Data Book

16bit Micro controller
TLCS-900/L1 series

TMP91C824F

REV2.6 July. 12, 2002

Table of Contents

TLCS-900/L1 Devices

TMP91C824F

Data Book modification history

Rev./Date	page	Modification item	Reason
Rev12/Mar-15-2001	109	Bank Operation S/W Example in MMU	
	218	Recommended Crystal oscillation Circuit	
	165	Stop condition generation	
	166	Timing diagram for TMP91C824 Restart	
Rev14/July-16-2001	12-24	DFM(triple cock modify)	
Rev14/July-26-2001	17,240	EMCCR3	
	159,168	Add ed the sentence and diagram at SBI	
	155	Modify bit6(TRX)	
	114	Added the diagram and sentence of MMU	
Rev21/August-21-2001	14	CPU: modify the figure	
Rev22/September-21-2001	222	Electric Specification	
Rev22/September-7-2001	226	Murata factory URL added	
Rev23/October-26-2001	217	33MHZ AC added	
Rev24/January-18-2002	216	Syntax err “2V SLOW ICC” -> “2V NORMAL ICC” Unit “uA” -> “mA”	
	221	Spec modify AD converter IREF max 1.20mA -> 1.35mA	
Rev2.5/June-21-2002	contents	Remove “TENTATIVE” of ELECTRICAL CHARACTERISTICS	
	1	Add the comment about JTMP91C824 Modify the frequency of Micro DMA	Change the reference
	2	Add the comment about package	Mistake
	16,240	Modify the input frequency of (DFMCR1)	Mistake
	29	Modify the name SBI0BR1→ SBI0BR0 in the table 3.3.2	Mistake
	30,251	Add the limitation about HALT mode	Mistake ***Important matter**
	48,236	Add the comment to the register IIMC bit6	Mistake
	55	Modify the comment in the figure	Mistake
	87	Add the figure for connecting SRAM	Shortage
	97,98	Modify W→R/W and remove the note	Mistake
	100,104 106	Modify the name of the register in the program P7→PB	Mistake
	146	Modify the name of the register in the program P9→PC	Mistake
	208	Add the limitation about RTC	Mistake ***Important matter**
	212	Modify the name of scale in the table	Change expression
	226	Change the comment about MURATA at the bottom	Change expression
	240	Remove the note	Mistake
Rev2.6/July-12-2002	15,239	Add the word “IDLE1” to the bit0 of (SYSCR2) register	Mistake
	30,251	Add the INTALM0 to 4 in the(note)	Mistake

※Each page number depends on the revision when it's made.

CMOS 16-Bit Microcontrollers
TMP91C824F/JTMP91C824-S

1. OUTLINE AND FEATURES

TMP91C824F is a high-speed 16-bit microcontroller designed for the control of various mid- to large-scale equipment.
TMP91C824F comes in a 100-pin flat package and a 100-pad Dice form. JTMP91C824-S is a chip form product.

Listed below are the features.

(1) High-speed 16-bit CPU (900/L1 CPU)

- Instruction mnemonics are upward-compatible with TLCS-90
- 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 (0.5 μs/2 bytes at 33 MHz)

(2) Minimum instruction execution time: 121 ns (at 33 MHz)

(3) Built-in RAM: 8 Kbytes
Built-in ROM: None

(4) External memory expansion

- Expandable up to 106M bytes (shared program/data area)
- Can simultaneously support 8-/16-bit width external data bus
  ··· Dynamic data bus syzing
- Separate bus system

(5) 8-bit timers: 4 channels

(6) General-purpose serial interface: 2 channels

- UART/Synchronous mode: 2 channels
- IrDA Ver.1.0 (115.2kbps) mode selectable: 1 channel

(7) Serial bus interface: 1 channel

- I2C bus mode/clock synchronous mode selectable

980508TBA1

- For a discussion of how the reliability of microcontrollers can be predicted, please refer to Section 1.3 of the chapter entitled Quality and Reliability Assurance / Handling Precautions.
- TOSHIBA is continually working to improve the quality and the 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 observe standards of safety, and to avoid situations in which a malfunction or failure of a TOSHIBA product could cause loss of human life, bodily injury or damage to property. In developing your designs, please ensure that TOSHIBA products are used within specified operating ranges as set forth in the most recent products specifications. Also, please keep in mind the precautions and conditions set forth in the TOSHIBA Semiconductor Reliability Handbook.
- The products described in this document are subject to foreign exchange and foreign trade laws.
- The information contained herein is presented only as a guide for the applications of our products. No responsibility is assumed by TOSHIBA CORPORATION for any infringements of intellectual property or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any intellectual property or other rights of TOSHIBA CORPORATION or others.
- The information contained herein is subject to change without notice.

(8) Timer for real-time clock (RTC)

- Based on TC8521A

(9) 10-bit A/D converter : 8 channels

(10) Watch dog timer

(11) Melody/Alarm generator

- Melody: Output of clock 4 to 5461Hz
- Alarm: Output of the 8 kinds of alarm pattern
- Output of the 5 kinds of interval interrupt

(12) Chip select/Wait controller: 4 channels

(13) Memory Management Unit

- Expandable up to 106M bytes (4 local area/8bank method)

(14) Interrupts: 37 interrupts

- 9 CPU interrupts: Software interrupt instruction and illegal instruction
- 23 internal interrupts: 7 priority levels are selectable
- 5 external interrupts: 7 priority levels are selectable (among 4 interrupts are selectable edge mode)

(15) Input/output ports: 35 pins (@External 16-bit data bus memory)

(16) Stand-by function
Three Halt modes: Idle2 (programmable), Idle1 and Stop

(17) Triple-clock controller

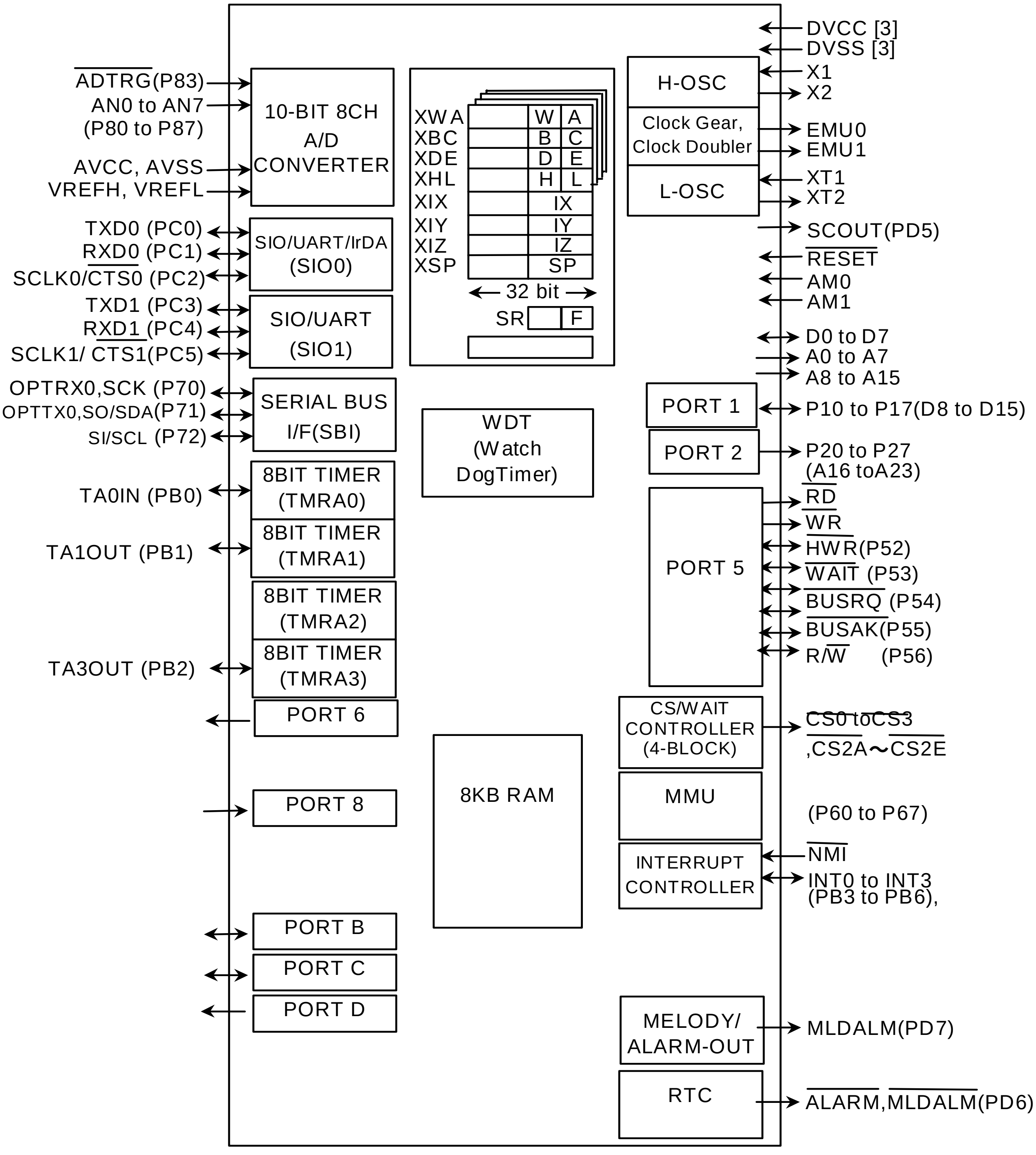
- Clock doubler (DFM) circuit is inside
- Clock gear function: Select a High-frequency clock fc/1 to fc/16
- RTC (fs=32.768kHz)

(18) Operating voltage

- VCC = 2.7 V to 3.6 V (fc max = 33 MHz)
- VCC = 1.8 V to 3.6 V (fc max = 10 MHz)

(19) Package

- 100-pin QFP: LQFP100 - P -1414 - 0.5D
- Chip form supply also available. For details, contact your local Toshiba sales representative.



( ) : Initial Function After Reset

Figure 1.1 TMP91C824F Block Diagram

2. PIN ASSIGNMENT AND PIN FUNCTIONS

The assignment of input/output pins for the TMP91C824F, their names and functions are as follows:

2.1 Pin Assignment Diagram

Figure 2.1.1 shows the pin assignment of the TMP91C824F.

