial transfe	r operating sta	atus monitor
							() Transf	er terminated	
								1 Transf	er in progress	

Serial Bus Interface Status Register

Figure 3.11.21 Register for SIO Mode







- (1) Serial clock
 - a. Clock source

SBI0CR1<SCK2:0> is used to select the following functions:

Internal clock

In internal clock mode one of seven frequencies can be selected. The serial clock signal is output to the outside on the SCK pin. The SCK pin goes high when data transfer starts. When the device is writing (in transmit mode) or reading (in receive mode), data cannot follow the serial clock rate, so an automatic wait function is executed which automatically stops the serial clock and holds the next shift operation until reading or writing has been completed.



Figure 3.11.23 Automatic Wait Function

External clock (<SCK2:0> = "111")

An external clock input via the SCK pin is used as the serial clock. In order to ensure the integrity of shift operations, both the high and low-level serial clock pulse widths shown below must be maintained. The maximum data transfer frequency is 1 MHz (when fc = 16 MHz).



Figure 3.11.24 Maximum Data Transfer Frequency When External Clock Input Used

b. Shift edge

Data is transmitted on the leading edge of the clock and received on the trailing edge.

Leading edge shift

Data is shifted on the leading edge of the serial clock (on the falling edge of the SCK pin input/output).

Trailing edge shift

Data is shifted on the trailing edge of the serial clock (on the rising edge of the SCK pin input/output).



*: Don't care

www.DataSheet4U.com

Figure 3.11.25 Shift Edge

(2) Transfer modes

The SBI0CR1<SIOM1:0> is used to select a transmit, receive or transmit/receive mode.

a. 8-bit transmit mode

Set a control register to a transmit mode and write transmit data to the SBI0DBR.

After the transmit data is written, set the SBI0CR1<SIOS> to "1" to start data transfer. The transmitted data is transferred from SBI0DBR to the shift register and output to the SO pin in synchronized with the serial clock, starting from the least significant bit (LSB). When the transmission data is transferred to the shift register, the SBI0DBR becomes empty. An INTSBI (Buffer empty) interrupt request is generated to request new data.

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

When the external clock is used, data should be written to SBI0DBR before new data is shifted. The transfer speed is determined by the maximum delay time between the time when an interrupt request is generated and the time when data is written to SBI0DBR by the interrupt service program.

When the transmit is started, after the SBI0SR<SIOF> goes "1" output from the SO pin holds final bit of the last data until falling edge of the SCK.

Transmitting data is ended by clearing the <SIOS> to "0" by the buffer empty interrupt service program or setting the <SIOINH> to "1". When the <SIOS> is cleared, the transmitted mode ends when all data is output. In order to confirm if data is surely transmitted by the program, set the <SIOF> (Bit3 of SBI0SR) to be sensed. The SBI0SR<SIOF> is cleared to "0" when transmitting is complete.

When the <SIOINH> is set to "1", transmitting data stops. SBI0SR<SIOF> turns "0".

When an external clock is used, it is also necessary to clear SBI0CR1<SIOS> to "0" before new data is shifted; otherwise, dummy data is transmitted and operation ends.



Figure 3.11.26 Transmission Mode







b. 8-bit receive mode

Set the control register to receive mode and set SBI0CR1<SIOS> to "1" for switching to receive mode. Data is received into the shift register via the SI pin and synchronized with the serial clock, starting from the least significant bit (LSB). When 8-bit data is received, the data is transferred from the shift register to SBI0DBR. An INTSBI (Buffer full) interrupt request is generated to request that the received data be read. The data is then read from SBI0DBR by the interrupt service program.

When an internal clock is used, the serial clock will stop and the automatic wait function will be in effect until the received data has been read from SBI0DBR.

When an external clock is used, since shift operation is synchronized with an external clock pulse, the received data should be read from SBI0DBR before the next serial clock pulse is input. If the received data is not read, any further data which is to be received is canceled. The maximum transfer speed when an external clock is used is determined by the delay time between the time when an interrupt request is generated and the time when the received data is read.

Receiving of data ends when <SIOS> is cleared to "0" by the buffer full interrupt service program or when <SIOINH> is set to "1". If <SIOS> is cleared to "0", received data is transferred to SBIODBR in complete blocks. The received mode ends when the transfer is complete. In order to confirm whether data is being received properly by the program, set SBIOSR<SIOF> to be sensed. <SIOF> is cleared to "0" when receiving has been completed. When it is confirmed that receiving has been completed, the last data is read. When <SIOINH> is set to "1", data receiving stops. <SIOF> is cleared to "0" (The received data becomes invalid, therefore no need to read it).

Note: When the transfer mode is changed, the contents of SBI0DBR will be lost. If the mode must be changed, conclude data receiving by clearing <SIOS> to "0", read the last data, then change the mode.



Figure 3.11.28 Receiving Mode (when an internal clock is used)

c. 8-bit transmit/receive mode

Set a control register to a transmit/receive mode and write data to SBI0DBR. After the data has been written, set SBI0CR1<SIOS> to "1" to start transmitting/receiving. When data is transmitted, the data is output via the SO pin, starting from the least significant bit (LSB) and synchronized with the leading edge of the serial clock signal. When data is received, the data is input via the SI pin on the trailing edge of the serial clock signal. 8-bit data is transferred from the shift register to SBI0DBR and an INTSBI interrupt request is generated. The interrupt service program reads the received data from the data buffer register and writes the data that is to be transmitted. SBI0DBR is used for both transmitting and receiving. Transmitted data should always be written after received data has been read.

When an internal clock is used, the automatic wait function will be in effect until the received data has been read and the next data has been written. When an external clock is used, since the shift operation is synchronized with the external clock, received data is read and transmitted data is written before a new shift operation is executed. The maximum transfer speed when an external clock is used is determined by the delay time between the time when an interrupt request is generated and the time at which received data is read and transmitted data is written.

When transmission is started, after the SBI0SR<SIOF> goes "1" output from the SO pin holds final bit of the last data until falling edge of the SCK.

Transmitting/receiving data ends when <SIOS> is cleared to "0" by the INTSBI interrupt service program or when SBI0CR1<SI0INH> is set to "1". When <SIOS> is cleared to "0", received data is transferred to SBI0DBR in complete blocks. The transmit/receive mode ends when the transfer is complete. In order to confirm whether data is being transmitted/received properly by the program, set SBI0SR<SIOF> to be sensed. <SIOF> is set to "0" when transmitting/receiving has been completed. When <SIOINH> is set to 1, data transmitting/receiving stops. SBI0SR<SIOF> is then cleared to 0.

www.DataSheet4U.com

Note: When the transfer mode is changed, the contents of SBI0DBR will be lost. If the mode must be changed, conclude data transmitting/receiving by clearing <SIOS> to "0", read the last data, then change the transfer mode.



Figure 3.11.29 Transmission/Receiving Mode (when an internal clock is used)





(Transmission/receiving mode)

3.12 Analog/Digital Converter

The TMP91CP27 incorporates a 10-bit successive approximation-type analog/digital converter (AD converter) with 4-channel analog input.

Figure 3.12.1 is a block diagram of the AD converter. The 4-channel analog input pins (AN0 to AN3) are shared with the input-only port 5 and can thus be used as an input port.

Note: When IDLE2, IDLE1 or STOP mode is selected, so as to reduce the power, with some timing the system may enter a stand-by mode even though the internal comparator is still enabled. Therefore be sure to check that AD converter operations are halted before a HALT instruction is executed.



Figure 3.12.1 Block Diagram of AD Converter

3.12.1 Control Register

The AD converter is controlled by the two AD mode control registers: ADMOD0 and ADMOD1. The AD conversion results are stored in 8 kinds of AD conversion data upper and lower registers: ADREG04H/L, ADREG15H/L, ADREG26H/L and ADREG37H/L. Figure 3.12.2 to Figure 3.12.5 shows the registers related to the AD converter.



AD Mode Control Register 0

Figure 3.12.2 Register for AD Converter

					AL		nii oi Keg	iste	I I			
		7		(6	5	4		3	2	1	0
ADMOD1	Bit symbol	VREF	-ON	12/	٩D	/		A	DTRGE	ADCH2	ADCH1	ADCH0
(02B1H)	Read/Write	R/V	N	R/	W		/			R/	W	
	After reset	0		(C		/		0	0	0	0
	Function	VREF		IDLE2	2			AD	external			
		applica	ition	0: Sto	р			trig	ger start	Analog input channel selectior		
		control		1: Ope	erate			con	trol			
		0: OFF						0: E	Disable			
		1: ON						1: E	nable			
									_ L		•	
											↓	
							A	Analo	og input cl	hannel selec	ction	
							<scai< td=""><td>N></td><td>U (Channal</td><td>、 </td><td>1 (Channal</td><td>\mathbf{r}</td></scai<>	N>	U (Channal	、 	1 (Channal	\mathbf{r}
							.0-		fixed		scanned)
							000				Cocarnieu	
							001		AN1	$ANO \rightarrow A$	N1	
						010 011 (Note)			AN2	$ANO \rightarrow A$	$N1 \rightarrow AN2$	
									AN3	$ANO \rightarrow A$	$N1 \rightarrow AN2 -$	→ AN3
							100					
						-	101			Don'	t select.	
							110					
							111					
								≻ A	D conver	sion start co	ontrol by exte	rnal trigger
								(ADTRG ir	nput)	-	
									D Disab	led		
									1 Enabl	ed		
								<u>→ [[</u>	DLE2 con	trol		
									0 Stopp	ed		
									1 In ope	eration		
		L						→ C	ontrol of	application of	of reference v	oltage to
								A	D conver	ter		
									0 OFF			
et411.com									1 ON			
CL-10.0011								Р	oforo otor	ting conver	ion (Poforo)	writing "1" to

AD Mode Control Register 1

www.DataSheet4U.com

Before starting conversion (Before writing "1" to ADMOD0<ADS>), set the <VREFON> bit to "1".

Note: As pin AN3 also functions as the \overline{ADTRG} input pin, do not set $\langle ADCH2:0 \rangle = "011"$ when using \overline{ADTRG} with $\langle ADTRGE \rangle = "1"$.

Figure 3.12.3 Register for AD Converter

		7	6	5	4	3	2	1	0
ADREG04	Bit symbol	ADR01	ADR00						ADR0RF
(02A0H)	Read/Write	F	र						R
	After reset	Unde	fined	/					0
	Function	Stores low	er 2 bits of						AD conversion
		AD conver	sion result						data storage
									flag
									1: Conversion
									result stored
			AD Con	version Da	ata Upper	Register 0	/4		
		7	6	5	4	3	2	1	0
ADREG04H	Bit symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
(02A1H)	Read/Write				F	र			
	After reset				Unde	fined			
	Function			Stores u	upper 8 bits A	AD conversio	n result.		
			AD Con	version Da	ata Lower	Register 1	/5		
		7	6	5	4	3	2	1	0
ADREG15L	Bit symbol	ADR11	ADR10						ADR1RF
(02A2H)	Read/Write	F	१						R
	After reset	Unde	fined						0
	Function	Stores low	er 2 bits of						AD conversion
		AD conver	sion result						result flag
									1: Conversion
									result stored
			AD Con	version Da	ata Upper	Register 1	/5		
		7	6	5	4	3	2	1	0
	Bit symbol	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
(02A3H)	Read/Write				ſ	२	•	•	<u> </u>
	After reset				Unde	efined			
	Function			Stores up	oper 8 bits of	AD conversi	on result.		
www.DataSheet4U.com									
	Channely		98	7 6 5	4 3	2 1 0)		
	Conversion r	esult							
			ADREGxH	¥		↓		ADF	₹EGxL
			76	5 4 3	2 1 0	76	54	3 2 1	0
									;
								γ	
				•	Bits 5 to 1 a	re always rea	nd as "1".		
				•	Bit0 is the Al	D conversion	data storage	e flag <adr< td=""><td>xRF>. When the</td></adr<>	xRF>. When the
					the registers	(ADREGxH	, ADREGxL)	is read, the	flag is cleared to
					"0".		,		2

AD Conversion Data Lower Register 0/4

Figure 3.12.4 Register for AD Converter

		7	6	5	4	3	2	1	0
	Bit symbol	ADR21	ADR20						ADR2RF
(02A4H)	Read/Write	F	۰	\sim	\sim	\sim	\sim	\sim	R
(02/11/)	After reset	Unde	efined	\sim	\sim	\sim	\sim	\sim	0
	Function	Stores lower 2 bits of AD conversion result.							AD conversion data storage flag 1: Conversion result stored
			AD Con	version Da	ata Upper	Register 2	2/6		
		7	6	5	4	3	2	1	0
	Bit symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
(02A5H)	Read/Write				F	२			
· · · · · · · · · · · · · · · · · · ·	After reset				Unde	efined			
	Function			Stores up	oper 8 bits of	AD conversi	on result.		
		L	AD Con	version D:	ata Lower	Register 3	3/7		
		7	6	5	4	3	2	1	0
	Bit symbol	ADR31	ADR30		· ·	<u> </u>			
ADREG37L	Read/Write	F	2						R
(02/(01))	After reset	Unde	fined						0
Function		Stores low Ad conver	ver 2 bits of sion result.						AD cnversion data storage flag 1: Conversion result stored
			AD Conv	version Re	sult Upper	Register	3/7		
		7	6	5	4	3	2	1	0
	Bit symbol	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
(02A7H)	Read/Write				F	२			
(02/11/1)	After reset				Unde	fined			
	Function			Stores up	oper 8 bits of	AD conversi	on result.		
www.DataSheet4U.com	Channel x Conversion re	P 8 7 6 5 4 3 2 1 0 result							RECYL
			76	5 4 3	2 1 0	7 6	5 4	<u>3 2 1</u>	0

AD Conversion Result Lower Register 2/6



"0".

