Toshiba Semiconductor and Storage - TMP91FY42FG(C,JZ) Datasheet

Welcome to E-XFL.COM

What is "Embedded - Microcontrollers"?

"Embedded - Microcontrollers" refer to small, integrated circuits designed to perform specific tasks within larger systems. These microcontrollers are essentially compact computers on a single chip, containing a processor core, memory, and programmable input/output peripherals. They are called "embedded" because they are embedded within electronic devices to control various functions, rather than serving as standalone computers. Microcontrollers are crucial in modern electronics, providing the intelligence and control needed for a wide range of applications.

Applications of "Embedded - Microcontrollers"

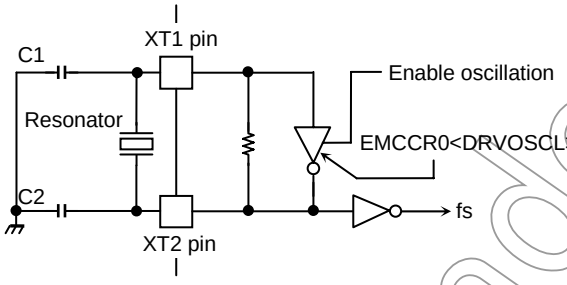
Details	
Product Status	Obsolete
Core Processor	900/L1
Core Size	16-Bit
Speed	27MHz
Connectivity	EBI/EMI, I²C, IrDA, UART/USART
Peripherals	DMA, PWM, WDT
Number of I/O	81
Program Memory Size	256KB (256K x 8)
Program Memory Type	FLASH
EEPROM Size	-
RAM Size	16K x 8
Voltage - Supply (Vcc/Vdd)	2.7V ~ 3.6V
Data Converters	A/D 8x10b
Oscillator Type	Internal
Operating Temperature	-40°C ~ 85°C (TA)
Mounting Type	Surface Mount
Package / Case	100-LQFP
Supplier Device Package	100-LQFP (14x14)
Purchase URL	https://www.e-xfl.com/product-detail/toshiba-semiconductor-and-storage/tmp91fy42fg-c-jz

(2) Reduced drivability for low-frequency oscillator

(Purpose)

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

(Block diagram)



(Setting method)

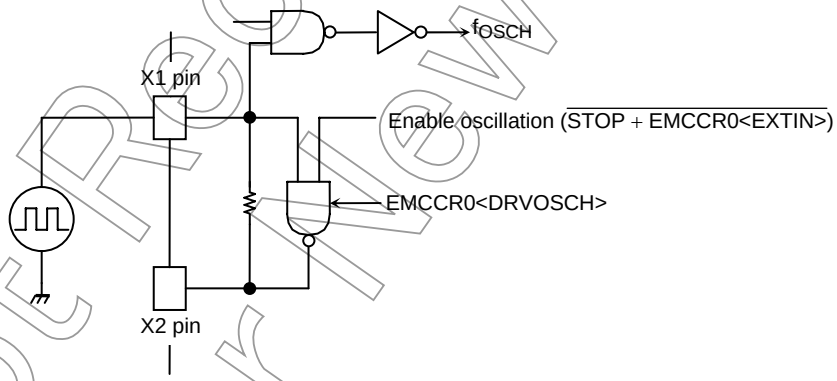
The drivability of the oscillator is reduced by writing 0 to the EMCCR0<DRVOSCL> register. By reset, <DRVOSCL> is initialized to 1.

(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)



(Setting method)

The oscillator is disabled and starts operation as buffer by writing 1 to EMCCR0<EXTIN> register. X2 pin is always outputted 1.

By reset, <EXTIN> is initialized to 0.

Note: Do not write EMCCR0<EXTIN> = "1" when using external resonator.

Table 3.3.5 Source of Halt State Clearance and Halt Clearance Operation

Status of Received Interrupt			Interrupt Enabled (Interrupt level) ≥ (Interrupt mask)			Interrupt Disabled (Interrupt level) < (Interrupt mask)		
HALT mode			IDLE2	IDLE1	STOP	IDLE2	IDLE1	STOP
Source of halt state clearance	Interrupt	NMI	♦	♦	♦ *1	–	–	–
		INTWD	♦	×	×	–	–	–
		INT0~INT4 (Note 1)	♦	♦	♦ *1	○	○	○ *1
		INTRTC	♦	♦	×	○	○	×
		INT5~INT8	♦ (Note2)	×	×	×	×	×
		INTTA0~INTTA7	♦	×	×	×	×	×
		INTTB00, INTTB01, INTTB10, INTTB11,INTTBOF0, INTTBOF1	♦	×	×	×	×	×
		INTRX0~INTRX1, INTTX0~INTTX1	♦	×	×	×	×	×
		INTSBI	♦	×	×	×	×	×
		INTAD	♦	×	×	×	×	×
		RESET	Initialize LSI.					

♦: After clearing the HALT mode, CPU starts interrupt processing.

○: After clearing the HALT mode, CPU resumes executing starting from instruction following the HALT instruction.

×: 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.

Note1: When the HALT mode is cleared by an INT0 interrupt of the level mode in the interrupt enabled status, hold level H until starting interrupt processing. If level L is set before holding level L, interrupt processing is correctly started.

Note2: When the external interrupts INT5 to INT8 are used during IDLE2 mode, set to 1 for TB0RUN<I2TB0> and TB1RUN<I2TB1>.