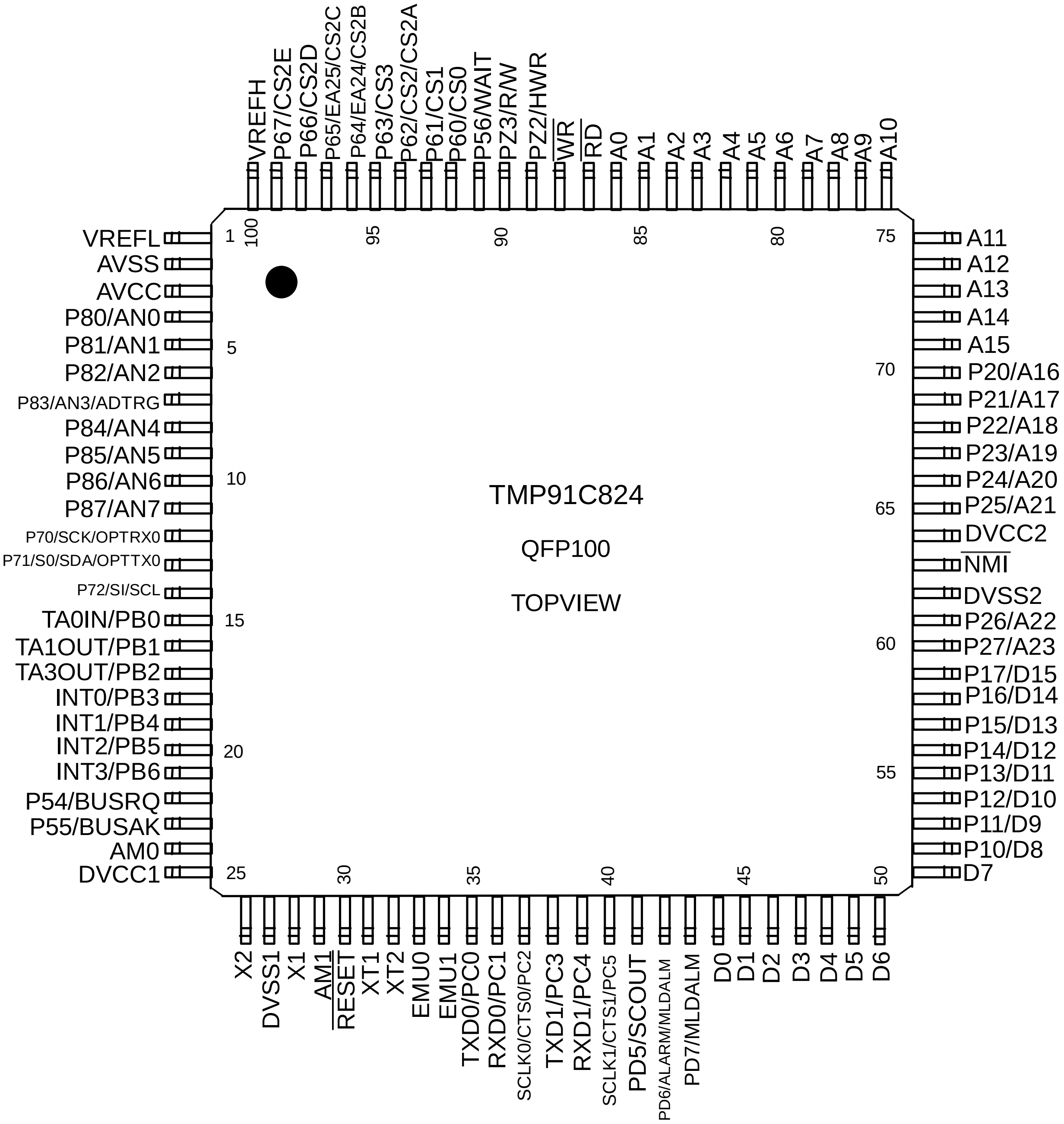


Figure 2.1.1 Pin assignment diagram (100-pin QFP)

2.2 Pad Layout

(Chip size 4.37mm ×4.37mm) unit (μm)

PIN	Name	X	Y	PIN	Name	X	Y	PIN	Name	X	Y
no		point	point	No		point	point	No		point	point
1	VREFL	-2050	1721	35	PC0	-140	-2050	69	P21	2045	939
2	AVSS	-2050	1596	36	PC1	-14	-2050	70	P20	2045	1075
3	AVCC	-2050	1470	37	PC2	112	-2050	71	A15	2045	1207
4	P80	-2050	1337	38	PC3	238	-2050	72	A14	2045	1337
5	P81	-2050	1209	39	PC4	365	-2050	73	A13	2045	1464
6	P82	-2050	1076	40	PC5	491	-2050	74	A12	2045	1592
7	P83	-2050	943	41	PD5	618	-2050	75	A11	2045	1721
8	P84	-2050	810	42	PD6	744	-2050	76	A10	1720	2045
9	P85	-2050	677	43	PD7	871	-2050	77	A9	1591	2045
10	P86	-2050	544	44	D0	998	-2050	78	A8	1464	2045
11	P87	-2050	416	45	D1	1124	-2050	79	A7	1337	2045
12	P70	-2050	148	46	D2	1251	-2050	80	A6	1197	2045
13	P71	-2050	15	47	D3	1377	-2050	81	A5	1058	2045
14	P72	-2050	-118	48	D4	1504	-2050	82	A4	918	2045
15	PB0	-2050	-251	49	D5	1630	-2050	83	A3	778	2045
16	PB1	-2050	-384	50	D6	1757	-2050	84	A2	639	2045
17	PB2	-2050	-517	51	D7	2045	-1750	85	A1	499	2045
18	PB3	-2050	-650	52	P10	2045	-1614	86	A0	359	2045
19	PB4	-2050	-783	53	P11	2045	-1478	87	/RD	219	2045
20	PB5	-2050	-916	54	P12	2045	-1341	88	/WR	80	2045
21	PB6	-2050	-1049	55	P13	2045	-1205	89	PZ2	-59	2045
22	P54	-2050	-1182	56	P14	2045	-1069	90	PZ3	-199	2045
23	P55	-2050	-1315	57	P15	2045	-933	91	P56	-338	2045
24	AM0	-2050	-1448	58	P16	2045	-796	92	P60	-478	2045
25	VCC	-2050	-1581	59	P17	2045	-660	93	P61	-618	2045
26	X2	-1551	-2050	60	P27	2045	-524	94	P62	-757	2045
27	VSS	-1330	-2050	61	P26	2045	-388	95	P63	-897	2045
28	X1	-1205	-2050	62	VSS	2045	-234	96	P64	-1037	2045
29	AM1	-1075	-2050	63	/NMI	2045	-80	97	P65	-1176	2045
30	/RESET	-948	-2050	64	VCC	2045	240	98	P66	-1316	2045
31	XT1	-822	-2050	65	P25	2045	394	99	P67	-1456	2045
32	XT2	-520	-2050	66	P24	2045	530	100	VREFH	-1725	2045
33	EMU0	-394	-2050	67	P23	2045	666				
34	EMU1	-267	-2050	68	P22	2045	803				

2.3 Pin Names and Functions

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

Table 2.2 Pin names and functions.

Pin Name	Number of Pins	I/O	Functions
D0 to D7	8	I/O	Data (lower): bits 0 to 7 of data bus
P10 to P17 D8 to D15	8	I/O I/O	Port 1: I/O port that allows I/O to be selected at the bit level (When used to the external 8bit bus) Data (upper): bits 8 to15 of data bus
P20 to P27 A16 to A23	8	Output Output	Port 2: Output port Address: bits 16 to 23 of address bus
A8 to A15	8	Output	Address: bits 8 to 15 of address bus
A0 to A7	8	Output	Address: bits 0 to 7 of address bus
/RD	1	Output	Read: strobe signal for reading external memory
/WR	1	Output	Write: strobe signal for writing data to pins D0 to D7
PZ2 /HWR	1	I/O Output	Port Z2: I/O port (with pull-up resistor) High Write: strobe signal for writing data to pins D8 to D15
PZ3 $\overline{\text{R/W}}$	1	I/O Output	Port Z3: I/O port (with pull-up resistor)
P54 /BUSRQ	1	I/O Input	Port 54: I/O port (with pull-up resistor) Bus Request: High-Impedance used to request Bus Release Read/Write: 1 represents Read or Dummy cycle; 0 represents write cycle.
P55 /BUSAK	1	I/O Output	Port 55: I/O port (with pull-up resistor) Bus Acknowledge: signal used to acknowledge Bus Release
P56 /WAIT	1	I/O Input	Port 56: I/O port (with pull-up resistor) Wait: pin used to request CPU bus wait
P60 /CS0	1	Output Output	Port 60:Output port Chip select 0: Outputs “0” when address is within specified address area.
P61 /CS1	1	Output Output	Port 61:Output port Chip Select 1: outputs “0” when address is within specified address area
P62 /CS2 /CS2A	1	Output Output Output	Port 62: Output port Chip Select 2: outputs “0” when address is within specified address area Expand Chip Select: 2A: outputs 0 when address is within specified address area
P63 /CS3	1	Output Output	Port 63:Output port Chip Select 3: outputs “0” when address is within specified address area
P64 EA24 /CS2B	1	Output Output Output	Port 64: Output port (Address 24: Expand address) (Expand Chip Select: 2B: outputs “0” when address is within specified address area)
P65 EA25 /CS2C	1	Output Output Output	Port 65: Output port (Address 25: Expand address) (Expand Chip Select: 2B: outputs “0” when address is within specified address area)
P66 /CS2D	1	Output Output	Port 66: Output port (Expand Chip Select: 2D: outputs “0” when address is within specified address area)
P67 /CS2E	1	Output Output	Port 67: Output port (Expand Chip Select: 2E: outputs “0” when address is within specified address area)

.

Pin Name	Number of Pins	I/O	Functions
P70 SCK OPTRX0	1	I/O I/O Input	Port 70: I/O port Serial bus interface clock I/O data at SIO mode Serial recive data “0”
P71 S0 SDA OPTTX0	1	I/O Output I/O Output	Port 71: I/O port Serial bus interface send data at SIO mode Serial bus interface send/recive data at I2C mode Open drain output mode by programmable (with pull up) Serial send data “0”
P72 SI SCL	1	I/O Input I/O	Port 72I/O port Serial bus interface recive data at SIO mode Serial bus interface clock I/O data at I2C mode Open drain output mode by programmable (with pull up)
P80 to P87 AN0 to AN7 /ADTRG	8	Input Input Input	Port 80 to 87 port: Pin used to input ports Analog input 0 to 7: Pin used to Input to A/D conveter A/D trigger: Signal used to request A/D start (with used to P83)
PB0 TA0IN	1	I/O Input	Port B0: I/O port 8bit timer 0 input: Timer 0 input
PB1 TA1OUT	1	I/O Output	Port B1: I/O port 8bit timer 1 output: Timer 0 input or Timer 1 output
PB2 TA3OUT	1	I/O Output	Port B2: I/O port 8bit timer 3 output: Timer 2 input or Timer 3 output
PB3 INT0	1	I/O input	Port B0: I/O port Interrupt request pin0: Interrupt request pin with programmable level / rising / falling edge
PB4 to PB6 INT1 to INT3	3	I/O input	Port B4 to B6: I/O port Interrupt request pin1 to 3: Interrupt request pin with programmable level / rising /falling edge
PC0 TXD0	1	I/O Output	Port C0: I/O port Serial 0 send data: Open drain output pin by programmable
PC1 RXD0	1	I/O Input	Port C1: I/O port Serial 0 recive data

Pin Name	Number of Pins	I/O	Functions
PC2 SCLK0 /CTS0	1	I/O I/O Input	Port C2: I/O port Serial clock I/O 0 Serial data send enable 0 (Clear to Send)
PC3 TXD1	1	I/O Output	Port C3: I/O port Serial send data 1 Open drain output pin by programmable
PC4 RXD1	1	I/O Input	Port C4: I/O port Serial recive data 1
PC5 SCLK1 /CTS1	1	I/O I/O Input	Port C5: I/O port Serial clock I/O 1 Serial data send enable 1 (Clear to Send)
XT1	1	Input	Low Frequency Oscillator connecting pin
XT2	1	Output	Low Frequency Oscillator connecting pin
PD5 SCOUT	1	Output Output	Port D5: Output port System clock output: f_{SYS} or f_S output
PD6 /ALARM /MLDALM	1	Output Output Output	Port D6: Output port RTC alarm output pin
PD7 MLDALM	1	Output Output	Port D7: Output port Melody / Alarm output pin
/NMI	1	Input	Non-Maskable Interrupt Request Pin: interrupt request pin with programmable falling edge level or with both edge levels programmable
AM0 to 1	2	Input	Operation mode: Fixed to AM1=”0”,AM0=”1” 16-bit external bus or 8/16-bit dynamic sizing. Fixed to AM1=”0”,AM0=”0” 8-bit external bus fixed.
EMU0	1	Output	Open pin
EMU1	1	Output	Open pin
/RESET	1	Input	Reset: initializes TMP91C824. (With pull-up resistor)
VREFH	1	Input	Pin for reference voltage input to AD converter (H)
VREFL	1	Input	Pin for reference voltage input to AD converter (L)
AVCC	1		High-frequency oscillator connection pins
AVSS	1		Power supply pin for AD converter
X1/X2	2		GND pin for AD converter (0 V)
DVCC	3		Power supply pins (All Vcc pins should be connecyed with the power Supply pin).
DVSS	3		GND pins (0 V) (All pins shuold be connected with GND(0V).