3.12.2 Operation

(1) Analog reference voltage

A high-level analog reference voltage is applied to the AVCC pin; a low-level analog reference voltage is applied to the AVSS pin. To perform AD conversion, the reference voltage as the difference between AVCC and AVSS, is divided by 1024 using string resistance. The result of the division is then compared with the analog input voltage.

To turn off the switch between AVCC and AVSS, write "0" to ADMOD1<VREFON> in AD mode control register 1. To start AD conversion in the OFF state, first write "1" to ADMOD1<VREFON>, wait 3 μ s until the internal reference voltage stabilizes (this is not related to fc), then set ADMOD0<ADS> to "1".

(2) Analog input channel selection

The analog input channel selection varies depends on the operation mode of the AD converter.

- In analog input channel fixed mode (ADMOD0<SCAN> = "0") Setting ADMOD1<ADCH2:0> selects one of the analog input pins AN0 to AN3 as the input channel.
- In analog input channel scan mode (ADMOD0<SCAN> = "1") Setting ADMOD1<ADCH2:0> selects one of the 4 scan modes.

Table 3.12.1 Illustrates analog input channel selection in each operation mode.

After Reset, ADMOD0<SCAN> = "0" and ADMOD1<ADCH2:0> = "000". Thus pin AN0 is selected as the fixed input channel. Pins not used as analog input channels can be used as standard input port pins.

<adch2:0></adch2:0>	Channel Fixed <scan> = "0"</scan>	Channel Scan <scan> = "1"</scan>
000	AN0	AN0
001	AN1	$AN0 \rightarrow AN1$
010	AN2	$AN0 \to AN1 \to AN2$
011	AN3	$\text{AN0} \rightarrow \text{AN1} \rightarrow \text{AN2} \rightarrow \text{AN3}$
100		
101	Don	't select
110	Don	1 1 301001.
111		

Table 3.12.1 Analog Input Channel Selection

(3) Starting AD conversion

To start AD conversion, write "1" to ADMOD0<ADS> in AD mode control register 0, or ADMOD1<ADTRGE> in AD mode control register 1 and input falling edge on ADTRG pin. When AD conversion starts, the AD conversion busy flag ADMOD0<ADBF> will be set to "1", indicating that AD conversion is in progress.

Writing "1" to ADMOD0<ADS> during AD conversion restarts conversion. At that time, to determine whether the AD conversion results have been preserved, check the value of the conversion data storage flag ADREGxL<ADRxRF>.

During AD conversion, a falling edge input on the $\overline{\text{ADTRG}}$ pin will be ignored.

(4) AD conversion modes and the AD conversion end interrupt

The 4 AD conversion modes are:

- Channel fixed single conversion mode
- Channel scan single conversion mode
- Channel fixed repeat conversion mode
- Channel scan repeat conversion mode

The ADMOD0<REPEAT> and ADMOD0<SCAN> settings in AD mode control register 0 determine the AD mode setting.

Completion of AD conversion triggers an AD conversion end interrupt request INTAD. Also, ADMOD0<EOCF> will be set to "1" to indicate that AD conversion has been completed.

a. Channel fixed single conversion mode

Setting ADMOD0<REPEAT> and ADMOD0<SCAN> to "00" selects channel fixed single conversion mode.

In this mode, data on one specified channel is converted once only. When the conversion has been completed, the ADMOD0<EOCF> flag is set to "1", ADMOD0<ADBF> is cleared to "0", and an INTAD interrupt request is generated.

b. Channel scan single conversion mode

Setting ADMOD0<REPEAT> and ADMOD0<SCAN> to "01" selects channel scan single conversion mode.

In this mode, data on the specified scan channels is converted once only. When scan conversion has been completed, ADMOD0<EOCF> is set to "1", ADMOD0<ADBF> is cleared to "0", and an INTAD interrupt request is generated.

www.DataSheet4U.com

c. Channel fixed repeat conversion mode

Setting ADMOD0<REPEAT> and ADMOD0<SCAN> to "10" selects channel fixed repeat conversion mode.

In this mode, data on one specified channel is converted repeatedly. When conversion has been completed, ADMODO<EOCF> is set to "1" and ADMODO<ADBF> is not cleared to "0" but held "1". INTAD interrupt request generation timing is determined by the setting of ADMODO<ITMO>.

Clearing <ITM0> to "0" generates an interrupt request every time an AD conversion is completed.

Setting <ITM0> to "1" generates an interrupt request on completion of every fourth conversion.

d. Channel scaen repeat conversion mode

Setting ADMOD0<REPEAT> and ADMOD0<SCAN> to "11" selects channel scan repeat conversion mode.

In this mode, data on the specified scan channels is converted repeatedly. When each scan conversion has been completed, ADMOD0<EOCF> is set to "1" and an INTAD interrupt request is generated. ADMOD0<ADBF> is not cleared to "0" but held "1".

To stop conversion in a repeat conversion mode (e.g., in cases c. and d.), program a "0" to ADMOD0<REPEAT>. After the current conversion has been completed, the repeat conversion mode terminates and ADMOD0<ADBF> is cleared to "0".

Switching to a halt state (IDLE2 mode with ADMOD1<I2AD> cleared to "0", IDLE1 mode or STOP mode) immediately stops operation of the AD converter even when AD conversion is still in progress. In repeat conversion modes (e.g., in cases c. and d.), when the halt is released, conversion restarts from the beginning. In single conversion modes (e.g., in cases a. and b.), conversion does not restart when the halt is released (The converter remains stopped).

Table 3.12.2 shows the relationship between the AD conversion modes and interrupt requests.

Mode	Generation of	ADMOD0			
Mode	Interrupt Request	<itm0></itm0>	<repeat></repeat>	<scan></scan>	
Channel fixed single conversion mode	After completion of conversion	Х	0	0	
Channel scan single conversion mode	After completion of scan conversion	Х	0	1	
Channel fixed repeat	Every conversion	0	1	0	
conversion mode	Every forth conversion	1	I	0	
Cannel scan repeat conversion mode	After completion of every scan conversion	Х	1	1	

Table 3.12.2 Relationship between the AD Conversion Modes and Interrupt Requests AD

X: Don't care

(5) AD conversion time

84 states (10.5 μs @ fFPH = 16MHz) are required for the AD conversion for one channel.

(6) Storing and reading the results of AD conversion

The AD conversion data upper and lower registers (ADREG04H/L to ADREG37H/L) store the AD conversion results. (ADREG04H/L to ADREG37H/L are read-only registers.)

In channel fixed repeat conversion mode with ADMODO<ITMO> = "1", the conversion results are stored successively in registers ADREG04H/L to ADREG37H/L. In other modes, the AN0, AN1, AN2 and AN3 conversion results are stored in ADREG04H/L, ADREG15H/L, ADREG26H/L and ADREG37H/L respectively.

Table 3.12.3 shows the correspondence between the analog input channels and the registers that are used to hold the results of AD conversion.

Table 3.12.3 Correspondence between Analog Input Channel and AD Conversion Result Register

	AD Conversion Result Register				
Analog Input Channel (Port 5)	Conversion Mode Other	Channel Fixed Repeat Conversion Mode			
	Than Right	(Every forth conversion)			
AN0	ADREG04H/L	ADREG04H/L			
AN1	ADREG15H/L	ADREG15H/L			
AN2	ADREG26H/L	ADREG26H/L			
AN3	ADREG37H/L	ADREG37H/L			

www.DataSheet4U.com

<ADRxRF> "bit0" of the AD conversion data lower register is used as the AD conversion data storage flag. The storage flag indicates whether the AD conversion result register has been read or not. When a conversion result is stored in the AD conversion result register, the flag is set to "1". When either of the AD conversion result registers (ADREGxH or ADREGxL) is read, the flag is cleared to "0".

Reading the AD conversion result also clears the AD conversion end flag ADMOD0<EOCF> to "0".

Example:

a. Convert the analog input voltage on the AN3 pin and write the result, to memory address 1800H using the AD interrupt (INTAD) processing routine.

Setting of r	nain routine	
_	76543210	
INTE0AD	← X 1 0 0	Enable INTAD and set it to interrupt level 4.
ADMOD1	← 1 1 X X 0 0 1 1	Set pin AN3 to the analog input channel.
_ADMOD0	$\leftarrow X X 0 0 0 0 0 1$	Start conversion in channel fixed single conversion mode.
Interrupt ro	utine processing example	
[WA	← ADREG37	Read value of ADREG37L, ADREG37H to general purpose register WA (16-bit).
WA	>>6	Shift contents read into WA six times to right and zero-fill upper bits.
(1800H)	\leftarrow WA	Write contents of WA to memory address 1800H.

b. Converts repeatedly the analog input voltages on the three pins AN0, AN1 and AN2, using channel scan repeat conversion mode.

[INTE0AD	\leftarrow X 0 0 0 – – – –	Disable INTAD.
ADMOD1	← 1 1 X X 0 0 1 0	Set pins AN0 to AN2 to be the analog input channels.
ADMOD0	← X X 0 0 0 1 1 1	Start conversion in channel scan repeat conversion mode.

X: Don't care, -: No change

3.13 Watchdog Timer (Runaway detection timer)

The TMP91CP27 features a watchdog timer for detecting runaway.

The watchdog timer (WDT) is used to return the CPU to 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. Connecting the watchdog timer out to the reset pin internally forces a reset.

3.13.1 Configuration

Figure 3.13.1 is a block diagram of watchdog timer.



Figure 3.13.1 Block Diagram of Watchdog Timer

Note: It needs to care designing the machine set, because watchdog timer can't operate completely by external noise.

The watchdog timer consists of a 22-stage binary counter which uses the system clock (fSYS) as the input clock. The binary counter can output fSYS/2¹⁵, fSYS/2¹⁷, fSYS/2¹⁹ and fSYS/2²¹. Selecting one of the outputs using WDMOD<WDTP1:0> generates a watchdog interrupt and outputs watchdog timer out when an overflow occurs as shown in Figure 3.12.2.



Figure 3.13.2 Normal Mode

The runaway detection result can also be connected to RESET pin internally.

In this case, the reset time will be between 22 and 29 states (44 to $58 \ \mu s$ at fOSCH = 16 MHz, fFPH = 1 MHz) as shown Figure 3.13.3. Also, system clock fSYS (1 cycle = 1 state) which generated clock by dividing it into 2, that clock fFPH divide clock fOSCH high-frequency oscillator into 16 is used to resetting.





3.13.2 Control Register

The watchdog timer WDT is controlled by two controls registers WDMOD and WDCR.

- (1) Watchdog timer mode register (WDMOD)
- a. Setting the detection time for the watchdog timer in <WDTP1:0>

This 2-bit register is used for setting the watchdog timer interrupt time used when detecting runaway. During resetting, this register is initialized to WDMOD<WDTP1:0> = "00".

The detection times for WDT are shown in Figure 3.13.4.

b. Watchdog timer enable/disable control register <WDTE>

During resetting, WDMOD<WDTE> is initialized to "1", enabling the watchdog timer.

To disable the watchdog timer, it is necessary to set this bit to "0" and to write the disable code (B1H) to 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 the watchdog timer from the disabled state to the enabled state merely by setting <WDTE> to "1".

c. Watchdog timer out reset connection <RESCR>

This register is used to connect the output of the watchdog timer with the RESET terminal internally. Since WDMOD<RESCR>is initialized to "0" on 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 the binary counter for the watchdog timer.

• Disable control

Disable control the watchdog timer can be disabled by clearing WDMOD<WDTE> to "0" and then writing the disable code (B1H) to the WDCR register.

WDMOD \leftarrow 0 - - 0 0 - - 0Clear WDMOD<WDTE> to "0".WDCR \leftarrow 1 0 1 1 0 0 0 1Write the disable code (B1H).

• Enable control

Set WDMOD<WDTE> to "1".

• Watchdog timer clear control

To clear the binary counter and cause counting to resume, write the clear code (4EH) to the WDCR register.

WDCR \leftarrow 0 1 0 0 1 1 1 0 Write the clear code (4EH).



www.DataSheet4U.com

→ Watchdog timer enable/disable control

0	Disable
1	Enable

Figure 3.13.4 Watchdog Timer Mode Register



Figure 3.13.5 Watchdog Timer Control Register

3.13.3 Operation

The watchdog timer generates an INTWD interrupt when the detection time set in the WDMOD<WDTP1:0> has elapsed. The binary counter for the watchdog timer must be cleared to "0" by software (Instruction) before an INTWD interrupt will be generated. If the CPU malfunctions (e.g., if runaway occurs) due to causes such as noise and does not execute the instruction used to clear the binary counter, the binary counter will overflow and an INTWD interrupt will be generated. The CPU will detect malfunction (Runaway) due to the INTWD interrupt and in this case it is possible to return the CPU to normal operation by means of an anti-malfunction program.

The watchdog timer works immediately after reset.

The watchdog timer does not operate in IDLE1 or STOP mode. When the device is in IDLE2 mode, the operation of WDT depends on the WDMOD<I2WDT> setting. Ensure that WDMOD<I2WDT> is set before the device enters IDLE2 mode.