(Example releasing IDLE1 mode)

An INT0 interrupt clears the halt state when the device is in IDLE1 mode.

```
Address
8200H   LD      (P6FC), 08H     ; Sets P63 to INT0.
8203H   LD      (IIMC), 00H     ; Selects INT0 interrupt rising edge.
8206H   LD      (INTE0AD), 06H  ; Sets INT0 interrupt level to 6.
8209H   EI      5               ; Sets interrupt level to 5 for CPU.
820BH   LD      (SYSCR2), 88H   ; Sets HALT mode to IDLE1 mode.
820EH   HALT                    ; Halts CPU.

INT0                                          INT0 interrupt routine
                                                        RETI
820FH   LD      XX, XX
```

(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 (Note) 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 avobe plogram, 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 (e.g., "NOP" × 1 times).

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
	EI

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 writing INTCLR register.

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 → L)

INTRX: Instruction which read the receive buffer

		7	6	5	4	3	2	1	0
B0CS (00C0H)	Bit symbol	B0E		B0OM1	B0OM0	B0BUS	B0W2	B0W1	B0W0
	Read/Write	W		W					
	After reset	0		0	0	0	0	0	0
Read-modify-write instructions are prohibited.	Function	0: Disable 1: Enable		Chip select output waveform selection 00: For ROM/SRAM 01: 10: 11: Don't care		Data bus width 0: 16 bits 1: 8 bits	Number of waits 000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits		100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits
B1CS (00C1H)	Bit symbol	B1E		B1OM1	B1OM0	B1BUS	B1W2	B1W1	B1W0
	Read/Write	W		W					
	After reset	0		0	0	0	0	0	0
Read-modify-write instructions are prohibited.	Function	0: Disable 1: Enable		Chip select output waveform selection 00: For ROM/SRAM 01: 10: 11: Don't care		Data bus width 0: 16 bits 1: 8 bits	Number of waits 000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits		100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits
B2CS (00C2H)	Bit symbol	B2E	B2M	B2OM1	B2OM0	B2BUS	B2W2	B2W1	B2W0
	Read/Write	W							
	After reset	1	0	0	0	0	0	0	0
Read-modify-write instructions are prohibited.	Functions	0: Disable 1: Enable	CS2 area selection 0: 16-Mbyte area 1: CS area	Chip select output waveform selection 00: For ROM/SRAM 01: 10: 11: Don't care		Data bus width 0: 16 bits 1: 8 bits	Number of waits 000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits		100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits
B3CS (00C3H)	Bit symbol	B3E		B3OM1	B3OM0	B3BUS	B3W2	B3W1	B3W0
	Read/Write	W		W					
	After reset	0		0	0	0	0	0	0
Read-modify-write instructions are prohibited.	Functions	0: Disable 1: Enable		Chip select output waveform selection 00: For ROM/SRAM 01: 10: 11: Don't care		Data bus width 0: 16 bits 1: 8 bits	Number of waits 000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits		100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits
BEXCS (00C7H)	Bit symbol					BEXBUS	BEXW2	BEXW1	BEXW0
	Read/Write					W			
	After reset					0	0	0	0
Read-modify-write instructions are prohibited.	Functions					Data bus width 0: 16 bits 1: 8 bits	Number of waits 000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits		100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits

Master enable bit

0	Enable
1	Disable

CS2 area selection

0	16-Mbyte area
1	Specified address area

Chip select output waveform selection

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

Number of address area waits
(See 3.6.2, (3) Wait control.)