3. OPERATION

This following describes block by block the functions and operation of the TMP91C824F.
Notes and restrictions for eatch book are outlined in " 7, Precautions and Restrictions at the end of this manual.

3.1 CPU

The TMP91C824 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 TMP91C824; these functions are not covered in the TLCS-900/L1 CPU section.

3.1.1 Reset

When resetting the TMP91C824 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 /RESET input to Low level at least for 10 system clocks (ten states: 80 µs at 4 MHz).

After Reset, Clock doubler circuit is set to x1 mode, and also Clock gear is set to x1/16 mode. It means that the initial clock mode starts x1/64 speed mode against the maximum speed of TMP91C824.

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<0 to 7> ← value at FFFF00H address
  PC<15 to 8> ← value at FFFF01H address
  PC<23 to 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).
  (Note: As this product does not support MIN mode, do not write a 0 to the <MAX>)
- 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.

(Note1) The CPU internal register (except to PC,SR,XSP) and internal RAM data do not change by resetting.

Figure 3.1.1 is a reset timing chart of the TMP91C824.

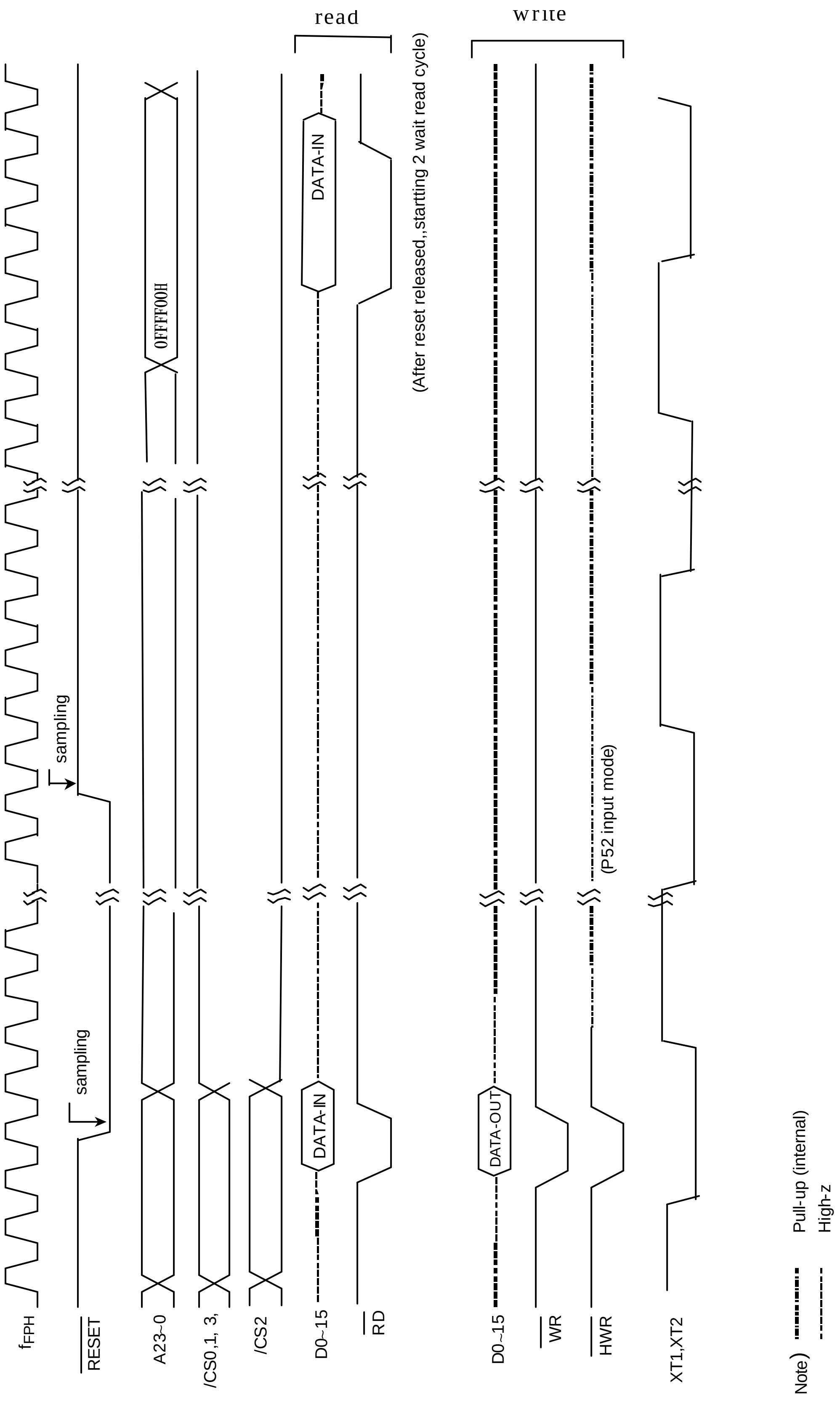


Figure 3.1.1 TMP91C824 Reset Timing Chart

3.2 Memory Map

Figure 3.2.1 is a memory map of the TMP91C824.

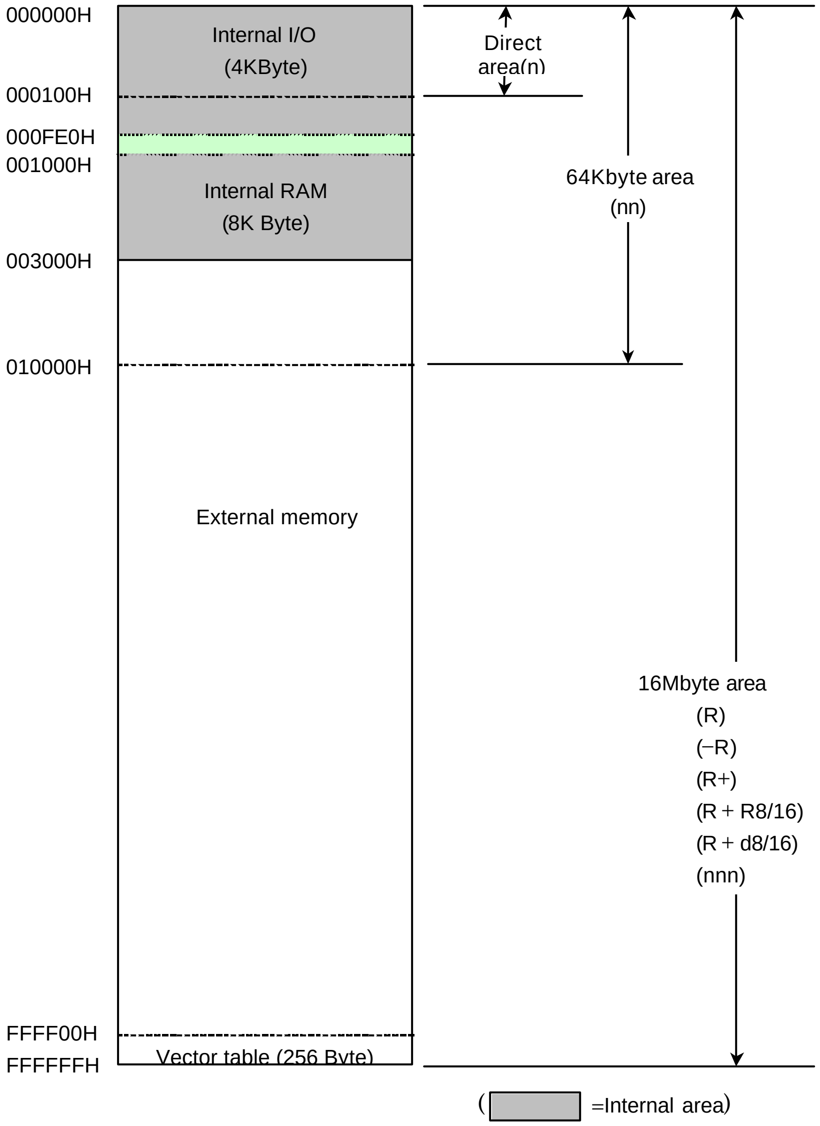


Figure 3.2 1 Memory Map

Note : Address 000FE0H – 00FFFH is assigned for the TOSHIBA reserve area, user can’t use.