Example:

a.	Clear binary counter.
	WDCR \leftarrow 0 1 0 0 1 1 1 0 Write the clear code (4EH).
b.	Set watchdog timer detection time to 2 ¹⁷ /fsys.
	WDMOD \leftarrow 1 0 1 0 0 0
c.	Disable watchdog timer.
	WDMOD $\leftarrow 0 0 0 0$ Clear <wdte> to "0".</wdte>
	WDCR \leftarrow 1 0 1 1 0 0 0 1 Write the disable code (B1H).

3.14 Timer for Real Time Clock (RTC)

The TMP91CP27 includes a timer that is used for a clock operation.

An interrupt (INTRTC) can be generated each 0.0625 [s] or 0.125 [s] or 0.25 [s] or 0.50 [s] by using a low frequency clock of 32.768 kHz. A clock function can be easily used.

A timer for real time clock can operate in all modes in which a low-frequency oscillation is operated.

In addition, INTRTC can return from each standby mode except STOP mode.



Figure 3.14.1 Block Diagram for Real Time Clock

The timer for real time clock is controlled by the real time clock control register (RTCCR) as shown in Figure 3.14.2.

RTCCR	
(0310H)	

		1	6	5	4	3	2	1	0
D	Bit symbol	Ι	/	/	/	/	RTCSEL1	RTCSEL0	RTCRUN
ĸ	Read/Write	R/W		/	/	/	R/	W	R/W
(ר	After reset	0		/	/	/	0	0	0
	Function	Write "0".					00: 2 ¹⁴ /fs 01: 2 ¹³ /fs 10: 2 ¹² /fs 11: 2 ¹¹ /fs		0: Stop & clear 1: Count

www.DataSheet4U.com



(13 –	(13 - 32.700 KHz)				
00	0.50 s				
01	0.25 s				
10	0.125 s				
11	0.0625 s				

Figure 3.14.2 Real Time Clock Control Register

4. Electrical Characteristics

4.1 Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit	
Power supply voltage	Vcc	-0.5 to 4.0	V	
Input voltage	VIN	-0.5 to Vcc + 0.5	v	
Output current (1 pin)	IOL	2		
Output current (1 pin)	IOH	-2	m ^	
Output current (Total)	ΣΙΟΓ	80	IIIA	
Output current (Total)	ΣΙΟΗ	-80		
Power dissipation (Ta = 85°C)	PD	600	mW	
Soldering temperature (10 s)	TSOLDER	260		
Storage temperature	TSTG	-65 to 150	°C	
Operation temperature	TOPR	-40 to 85		

Note: The absolute maximum ratings are rated values that 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 that include this device, ensure that no absolute maximum rating value will ever be exceeded.

	Parameter	Symbol	Condit	ion	Min	Typ.(Note)	Max	Unit
Power supply voltage			fc = 4 to 27 MHz	fs = 30 to	2.7			
	VSS = DVSS = 0V	VCC	fc = 2 to 10 MHz	34 kHz	1.8		3.6	V
	P00 to P17	VII	$Vcc \ge 2.7 V$				0.6	
	(AD0 to AD15)	VIL	Vcc < 2.7 V				0.2 Vcc	
e	P20 to P07 (Except P62)	\/II_1	$Vcc \geq 2.7 \ V$				0.3 Vcc	
oltaç	F2010 F97 (Except F03)	VILI	Vcc < 2.7 V				0.2 Vcc	
N VG	RESET, NMI		$Vcc \geq 2.7 \ V$				0.25 Vcc	
t lo	P63 (INT0)	VILZ	Vcc < 2.7 V		-0.3		0.15 Vcc	V
ndu	AMO and AMI	V/II 0	$Vcc \ge 2.7 V$				0.3	
-	AMU and AMI	VIL3	Vcc < 2.7 V				0.3	
	X1	VIL4	$Vcc \ge 2.7 V$				0.2 Vcc	
			Vcc < 2.7 V				0.1 Vcc	
	P00 to P17	УШ	$Vcc \geq 2.7 \ V$		2.0			
	(AD0 to AD15)	VIII	Vcc < 2.7 V		0.7 Vcc			
ge	P20 to P97 (Except P63)	VIH1	$Vcc \ge 2.7 V$		0.7 Vcc			
olta			Vcc < 2.7 V		0.8 Vcc			
ہ طو	RESET, NMI,	1/11/0	$Vcc \ge 2.7 V$		0.75 Vcc			Ň
t hiç	P63 (INT0)	VINZ	Vcc < 2.7 V		0.85 Vcc		VCC + 0.3	v
ndu	AMO and AMI	\//11/2	$Vcc \ge 2.7 V$		Vcc - 0.3			
_	AMU and AMI	VIH3	Vcc < 2.7 V		Vcc - 0.3			
	VA	\/ILLA	$Vcc \ge 2.7 V$		0.8 Vcc			
		VIH4	Vcc < 2.7 V		0.9 Vcc			
Output	low voltage	VOI	IOL = 1.6mA	$Vcc \geq 2.7 \ V$			0.45	
Output	low voltage	VOL	IOL = 0.4 mA	Vcc < 2.7 V			0.15 Vcc	V
Output	high voltage	ЛОН	$IOH = -400 \ \mu A$	$Vcc \ge 2.7 V$	Vcc - 0.3			v
Culput	nigh vollage	vОП	$IOH = -200 \ \mu A$	Vcc < 2.7 V	0.8 Vcc			

4.2 DC Characteristics (1/2)

Note: Typical values are for when Ta = 25 $^\circ\text{C}$ and V_{CC} = 3.0 V uncles otherwise noted.

4.2 DC Characteristics (2/2)

Parameter	Symbol	Condition Min		Typ. (Note1)	Max	Unit
Input leakage current	ILI	$0.0 \leq V_{IN} \leq Vcc$		0.02	±5	
Output leakage current	ILO	$0.2 \leq V_{IN} \leq Vcc - 0.2$		0.05	±10	μA
Power down voltage (@STOP, RAM back up)	VSTOP	V IL2 = 0.2 Vcc, V IH2 = 0.8 Vcc	1.8		3.6	V
	DDOT	Vcc = 2.7 V to 3.6 V	100		400	
RESET pull-up resistor	RROI	$Vcc = 2 V \pm 10\%$	200		1000	kΩ
Pin capacitance	CIO	fc = 1 MHz			10	PF
Schmitt width		$Vcc \ge 2.7 V$	0.4	1.0		V
RESET, NMI, INTO	VIH	Vcc < 2.7 V	0.3	0.8		v
Drogrommoble pull up register	риц	Vcc = 2.7 V to 3.6 V	100	100 40		
Programmable pull-up resistor	ККП	$Vcc = 2 V \pm 10\%$	200		1000	kΩ
NORMAL (Note 2), (Note 3)		Vcc = 2.7 V to 3.6 V fc = 27 MHz		11.5 (10.8)	19.0	mA
IDLE2 (Note 3)				5.5 (4.8)	8.0	
IDLE1 (Note 3)				2.5 (1.8)	4.0	
NORMAL (Note 2), (Note 3)		$\sqrt{22} - 3 \sqrt{1 + 10\%}$		11.5 (10.8)	16.0	mA
IDLE2 (Note 3)		fc = 27 MHz		5.5 (4.8)	7.5	
IDLE1 (Note 3)				2.5 (1.8)	3.5	
NORMAL (Note 2), (Note 3)		$Vcc = 2 V \pm 10 \%$		3.5 (3.0)	5.0	
IDLE2 (Note 3)	1	fc = 10 MHz		2.0 (1.5)	3.0	mA
IDLE1 (Note 3)	ICC	(Typ. Vcc = 2.0 V)		0.9 (0.4)	1.8	
SLOW (Note 2)				14.5	30	
IDLE2		VCC = 2.7 V to 3.6 V		7.0	19	μΑ
IDLE1		15 - 52.700 KHZ		5.0	15	
SLOW (Note 2)	l	$Vcc = 2 V \pm 10 \%$		10	20	
IDLE2		fs = 32.768 kHz		5.0	13	μA
IDLE1		(Typ. Vcc = 2.0 V)		3.0	10	
STOP		Vcc = 1.8 V to 3.6 V		0.1	10	μA

Note 1: Typical values are for when Ta = 25° C and V_{CC} = 3.0 V unless otherwise noted. www.DataSheet4U.com

Note 2: Icc measurement conditions (NORMAL, SLOW):

All functions are operational; output pins are open and input pins are fixed.

Note 3: Power supply cuurent from AVCC pin is included in power supply current of VCC pin. Also, AVCC pin share with AD reference power supply in TMP91CP27. Therefore, it is included in power supply current of VCC pin that not only power supply current from AVCC pin but also current to ladder resitster. Insert of () is current value when VREF is Off.

4.3 AC Characteristics

(1) Vcc = 2.7 V to 3.6 V

No. Parameter		Symbol	Valiable		f _{FPH} = 2	27 MHz	Llnit
NO.	Falameter	Symbol	Min	Max	Min	Max	Onit
1	f _{FPH} period (=x)	t _{FPH}	37.0	31250	37.0		ns
2	A0 to A15 valid \rightarrow ALE falling	t _{AL}	0.5x - 6		12		ns
3	ALE falling \rightarrow A0 to A15 hold	t _{LA}	0.5x – 16		2		ns
4	ALE high pulse width	tLL	x - 20		17		ns
5	ALE falling $\rightarrow \overline{\text{RD}} / \overline{\text{WR}}$ falling	t _{LC}	0.5x – 14		4		ns
6	\overline{RD} rising $\rightarrow ALE$ rising	t _{CLR}	0.5x - 10		8		ns
7	\overline{WR} rising \rightarrow ALE rising	t _{CLW}	x – 10		27		ns
8	A0 to A15 vlalid $\rightarrow \overline{\text{RD}}$ / $\overline{\text{WR}}$ falling	t _{ACL}	x – 23		14		ns
9	A0 to A21 valid $\rightarrow \overline{\text{RD}} / \overline{\text{WR}}$ falling	t _{ACH}	1.5x – 26		29		ns
10	$\overline{\text{RD}}$ rising \rightarrow A0 to A21 hold	t _{CAR}	0.5x – 13		5		ns
11	$\overline{\text{WR}}$ rising \rightarrow A0 to A21 hold	t _{CAW}	x – 13		24		ns
12	A0 to A15 valid \rightarrow D0 to D15 input	t _{ADL}		3.0x - 38		73	ns
13	A0 to A21 valid \rightarrow D0 to D15 input	t _{ADH}		3.5x – 41		88	ns
14	$\overline{\text{RD}}$ falling \rightarrow D0 to D15 input	t _{RD}		2.0x - 30		44	ns
15	RD low puse width	t _{RR}	2.0x - 15		59		ns
16	$\overline{\text{RD}}$ rising \rightarrow D0 to D15 hold	t _{HR}	0		0		ns
17	$\overline{\text{RD}}$ rising \rightarrow A0 to A15 output	t _{RAE}	x – 15		22		ns
18	WR low pulse width	t _{WW}	1.5x – 15		40		ns
19	D0 to D15 valid $\rightarrow \overline{\text{WR}}$ rising	t _{DW}	1.5x – 35		20		ns
20	$\overline{\text{WR}}$ rising \rightarrow D0 to D15 hold	t _{WD}	x – 25		12		ns
21	A0 to A21 valid \rightarrow Port input	t _{APH}		3.5x - 89		40	ns
22	A0 to A21 valid \rightarrow Port hold	t _{APH2}	3.5x		129		ns
23	A0 to A21 valid \rightarrow Port valid	t _{AP}		3.5x + 80		209	ns

AC measurement conditions

+ Output level: High 0.7 \times Vcc/Low 0.3 \times V_{CC}, C_L = 50 pF

- Input level: High 0.9 \times Vcc/Low 0.1 \times V_{CC}

Note: Symbol [x] in the above table means the period of clock f_{FPH} . It's half period the system clock www.DataSheet4U.com f_{SYS} for CPU core.

The period of clock f_{FPH} depends on the clock gear setting or the selection of high/low oscillator frequency.

(2) 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{RD} and \overline{CS} are not enabled. Therefore, the above waveform diagram should be regarded as depicting internal operation. Please also note that the timing and AC characteristics of port input/output shown above are typical representation. For details, contact your local Toshiba sales representative.

(3) Write cycle



www.DataSheet4U.com

Note: Since the CPU accesses the internal area to write data to a port, the control signals of external pins such as \overline{WR} and \overline{CS} are not enabled. Therefore, the above waveform diagram should be regarded as depicting internal operation. Please also note that the timing and AC characteristics of port input/output shown above are typical representation. For details, contact your local Toshiba sales representative.

4.4 AD Conversion Characteristics

AVCC = VCC, AVSS = VSS									
Parameter	Symbol	Condition	Min	Тур.	Max	Unit			
Analog intput voltage	VAIN		AVSS		AVCC	V			
Error		V_{CC} = 2.7 V to 3.6 V		±1.0	±4.0	ISB			
(Not including quantization errors)	_	$V_{CC}=2~V\pm10\%$		±1.0	±4.0	LSB			

Note 1:1 LSB = (AVCC - AVSS)/1024 [V]

Note 2: Minimum operation frequency:

The operaion of AD converter is guranteed only using fc (High frequency oscillator). fs (Low frequency oscillator) is not guranteed. But When frequency of clock selected by clock gear is more than and eqaull 4 MHz in using fc, it is guranteed ($f_{FPH} \ge 4 \text{ MHz}$).