Data bus width selection

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

Figure 3.6.5 Chip Select/Wait Control Registers

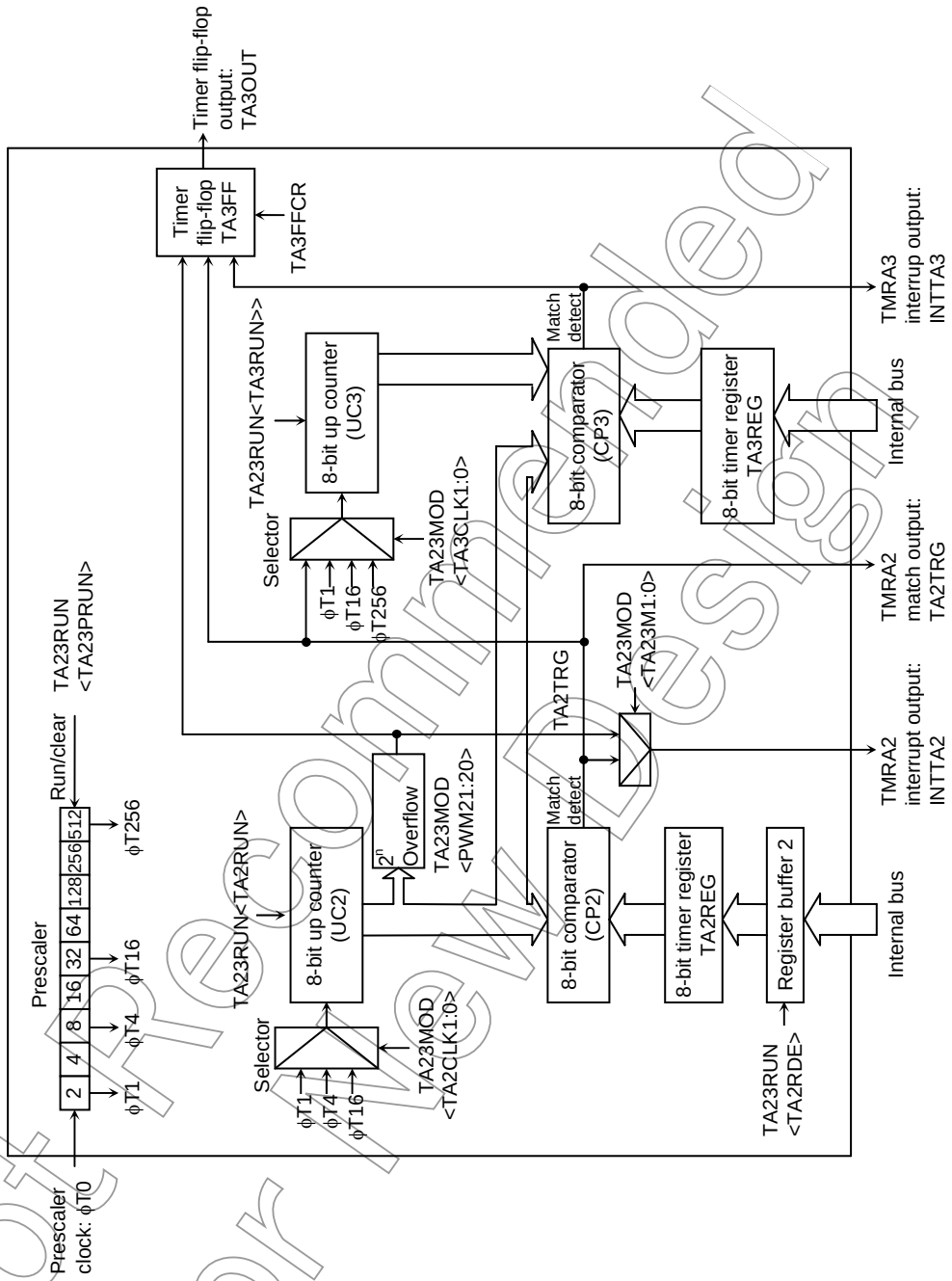


Figure 3.7.2 TMRA23 Block Diagram

(4) Comparator (CP0)

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<TA1FFIE> in the timer flip-flop control register.

A reset clears the value of TA1FF1 to 0.

Writing 01 or 10 to TA1FFCR<TA1FFC1:0> sets TA1FF to 0 or 1. Writing 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 function register P7CR, P7FC.

Table 3.7.3 PWM Cycle

at fc = 33 MHz, fs = 32.768 kHz

Select System Clock <SYSCK>	Select Prescaler Clock <PRCK1:0>	Gear Value <GEAR2:0>	PWM Cycle								
			2^6			2^7			2^8		
			φT1	φT4	φT16	φT1	φT4	φT16	φT1	φT4	φT16
1 (fs)	00 (f_{FPH})	XXX	15.6 ms	62.5 ms	250 ms	31.3 ms	125.0 ms	500 ms	62.5 ms	250ms	1000 ms
0 (fc)		000 (fc)	19.0 μs	75.9 μs	303.4 μs	37.9 μs	151.7 μs	606.8 μs	75.9 μs	303.4 μs	1311 μs
		001 (fc/2)	37.9 μs	151.7 μs	606.8 μs	75.9 μs	303.4 μs	1213.6 μs	151.7 μs	606.8 μs	2621 μs
		010 (fc/4)	75.9 μs	303.4 μs	1213.6 μs	151.7 μs	606.8 μs	2427.3 μs	303.4 μs	1213.6 μs	5243 μs
		011 (fc/8)	151.7 μs	606.8 μs	2427.3 μs	303.4 μs	1213.6 μs	4854.5 μs	606.8 μs	2427.3 μs	9709.0 μs
		100 (fc/16)	303.4 μs	1213.6 μs	4854.5 μs	606.8 μs	2427.3 μs	9709.0 μs	1213.6 μs	4854.5 μs	19418 μs
	10 (fc/16 clock)	XXX	303.4 μs	1213.6 μs	4854.5 μs	606.8 μs	2427.3 μs	9709.0 μs	1213.6 μs	4854.5 μs	19418 μs

XXX: Don't care

(5) Settings for each mode

Table 3.7.4 shows the SFR settings for each mode.

Table 3.7.4 Timer Mode Setting Registers

Register Name	TA01MOD				TA1FFCR
<Bit Symbol>	<TA01M1:0>	<PWM01:00>	<TA1CLK1:0>	<TA0CLK1:0>	TA1FFIS
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)	–
8-bit PWM × 1 channel	11	2^6, 2^7, 2^8 (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

–: Don't care

(6) Comparators (CP10 and CP11, CP12 and CP13)

CP0 and CP1 are 16-bit comparators which compare the value in the up counter UC0 value with the value set of TB0RG0H/L or TB0RG1H/L 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 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.

TMRB0 Flip-Flop Control Register

TB0FFCR (0183H)

Read-Modify-write instruction is prohibited

	7	6	5	4	3	2	1	0
Bit symbol	TB0FF1C1	TB0FF1C0	TB0C1T1	TB0C0T1	TB0E1T1	TB0E0T1	TB0FF0C1	TB0FF0C0
Read/Write	W*		R/W				W*	
After reset	1	1	0	0	0	0	1	1
Function	TB0FF1 control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11".		TB0FF0 inversion trigger 0: Trigger disable 1: Trigger enable				TB0FF0 Control 00: Invert 01: Set 10: Clear 11: Don't care * Always read as "11".	
			Invert when the UC10 value is loaded in to TB0CP1H/L.	Invert when the UC10 value is loaded in to TB0CP0H/L.	Invert when the UC10 matches with TB0RG1H/L.	Invert when the UC10 match with TB0RG0H/L.		