3.3 Triple Clock Function and Standby Function

TMP91C824 contains (1) a clock gear, (2) clock doubler (DFM), (3) stand-by controller and (4) noise-reduction circuit. It is used for low-power, low-noise systems.
This chapter is organized as follows:

3.3.1 Block diagram of system clock

3.3.2 SFRs

3.3.3 System clock controller

3.3.4 Prescaler clock controller

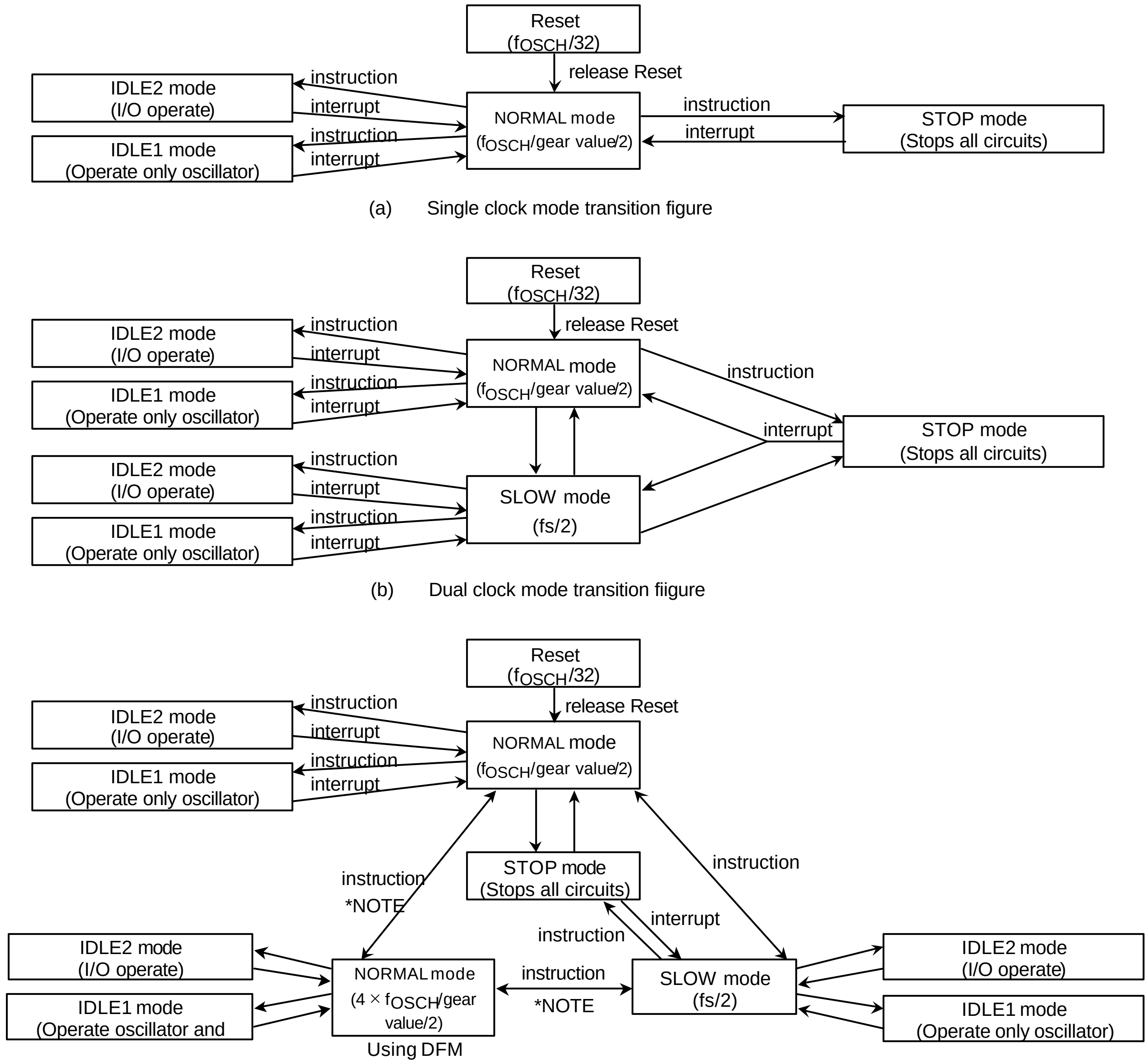
3.3.5 Clock doubler (DFM)

3.3.6 Noise-reducing circuit

3.3.7 Stand-by controller

The clock operating modes are as follows: (a) Single Clock Mode (X1, X2 pins only), (b) Dual Clock Mode (X1, X2, XT1 and XT2 pins) and (c) Triple Clock Mode (the X1, X2, XT1 and XT2 pins and DFM).

Figure 3.3.1 shows a transition figure.



(a) Single clock mode transition figure

(b) Dual clock mode transition fiigure

(c) Triple clock mode trasision Figure

***NOTE)**

- It’s prohibited to control DFM in SLOW mode when shifting from SLOW mode to NORMAL mode with use of DFM.（DFM Start up/Stop/Change Write to DFMCR0<ACT1:0> resister）
- If you shift from NORMAL mode with use of DFM to NORMAL mode, the instruction should be separated into two procedures as below. Change CPU clock ->Stop DFM circuit
- It’s prohibited to shift from NORMAL mode with use of DFM to STOP mode directly. You should set NORMAL mode once, and then shift to STOP mode.(You should stop high frequency oscillator after you stop DFM.)

Figure 3.3.1 System clock block diagram

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

3.3.1 Block diagram of system clock

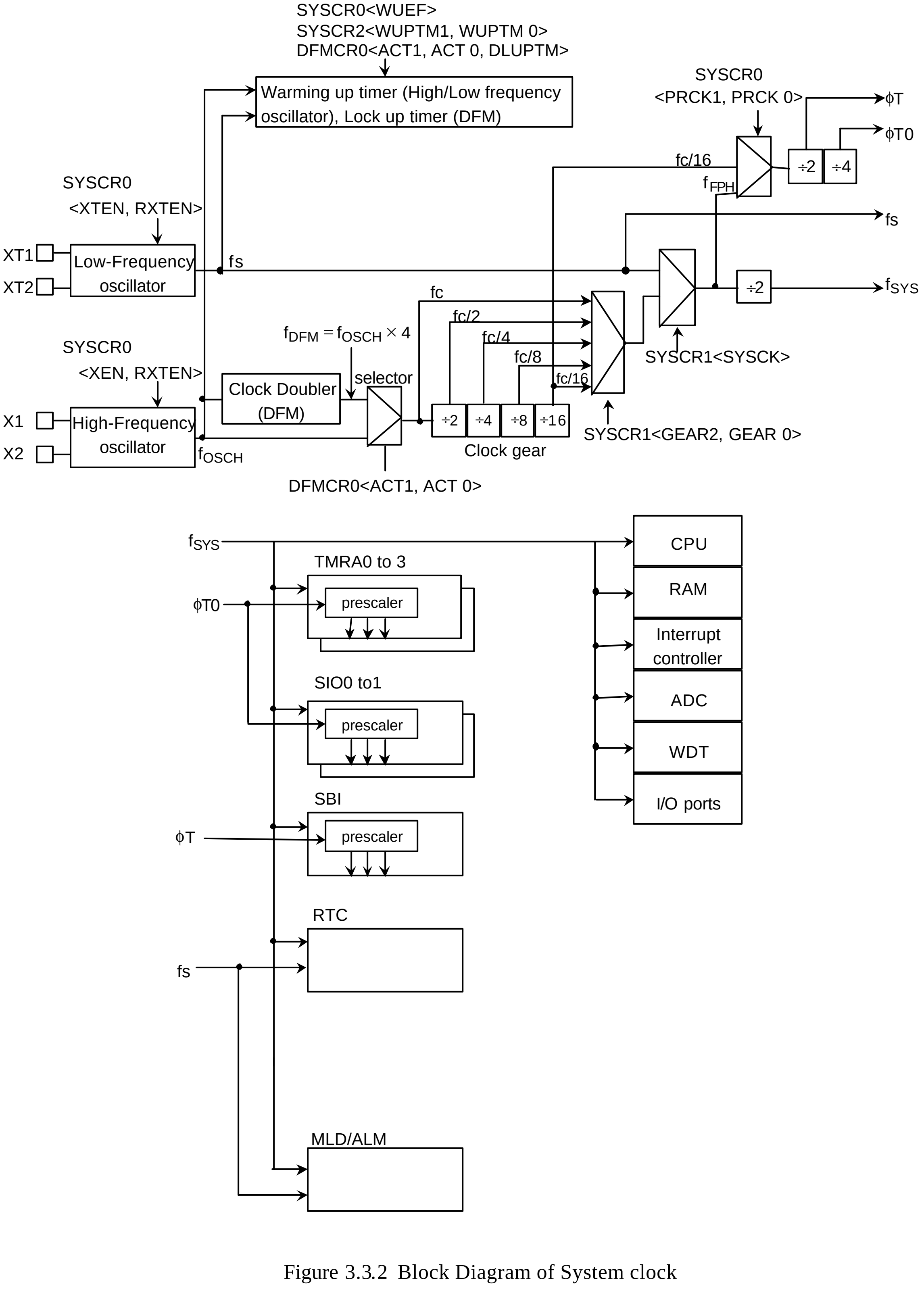


Figure 3.3.2 Block Diagram of System clock

3.3.2 SFR

		7	6	5	4	3	2	1	0
SYSCR0 (00E0H)	bit Symbol	XEN	XTEN	RXEN	RXTEN	RSYSCK	WUEF	PRCK1	PRCK0
	Read/Write	R/W							
	After reset	1	1	1	0	0	0	0	0
	Function	High-frequency oscillator (fc) 0: Stop 1: Oscillation	Low-frequency oscillator (fs) 0: Stop 1: Oscillation (note1)	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 0: Write Don't care 1: Write start timer 0: Read end warm-up 1: Read do not end warm-up	Select prescaler clock 00: f_{FPH} (note2) 01: reserved 10: fc/16 11: reserved	

		7	6	5	4	3	2	1	0
SYSCR1 (00E1H)	bit Symbol					SYSCK	GEAR2	GEAR1	GEAR0
	Read/Write					R/W			
	After reset					0	1	0	0
	Function					Select system clock 0: fc 1: fs	Select gear value of high frequency (fc) 000: fc 001: fc/2 010: fc/4 011: fc/8 100: fc/16 101: (reserved) 110: (reserved) 111: (reserved)		

		7	6	5	4	3	2	1	0
SYSCR2 (00E2H)	bit Symbol		SCOSEL	WUPTM1	WUPTM0	HALTM1	HALTM0	SELDRV	DRVE
	Read/Write		R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset		0	1	0	1	1	0	0
	Function		0: fs 1: f_{SYS}	Warm-Up Timer 00: reserved 01: 2^8/inputted frequency 10:2^{14} 11:2^{16}		HALT mode 00: reserved 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		<DRVE> mode select 0: STOP 1: IDLE1	Pin state control in STOP/IDLE1 mode 0: I/O off 1: Remains the state before HALT

(note1) By Reset, low-frequency oscillator is enable.

(note2) In case of using built-in SBI circuit, it must set SYSCR0<PRCK1:0> to '00'.

Figure 3.3.3 SFR for system clock

Figure 3.3.4 SFR for DFM

Symbol	Name	Address	7	6	5	4	3	2	1	0
DFMCR0	DFM Control Register 0	E8H	ACT1	ACT0	DLUPFG	DLUPTM				
			R/W	R/W	R	R/W				
			0	0	0	0				
			DFM LUP select f_{FPH} 00 STOP STOP f_{OSCH} 01 RUN RUN f_{OSCH} 10 RUN STOP f_{DFM} 11 RUN STOP f_{OSCH}		Lock up Status Flag 0: **end** 1: **not end**	Lock-up Time 0: $2^{12}/f_{OSCH}$ 1: $2^{10}/f_{OSCH}$				
DFMCR1	DFM Control Register 1	E9H								
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	1	0	0	1	1
			DFM revision Input frequency 4~6.75MHz(@2.7V~3.6V) : write “0BH” Input frequency 2~2.5MHz(@2.0±10%) : write “1BH”							

Limitation point on the use of DFM

1. It’s prohibited to execute DFM enable/disable control in the SLOW mode(fs) (write to DFMCR0<ACT1:0>=”10”). You should control DFM in the NORMAL mode.

2. If you stop DFM operation during using DFM(DFMCR0<ACT1:0>=”10”) , you shouldn’t execute that change the clock f_{DFM} to f_{OSCH} and stop the DFM at the same time. Therefore the above executions should be separated into two procedures as showing below.

```
LD      (DFMCR0),C0H    ;   change the clock fDFM to fOSCH
LD      (DFMCR0),00H    ;   DFM stop
```

3. If you stop high frequency oscillator during using DFM (DFMCR0<ACT1:0>=”10”) , you should stop DFM before you stop high frequency oscillator.