Note 3: The value for Icc (Current of VCC pin) includes the current which flows through the AVCC pin.

4.5 Serial Channel Timing (I/O interface mode)

Parameter	Symbol	Valiab	10 MHz		27 MHz		Linit	
i alametei	Symbol	Min	Max	Min	Max	Min	Max	Unit
SCLK period	tscy	16X		1.6		0.59		μS
Output data → SCLK rising/falling	toss	t _{SCY} /2 – 4X – 110 (V _{CC} = 2.7 V to 3.6 V)		290		38		
		$t_{SCY}/2 - 4X - 180$ (V _{CC} = 2 V ± 10%)		220		Ι		ns
SCLK rising/falling \rightarrow Output data hold	t _{OHS}	$t_{SCY}/2 + 2X + 0$		1000		370		ns
SCLK rising/falling \rightarrow Input data hold	t _{HSR}	3X + 10		310		121		ns
SCLK rising/falling \rightarrow Valid data input	tSRD		$t_{\text{SCY}} - 0$		1600		592	ns
Valid data input \rightarrow SCLK rising/falling	t _{RDS}	0		0		0		ns

(1) SCLK input mode

(2) SCLK ouptut mode

Paramotor	Symbol	Valial	10 MHz		27 MHz		Linit	
Falameter	Symbol	Min	Max	Min	Max	Min	Max	Unit
SCLK period	tSCY	16X	8192X	1.6	819	0.59	303	μS
Output data \rightarrow SCLK rising/falling	toss	t _{SCY} /2 - 40		760		256		ns
SCLK rising/falling \rightarrow Output data hold	t _{OHS}	t _{SCY} /2 - 40		760		256		ns
SCLK rising/falling \rightarrow Input data hold	^t HSR	0		0		0		ns
SCLK rising/falling \rightarrow Valid data input	t _{SRD}		t _{SCY} – 1X – 180		1320		375	ns
Valid data input \rightarrow SCLK rising/falling	t _{RDS}	1X + 180		280		217		ns

www.DataSheet4U.cor

Note 1: SCLK rising/falling: The rising edge is used in SCLK rising mode.

The falling edge is used in SCLK falling mode.

Note 2: 27 MHz and 10 MHz values are calculated from $t_{SCY} = 16X$ case.

Note 3: Symbol [x] in the above table means the period of clock f_{FPH} . It's half period the system clock f_{SYS} for CPU core.

The period of clock f_{FPH} depends on the clock gear setting or the selection of high/low oscillator frequency.



4.6 Event Counter (TA0IN, TA4IN, TB0IN0 and TB0IN1)

Parameter	Symbol	Valiable		10 MHz		27 MHz		Linit
Falameter	Symbol	Min	Max	Min	Max	Min	Max	Onit
Clock period	t _{VCK}	8X + 100		900		396		ns
Clock low level pulse width	t _{VCKL}	4X + 40		440		188		ns
Clock high level pulse width	t _{VCKH}	4X + 40		440		188		ns

4.7 Interrupt and Capture

(1) $\overline{\text{NMI}}$ and INT0 Interrupts

Parameter	Symbol	Variable		10 MHz		27 MHz		Unit	
	0,	Min	Max	Min	Max	Min	Max		
MMI and INT0 low level pulse width	t _{INTAL}	4X + 40		440		188		ns	
MI and INT0 high level pulse width	^t INTAH	4X + 40		440		188		ns	

(2) INT5 and INT6 interrupts, capture

INT5 and INT6 input pulse width depend on the system clock selection and clock selection for prescaler. Below table show pulse width of each operation clock.

System Clock	Clock Selection	t _{IN}	ITBL	t		
Selection	for Prescaler	(INT5 and INT6 lo	w level pulse width)	(INT5 and INT6	Linit	
SYSCR1	SYSCR0	Valiable	f _{FPH} = 27 MHz	Valiable	f _{FPH} = 27MHz	Unit
<sysck></sysck>	<prck1:0></prck1:0>	Min	Min	Min	Min	
0 (fa)	00 (f _{FPH})	8X + 100	396	8X + 100	396	ns
0 (IC)	10 (fc/16)	128Xc + 0.1	4.8	128Xc + 0.1	4.8	
1 (fs)	00 (f _{FPH})	8X + 0.1	244.3	8X + 0.1	244.3	μs

www.DataSheet4U.comNote 1: "Xc" shows period of clock fc in high frequency oscillator.

Note 2: Symbol [x] in the above table means the period of clock f_{FPH} . It's half period the system clock f_{SYS} for CPU core.

The period of clock f_{FPH} depends on the clock gear setting or the selection of high/low oscillator frequency.
4.8 Recommended Oscillation Circuit

TMP91CP27 has been evaluated by murata manufacturing Co., Ltd. Please refer to murata manufacturing Co., Ltd.

- Note 1: Total loads value of oscillator is sum of external loads (C1 and C2) and floating loads of actual assemble board. There is a possibility of miss-operating. When designing board, it should design minimum length pattern around oscillator. And we recommend that oscillator evaluation try on your actual using board.
- Note 2: The product numbers and specifications of the resonators by murata manufacturing Co., Ltd. are subject to change. For up-to-date information, please refer to the following URL:

http://www.murata.co.jp/search/index.html

(1) Connection example



High-frequency oscillation connection



Low-frequency oscillation connection

(2) Recommended ceramic oscillator

Oscillation		Р	arameter o	of Element	S	Running	Condition
Frequency [MHz]	Item of Oscillator	C1 [pF]	C2 [pF]	Rf [Ω]	Rd [Ω]	Voltage of Power [V]	Temperature [°C]
4.0	CSTLS4M00G56-B0 (CSTS0400MG06)	(47)	(47)	Open	0		
8.0	CSTLS8M00G56-B0 (CSTS0800MG06)	(47)	(47)	Open	0		
10.0	CSTLS10M0G53-B0 (CSTS1000MG03)	(15)	(15)	Open	0		
12.0	CSTLA12M0T55-B0 (CST12. 0MTW)	(30)	(30)	Open	0	2.7~3.6	-40~85
16.0	CSTLS16M0X51-B0 (New and old is same product No.)	(5)	(5)	Open	0		
20.0	CSTLS20M0X51-B0 (New and old is same product No.)	(5)	(5)	Open	0		
27.0	CSALS27M0X51-B0 (New and old is same product No.)	Open	Open	10 k	0		

www.DataSheet4U.com

Oppillation		P	arameter	of Element	Running Condition		
Frequency [MHz]	Item of Oscillator	C1 [pF]	C2 [pF]	Rf [Ω]	Rd [Ω]	Voltage of Power [V]	Temperature [°C]
4.0	CSTLS4M00G56U-B0 (CSTS0400MG06-951)	(47)	(47)	Open	0		
8.0	CSTLS8M00G56U-B0 (CSTS0800MG06-951)	(47)	(47)	Open	0	1.8~2.2	-40~85
10.0	CSTLS10M0G53U-B0 (CSTS1000MG03-951)	(15)	(15)	Open	0		

Note: In CST*** type oscillator, capacitance C1, C2 is builtin.

5. Table of SFRs

The SFRs (Special function registers) include the I/O ports and peripheral control registers allocated to the 4-Kbyte address space from 000000H to 000FFFH.

- (1) I/O port
- (2) I/O port control
- (3) Interrupt control
- (4) Chip select/wait control
- (5) Clock control
- (6) 8-bit timer control
- (7) 16-bit timer control
- (8) UART/serial channel control
- (9) I²C bus/serial channel control
- (10) AD converter control
- (11) Watchdog timer control
- (12) RTC (Real time clock) control

Table layout

Symbol	Name	Address	7	6	$\left(\right)$		1	0	
)			→ Bit symbol → Read/Write → Initial value after reset → Remarks
					·				

Note: "Prohibit RMW" in the table means that you cannot use RMW instructions on these registers.

www.DataSheet4U.com Example: When setting only bit0 of the register POCR to "1", the instruction "SET 0, (0002H)" cannot be used. The LD (Transfer) instruction must be used to write all eight bits.

Read/Write

R/W: Both read and write are possible.

- R: Only read is possible
- W: Only write is possible
- W*: Both read and write are possible (when this bit is read as 1.)
- Prohibit RMW: Read-modify-write instructions are prohibited. (The EX, ADD, ADC, BUS, SBC, INC, DEC, AND, OR, XOR, STCF, RES, SET, CHG, TSET, RLC, RRC, RL, RR, SLA, SRA, SLL, SRL, RLD and RRD instruction are read-modify-write instructions.)

*R/W: Read-modify-write instructions are prohibited when controlling the pull-up resistor.

[1] Port	
Address	Name
0000H	P0
1H	P1
2H	P0CR
3H	
4H	P1CR
5H	P1FC
6H	P2
7H	P3
8H	P2CR
9H	P2FC
AH	P3CR
BH	P3FC
CH	P4
DH	P5
EH	P4CR
FH	P4FC

Table 5.1 Address Map for SFRs

Address	Name
0010H	
1H	
2H	P6
3H	P7
4H	P6CR
5H	P6FC
6H	P7CR
7H	P7FC
8H	P8
9H	P9
AH	P8CR
BH	P8FC
CH	P9CR
DH	P9FC
EH	
FH	

Address	Name
0020H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
СН	
DH	
EH	
FH	ODE

[2] INTC

Address	Name
0080H	DMA0V
1H	DMA1V
2H	DMA2V
3H	DMA3V
4H	
5H	
6H	
7H	
8H	INTCLR
9H	DMAR
AH	DMAB
BH	
CH	IIMC
DH	
EH	
FH	

Address	Name
0090H	INTE0AD
1H	
2H	
3H	INTE56
4H	
5H	INTETA01
6H	INTETA23
7H	INTETA45
8H	
9H	INTETB0
AH	
BH	INTETB01V
CH	INTES0
DH	INTES1
EH	INTES2RTC
FH	

Address	Name
00A0H	INTETC01
1H	INTETC23
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
СН	
DH	
EH	
FH	

www.DataSheet4U.com

[3] CS/WAIT	
Address	Name
00C0H	B0CS
1H	B1CS
2H	B2CS
3H	B3CS
4H	
5H	
6H	
7H	BEXCS
8H	MSAR0
9H	MAMR0
AH	MSAR1
BH	MAMR1
CH	MSAR2
DH	MAMR2
EH	MSAR3
FH	MAMR3

Note: Do not access to the unnamed addresses, e.g., addresses to which no register has been allocated.

[4] CGEAR

4] CGEAR	
Address	Name
00E0H	SYSCR0
1H	SYSCR1
2H	SYSCR2
3H	EMCCR0
4H	EMCCR1
5H	
6H	
7H	
8H	
9H	
AH	
BH	
СН	
DH	
EH	
FH	

Address	Name
00F0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Table 5.2 Address Map for SFRs

[5] TMRA

Address	Name
0100H	TA01RUN
1H	
2H	TAOREG
3H	TA1REG
4H	TA01MOD
5H	TA1FFCR
6H	
7H	
8H	TA23RUN
9H	
AH	TA2REG
BH	TA3REG
CH	TA23MOD
DH	TA3FFCR
EH	
FH	

Address	Name
0110H	TA45RUN
1H	
2H	TA4REG
3H	TA5REG
4H	TA45MOD
5H	TA5FFCR
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

www.DataSheet4U.com

[6] TMRB		_		
Address	Name		Address	Name
0180H	TBORUN		0190H	
1H			1H	
2H	TB0MOD		2H	
3H	TB0FFCR		3H	
4H			4H	
5H			5H	
6H			6H	
7H			7H	
8H	TB0RG0L		8H	
9H	TB0RG0H		9H	
AH	TB0RG1L		AH	
BH	TB0RG1H		BH	
CH	TB0CP0L		CH	
DH	TB0CP0H		DH	
EH	TB0CP1L		EH	
FH	TB0CP1H		FH	

Note: Do not access to the unnamed addresses, e.g., addresses to which no register has been allocated.

[7] UAK 1/310	
Address	Name
0200H	SC0BUF
1H	SC0CR
2H	SC0MOD0
3H	BR0CR
4H	BR0ADD
5H	SC0MOD1
6H	
7H	SIRCR
8H	SC1BUF
9H	SC1CR
AH	SC1MOD0
BH	BR1CR
CH	BR1ADD
DH	SC1MOD1
EH	
FH	

Table 5.3 Address Map for SFRs

)
Address	Name
0240H	SBI0CR1
1H	SBIODBR
2H	I2C0AR
3H	SBI0CR2/SBI0SR
4H	SBI0BR0
5H	SBI0BR1
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[9] 10-bit ADC

Address	Name
02A0H	ADREG04L
1H	ADREG04H
2H	ADREG15L
3H	ADREG15H
4H	ADREG26L
5H	ADREG26H
6H	ADREG37L
7H	ADREG37H
8H	
9H	
AH	
BH	
СН	
DH	
EH	
FH	

Address	Name
02B0H	ADMOD0
1H	ADMOD1
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

www.DataSheet4U.com

[10] WDT		[11] RTC	
Address	Name	Address	Name
0300H	WDMOD	0310H	RTCCR
1H	WDCR	1H	
2H		2H	
3H		3H	
4H		4H	
5H		5H	
6H		6H	
7H		7H	
8H		8H	
9H		9H	
AH		AH	
BH		BH	
СН		CH	
DH		DH	
EH		EH	
FH		FH	

Note: Do not access to the unnamed addresses, e.g., addresses to which no register has been allocated.