→ Timer Flip-Flop (TB0FF0) control

00	Invert to TB0FF0 (Software inversion).
01	Set TB0FF0 to "1".
10	Clear TB0FF0 to "0".
11	Don't care

→ Inversion trigger of TB0FF0 when the UC10 match with TB0RG0H/L

0	Trigger disable (Disable inversion)
1	Trigger enable (Enable inversion)

→ Inversion trigger of TB0FF0 when the UC10 match with TB0RG1H/L

0	Trigger disable (Disable inversion)
1	Trigger enable (Enable inversion)

→ Inversion trigger of TB0FF0 When the UC10 value is loaded in to TB0CP0H/L

0	Trigger disable (Disable inversion)
1	Trigger enable (Enable inversion)

→ Inversion trigger of TB0FF0 When the UC10 value is loaded in to TB0CP1H/L

0	Trigger disable (Disable inversion)
1	Trigger enable (Enable inversion)

Figure 3.8.6 Register for TMRB

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 interrupt. In the transfer interrupt program, The receiving operation should be done like the above example before setting the next transfer data.

Example: Channel 0, SCLK output
Baud rate = 9600 bps
fc = 14.7456 MHz

* Clock state

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

Main routine

	7	6	5	4	3	2	1	0	
									Set the INTTX0 level to 1.
INTES0	–	0	0	1	–	0	0	0	Set the INTRX0 level to 0.
P9CR	–	–	–	–	–	1	0	1	Set P90, P91 and P92 to function as the TXD0, RXD0 and SCLK0 pins respectively.
P9FC	X	X	–	X	–	1	X	1	
SC0MOD0	–	–	–	–	0	0	–	–	Select I/O interface mode.
SC0MOD1	1	1	X	X	X	X	X	X	Select full duplex mode.
SC0CR	–	–	–	–	–	–	0	–	SCLK output, transmit on negative edge, receive on positive edge
BR0CR	0	0	1	1	0	0	1	1	Baud rate = 9600 bps
SC0MOD0	–	–	1	–	–	–	–	–	Enable receiving
SC0BUF	*	*	*	*	*	*	*	*	Set the transmit data and start.

INTTX0 interrupt routine

Acc	SC0BUF								Read the receiving buffer.
SC0BUF	*	*	*	*	*	*	*	*	Set the next transmit data.

X: Don't care, –: No change

3.10.6 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 <SA6:0> and the <ALS> (<ALS> = 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 <BB> = 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 INTSBI 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 which 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 occurs 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.

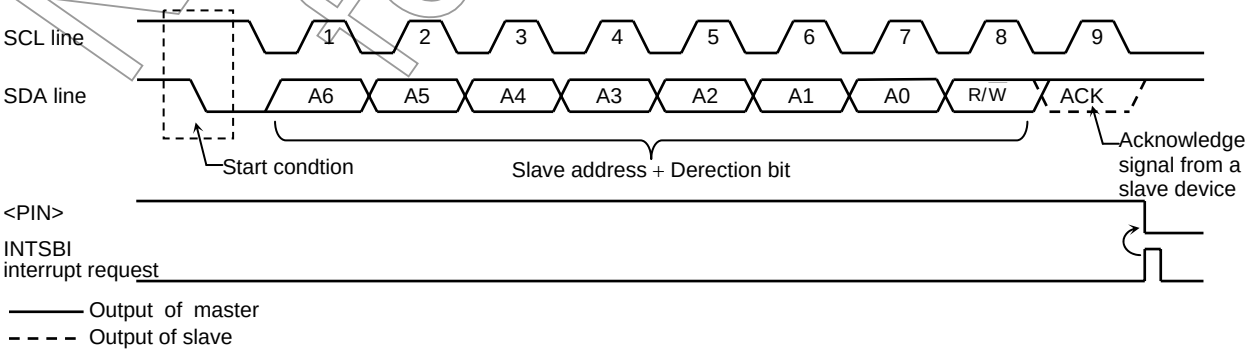


Figure 3.10.13 Start Condition Generation and Slave Address Transfer

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 occurs and the <PIN> becomes 0, Then the TMP91FY42F pulls down the SCL pin to the low level. The TMP91FY42 outputs a clock pulse for 1 word of data transfer and the acknowledge signal each time that received data is read from the SBI0DBR.