Please refer to 3.3.5 Clock Doubler (DFM) for the details.

		7	6	5	4	3	2	1	0
EMCCR0 (00E3H)	bit Symbol	PROTECT	TA3LCDE	–	–	–	EXTIN	DRVOSCH	DRVOSCL
	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	LCDC source CLK 0: 32KHz 1: TA3OUT	Write "1"	Write "0"	Write "0"	1: External clock	fc oscillator driver ability 1: NORMAL 0: WEAK	fs oscillator driver ability 1: NORMAL 0: WEAK
EMCCR1 (00E4H)	bit Symbol	Switching the protect ON/OFF by write to following 1st-KEY,2nd-KEY 1st-KEY: EMCCR1=5AH,EMCCR2=A5H in succession write 2nd-KEY: EMCCR1=A5H,EMCCR2=5AH in succession write							
	Read/Write								
	After reset								
	Function								
EMCCR2 (00E5H)	bit Symbol								
	Read/Write								
	After reset								
	Function								
EMCCR3 (00E6H)	bit Symbol		ENFROM	ENDROM	ENPROM		FFLAG	DFLAG	PFLAG
	Read/Write		R/W	R/W	R/W		R/W	R/W	R/W
	After reset		0	0	0		0	0	0
	Function		CS1A area detect control 0: disable 1: enable	CS2B-2G area detect control 0: disable 1: enable	CS2A area detect control 0: disable 1: enable		CS1A write Operation flag	CS2B-2G write operation Flag	CS2A write Operation Flag
							When reading "0" : not written "1" : written When writing "0" : clear flag		

(note) In case of Vcc=2V±10% use, fixed to EMCCR0<DRV0SCH>='1'.

Figure 3.3.5 SFR for noise-reduction

3.3.3 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<GEAR0 to GEAR2> 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 <XEN> = 1, <XTEN> = 0, <SYSCK> = 0 and <GEAR0~GEAR2> = 100 will cause the system clock (f_{SYS}) to be set to fc/32 (fc/16× 1/2) after a Reset.

For example, f_{SYS} is set to 0.5 MHz when the 16-MHz oscillator is 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<WUPTM0,WUPTM1>.

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

Table 3.3.1 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.

Table 3.3.1 Warming-up times

Warming-up Time SYSCR2 <WUPTM1,WUPTM0>	Change to Normal Mode	Change to Slow Mode
01 (2^8 / frequency)	16 (μs)	7.8 (ms)
10 (2^{14} / frequency)	1.024 (ms)	500 (ms)
11 (2^{16} / frequency)	4.096 (ms)	2000 (ms)

at f_{OSCH} = 16 MHz,
fs = 32.768 kHz

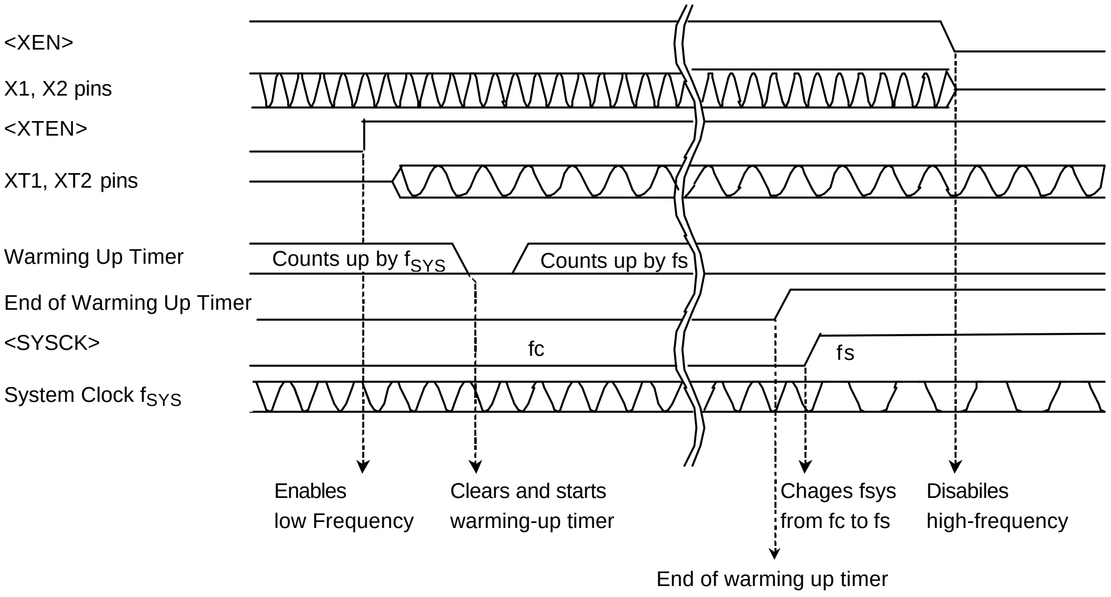
Example 1-Setting the clock

Changing from high frequency (fc) to low frequency (fs).

```
SYSCR0   EQU   00E0H
SYSCR1   EQU   00E1H
SY SCR2  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)            ; ┐
         JR    NZ, WUP                ; ┘ Detects stopping of warm-up timer.
         SET   3, (SYSCR1)            ;   Changes fSYS from fc to fs.
         RES   7, (SYSCR0)            ;   Disables high-frequency oscillation.
```

(note) “x” means don’t care

“-“ means no change



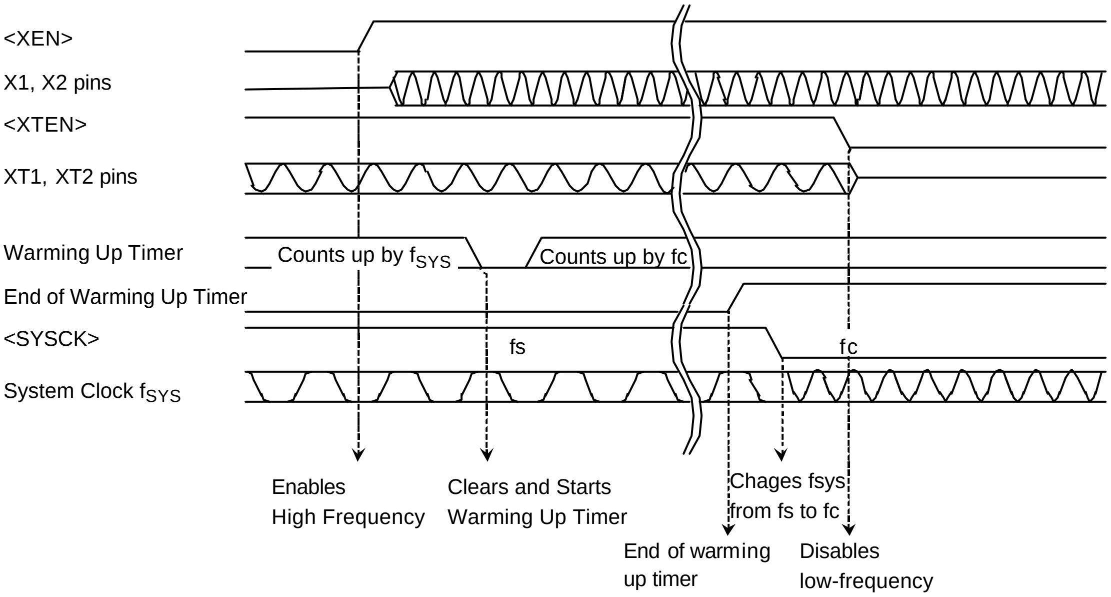
Example 2-Setting the clock

Changing from low frequency (fs) to high frequency (fc).

```
SYSCR0   EQU   00E0H
SYSCR1   EQU   00E1H
SYSCR2   EQU   00E2H
         LD    (SYSCR2), X–10– –X–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)            ; }
         JR    NZ, WUP                ; } Detects stopping of warm-up timer.
         RES   3, (SYSCR1)            ;   Changes fSYS from fs to fc.
         RES   6, (SYSCR0)            ;   Disables low-frequency oscillation.
```

(note) “x” means don’t care

“-“ means no change



(2) Clock gear controller

When the high-frequency clock fc is selected by setting SYSCR1<SYSCK> = 0, f_{FPH} is set according to the contents of the Clock Gear Select Register SYSCR1<GEAR0 to GEAR2> to either fc, fc/2, fc/4, fc/8 or fc/16. Using the clock gear to select a lower value of f_{FPH} reduces power consumption.

Example 3

Changing to a high-frequency gear

```
SYSCR1    EQU    00E1H
          LD     (SYSCR1), XXXX0000B    ;   Changes fSYS to fc/2.
          LD     (SYSCR1), XXXX0100B    ;   Changes fSYS to fc/32.
```

X: Don't care

(High-speed clock gear changing)

To change the clock gear, write the register value to the SYSCR1<GEAR2-0> register.It is necessary the warmming 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)
SYSCR1    EQU    00E1H
          LD     (SYSCR1), XXXX0001B    ;   Changes fSYS to fc/4.
          LD     (DUMMY), 00H           ;   Dummy instruction
         | Instruction to be executed after clock gear has changed |
```

(3) Internal clock terminal out function

It can out internal clock(f_{SYS} or f_S) from PD5/SCOUT.

PD5 pin function is set to SCOUT output by the following bit setting.

: PDFC<PD5F>='1'

Output clock select

:Refer to SYSCR2<SCOSEL> bit setting

HALT mode SCOUT select	NORMAL SLOW	HALT mode		
		IDLE2	IDEL1	STOP
<SCOSEL>='0'	f_S clock out			'0' or '1' fix out
<SCOSEL>='1'	f_{SYS} clock out		'0' or '1' fix out	

3.3.4 Prescaler clock controller

For the internal I/O (TMRA01 to 23, SIO0 to 1) there is a prescaler which can divide the clock.

The φT0 clock input to the prescaler is either the clock f_{FPH} divided by 4 or the clock fc/16 divided by 4. The setting of the SYSCR0 <PRCK0 to PRCK1> register determines which clock signal is input.

3.3.5 Clock doubler (DFM)

DFM outputs the f_{DFM} clock signal, which is four times as fast as f_{OSCH}. It can use the low-frequency oscillator, even though the internal clock is high-frequency.

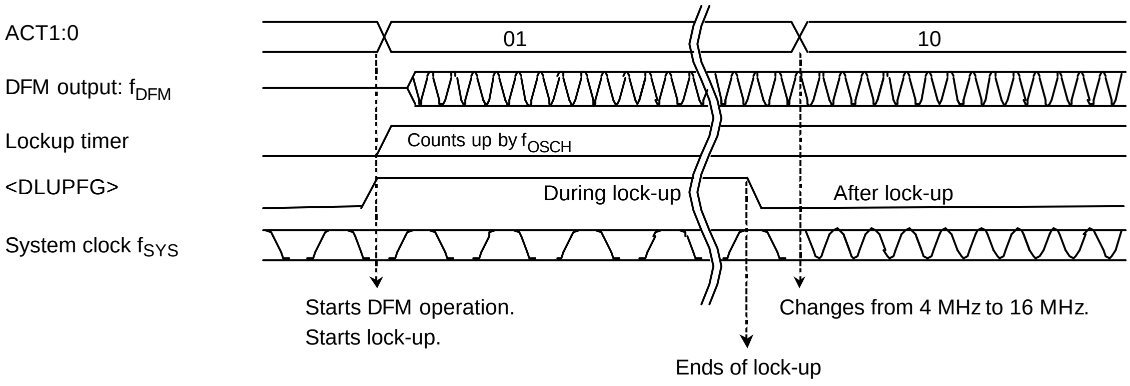
A Reset initializes DFM to Stop status, setting to DFMCR0-register is needed before use.

Like an oscillator, this circuit requires time to stabilize. This is called the lock-up time.

The following example shows how DFM is used.