(1) I/O port

	Symbol	Name	Address	7	6	5	4	3	2	1	0		
				P07	P06	P05	P04	P03	P02	P01	P00		
	P0	Port 0	00H	R/W									
				Data from external port (Output latch register is undefined)									
				P17	P16	P15	P14	P13	P12	P11	P10		
	P1	Port 1	01H	R/W									
					Data f	rom external	port (Output	latch registe	er is cleared	to "0")			
						P25	P24	P23	P22	P21	P20		
	P2	Port 2	06H					R/	W				
				/		Data	a from exterr	al port (Outp	out latch regis	ster is set to	"1")		
									P32	P31	P30		
			0711		/	/	/	/		*R/W			
			07日						Data from				
									external	1	1		
	P3	Port 3	(Prohibit						port Note1				
			RMW)						Pull-up				
			,						resistor	_	_		
									0: OFF				
						/		/	1: ON				
			0CH (Prohibit				/		P42	P41	P40		
	5.4									*R/W			
	P4 P0	Port 4							Data from	n external po	ort Note1		
			RIVIVV)										
								DEO	0:	OFF 1: 0	N DEO		
	DE	Dort 5		$\langle \rangle$				P53	P52	P51	P50		
	P0	Port 5	UDH				/	K					
								DCO	Data from e	xternal port	DCO		
				\backslash			\backslash	P03	P02	P01	P60		
	P6	Port 6	12H						R/				
								(Out	Data ITOITI e	istor is set to	"1")		
				\backslash			P74	(Out		D71	P70		
							F / 4	F75	R/M	Γ/Ι	F70		
	P7	Port 7	13H					Data	from externa	Inort			
www.DataSheet	4U.com							Output lat	ch register is	set to "1")			
				\frown	\frown	\frown		P83	P82	P81	P80		
				\backslash	\sim	\sim	\backslash	1.00	R/	W	1.00		
	P8	Port 8	18H						Data from e	xternal port			
								(Out	put latch real	ister is set to	· "1")		
				P97	P96	P95	P94	P93	P92	P91	, P90		
	P9	Port 9	19H	R/W	R/W			R/	W				
				1	1	Data	a from exterr	al port (Out	out latch regis	ster is set to	"1")		
										,			

Note: Output latch is set to "1", and pull-up resistor is connected.

	(4)	no por	CONTROL							1			
	Symbol	Name	Address	7	6	5	4	3	2	1	0		
				P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C		
		Dort 0	02H	W									
	P0CR	Poil 0	(Prohibit	0	0	0	0	0	0	0	0		
		CONTION	RMW)				0: Input	1: Output					
				(Wh	en access to	external, be	ecome AD7 t	o AD0 and t	his register i	s cleared to	"0".)		
			0.411	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C		
	Port 1	Port 1	04H (Drahihit				V	V					
	PICK	control		0	0	0	0	0	0	0	0		
					< <refer column="" of="" p1fc="" to="">></refer>								
			0511	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F		
	DAEO	Port 1	05H				V	V					
	PIFC	function		0	0	0	0	0	0	0	0		
			RIVIVV)	Pí	IFC/P1CR =	00: Input po	ort, 01: Outpu	it port, 10: A	D8 to AD15,	11: A8 to A	15		
				/		P25C	P24C	P23C	P22C	P21C	P20C		
	DOOD	Port 2	08H		/			V	V				
	P2CR	CR control				0	0	0	0	0	0		
			RIVIVV)			< <refer column="" of="" p2fc="" to="">></refer>							
					/	P25F	P24F	P23F	P22F	P21F	P20F		
		Port 2	2 (Prohibit on RMW)		/			V	V				
	P2FC	function		\sim		0	0	0	0	0	0		
						P2FC/P2CF	R = 00: Input	port, 01: Out	put port, 10:	A0 to A5, 11	: A16 to A21		
						/			P32C				
			0AH	\sim		/		\sim	W				
	P3CR	Port 3	(Prohibit			/			0				
		control	RMW)						0: Input				
			,						1: Output				
				_		/			P32F	P31F	P30F		
			0BH	W			/			W			
	P3FC	Port 3	(Prohibit	0			/		0	0	0		
		function	RMW)	Write "0".					0: Port	0: Port	0: Port		
									1: HWR	1: WR	1: RD		
									P42C	P41C	P40C		
		Port 4	0EH	\sim		/		\sim		W			
www.DataSheet4	URAGR	control	(Prohibit			/			0	0	0		
			KIVIVV)						0:	Input 1: Out	put		
									P42F	P41F	P40F		
			0FH			/	//			W			
	P4FC	Port 4	(Prohibit	$\overline{}$	\sim			$\overline{}$	0	0	0		
		function	RMW)						0 [.] Port	0 [.] Port	0 [.] Port		
									1: CS2	1: CS1	1: CS0		

(2) I/O port control (1/2)

Note 1: When port 2 is used as address bus A21 to A16 or A5 to A0, set P2FC after set P2CR.

Note 2: "L" level is outputted from P30 pin also during reading internal area by setting P3<P30> to "0", set P3FC<P30F> to "1".

Note 3: When port 4 is used as chip select signal $\overline{CS0}$ to $\overline{CS2}$ set P4CR to "1" after set P4FC to "1".

		- I									
	Symbol	Name	Address	7	6	5	4	3	2	1	0
			4.41.1					P63C	P62C	P61C	P60C
	DECP	Port 6	I4⊓ (Prohihit						V	V	
	FUCI	control	(FTOHIDIC RMW)					0	0	0	0
			(WWV)	_	_				0: Input	1: Output	
								P63F	P62F	P61F	P60F
		Dort 6	15H						V	V	
	P6FC	function	(Prohibit					0	0	0	0
		Tunction	RMW)					0: Port	0: Port	0: Port	0: Port
								1: INT0	1: SCL	1: SDA/SO	1: SCK out
				/			P74C	P73C	P72C	P71C	P70C
	DZCD	Port 7	16H (Drohihit	/	/	/			W		
	FICK	control		/	/	/	0	0	0	0	0
								0: I	nput 1: Out	out	
							P74F		P72F	P71F	
			174	\backslash	/	\sim	W	\sim	W	W	
	P7FC	Port 7	I/II (Prohibit		/	\sim	0	\sim	0	0	
	1710	function	RMW)				0: Port		0: Port	0: Port	
			,				1: TA5OUT		1: TA3OUT	1: TA1OUT	
				/	/			P83C	P82C	P81C	P80C
		Port 8	1AH (Drohihit						V	V	
	POCK	control						0	0	0	0
			IXIVIVV)						0: Input	1: Output	
				/	/	/		P83F	P82F	P81F	P80F
			100		/				V	V	
	P8FC	Port 8	(Prohibit	/				0	0	0	0
	1010	function	RMW)					0: Port	0: Port	0: Port	0: Port
			,					1: TB0OUT1	1: TB0OUT0	1: INT6/	1: INT5/
										TB0IN1	TB0IN0
			1CH	P97C	P96C	P95C	P94C	P93C	P92C	P91C	P90C
	P9CR	Port 9	(Prohibit	W	W		1	W	1		
	1001	control	RMW)	1	1	0	0	0	0	0	0
			,	/	<u> </u>	I	0: Input	1: Output		<u> </u>	
						P95F		P93F	P92F		P90F
www.DataShor	t/II com	Port 9	1DH			W		W	W		W
www.DataShed	P9FC	function	(Prohibit			0		0	0		0
		lanouon	RMW)			0: Port		0: Port	0: Port		0: Port
						1: SCLK1		1: TXD1	1: SCLK0		1: TXD0
								ODE62	ODE61	ODE93	ODE90
	ODE	Open-drain	2EH					R/W	R/W	R/W	R/W
	ODL	enable	2111		/			0	0	0	0
								1: P62ODE	1: P610DE	1: P930DE	1: P900DE

I/O port control (2/2)

Note 1: External interrupt INT0:

Input enable is controlled by P6FC<P63F>. Level/edge selection and rising/falling selection is controlled by IIMC<I0LE, I0EDGE>.

Note 2: External interrupts INT5 and INT6:

Input enable is set by P8FC<P81F, P80F>. The setting of edge is controlled by TB0MOD.

- Note 3: When P70 and P73 is used as an input port, the input signal is inputted to 8bit-timer (TMRA0 and TMRA4) as TA0IN and TA4IN inputs.
- Note 4: When P91 and P94 is used as an input port, the input signal is inputted to SIO as serial receiving data RXD0 and RXD1.

· - /		-p	- (=: =;								
Symbol	Name	Address	7	6	5	4	3	2	1	0	
INTE0AD	Interrupt			INT	AD		INT0				
	enable		IADC	IADM2	IADM1	IADM0	I0C	10M2	I0M1	I0M0	
	INT0	90H	R		R/W		R		R/W		
	& AD		0	0	0	0	0	0	0	0	
			1: INTAD	Inter	rupt request	level	1: INT0	Inter	rupt request	level	
INTE56	Interrupt			IN.	Т6			IN	T5		
	enable		I6C	I6M2	I6M1	16M0	I5C	I5M2	I5M1	15M0	
	INT6/5	93H	R		R/W		R		R/W		
			0	0	0	0	0	0	0	0	
			1: INT6	Interru	put requese	t level	1: INT5	Interrupt request level			
INTETA01	INTETA01 Interrupt		INTTA1 (TMRA1)			INTTA0 (TMRA0)					
	enable	ble RA 95H	ITA1C	ITA1M2	ITA1M1	ITA1M0	ITA0C	ITA0M2	ITA0M1	ITA0M0	
	TMRA		R	R/W		R		R/W			
	1/0		0	0	0	0	0	0	0	0	
			1: INTTA1	Interrupt request level			1: INTTA0	Inter	rupt request	level	
INTETA23	Interrupt		INTTA3 (TMRA3)				INTTA2 (TMRA2)				
	enable		ITA3C	ITA3M2	ITA3M1	ITA3M0	ITA2C	ITA2M2	ITA2M1	ITA2M0	
	TMRA	96H	R		R/W		R		R/W		
	3/2		0	0	0	0	0	0	0	0	
			1: INTTA3	Inter	rupt request	level	1: INTTA2	Inter	rupt request	level	
INTETA45	Interrupt			INTTA5 ((TMRA5)			INTTA4	(TMRA4)		
	enable		ITA5C	ITA5M2	ITA5M1	ITA5M0	ITA4C	ITA4M2	ITA4M1	ITA4M0	
	TMRA	97H	R		R/W		R		R/W		
	5/4		0	0	0	0	0	0	0	0	
			1: INTTA5	Inter	rupt request	level	1: INTTA4	Inter	rupt request	level	

(3) Interrupt control (1/3)

				-				1			
	Symbol	Name	Address	7	6	5	4	3	2	1	0
					INTTB01	(TMRB0)			INTTB00	(TMRB0)	
		Interrupt		ITB01C	ITB01M2	ITB01M1	ITB01M0	ITB00C	ITB00M2	ITB00M1	ITB00M0
	INTETB0	enable	99H	R		R/W		R		R/W	
		TMRB0		0	0	0	0	0	0	0	0
				1: INTTB01	Inter	rupt request	level	1: INTTB00	Inter	upt request	level
		1-1						INT	TBOF0 (TN	IRB0 over flo	ow)
		Interrupt		-	-	-	-	ITF0C	ITF0M2	ITF0M1	ITF0M0
	INTETB01V		9BH	R		R/W		R		R/W	
				0	0	0	0	0	0	0	0
		(Oveniow)				Write "0".		1: INTTBOF0 Interrupt request level			
					INT	TX0			INT	RX0	
		Interrupt		ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
	INTES0	enable	9CH	R		R/W		R		R/W	
		serial 0		0	0	0	0	0	0	0	0
				1: INTTX0	Inter	rupt request	level	1: INTRX0	Inter	upt request	level
					INT	TX1			INT	RX1	
	Inte	Interrupt		ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
	INTES1	enable	nable 9DH	R		R/W		R		R/W	
		serial 1		0	0	0	0	0	0	0	0
				1: INTTX1 Interrupt request level			1: INTRX1 Interrupt request level				
					INTI	RTC			INT	SBI	
		Interrupt		IRTCC	IRTCM2	IRTCM1	IRTCM0	IS2C	IS2M2	IS2M1	IS2M0
	INTES2RTC	enable	9EH	R		R/W		R		R/W	
		SBI/RTC		0	0	0	0	0	0	0	0
				1: INTRTC	Inter	rupt request	level	1: INTSBI	Inter	upt request	level
		Interrupt			INT	TC1			INT	TC0	
	INTETC01	enable	АОН	ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0
	INTETOOT	INTTC0/1	AUT	R		R/W		R		R/W	
				0	0	0	0	0	0	0	0
		Interrent			INT	ТСЗ			INT	TC2	
	INITETCOS	enable	۸1H	ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
	10121023	INTTC2/3	A1H	R		R/W		R		R/W	
www.DataShee	t4U.com			0	0	0	0	0	0	0	0

Interrupt control (2/3)

	Symbol	Name	Address	7	6	5	4	3	2	1	0	
					/	DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0	
	DMAOV	DMA0	80H	/	/			R	z/W			
	DIVIAUV	start vector				0	0	0	0	0	0	
								DMA0 s	tart vector			
						DMA1V5	DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0	
	DMA 41/	DMA1	81H					R	2/W			
	DIVIATV	start vector				0	0	0	0	0	0	
								DMA1 s	tart vector			
						DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0	
	DMAOV	DMA2	82H					R	2/W			
	DMA2V	start vector		/	/	0	0	0	0	0	0	
								DMA2 s	tart vector			
						DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0	
	DMAOV	DMA3	83H		/		R/W					
	DIVIA3V	start vector		/	/	0	0	0	0	0	0	
								DMA3 s	tart vector			
		Interrupt 88H INTCLR clear (Prohibit	ot 88H (Prohibit		/	CLRV5	CLRV4	CLRV3	CLRV2	CLRV1	CLRV0	
				/	/				W			
	INTELR		/	/	0	0	0	0	0	0		
		control	RIVIVV)			Clear interrupt request flag by writing DMA start vector						
		DMA		/	/	/	/	DMAR3	DMAR2	DMAR1	DMAR0	
	DIAD	software	0011			/	/	R/W	R/W	R/W	R/W	
	DIVIAR	request	89H	/	/	/	/	0	0	0	0	
		register							1: DMA requ	est in softwa	re	
								DMAB3	DMAB2	DMAB1	DMAB0	
			0.4.1					R/W	R/W	R/W	R/W	
	DIVIAB	request	бАП	/	/	/	/	0	0	0	0	
		register						1:	DMA reques	st on burst n	node	
				-	-	-	_	-	I0EDGE	IOLE	NMIREE	
				W	W	W	W	W	W	W	W	
		Interrupt	8CH	0	0	0	0	0	0	0	0	
	IIMC	input mode	(Prohibit						INT0	INT0	1: Operation	
MANAN DataSha	t411.com	control	RMW)			Mrita "O"			edge	0: Edge	even on	
www.DataSHe	thu.com					write U.			0: Rising	1: Level	NMI rising	
									1: Falling		edae	

Interrupt control (3/3)

Note: Only one channel can be set once for DMAR register. (Don't write "1" to plural bits.)