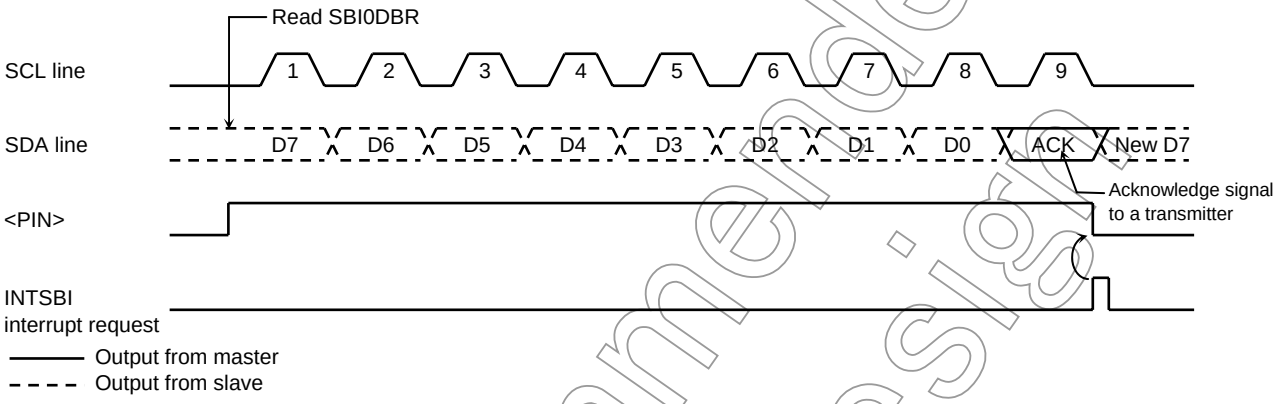


Figure 3.10.15 Example of when <BC2:0> = 000, <ACK> = 1 in Receiver Mode

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 BC<2:0> to 001 and read the data. The TMP91FY42 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 interprets 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 TMP91FY42 generates a stop condition (See Section 3.10.6 (4)) and terminates data transfer.

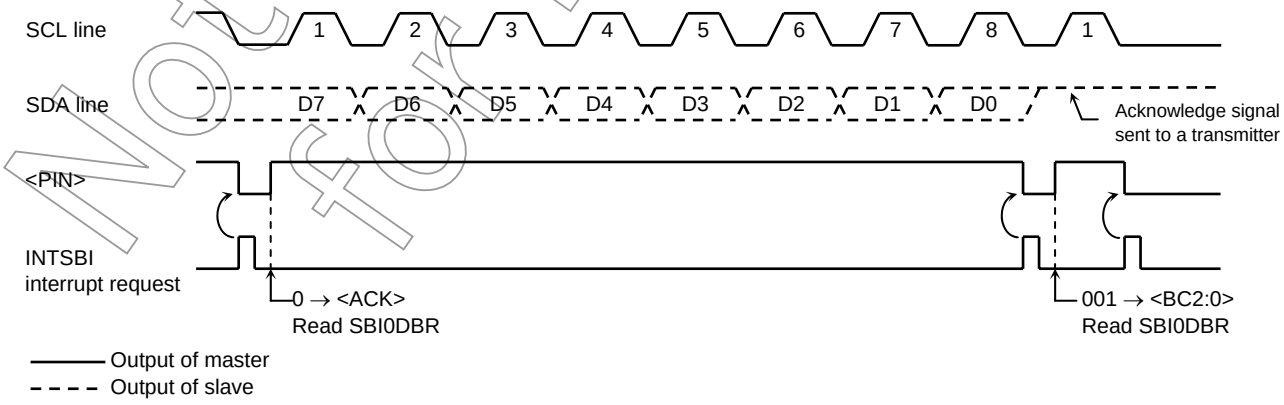


Figure 3.10.16 Termination of Data Transfer in Master Receiver Mode

(4) Stop condition generation

When SBI0SR<BB> = 1, the sequence for generating a stop condition can be initiated 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 TMP91FY42 generates a stop condition when the other device has released the SCL line.

When SBI0CR2<MST, TRX, PIN> are written 1 and <BB> is written 0, <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.

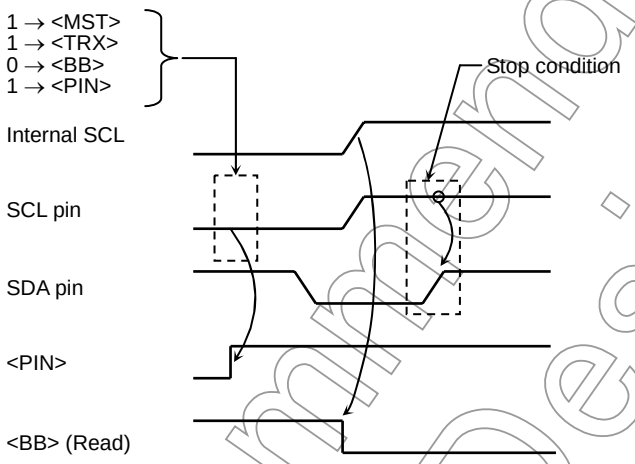


Figure 3.10.17 Stop Condition Generation (Single master)

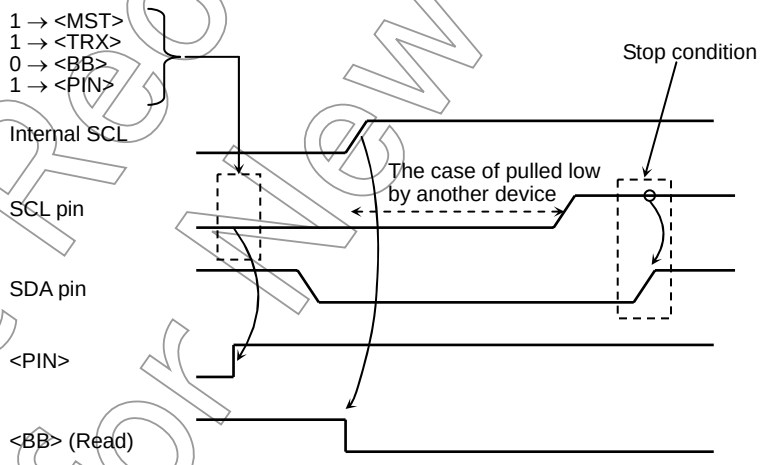


Figure 3.10.18 Stop Condition Generation (Multi master)

3.13 Special timer for CLOCK

The TMP91FY42 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.