```
DFMCR0    EQU    00E8H
DFMCR1    EQU    00E9H
          LD     (DFMCR1),00001011B          DFM parameter setting
          LD     (DFMCR0), 01X0XXXXB    ;    Set lock-up time to 2^12/4 MHz
                                             Enables DFM operation and starts lock-up.
LUP:      BIT    5, (DFMCR0)            ;  }
          JR     NZ, LUP                ;  } Detects end of lock-up
          LD     (DFMCR0), 10X0XXXXB    ;    Changes fc from 4 MHz to 16 MHz.
```

X: Don't care



(note) Input frequency limitation and correction for DFM

Recommend to use Input frequency(High speed oscillation) for DFM in the following condition.

f_{OSCH} = 4 ~ 6.75MHz (Vcc = 2.7~ 3.6V) : write 0BH to DFMCR1

f_{OSCH} = 2 ~ 2.5MHz (Vcc = 2.0V ±10%) : write 1BH to DFMCR1

Limitation point on the use of DFM

1. It’s prohibited to execute DFM enable/disable control in the SLOW mode(fs) (write to DFMCR0<ACT1:0>=”10”). You should control DFM in the NORMAL mode.

2. If you stop DFM operation during using DFM (DFMCR0<ACT1:0>=”10”) , you shouldn’t execute the commands that change the clock f_{DFM} to f_{OSCH} and stop the DFM at the same time. Therefore the above execution should be separated into two procedures as showing below.

```
        LD      (DFMCR0),C0H    ;  Change the clock fDFM to fOSCH
        LD      (DFMCR0),00H    ;  DFM stop
```

3. If you stop high frequency oscillator during using DFM(DFMCR0<ACT1:0>=”10”) , you should stop DFM before you stop high frequency oscillator.

Examples of settings are below.

(1) Start Up / Change Control

(OK) Low frequency oscillator operation mode(f_s) (high frequency oscillator STOP)
→ High frequency oscillator start up → High frequency oscillator operation mode(f_{OSCH})→DFM start up → DFM use mode (f_{DFM})

```
        LD      (SYSCR0),  11---1--B        ; High frequency oscillator start up/ Warming up start
WUP:    BIT     2,(SYSCR0)                  ;
        JR      NZ,WUP                      ;  } Check for the flag of warming up end
        LD      (SYSCR1),  ----0---B        ; Change the system clock fs to fOSCH
        LD      (DFMCR0),01-0----B          ; DFM start up / lock up start
LUP:    BIT     5, (DFMCR0)                 ;
        JR      NZ,LUP                      ;  } Check for the flag of lock up end
        LD      (DFMCR0),10-0----B          ; Change the system clock fOSCH to fDFM
```

(OK) Low frequency oscillator operation mode(f_s) (high frequency oscillator Operate)
→ High frequency oscillator operation mode(f_{OSCH})→DFM start up → DFM use mode (f_{DFM})

```
        LD      (SYSCR1),  ----0---B        ; Change the system clock fs to fOSCH
        LD      (DFMCR0), 01-0----B         ; DFM start up / lock up start
LUP:    BIT     5, (DFMCR0)                 ;
        JR      NZ,LUP                      ;  } Check for the flag of lock up end
        LD      (DFMCR0),10-0----B          ; Change the system clock fOSCH to fDFM
```

(NG) Low frequency oscillator operation mode(f_s) (high frequency oscillator STOP)
→ High frequency oscillator start up → DFM start up → DFM use mode (f_{DFM})

```
        LD      (SYSCR0),11---1--B ; High frequency oscillator start up/ Warming up start
WUP:    BIT     2,(SYSCR0)                  ;
        JR      NZ,WUP                      ;  } Check for the flag of warming up end
        LD      (DFMCR0),01-0----B          ; DFM start up / lock up start
LUP:    BIT     5, (DFMCR0)                 ;
        JR      NZ,LUP                      ;  } Check for the flag of lock up end
        LD      (DFMCR0),10-0----B          ; Change the internal clock fOSCH to fDFM
        LD      (SYSCR1),   -----0---B      ; Change the system clock fs to fDFM
```

（2） Change / Stop Control

(OK) DFM use mode (f_{DFM})→ High frequency oscillator operation mode(f_{OSCH})→ DFM Stop → Low frequency oscillator operation mode(f_s)→High frequency oscillator stop

```
LD      (DFMCR0),11------B      ; Change the system clock fDFM to fOSCH
LD      (DFMCR0),00------B      ; DFM stop
LD      (SYSCR1), ----1---B     ; Change the system clock fOSCH to fs
LD      (SYSCR0), 0-------B     ; High frequency oscillator stop
```

(NG) DFM use mode (f_{DFM})→ Low frequency oscillator operation mode(f_s)→ DFM stop →High frequency oscillator stop

```
LD      (SYSCR1), ----1---B     ; Change the system clock fDFM to fS
LD      (DFMCR0),11------B      ; Change the internal clock (fC) fDFM to fOSCH
LD      (DFMCR0),00------B      ; DFM stop
LD      (SYSCR0), 0-------B     ; High frequency oscillator stop
```

(OK) DFM use mode (f_{DFM})→Set the STOP mode →High frequency oscillator operation mode (f_{OSCH})→ DFM stop→ HALT(High frequency oscillator stop)

```
LD      (SYSCR2), ----01--B     ; Set the STOP mode
                                 (This command can execute before use of DFM)
LD      (DFMCR0),11------B      ; Change the system clock fDFM to fOSCH
LD      (DFMCR0),00------B      ; DFM stop
HALT                            ; Shift to STOP mode
```

(NG) DFM use mode (f_{DFM})→Set the STOP mode→ HALT(High frequency oscillator stop)

```
LD      (SYSCR2), ----01--B     ; Set the STOP mode
                                (This command can execute before use of DFM)
HALT                            ; Shift to STOP mode
```

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) SFR protection of register contents

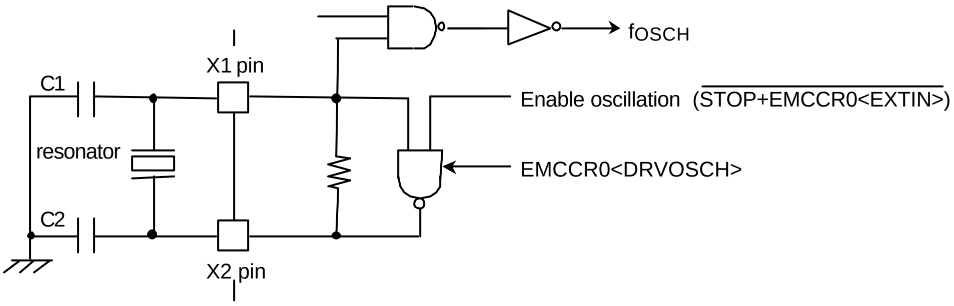
(5) ROM protection of register contents

(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.

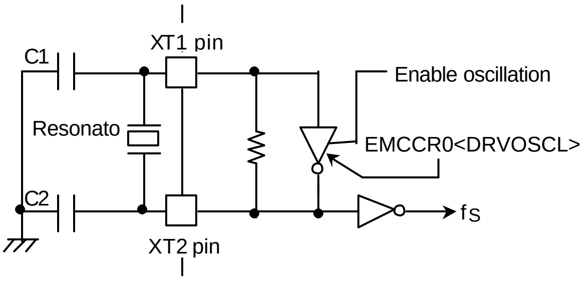
Don’t set <DRVOSCH> to “0” at Vcc=2V ± 10%.

(2) Reduced drivability for low-frequency oscillator

(Purpose)

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

(Block diagram)



(Setting method)

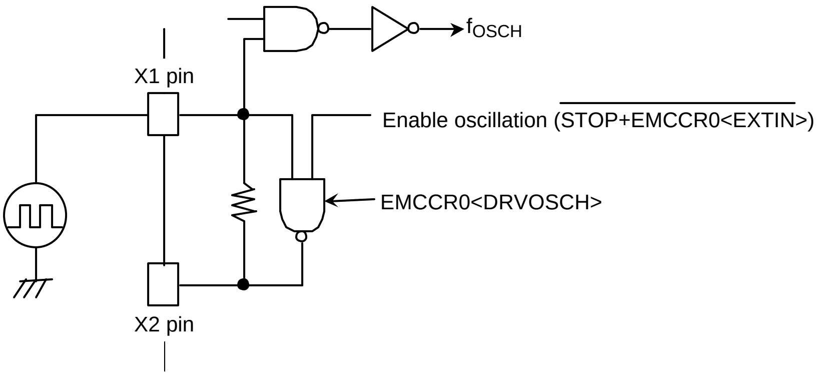
The drivability of the oscillator is reduced by writing 0 to the EMCCR0<DRVOSCL> register. By Reset, <DRVOSCL> is initialized to "1".

(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".

(4) 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, MMU) is changed.

And error handling in runaway becomes easy by INTP0 interruption.

Specified SFR list

1. CS/WAIT controller

   B0CS, B1CS, B2CS, B3CS, BEXCS,
   MSAR0, MSAR1, MSAR2, MSAR3,
   MAMR0, MAMR1, MAMR2, MAMR3

2. MMU

   LOCAL0/1/2/3

3. Clock gear

   SYSCR0, SYSCR1, SYSCR2, EMCCR0,EMCCR3

4. DFM

   DFMCR0, DFMCR1

(Operation explanation)

Execute and release of protection (write operation to specified SFR) become possible by setting up a double key to EMCCR1 and EMCCR2 register.

（Double key)

1st-KEY ： Succession writes in 5AH at EMCCR1 and A5H at EMCCR2

2nd-KEY ： Succession writes in A5H at EMCCR1 and 5AH at EMCCR2

A state of protection can be confirmed by reading EMCCR0<PROTECT>.

By reset, protection becomes OFF.

And INTP0 interruption occurs when write operation to specified SFR was executed with protection ON state.

(5) Runaway provision with ROM protection register

(Purpose)

Provision in runaway of program by noise mixing.

(Operation explanation)

When write operation was executed for external three kinds of ROM by runaway of program, INTP1 is occurred and detects runaway function.

Three kinds of ROM is fixed as for Flash-ROM(Option-Program ROM), Data-ROM, Program-ROM are as follows on the logical address memory map.

1. Flash-ROM : Address 400000H-7FFFFFH
2. Data-ROM : Address 800000H-BFFFFFH
3. Program-ROM : Address C00000H-FFFFFFH

For these address, admission / prohibition of detection of write operation sets it up with EMCCR3<ENFROM,ENDROM,ENPROM>. And INTP1 interruption occurred within which ROM can confirm each with EMCCR3<FFLAG,DFLAG,PFLAG>. This flag is cleared when write in "0".

3.3.6 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,HALTM0> register.

The subsequent actions performed in each mode are as follows:

① IDLE2: Only the CPU halts.

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

Table 3.3 2 Shows the registers of setting operation during IDLE2 mode.

Table 3.3.2 SFR seting operation during IDLE2 mode

Internal I/O	SFR
TMRA01	TA01RUN<I2TA01>
TMRA23	TA23RUN<I2TA23>
SIO0	SC0MOD1<I2S0>
SIO1	SC1MOD1<I2S1>
A/D converter	ADMOD1<I2AD>
WDT	WDMOD<I2WDT>
SBI	SBI0BR0<I2SBI0>

② Idle1: Only the oscillator and the RTC (real-time clock) continue to operate.

③ Stop: All internal circuits stop operating.

The operation of each of the different Halt Modes is described in Table 3.3.3.

Table 3.3.3 I/O operation during Halt Modes

Halt Mode		IDLE2	IDLE1	STOP
SYSCR2 <HALTM1:0>		11	10	01
Block	CPU	Stop		
	I/O ports	Keep the state when the HALT instruction was executed.	See table 3.3.6	
	TMRA	Available to select operation block	Stop	
	SIO, SBI			
	A/D converter			
	WDT			
	RTC,MLD		Possible to operate	

(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 3.3 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 mask register.) However only for INT0~INT4 and RTC interrupts,even if the interrupt request level set before executing the HALT instruction is less than the value of the interrupt mask register, releasing the 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 (/NMI,INT0-3,INTKEY,INTRTC,INTALM0 to 4) 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 isnot applicable).(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 3.3.5) 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.)

Table 3.3.4 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	–	–	–
		INTWDT	◎	×	×	–	–	–
		INT0 ? ? 3 (Note1)	◎	◎	◎*1	○	○	○*1
		INTALM0 to 4	◎	◎	×	○	○	×
		INTTA0 to 3	◎	×	×	×	×	×
		INTRX0 to 1,TX0 to 1	◎	×	×	×	×	×
		INTAD	◎	×	×	×	×	×
		INTKEY	◎	◎	◎*1	○	○	○*1
		INTRTC	◎	◎	×	○	○	×
		INTSBI	◎	×	×	×	×	×
	RESET		◎	◎	◎	◎	◎	◎

◎: After clearing the Halt mode, CPU starts interrupt processing. (RESET initializes the microcont.)

○: 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.

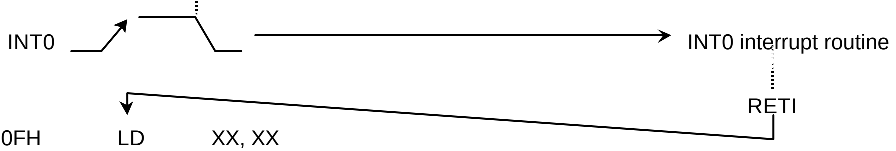
*1: Releasing the halt mode is executed after passing the warmming-up time.

Note 1: 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.

(Example - clearing Idle1 Mode)

An INT0 interrupt clears the Halt state when the device is in Idle1 Mode.