	(1)	emp sei	000 mare 0	01101 01 (1)	/						
	Symbol	Name	Address	7	6	5	4	3	2	1	0
				B0E		B0OM1	B0OM0	B0BUS	B0W2	B0W1	B0W0
				W		W	W	W	W	W	W
		Block 0		0		0	0	0	0	0	0
		CS/WAIT	COH	0. Disable		00: ROM/S	RAM	Data bus	Sot number of	of wait	0
	BOCS	control	(Prohibit	1: Enable		00.100000		Data Dus	Set number (100: Dee	a m ca al
		register	RMW)	1. Enable		10: C Booo	nicd	width	000: 2 waits	100: Res	erved
		regiotor				10. Rese	lived		001: 1 wait	101: 3 Wa	aits
						11. 2		0. TO DILS	010:(1 + N)	vaits 110: 4 wa	alts
				545		D 40144	D.C.M.	1: 8 DILS	011: 0 waits	111: 8 wa	aits
				B1E		B1OM1	BIOMO	BIBUS	B1W2	B1W1	B1W0
				W		W	W	W	W	W	W
		Block 1	C1H	0		0	0	0	0	0	0
	B1CS	CS/WAIT	(Prohibit	0: Disable		00: ROM/S	RAM	Data bus	Set number of	of wait	
	2.00	control	RMW)	1: Enable		01:]		width	000: 2 waits	100: Res	erved
		register)			10: Rese	erved	selection	001: 1 wait	101: 3 wa	aits
						11: J		0: 16 bits	010: (1 + N) v	waits110:4 wa	aits
								1: 8 bits	011: 0 waits	111: 8 wa	aits
				B2E	B2M	B2OM1	B2OM0	B2BUS	B2W2	B2W1	B2W0
				W	W	W	W	W	W	W	W
		Block 2	0.011	1	0	0	0	0	0	0	0
		CS/WAIT	C2H	0 [.] Disable	-	00' ROM/S	RAM	Data bus	Set number of	- of wait	-
	B2CS	control	(Prohibit	1: Enable	0:16-MB	01.		width	000: 2 waits	100: Pee	enved
		register	RMW)	1. Enable	area	10 Rese	erved	selection	000: 2 waits	101:3 ws	aite
		- 5			1: Area	11.	ived	0. 16 hits	010: (1 + NI)	101.5 wa	aits
					setting	11. *		1. 8 hite	010: (1 + 1)	111: 9 wa	aito
				B3E	/	B3OM1	B3OM0	B2BLIS	B3W2	B3\//1	B3\//0
						0301011	DJOIVIO M/	03003	103002	100001	100
		Block 2	C3H (Prohibit RMW)	 	/	VV O	VV O	VV	VV	VV O	VV O
					/			U Data kwa	0	0	0
	B3CS	control					RAIN	Data bus	Set number o	of wait	
		register		1: Enable			mind	width	000: 2 waits	100: Res	erved
		register				10: Rese	erved	selection	001: 1 wait	101: 3 wa	aits
						11: *		0: 16 DIts	010: (1 + N) V	waits110:4 wa	aits
						<u> </u>	\sim	1: 8 DILS	011: 0 waits	111:8 wa	aits
								BEXBUS	BEXW2	BEXW1	BEXWO
								VV	VV	W	W
		External	C7H					0	0	0	0
	BEXCS	CS/WAIT	(Prohibit					Data bus	Set number of	of wait	
www.DataSheet	4U.com	control	RMW)					width	000: 2 waits	100: Res	erved
		register	,					selection	001: 1 wait	101: 3 wa	aits
								0: 16 bits	010: (1 + N) v	waits110:4 wa	aits
								1: 8 bits	011: 0 waits	111: 8 wa	aits
		Memory		S23	S22	S21	S20	S19	S18	S17	S16
	MSAR0	start	C8H			1	R/	W	1		
		address	0011	1	1	1	1	1	1	1	1
		register 0				Star	t address A2	23 to A16 se	tting		
		Memory		V20	V19	V18	V17	V16	V15	V14~9	V8
		address	Сон				R	W	-		
		mask	0311	1	1	1	1	1	1	1	1
		register 0				CS0 area	size setting	0: Address c	comparison		
		Memory		S23	S22	S21	S20	S19	S18	S17	S16
		start	CALL				R	W			
	WISAK'I	address	CAH	1	1	1	1	1	1	1	1
		register 1				Star	t address A2	23 to A16 se	tting		
		Memory		V21	V20	V19	V18	V17	V16	V15~9	V8
		address	05.1		-	-	R/	Ŵ	-		-
	MAMR1	mask	CBH	1	1	1	1	1	1	1	
		register 1				CS1 area	size settina	0: Address o	comparison		
		-					g				

⁽⁴⁾ Chip select/wait control (1/2)

Note: TMP91CP27 don't include \overline{WAIT} pin. Therefore, when select "(1 + N) waits", operation is same with "1 wait".

	-													
Symbol	Name	Address	7	6	5	4	3	2	1	0				
	Memory		S23	S22	S21	S20	S19	S18	S17	S16				
	start	0011		R/W										
MSAR2	address	ССН	1	1	1	1	1	1	1	1				
	register 2		Start address A23 to A16 setting											
	Memory		V22	V21	V20	V19	V18	V17	V16	V15				
	address	0011		R/W										
MAMRZ	mask	CDH	1	1	1	1	1	1	1	1				
register 2				CS2 area s	size setting	0: Address o	omparison							
	Memory		S23	S22	S21	S20	S19	S18	S17	S16				
	start	0511	R/W											
MSAR3	address	CEH	1	1	1	1	1	1	1	1				
	register 3		Start address A23 to A16 setting											
	Memory		V22	V21	V20	V19	V18	V17	V16	V15				
	address	0511				R/	W							
MAMR3	mask	CFH	1	1	1	1	1	1	1	1				
	register 3		CS3 area size setting 0: Address comparison											

Chip select/wait control (2/2)

	Symbol	Name	Address	7	6	5	4	3	2	1	0
				XEN	XTEN	RXEN	RXTEN	RSYSCK	WUEF	PRCK1	PRCK0
							R/	W			
				1	0	1	0	0	0	0	0
				High-	Low-	High-	Low-	Clock after	Warm-up	Select presca	aler clock
				frequency	frequency	frequency	frequency	release of	(WUP)	00: f _{FPH}	
		System		oscillator	oscillator	oscillator(fc)	oscillator(fs)	STOP mode	0 write:	01: Reserved	1
		clock		(fc)	(fs)	after release	after release		Don't care	10: fc/16	
	SYSCR0	control	E0H	0: Stopped	0: Stopped	of stop mode	of stop mode		1 write:	11: Reserved	1
		register 0		1: Oscillation	1: Oscillation	0: Stopped	0: Stopped	0: fc	Warm-up		
						1: Oscillation	1: Oscillation	1: fs	start		
									0 read:		
									End of WUP		
									1 read:		
									Don't end		
									WUP		
								SYSCK	GEAR2	GEAR1	GEAR0
									R/	W	
								0	1	0	0
		System						Clock	Select gear	of high freq	uency clock
	SYSCR1	CIOCK	E1H					selection	000: fc		
		control						0: fc	001: fc /2		
		register 1						1: fs	010: fc /4		
									011: tc /8		
									100: tc /16		
									Others: Res	served	
					-	WUPIM1	WUPTM0	HALIM1	HALIMO		DRVE
		Quatara			R/W	R/W	R/W	R/W	R/W	\sim	R/W
		System			0	1	0	1	1		0
	SYSCR2	control	E2H		Write "0".	WUP time t	or oscillator	HALI mode	9 !		1: Drive the
		rogistor 2					ea	00: Reserve	ea		pin in the
						10: 2/input	frequency		node		stop
						10: 2 /inpu	t frequency		node		mode
				DROTEOT		11.2 /inpu	it frequency		EVTIN	DDV 000U	
www.DataShe	et4U.com			PROTECT						DRVUSCH	DRV03CL DAM
		FMC		0	0	1	0	0	0	1	1
	EMCCR0	control	E3H	Protect flag	Write "0".	Write "1"	Write "0".	AI F Output	1: fc	fc oscillator	fs oscillator
		register 0		0: OFF				0: Disable	external	drive ability	drive abillity
				1: ON				1: Enable	clock	1: Normal	1: Normal
										0: Weak	0: Weak
		EMC									
	EMCCR1	control	E4H	Write "1FF	I": Protect O						
Write except "1FH": Protect ON											

(5) Clock control

Note 1: If protection is on by writing except "1FH" code to EMCCR1 register, write operations to the following SFRs are not possible.

- (1) CS/WAIT controller
 B0CS, B1CS, B2CS, B3CS, BEXCS,
 MSAR0, MSAR1, MSAR2, MSAR3,
 MAMR0, MAMR1, MAMR2, MAMR3
- (2) Clock gear (EMCCR1 can be written to) SYSCR0, SYSCR1, SYSCR2, EMCCR0

Note 2: When using internal SBI, set SYSCR0<PRCK1:0> to "00".

(6) 8-bit timer control (1/2)

(6 –	1)	TMRA01
(0 -	.,	

Symbol	Name	Address	7	6	5	4	3	2	1	0		
			TA0RDE	/	/	/	I2TA01	TA01PRUN	TA1RUN	TAORUN		
			R/W				R/W	R/W	R/W	R/W		
	TMRA01		0				0	0	0	0		
TA01RUN	RUN	100H	Double				IDLE2	8-bit timer ru	un/stop control			
	_		buffer				0: Stopped	0: Stop and	clear			
			0: Disable				1: Operation	1: Run (Cou	nt up)			
			1: Enable									
-	TMRA0	102H				-	-					
TAOREG	register	(Prohibit				V	/					
	•	RIVIVV)				Unde	fined					
TA 4050	TMRA1	103H				-	-					
TATREG	register	(Prohibit				V	V					
	-	RIVIVV)	TAGANA	TAGANA	DW/MAC 4	Unde	fined	TATOLICO	TABOLIKA	TAGOLIKA		
		TA01M1										
			-			R/	W		0	0		
	TMRA01		0	0	0	0	0	0	0	0		
	source	104H	Operation m	lode	PWM cycle	al	Source clock for TMRA1 Source clock for					
TATINOD	CLK &	10411	00: 8-Dit time	er	00: Reserve	a		2		in input		
	mode		10: 8-bit PP	G	10.2 - 1		10· 4T16		00. TAUIN p	in input		
			11: 8-bit PW	(M	10.2 - 1 $11.2^8 - 1$		10. φ110 11: dT256		10· φT4			
							11. 41200		11: b T16			
				\sim			TAFF1C1	TAFF1C0	TAFF1IE	TAFF1IS		
							R/	Ŵ	R/	N		
	TMRA01	105H				\vee	1	1	0	0		
TA1FFCR	flip-flop	(Prohibit					00: Invert T	A1FF	1: TA1FF	Invertion		
	control	RMW)					01: SET TA	1FF	invert	by		
							10: Clear T	A1FF	enable	0: TMRA0		
							11: Don't ca	are		1: TMRA1		