Special timer for 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.

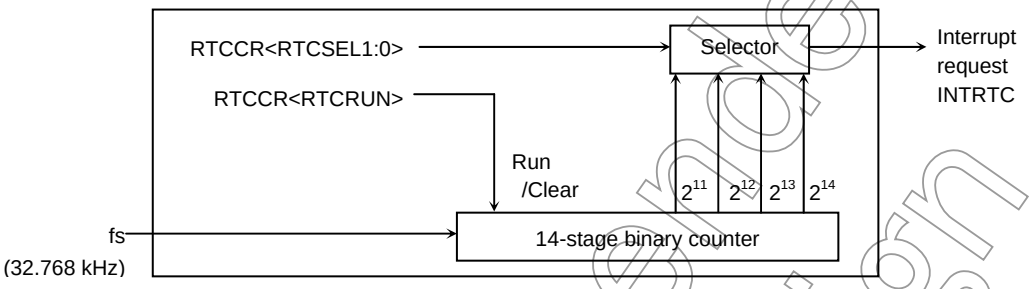


Figure 3.13.1 Block Diagram for Special timer for CLOCK

The Special timer for CLOCK is controlled by the Special timer for CLOCK control register (RTCCR) as shown in .

RTCCR (0310H)

	7	6	5	4	3	2	1	0
Bit symbol	–					RTCSEL1	RTCSEL0	RTCRUN
Read/Write	R/W					R/W		
After reset	0					0	0	0
Function	Always write "0".					00: 2^{14}/fs 01: 2^{13}/fs 10: 2^{12}/fs 11: 2^{11}/fs		0: Stop & clear 1: Count

Counting operation

0	Stop & clear
1	Count

Interrupt generation cycle (fs = 32.768 kHz)

00	0.50 s
01	0.25 s
10	0.125 s
11	0.0625 s

Figure 3.13.2 Special timer for CLOCK Control Register

Table 3.14.8 Transfer Format of Single Boot Program [RAM Transfer]

	Transfer Byte Number	Transfer Data from Controller to Device	Baud Rate	Transfer Data from Device to Controller
Boot ROM	1st byte	Baud rate setting UART 86H	Desired baud rate (Note 1)	—
	2nd byte	—		ACK response to baud rate setting Normal (baud rate OK) ・UART 86H (If the desired baud rate cannot be set, operation is terminated.)
	3rd byte	Operation command data (10H)		—
	4th byte	—		ACK response to operation command (Note 2) Normal 10H Error x1H Protection applied (Note 4) x6H Communications error x8H
	5th byte to 16th byte	Password data (12 bytes) (04FEF4H to 04FEFFH)		—
	17th byte	CHECKSUM value for 5th to 16th bytes		—
	18th byte	—		ACK response to CHECKSUM value (Note 2) Normal 10H Error 11H Communications error 18H
	19th byte	RAM storage start address 31 to 24 (Note 3)		—
	20th byte	RAM storage start address 23 to 16 (Note 3)		—
	21st byte	RAM storage start address 15 to 8 (Note 3)		—
	22nd byte	RAM storage start address 7 to 0 (Note 3)		—
	23rd byte	RAM storage byte count 15 to 8 (Note 3)		—
	24th byte	RAM storage byte count 7 to 0 (Note 3)		—
	25th byte	CHECKSUM value for 19th to 24th bytes (Note 3)		—
	26th byte	—		ACK response to CHECKSUM value (Note 2) Normal 10H Error 11H Communications error 18H
	27th byte to m'th byte	RAM storage data		—
	(m+1)th byte	CHECKSUM value for 27th to m'th bytes		—
	(m+2)th byte	—		ACK response to CHECKSUM value (Note 2) Normal 10H Error 11H Communications error 18H
RAM	(m+3)th byte	—		Jump to RAM storage start address

Note 1: For the desired baud rate setting, see Table 3.14.6.
Note 2: After sending an error response, the device waits for operation command data (3rd byte).
Note 3: The data to be transferred in the 19th to 25th bytes should be programmed within the RAM address range of 001000H to 004DFFH (15.8 Kbytes).
Note 4: When read protection or write protection is applied, the device aborts the received operation command and waits for the next operation command data (3rd byte).

3.14.4.16 Password

When the RAM Transfer command (10H) or the Flash Memory Protect Set command (60H) is received as operation command data, password verification is performed. First, the device echoes back the operation command data (10H to 60H) and checks the data (12 bytes) in the password area (addresses 04FEF4H to 04FEFFH).

Then, the device verifies the password data received in the 5th to 16th bytes against the data in the password area as shown in Table 3.14.22.

Unless all the 12 bytes are verified correctly, a password error will occur.

A password error will also occur if all the 12 bytes of password data contain the same value. Only exception is when all the 12 bytes are "FFH" and verified correctly and the reset vector area (addresses 04FF00H to 04FF02H) is all "FFH". In this case, a blank device will be assumed and no password error will occur.

If a password error has occurred, the device returns the ACK response data for password error in the 18th byte.

Table 3.14.22 Password Verification Table

Receive data	Data to be verified against
5th byte	Data at address 04FEF4H
6th byte	Data at address 04FEF5H
7th byte	Data at address 04FEF6H
8th byte	Data at address 04FEF7H
9th byte	Data at address 04FEF8H
10th byte	Data at address 04FEF9H
11th byte	Data at address 04FEFAH
12th byte	Data at address 04FEFBH
13th byte	Data at address 04FEFCH
14th byte	Data at address 04FEFDH
15th byte	Data at address04FEFEH
16th byte	Data at address 04FEFFH

Example of data that cannot be specified as a password

For blank products (Note)

・The password of a blank product must be all "FFH" (FFH, FFH, FFH, FFH, FFH, FFH, FFH, FFH, FFH, FFH, FFH, FFH).