```
Address
8203H     LD     (IIMC), 00H          ; Selects INT0 interrupt rising edge.
8206H     LD     (INTE0AD), 06H       ; Sets INT0 interrupt level to 6.
8209H     EI     5                    ; Sets interrupt level to 5 for CPU.
820BH     LD     (SYSCR2), 28H        ; Sets Halt mode to Idle1 Mode.
820EH     HALT                        ; Halts CPU.
```



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

(3) Operation

① 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 3.3 6 illustrates an example of the timing for clearance of the Idle2 Mode Halt state by an interrupt.

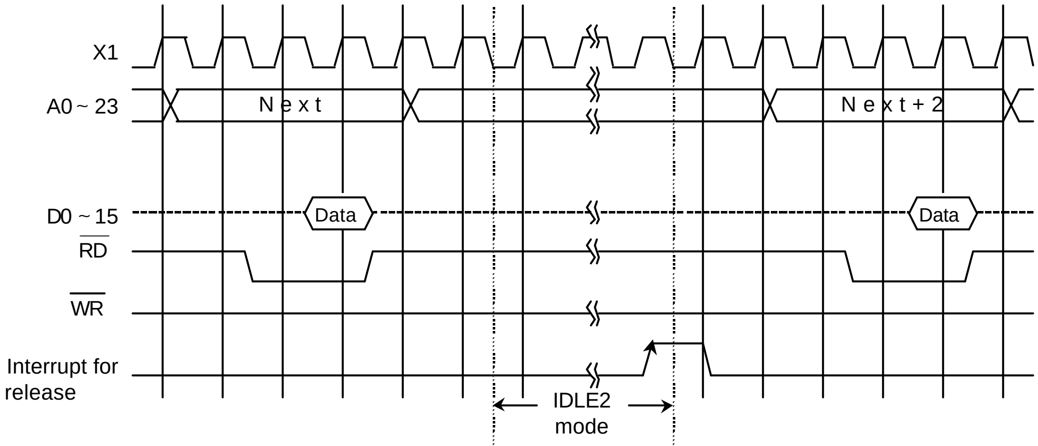


Figure 3.3.6 Timing chart for Idle2 Mode Halt state cleared by interrupt

② Idle1 Mode

In Idle1 Mode, only the internal oscillator and the RTC,MLD continue to operate. The system clock in the MCU stops. The pin status in the IDLE1 mode is depended on setting the register SYSCR2<SELDRV,DRVE>. Table 3.3 6 summarizes the state of these pins in the IDLE mode1.

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

Figure 3.3 7 illustrates the timing for clearance of the Idle1 Mode Halt state by an interrupt.

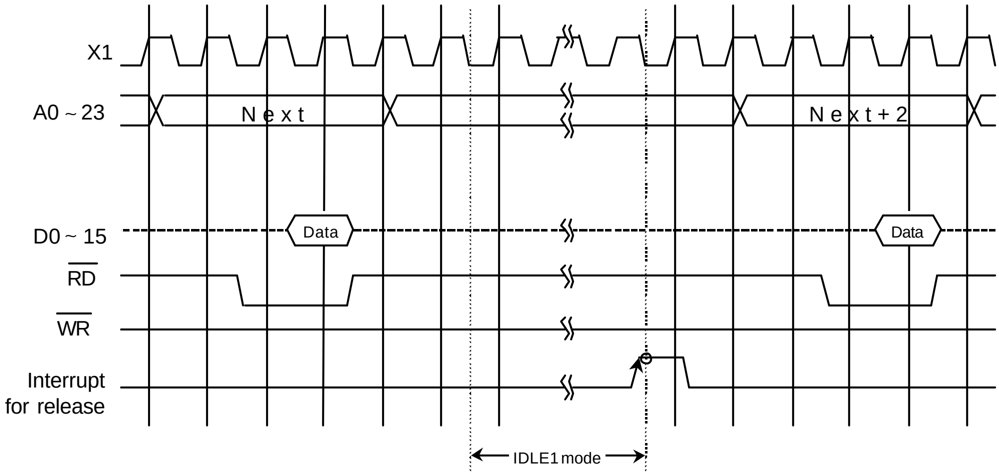


Figure 3.3.7 Timing chart for Idle1 Mode Halt state cleared by interrup

③ 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 3.3.6 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 3.3.5.

Figure 3.3.8 illustrates the timing for clearance of the Stop Mode Halt state by an interrupt.

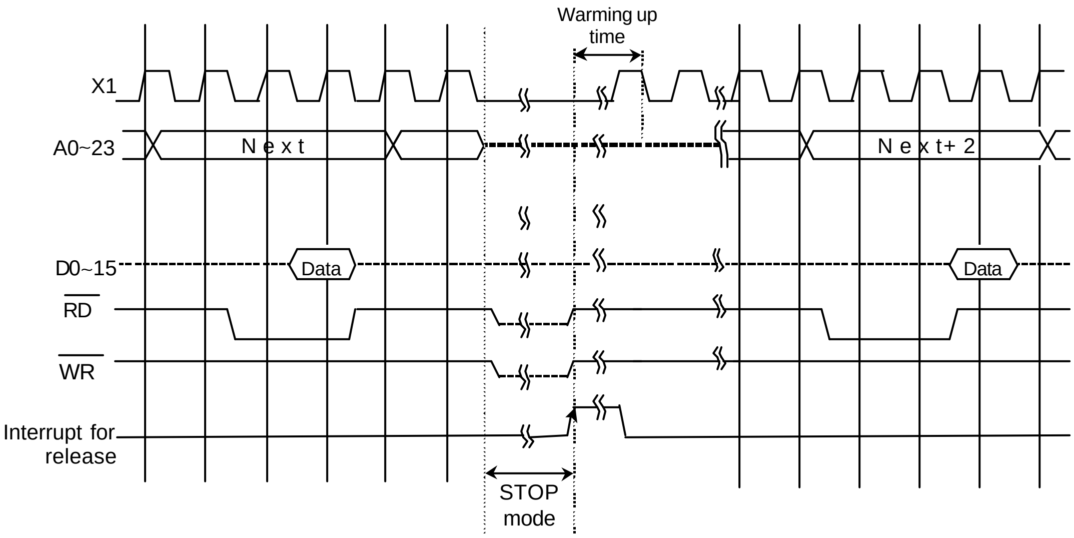


Figure 3.3.8 Timing chart for Stop Mode Halt state cleared by interrupt

Table 3.3.5 Sample warm-up times after clearance of Stop Mode

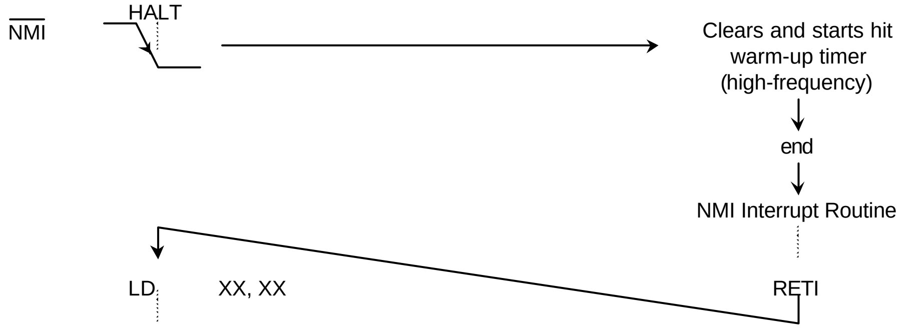
@f_{OSCH} =16 MHz, fs =32.768 kHz

SYSCR0 <RSYSCK>	SYSCR2<WUPTM1,WUPTM0>		
	01 (2^8)	10 (2^{14})	11 (2^{16})
0 (fc)	16 ? s	1.024 ms	4.096 ms
1 (fs)	7.8 ms	500 ms	2000 ms

(Setting 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
          SYSCR2   EQU    00E2H
8FFDH              LD     (SYSCR1), 08H         ; fSYS = fs/2
9000H              LD     (SYSCR2), X−1001X1B   ; Sets Warming Up Time to 2^14/fOSCH
9002H              LD     (SYSCR0), 011000 − −B ; Operates High Frequency after released.
                                                    − : no change
9005H              HALT
9006H              LD     XX, XX
```



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 8 state).In the system which accepts the interrupts during execution ”HALT” instruction, set the same operation mode before and after the STOP mode.