(6 – 2) TMRA23
--------	----------

	Symbol	Name	Address	7	6	5	4	3	2	1	0	
				TA2RDE		/		I2TA23	TA23PRUN	TA3RUN	TA2RUN	
				R/W			/	R/W	R/W	R/W	R/W	
		TMRA23		0				0	0	0	0	
	TA23RUN	RUN	108H	Double				IDLE2	8-bit timer ru	un/stop contr	ol	
		-		buffer				0: Stopped	0: Stop and	clear		
www.DotoCho	ot 11 o o m			0: Disable				1: Operation	1: Run (Cou	nt up)		
www.DataSne	et4U.com		40411	1: Enable								
	TAODEC	TMRA2	10AH (Prohibit									
	TAZKEG	register	(FIOHIDIC RMW)				۷ Linde	v				
			1084				Unde					
	TA3REG	TMRA3	(Prohibit				V	V				
		register	RMW)		Lindefined							
				TA23M1	TA23M0	PWM21	PWM20	TA3CLK1	TA3CLK0	TA2CLK1	TA2CLK0	
					R/W							
		TMRA23		0	0	0	0	0	0	0	0	
		source		Operation n	node	PWM cycle		Source clock for		Source clock for		
	TA23MOD	CLK &	10CH	00: 8-bit tim	er	00: Reserved		TMRA3		TMRA2		
		mode		01: 16-bit til	mer	$01:2^{\circ}-1$		00: TA2TR	3	00: Reserve	ed	
				10: 8-DIT PP	G	$10: 2^{\circ} - 1$ 11: 2 ⁸ 1		01: φ11 10: ↓T16		01: φ11 10: μ Τ 4		
				TT. O-DIL PV	VIVI	11.2 - 1		10. φ1 10 11· φT256		10. φ14 11· Δ Τ16		
							\sim	TAFE3C1	TAFE3C0	TAFE3IE	TAFE3IS	
						/		R/	W	R/	W	
		TMRA23	10DH		\sim	/		1	1	0	0	
	TA3FFCR	flip-flop	(Prohibit					00: Invert T	A3FF	1:TA3FF	Invert by	
		control	RMW)					01: SET TA	3FF	invert	0: TMRÁ2	
								10: Clear T	A3FF	enable	1: TMRA3	
								11: Don't ca	are			

8-bit timer control (2/2)

(6 – 3) IMRA45	45	TMR/	- 3)	(6
----------------	----	------	------	----

Symbol	Name	Address	7	6	5	4	3	2	1	0
			TA4RDE				I2TA45	TA45PRUN	TA5RUN	TA4RUN
			R/W				R/W	R/W	R/W	R/W
			0				0	0	0	0
TA45RUN	DUN	110H	Double				IDLE2	8-bit timer ru	in/stop contro	bl
	NON		buffer				0: Stopped	0: Stop and	clear	
			0: Disable				1: Operation	1: Run (Cou	nt up)	
			1: Enable							
		112H					_			
TA4REG	rogistor	(Prohibit				V	V			
	register	RMW)				Unde	efined			
		113H				-	_			
TA5REG	register	(Prohibit				V	V			
	register	RMW)				Unde	efined			
			TA45M1	TA45M0	PWM41	PWM40	TA5CLK1	TA5CLK0	TA4CLK1	TA4CLK0
				-		R/	W	-		-
	TMR 4/5		0	0	0	0	0	0	0	0
	source		Operation n	node	PWM cycle		Source cloc	k for TMRA5	Source clock for	
TA45MOD	CLK &	114H	00: 8-bit tim	er	00 : Reserve	ed	00: TA4TR0	G	TMRA4	
	mode		01: 16-bit tir	ner	01: 2 ⁶ – 1		01: φT1		00: TA4IN F	Pin input
	mode		10: 8-bit PP	G	10: 2 ⁷ – 1		10: φT16		01:	
			11: 8-bit PV	/M	11: 2 ⁸ – 1		11: φT256		10:	
								1	11:	
							TAFF5C1	TAFF5C0	TAFF5IE	TAFF5IS
							R	/W	R/	W
	TMRA45	115H					1	1	0	0
TA5FFCR	flip-flop	(Prohibit					00: Invert T	A5FF	1:TA5FF	Invert by
	control	RMW)					01: SET TA	5FF	Invert	U: IMRA4
							10: Clear TA5FF		enable	1: TMRA5
							11: Don't ca	re		

(7) 16-bit timer control

(7 – 1) TMRB0

				_	-	_		-	-		-
	Symbol	Name	Address	7	6	5	4	3	2	1	0
				TB0RDE	-		/	I2TB0	TB0PRUN		TB0RUN
				R/W	R/W			R/W	R/W		R/W
		TMRB0		0	0			0	0		0
	TBORUN	Run	180H	Double	Write "0".			IDLE2	16-bit timer r	un/stop conti	rol
		i turi		buffer				0: Stopped	0: Stop and o	clear	
				0: Disable				1: Operation	1: Run (Cour	nt up)	
				1: Enable							
				TB0CT1	TB0ET1	TB0CP0I	TB0CPM1	TB0CPM0	TB0CLE	TB0CLK1	TB0CLK0
				R/	W	W*			R/W		
		TMRB0	182H	0	0	1	0	0	0	0	0
	TRAMOR	source		TB0FF1 Inve	rsion trigger	0: Software	Capture timr	ning	1:UC0 clear	Source clocl	<
	IBOMOD	CLK &	(Prohibit	0: Disable		capture	00: Disable	C	enable	00: TB0IN0	input
		mode	RMW)	1: Enable		1: Undefined	01: ↑. ↑ (TB0II	N0. TB0IN1)		01: ₀ T1	
				Capture to	TB0RG1		10: ↑. ↓ (TB0II	N0)		10: _φ T4	
				TB0CP1	matching		11: ↑. ↓ (TA1C	OUT)		11: _φ T16	
				TB0FF1C1	TB0FF1C0	TB0C1T1	TB0C0T1	TB0E1T1	TB0E0T1	TB0FF0C1	TB0FF0C0
					/*		R/	W			/*
				1	1	0	0	0	0	1	1
			183H	00: Invort TE			ort trigger	0	0	00: Invort TI	
		TMRB0	10011				t: Enchlo			00. Invent TPO	
	TB0FFCR	flip-flop	(Prohihit	10: Clear TB		U. Disable		Invort when	Invert when	10: Clear TE	
		control	RMW)	11: Don't care					the LIC 11: Don't care		
			((iiii))	TI. DUITI Ca	le					TT: Don't care	
						value is	value is	matches	matches		
		THERE	40011			IDUCPI	IDUCPU	IDURGI	IDUKGU		
			100H (Drahihit				-				
	TBURGUL	register 0					V	/ 			
		LOW	RIVIVV)				Unde	rined			
		IMRB0	189H				-				
	TBORGOH	register 0	(Prohibit				V	1			
		High	RIVIVV)				Unde	fined			
		TMRB0	18AH				-				
www.DataShe	et#BORG1L	register 1	(Prohibit				V	/			
		Low	RMW)				Unde	fined			
		TMRB0	18BH				-				
	TB0RG1H	register 1	(Prohibit				V	/			
		High	RMW)				Unde	fined			
		TMRB0					-				
	TB0CP0L	capture register 0	18CH				F	1			
		Low					Unde	fined			
		TMRB0					-				
	TB0CP0H	capture	18DH				R				
		High					Linde	fined			
		TMRB0					Onde				
		capture	18 F H				-	,			
	I DOOF IL	register 1	IULII					fined			
		TMRB0					Unde	inieu			
	TDOOD41	capture					-				
	TBUCPIH	register 1	IOFH								
		High					Unde	tined			

Note: When programming "1" to TB0MOD<TB0CP0I> in condition of programmed "0", present value of up-counter is captured to TB0CP0 register.

(8) UART/serial channel control (1/2)

(8 – 1) UART/SIO channel 0

			1											
	Symbol	Name	Address	7	6	5	4	3	2	1	0			
		Serial	200H	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0			
	SC0BUF	channel 0	(prohibit		R (Receiving)/W (Transmission)									
		buffer	RMW)				Undef	ined		-				
				RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC			
		Sorial		R R/W R (Cle			R (Clea	red to 0 by r	eading)	R/W				
	SCOCR	channel 0	2011	Undefined	0	0	0	0	0	0	0			
	COUCIN	control	20111	Receiving	Parity	1: Parity		1: Error		0: SCLK0 ↑	1: SCLK0			
		Control		data bit8	0: Odd	enable	Quarrun	Dority	Froming	1: SCLK0 ↓	pin input			
					1: Even		Ovenun	Failty	Framing					
				TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0			
							R/\	V						
		Serial		0	0	0	0	0	0	0	0			
	SC0MOD0	channel 0	202H	Transmission	1: CTS	1: Receive	1: Wakeup	00: I/O inter	face	00: TA0TRG				
		mode0		data bit8	enable	enable	enable	01: 7-bit UA	RT	01: Baud rate	egenerator			
								10: 8-bit UA	RT	10: Internal d	ock f _{SYS}			
								11: 9-bit UA	RT	11: External of	clock SCLK0			
			-	BR0ADDE	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0				
		Sorial					R/\	V						
		channel 0		0	0	0	0	0	0	0	0			
	BR0CR	channel 0 baud rate	203H	Write "0".	1: (16 – K)/16	00:								
		control			divided	01:		Set the dividing value "N" (0 to			o F).			
					enable	10:								
					/	11:			r					
		Serial						BR0K3	BR0K2	BR0K1	BR0K0			
	BR0ADD	channel 0	204H						R/	W				
		K setting	-					0	0	0	0			
		register						S	et the value	of "K" (1 to F).			
				12S0	FDPX0									
				R/W	R/W	/								
		Carial		0	0									
	SCOMOD1	Serial	2051	IDLE2	I/O									
	SCONODI	modo1	2000	0: Stop	interface									
www.DataShe	et4U.com	nouer		1: Operation	1: Full duplox									
					0. Half									
					duploy									

(8 – 2) IrDA

Symbol	Name	Address	7	6	5	4	3	2	1	0			
			PLSEL	RXSEL	TXEN	RXEN	SIRWD3	SIRWD2	SIRWD1	SIRWD0			
ł		[R/W										
ł		[0	0	0	0	0	0	0	0			
SIRCR	control register	207H	Transmission pulse width 0: 3/16 1: 1/16	Receiving data logic 0: "H" pulse 1: "L" pulse	Transmission operation 0: Disable 1: Enable	Receiving operation 0: Disable 1: Enable	Select effect Pulse width "2x × (Settin Possible: 1 t	ive pulse wid of more than g value + 1)" to 14	th and equal				
			i I		1 1	1	Not possible	: 0, 15					

UART/serial channel control(2/2)

(8 - 3) UART/SIO channel 1

	Symbol	Name	Address	7	6	5	4	3	2	1	0			
		Serial	208H	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0			
	SC1BUF	channel 1	(Prohibit			R ((Receiving)/V	V (Transmiss	ion)					
		buffer	RMW)		Undefined									
			209H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC			
		Serial		R R/W			R (Cleared to 0 by reading)			R/W				
	SC1CR	channel 1		Undefined	0	0	0	0	0	0	0			
		control		Receiving	Parity	1: Parity		1: Error		0: SCLK1 ↑	1:SCLK1			
				data bit 8	0: Odd 1: Even	enable	Overrun	Parity	Framing	1: SCLK1 ↓	pin input			
				TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0			
							R	/W		-				
		Serial		0	0	0	0	0	0	0	0			
	SC1MOD0	channel 1	20AH	Transmissi	1: CTS	1: Receive	1: Wakeup	00: I/O interface 01: 7-bit UART		00: TA0TRG				
		mode 0		on data bit8	enable	enable	enable			01: Baud rate	e generator			
								10: 8-bit UAF	RT	10: Internal of	lock f _{SYS}			
						DD (0) ((554040	11: 9-bit UAF	रा	11: External	clock SCLK1			
		Serial		-	BR1ADDE	BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0			
				0	0	0	R O	///	0	0	0			
	BR1CR	channel 1	20BH	U Mrite "O"			0	0	0	0	0			
	BILION	baud rate	20011	while 0.	divided	00. φτο 01: μτ2								
		control			enable	10 [.] φT2		Set	the dividing \	/alue "N" (0 to	o F).			
						11: \dd T32								
		Serial		/				BR1K3	BR1K2	BR1K1	BR1K0			
		channel 1	00011	/	\sim				R/	W				
	BRIADD	Ksetting	20CH					0	0	0	0			
		register						S	Set the value	of "K" (1 to F).			
				I2S1	FDPX1	/	/	/	/					
				R/W	R/W	/	/		/					
		Serial		0	0									
	SC1MOD1	channel 1	20DH	IDLE2	I/O									
		mode1		0: Stop	interface									
www.DataSh	eet4U.com			1: On and the	1: Full duplex									
				Operation	u: Hair duplex									

Note 1: As all error flags SCxCR<OERR, PERR,FERR> are cleared after reading, do not test only a single bit with a bit-testing instruction.

Note 2: The baud rate genetrator can be set N = "1" when UART mode and disable + (16 - K)/16 division function. Don't use in I/O interface mode.

Note 3: Set BRxCR<BRxADDE> to "0" and disable + (16 - K)/16 division function in I/O interface mode.