Note: A blank product is a product in which all the bytes in the password area (addresses 04FEF4H to 04FEFFH) and the reset vector area (addresses 04FF00H to 04FF02H) are "FFH".

For programmed products

・The same 12 consecutive bytes cannot be specified as a password.

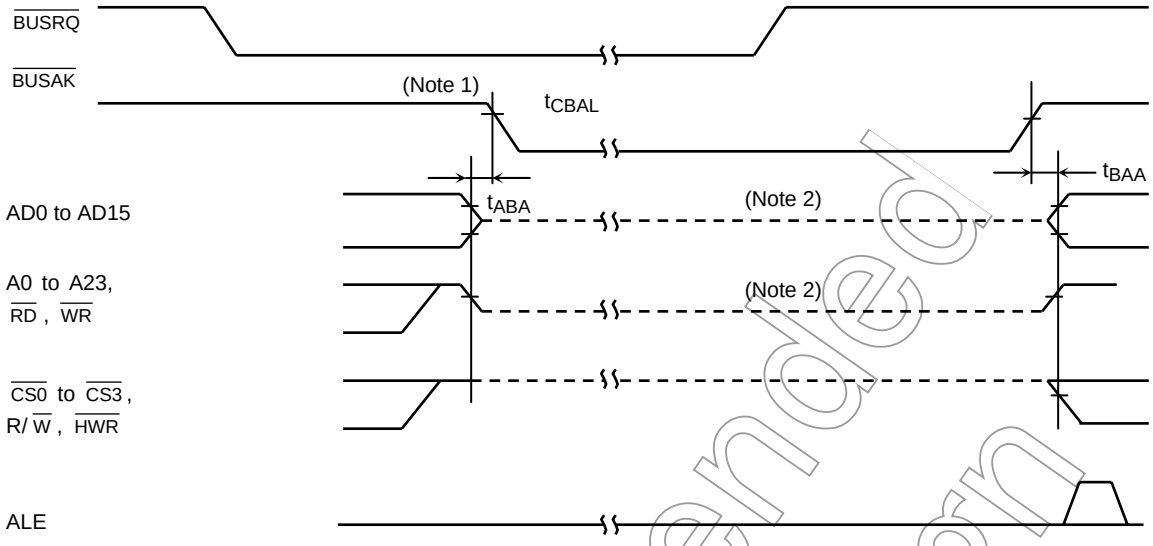
The table below shows password error examples.

Programmed product	1	2	3	4	5	6	7	8	9	10	11	12	Note
Error example 1	FFH	FFH	FFH	FFH	FFH	FFH	FFH	FFH	FFH	FFH	FFH	FFH	All "FF"
Error example 2	00H	00H	00H	00H	00H	00H	00H	00H	00H	00H	00H	00H	All "00"
Error example 3	5AH	5AH	5AH	5AH	5AH	5AH	5AH	5AH	5AH	5AH	5AH	5AH	All "5A"

(Example: Program to be loaded and executed in RAM)
Erase data at addresses 20000H to 20FFFH (sector erase) and then write 0706H to address 20000H.

```
;#### Flash memory sector erase processing ####
        ld      XIX, 0x20000                ; set start address
SECTORERASE:
        ld      (0x10AAA), 0xAA             ; 1st bus write cycle
        ld      (0x10554), 0x55             ; 2nd bus write cycle
        ld      (0x10AAA), 0x80             ; 3rd bus write cycle
        ld      (0x10AAA), 0xAA             ; 4th bus write cycle
        ld      (0x10554), 0x55             ; 5th bus write cycle
        ld      (XIX), 0x30                 ; 6th bus write cycle

        cal     TOGGLECHK                   ; check toggle bit

SECTORERASE_LOOP:
        ld      WA, (XIX+)                  ; read data from flash memory
        cp      WA, 0xFFFF                  ; blank data?
        j       ne, SECTORERASE_ERR         ; if not blank data, jump to error processing
        cp      XIX, 0x20FFF                ; end address (0x20FFF)?
        j       ULT, SECTORERASE_LOOP       ; check erased sector area and then end loop processing


;#### Flash memory program processing ####
        ld      XIX, 0x20000                ; set program address
        ld      WA, 0x0706                  ; set program data
PROGRAM:
        ld      (0x10AAA), 0xAA             ; 1st bus write cycle
        ld      (0x10554), 0x55             ; 2nd bus write cycle
        ld      (0x10AAA), 0xA0             ; 3rd bus write cycle
        ld      (XIX), WA                   ; 4th bus write cycle

        cal     TOGGLECHK                   ; check toggle bit

        ld      BC, (XIX)                   ; read data from flash memory
        cp      WA, BC
        j       ne, PROGRAM_ERR             ; if programmed data cannot be read, error is determined
        ld      BC, (XIX)                   ; read data from flash memory
        cp      WA, BC
        j       ne, PROGRAM_ERR             ; if programmed data cannot be read, error is determined

PROGRAM_END:
        j       PROGRAM_END                 ; program operation end


;#### Toggle bit (D6) check processing ####
TOGGLECHK:
        ld      L, (XIX)
        and     L, 0y01000000               ; check toggle bit (D6)
        ld      H, L                        ; save first toggle bit data
TOGGLECHK1:
        ld      L, (XIX)
        and     L, 0y01000000               ; check toggle bit (D6)
        cp      L, H                        ; toggle bit = toggled?
        j       z, TOGGLECHK2               ; If not toggled, end processing
        ld      H, L                        ; save current toggle bit state
        j       TOGGLECHK1                  ; Recheck toggle bit
TOGGLECHK2:
        ret


;#### Error processing ####
SECTORERASE_ERR:
        j       SECTORERASE_ERR             ;  sector erase error

PROGRAM_ERR:
        j       PROGRAM_ERR                 ; program error
```

4.9 Bus Request/Bus Acknowledge



Parameter	Symbol	Variable		f_{FPH} = 10 MHz		f_{FPH} = 27 MHz		Condition	Unit
		Min	Max	Min	Max	Min	Max		
Output buffer off to $\overline{BUSAK}$ low	t_{ABA}	0	80	0	80	0	80	Vcc = 2.7V to 3.6V	ns
$\overline{BUSAK}$ high to output buffer on	t_{BAA}	0	80	0	80	0	80	Vcc = 2.7V to 3.6V	ns

Note 1: Even if the $\overline{BUSRQ}$ signal goes low, the bus will not be released while the $\overline{WAIT}$ signal is low. The bus will only be released when $\overline{BUSRQ}$ goes low while $\overline{WAIT}$ is high.

Note 2: This line shows only that the output buffer is in the off state.
It does not indicate that the signal level is fixed.
Just after the bus is released, the signal level set before the bus was released is maintained dynamically by the external capacitance. Therefore, to fix the signal level using an external resister during bus release, careful design is necessary, since fixing of the level is delayed. The internal programmable pull-up/pull-down resistor is switched between the active and non-active states by the internal signal.

4.10 Flash Characteristics

(1) Rewriting

Parameter	Condition	Min	Typ	Max	Unit
Gurantee on Flash-memory rewriting	Vcc = 3.0V to 3.6V, fc = 4 to 27 MHz Ta = -10~40°C	—	—	100	Times

Clock gear (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
EMCCR0	EMC control register 0	E3H	PROTECT	–	–	–	ALEEN	EXTIN	DRVOSCH	DRVOSCL
			R	R/W						
			0	0	1	0	0	0	1	1
			Protection flag 0: OFF 1: ON	Always write “0”	Always write “1”	Always write “0”	1: ALE output enable	1: fc external clock	fc oscillator driver ability 1: Normal 0: Weak	fs oscillator driver ability 1: Normal 0: Weak
EMCCR1	EMC control register 1	E4H	Writing 1FH turns protections off. Writing any value other than 1FH turns protection on.							

Note: EMCCR1

When protect-ON is set to EMCCR1, It is prohibited that following SFRs are written.
(SFR that cannot write)

1. CS/WAIT controller
   B0CS, B1CS, B2CS, B3CS, BEXCS,
   MSAR0, MSAR1, MSAR2, MSAR3,
   MAMR0, MAMR1, MAMR2, MAMR3
2. Clock gear (only EMCCR1 is available to write)
   SYSCR0, SYSCR1, SYSCR2, EMCCR0
3. DFM
   DFMCR0

(6) DFM (Clock doubler)

Symbol	Name	Address	7	6	5	4	3	2	1	0
DFMCR0	DFM control register 0	E8H	ACT1	ACT0	DLUPFG	DLUPTM				
			R/W		R	R/W				
			0	0	0	0				
			Always write “0”							
DFMCR1	DFM control register 1	E9H	–	–	–	–	–	–	–	–
			R/W							
			0	0	0	1	0	0	1	1
			Don’t access this register							

Note: TMP91FY42 does not built-in Clock Doubler (DFM).

UART/Serial chanel (2/2)

(9-3) UART/SIO Channel1

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC1BUF	Serial channel 1 buffer	208H (Prohibit RMW)	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
			R (Receiving)/W (Transmission)							
			Undefined							
SC1CR	Serial channel 1 control	209H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Cleared to 0 when read)			R/W	
			Undefined	0	0	0	0	0	0	0
			Received data bit8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: Error Overrun	1: Error Parity	1: Error Framing	0: SCLK0↑ 1: SCLK0↓	0: Baud rate generator 1: SCLK1 pin input
SC1MOD0	Serial channel 1 mode	20AH	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Transmission data bit8	Handshake 0: CTS disable 1: CTS enable	Receive function 0: Receive disable 1: Receive enable	Wakeup function 0: Disable 1: Enable	Serial transmission mode 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial transmission clock (UART) 00: TA0TRG 01: Baud rate generator 10: Internal clock f_{SYS} 11: External clock SCLK1	
BR1CR	Baud rate control	20BH	–	BR1ADDE	BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
			R/W							
			0	0	0	0	0	0	0	0
			Always write “0”.	+ (16 − K)/16 division 0: Disable 1: Enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Divided frequency setting			
BR1ADD	Serial channel 1 K setting register	20CH					BR1K3	BR1K2	BR1K1	BR1K0
							R/W			
							0	0	0	0
							Set frequency divisor K (divided by N + (16 − K)/16).			
SC1MOD1	Serial channel 1 mode 1	20DH	I2S1	FDPX1						
			R/W							
			0	0						
			IDLE2 0: Stop 1: Operate	Duplex 0: Half 1: Full						