- Symbol Address Name 6 5 4 3 2 0 7 1 BC2 BC1 BC0 ACK SCK2 SCK1 SCK0 /SWRMON 240H R/W R/W W W W (I²C bus mode) 0 0 0 0 0 0 0/1 Acknowledge Select number of transfer bit Internal serial clock selector (When writing) (Prohibit mode 000: 8 001: 1 010:2 000: 5 001:6 010:7 0: Disable RMW) 011:3 100: 4 101:5 011:8 100: 9 101:10 Serial bus 1: Enable 110:6 111:7 110: 11 111: Reserved interface SBI0CR1 SIOINH control SIOM1 SCK0 SIOS SIOM0 SCK2 SCK1 register 1 W w W W W 240H (SIO 0 0 0 0 0 0 0 mode) Transfer Forcing Select transfer mode Select frequency of serial clock control stop of 00. 8-bit transmit 000:4 001:5 010.6 (Prohibit transfer 0: Stop 01. Reserved 011:7 100:8 101:9 RMW) 0: Continue 1: Start 10: 8-bit transmit/receiving 110:10 111: External SCK input 1: Stop 11: 8-bit receiving SBI DB7 DB6 DB5 DB4 DB3 DB2 DB1 DB0 241H data R (Receiving)/W (Transmission) SBI0DBR Prohibit buffer RMW) Undefined register SA6 SA5 SA4 SA3 SA2 SA1 SA0 ALS W 242H I²C bus 0 0 0 0 0 0 0 0 I2C0AR address (Prohibit Address register RMW) recognition Setting slave address 0: Enable 1: Disable
- (9) I²C bus/serial channel control (1/2)

Note 1: When use built-in SBI, set SYSCR0<PRCK1:0> to fFPH.

Note 2: Set the SBI0CR1<BC2:0> to "000" before switching to a clock-synchronous 8-bit SIO mode.

Note 3: Switch a mode to Port mode after confirming that the bus is free. And, switch from port mode to I^2C bus mode or SIO mode after confirming port conditon = "H".

www.DataSheet4U.com 4: Set the transfer mode and the serial clock in SIO mode after clearing SBI0CR1<SIOS> to "0" and <SIOINH> to "1".

Note 5: After reset, default value of SBI0CR1<SCK0> is cleared "0", and default value of <SWRMON> is set "1".

	Symbol	Nama	Addroop	7	6	F	4	2	2	1	0
	Symbol	Name	Audress	/		5		000044			
				MST	IRX	BB		SBIM1	SBIMO	SWRS11	SWRS10
			243H				W .				-
			(I ² C bus	0	0	0	1	0	0	0	0
			mode)	0: Slave	0: Receiver	0: Generates	1: Cancel	Operation mod	de selection	Software res	set control
				1: Master	1: Transmitter	the stop	INTSBI	00:Port mod	e	Write "10 " a	and "01" in
			(Prohibit			condition	interrupt	01:Clocked	synchronous	order, then a	an internal
		Serial bus	RMW)			1: Generates	request	8-bit SIO	mode	reset signal	is
	SBI0CR2	interface				the start		10:1 ² C bus m	node	generated.	
		control				condition	<u> </u>	11:Reserved	1		
	(In writing)	register 2						SBIM1	SBIM0	-	-
		-	243H					V	V	W	W
			(SIO					0	0	0	0
			mode)					Operation m	ode selection	Write	e "0".
								00: Port mod	e		
			(Prohibit					01:SIO mod	е		
			RMW)					10:I ² C bus m	node		
								11:Reserved	ł		
				MST	TRX	BB	PIN	AL	AAS	AD0	LRB
							R				
				0	0	0	1	0	0	0	0
			243H	0: Slave	0: Receiver	0: Bus free	Interrupt	0: Undetected	0: –	0: Undetected	Last
			(l ² C bus	1: Master	1: Transmitter	1: Bus busy	request	1: Detect	1: Detect	1: Detect	received
			(i o buo mode)				0: Requested	arbitration	slave	general	bit monitor
			mode)				1: Canceled	lost	address	call	0: "0"
		Sorial bus							match or		1: "1"
	SBIUSD	interface							general		
		etatue							call		
	(in reading)	register						SIOF	SEF		
		regiotor						F	२		
								0	0		
			243H					Transfer	Shift		
			(SIO					status	operation		
			mode)					monitor	status		
MANA Datas	hoot411.cor	0						0: Terminate	monitor		
www.Datao	1166140.001							1: In progress	0: Terminate		
									1: In progress		
				_	I2SBI0						
		Serial bus	244H	W	R/W	/					
		interface		0	0	/					
	SBIUBRU	baud rate	(Prohibit	Write "0".	IDLE2						
		register 0	RMW)		0: Stop						
					1: Operation						
				P4EN	_	/	/	/	/	/	/
				W	W	\sim	\sim	\sim	/	\sim	
		Serial bus	245H	0	0	\sim	\sim	\sim	\backslash	\sim	\backslash
	SBI0BR1	Interface	(D	Clock	Write "0"						
		baud rate	(Prohibit	control							
		register 1	KIMVV)	0: Stop							
				1: Operation							

I²C bus/serial channel control (2/2)

(10) AD converter control

	Symbol	Name	Address	7	6	5	4	3	2	1	0		
				EOCF	ADBF	-	-	ITM0	REPEAT	SCAN	ADS		
				F	२	R/W	R/W	R/W	R/W	R/W	R/W		
				0	0	0	0	0	0	0	0		
				End flag	Busy flag	Write "0".	Write "0".	Interrupt	0:Single	0: Channel	1: Start		
			280H	1: Conversion	1: Conversion			specification	conversion	fixed	conversion		
		mode		complete	in progress			channel	1:Repeat	mode			
		register 0	(Prohibit					fixed repeat	conversion	1: Channel			
		-	RMW)					0. Every		mode			
								conversion		mode			
								1: Every					
								fourth					
						<		conversion					
				VREFON	I2AD			ADIRGE	ADCH2	ADCH1	ADCH0		
				R/W	R/W			R/W		R/W			
				0	0			0	0	0	0		
				1: VREF on	IDLE2			External	Input channe	I selection			
		AD			0: Stop 1: Operation			trigger start		can In			
	ADMOD1	mode	2B1H		1. Operation			1: Enable	001: AN1/AN	I0 →AN1			
		register 1							010: AN2/AN	$0 \rightarrow AN1 \rightarrow A$	N2		
									011: AN3/AN	$0 \rightarrow AN1 \rightarrow A$	$AN2 \rightarrow AN3$		
									100:				
									101: } Dor	n't select			
									111:				
	ADREG04L	AD result		ADR01	ADR00			/			ADR0RF		
		register 0/4	2A0H	F	र		/			/	R		
		low		Unde	fined						0		
		AD result		ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02		
	ADREG04H	register 0/4 high	2A1H	R									
							Undefi	ned		-			
		AD result		ADR11	ADR10						ADR1RF		
	ADREGISE	register 1/5	2A2H	F	२						R		
www.DataShe	t4U.com	low		Unde	fined						0		
		AD result		ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12		
	ADREG15H	register 1/5	2A3H				R						
		high			[Undefi	ned		<			
		AD result		ADR21	ADR20						ADR2RF		
	ADREG26L	register 2/6	2A4H	F	2						R		
		IOW		Unde	fined						0		
		AD result		ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22		
	ADREG26H	register 2/6	2A5H				R	<u> </u>					
		nign				\sim	Undefi	ned	\sim	<u> </u>			
		AD result		ADR31	ADR30	>	>	>	>	>	ADR3RF		
	ADREG37L	register 3/7	2A6H	F	۲						R		
		IOW		Unde	etined						0		
		AD result	0.4 - ' '	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32		
	ADREG37H	register 3/7	2A7H				R						
		nign					Undefi	ned					

Note 1: ADMOD0<ADS> is always read as "0".

Note 2: When using $\overline{\text{ADTRG}}$ with ADMOD1<ADTRGE> = "1", do not set ADMOD1<ADCH2:0> = "011.

Note 3: When set ADMOD1<I2AD> to "0", operation is different by AD conversion mode after released Halt mode.

(11) Watchdog timer control

Symbol	Name	Address	7	6	5	4	3	2	1	0	
			WDTE	WDTP1	WDTP0	-	-	I2WDT	RESCR	-	
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	WDT		1	0	0	0	0	0	0	0	
WDMOD		300H	1: WDT	00: 215/f _{SYS}	;	Write "0".	Write "0".	IDLE2	1: Internaly	Write "0".	
registe	register	30011	enable	01: 217/f _{SYS}	;			0: Stop	connects		
	regiotor			10: 2 ¹⁹ /f _{SYS}	;			1: Operation	WDT out		
				11: 2 ²¹ /f _{SYS}					to the		
				0.0					reset pin		
		301H					_				
	WDT						W				
WDCR	control	(Prohibit					-				
		RMW)	B1H: WDT disable 4EH: WDT clear								

(12) RTC (Real time clock) control

Symbol	Name	Address	7	6	5	4	3	2	1	0
			-					RTCSEL1	RTCSEL0	RTCRUN
			R/W		/	/	/	R/	W	R/W
	RTC		0		/	/	/	0	0	0
RTCCR	control	310H	Write "0".					00: 2 ¹⁴ /fs		0: Stop
	register							01: 2 ¹³ /fs		and
								10: 2 ¹² /fs		clear
								11: 2 ¹¹ /fs		1: RUN

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 "74HC××" 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 (SYSCR2<HALTM1:0> = 0, 1) and the CPU executes the HALT instruction. When the drive enable bit SYSCR2<DRVE> is set to "1", 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), P2 (A16 to A21, A0 to A5), P60, P70 to P74, P80 to P83, P91 to P92 and P94 to P95



■ P30 (RD), P31 (WR)



■ P32, P40 to P42



P5 (AN0 to AN3)



P63 (INT0)



■ P61 (SO/SDA), P62 (SI/SCL), P90 (TXD0) and P93 (TXD1)



AM0 and AM1



ALE



RESET



X1 and X2



W---W

Ladder resistor

- AVSS

7. Points to Note and Restrictions

(1) Notation

- 1. The notation for built-in I/O registers is as follows register symbol<Bit symbol> Example: TA01RUN<TA0RUN> denotes bit TA0RUN of register TA01RUN.
- 2. Read-modify-write instructions

An instruction in which the CPU reads data from memory and writes the data to the same memory location in one instruction.

Example 1: SET 3, (TA01RUN) Set bit 3 of TA01RUN. Example 2: INC 1, (100H)..... Increment the data at 100H.

• Examples of read-modify-write instructions on the TLCS-900

Exchange instruction

EX (mem), R

Arithmetic operations

ADD	(mem), R/#	ADC (mem), R/#
SUB	(mem), R/#	SBC (mem), R/#
INC	#3, (mem)	DEC #3, (mem)

Logic operations

www.DataSheet4U.com

AND (mem), R/#	OR	(mem), R/#
XOR (mem), R/#		

Bit manipulation operations

STCF #3/A, (mem)	RES #3, (mem)
SET #3, (mem)	CHG #3, (mem)
TSET #3, (mem)	

Rotate and shift operations

RLC	(mem)	RRC	(mem)
RL	(mem)	\mathbf{RR}	(mem)
SLA	(mem)	SRA	(mem)
SLL	(mem)	SRL	(mem)
RLD	(mem)	RRD	(mem)

 $3.~f_{\rm OSCH,}\, fc,\, fs,\, f_{\rm FPH},\, f_{\rm SYS}$ and one state

The clock frequency input on ins X1 and X2 is called f_{OSCH} . TMP91CP27 have not DFM. Therefore, become fc equal for form It called fs that clock frequency is inputted from XT1/XT2 pin.

The clock selected by SYSCR1<SYSCK> is called f_{FPH}. The clock frequency give by f_{FPH} divided by 2 is called f_{SYS}.

One cycle of fSYS is referred to as one state.

- (2) Points to note
 - a. AM0 and AM1 pins

This pin is connected to the VCC pin. Do not alter the level when the pin is active.

b. 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 a time equivalent to the warm-up time elapses between input of the release request and output of the system clock.

c. Programmable pull-up resistor

The programmable pull-up resistor can be turned ON/OFF by a program when the ports are set for use as input ports. When the ports are set for use as output ports, they cannot be turned ON/OFF by a program.

The data registers (e.g., P4 register) are used to turn the pull-up resistors ON/OFF. Consequently read-modify-write instructions are prohibited. Therefore, use Transfer instruction.

d. Watchdog timer

The watchdog timer starts operation immediately after a reset is released. When the watchdog timer is not to be used, disable it.

e. AD converter

TMP91CP27 include function that can cut or connect the string resistor between the reference pins (Share with AVCC and AVSS pins in TMP91CP27) by program. (ADMOD1 <VREFON>)

When STOP mode is used as reduce consumption power supply, disable the resistor using the program before the HALT instruction is executed.

f. RTC

www.DataSheet4U.com

If set IDLE1 mode (Operate only oscillator) and execute HALT instruction, built-in RTC is operation enable condition. If want to stop operation, stop operation by setting control register RTCCR <RTCRUN> to "0".

g. CPU (Micro DMA)

Only the LDC cr, r and LDC r, cr instructions can be used to access the control registers in the CPU (e.g., the transfer source address register (DMASn)).

h. Undefined SFR

The value of an undefined bit in an SFR (Special function register) is undefined when read.

i. POP SR instruction

Please execute the POP SR instruction during DI condition.

j. Releasing the HALT mode by requesting an interruption

Usually, interrupts can release all halts status. However, the interrupts (\overline{NMI} , INTO, INTRTC) 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 5 clocks of fFPH) with IDLE1 or STOP mode (IDLE2 is not applicable to this case).(In this case, an interrupt request is kept on hold internally)

If another interrupts is generated after it has shifted to HALT mode completely, halt status can be released without difficulty. The priority of this interrupt is compared with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt. 8. Package Dimensions

P-LQFP64-1010-0.50D

Unit: mm