Table 3.3.6 Pin states in IDLE1/Stop Mode

Pin name	Input/Output		<DRVE> = 0	<DRVE> = 1
D0~7	I/O		–	–
P10~17(D8~15?	Input mode Output mode I/O		– – –	– Output –
P20~27(A16~23),A0~15,PD0~PD7	Output pin		—	Output
$\overline{RD}$, $\overline{WR}$	Output pin		–	'1' output
P52~56	Input mode Output mode		– –	Input Output
P60~P64	Output pin		–	Output
P70-71	Input mode Output mode		– –	Input Output
P72	Input mode Output mode		Input –	Input Output
P80~P87	Input pin		–	–
PB0~PB2,PC0~PC5	Input mode Output mode		– –	Input Output
PB3~PB6	Input mode Output mode		Input –	Input Output
$\overline{NMI}$	Input pin		Input	Input
$\overline{RESET}$	Input		Input	Input
AM0, AM1	Input		Input	Input
X1,XT1	Input	IDLE1	Input	Input
		STOP	–	–
X2,XT2	Output	IDLE1	Output	Output
		STOP	"H" Level output XT2 is Hi-Z	"H" Level output XT2 is Hi-Z

- : Input for input mode/input pin is invalid; output mode/output pin is at high impedance.

Input : Input gate in operation. Fix input voltage to "L" or "H" so that input pin stays constant.

Output: Output state

3.4 Interrupts

Interrupts are controlled by the CPU Interrupt Mask Register SR<IFF2:0> and by the built-in interrupt controller.

The TMP91C824 has a total of 37 interrupts divided into the following five types:

- Interrupts generated by CPU: 9 sources
  (Software interrupts,Illegal Instruction interrupt)
- Internal interrupts: 23 sources
- Interrupts on external pins ($\overline{\text{NMI}}$ and INT0 to INT3): 5 sources

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

One of seven (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 <IFF[2:0]>. 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 <IFF[2:0]> value can be updated using the value of the EI instruction (“EI num” sets <IFF[2:0]> data to num).

For example, specifying “EI 3” 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 (<IFF[2:0]> =”7”) is identical to the “EI 7” 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,TMP91C824 has software start function for micro DMA processing request by the software not by the hardware interrupt.

Figure 3.4.1 shows the overall interrupt processing flow.

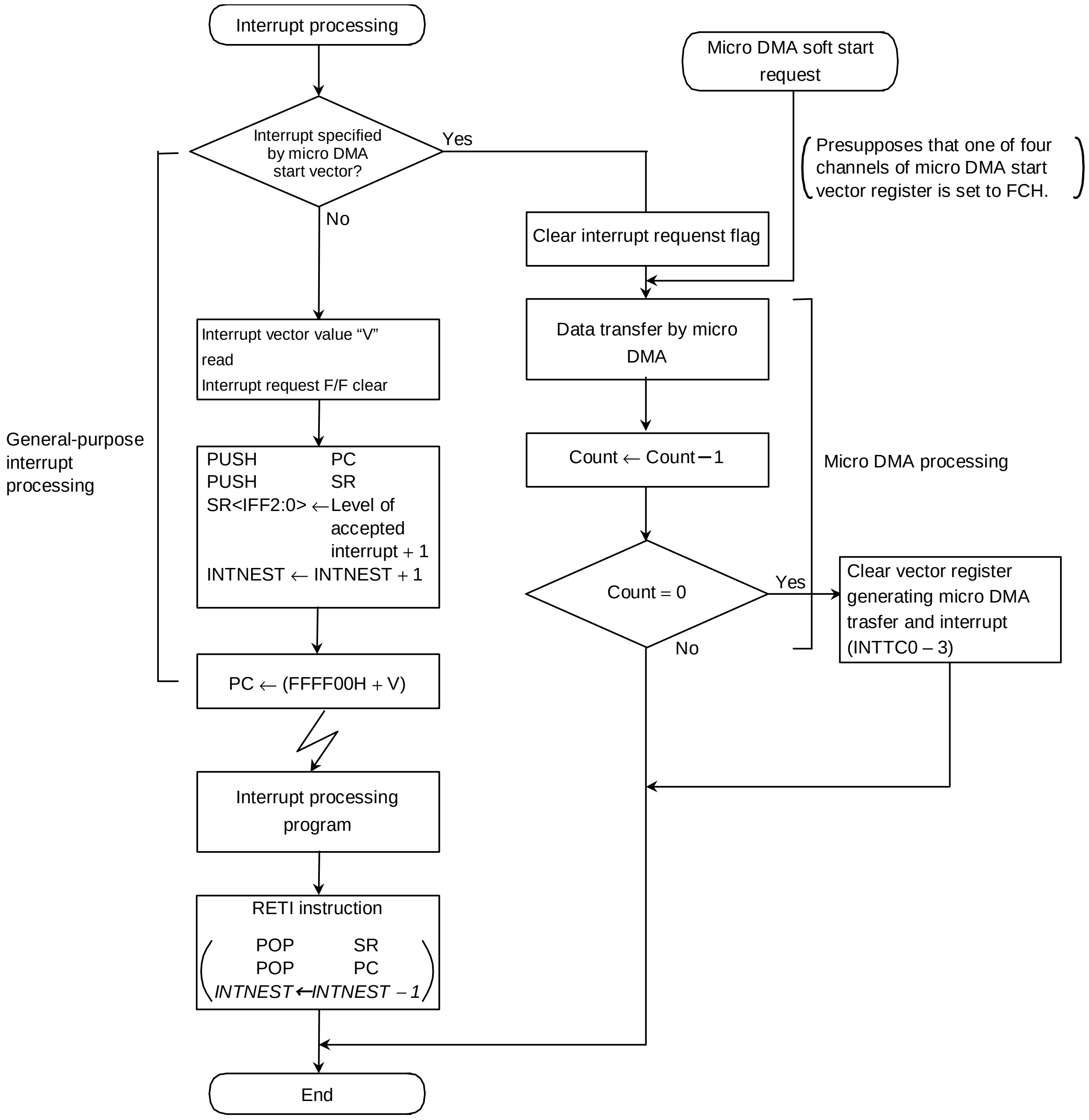

Interrupt processing
Interrupt specified by micro DMA start vector?
Yes
No
Micro DMA soft start request
Presupposes that one of four channels of micro DMA start vector register is set to FCH.
Clear interrupt requenst flag
Data transfer by micro DMA
Interrupt vector value "V" read
Interrupt request F/F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

Figure 3.4.1 Overall interrupt processing flow

3.4.1 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 <IFF[2:0]>. 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 vecto**r”** and starts the interrupt processing routine.

The above processing time is 18-states(2.25usec. at 16MHz) as the best case(16bits data-bus width and 0-wait).

When the CPU compled 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. (A priority level setting of 0 or 7 will disable an interrupt request.)

If an interrupt request which has a priority level equal to or greater than the value of the CPU Interrupt Mask Register <IFF[2:0]> comes out, the CPU accepts its interrupt. Then, the CPU Interrupt Mask Register <IFF[2:0]> 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 <IFF[2:0]> to “111”, disabling all maskable interrupts.

Table 3.4.1 shows the TMP91C824 interrupt vectors and micro DMA start vectors. The address FFFF00H to FFFFFFH (256 bytes) is assigned for the interrupt vector area.

Table 3.4.1 TMP91C824 interrupt vectors table

Default Priority	Type	Interrupt source and source of micro DMA request	Vector value(V)	Vector reference Address	Micro DMA start vector
1	Non-Maskable	"Reset" or「SWI 0」 instruction	0000H	FFFF00H	—
2		「SWI 1」 instruction	0004H	FFFF04H	—
3		INTUNDEF: illegal instruction or「SWI 2」 instruction	0008H	FFFF08H	—
4		「SWI 3」 instruction	000CH	FFFF0CH	—
5		「SWI 4」 instruction	0010H	FFFF10H	—
6		「SWI 5」 instruction	0014H	FFFF14H	—
7		「SWI 6」 instruction	0018H	FFFF18H	—
8		「SWI 7」 instruction	001CH	FFFF1CH	—
9		$\overline{\text{NMI}}$ pin	0020H	FFFF20H	—
10		INTWD: Watchdog timer	0024H	FFFF24H	—
–	Maskable	Micro DMA (MDMA)	—	—	—
11		INT0 pin	0028H	FFFF28H	0AH
12		INT1 pin	002CH	FFFF2CH	0BH
13		INT2 pin	0030H	FFFF30H	0CH
14		INT3 pin	0034H	FFFF34H	0DH
15		INTALM0: ALM0(8KHz)	0038H	FFFF38H	0EH
16		INTALM1: ALM1(512Hz)	003CH	FFFF3CH	0FH
17		INTALM2: ALM2(64Hz)	0040H	FFFF40H	10H
18		INTALM3: ALM3(2Hz)	0044H	FFFF44H	11H
19		INTALM4: ALM4(1Hz)	0048H	FFFF48H	12H
20		INTTA0 : 8 bit timer0	004CH	FFFF4CH	13H
21		INTTA1 : 8 bit timer1	0050H	FFFF50H	14H
22		INTTA2 : 8 bit rimer2	0054H	FFFF54H	15H
23		INTTA3 : 8 bit timer3	0058H	FFFF58H	16H
24		INTRX0 : serial reception (channel. 0)	005CH	FFFF5CH	17H
25		INTTX0 : serial transmission (channel. 0)	0060H	FFFF60H	18H
26		INTRX1 : serial reception (channel. 1)	0064H	FFFF64H	19H
27		INTTX1 : serial transmission (channel. 1)	0068H	FFFF68H	1AH
28		INTAD : A/D conversion end	006CH	FFFF6CH	1BH
29		INTRTC : RTC (alarm interrupt)	0074H	FFFF74H	1DH
30		INTSBI : SBI interrupt	0078H	FFFF78H	1EH
31		INTP0 : Protect0 (WR to special SFR)	0080H	FFFF80H	20H
32		INTP1 : Protect1 (WR to ROM)	0084H	FFFF84H	21H
33		INTTC0 : Micro DMA end (channel. 0)	0088H	FFFF88H	--
34		INTTC1 : Micro DMA end (channel. 1)	008CH	FFFF8CH	--
35		INTTC2 : Micro DMA end (channel. 2)	0090H	FFFF90H	--
36		INTTC3 : Micro DMA end (channel. 3)	0094H	FFFF94H	--
		(Reserved) : (Reserved)	0098H : 00FCH	FFFF98H : FFFFFCH	— : —

3.4.2 Micro DMA processing

In addition to general-purpose interrupt processing, the TMP91C824 supprots a micro DMA function. Interrupt requests set by micro DMA perform micro DMA processing at the highest priority level (level 6) among maskable interrupts, regardless of the priority level of the particular interrupt source. Micro. The micro DMA has 4 channels and is possible continuous transmission by specifing the say later burst mode.

Because the micro DMA function has been implemented with the cooperative operation of CPU, when CPU goes to a stand-by 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 level 6 and starts processing the request in spite of any interrupt source’s level. The micro DMA is ignored on <IFF[2:0]>=”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 (INTTC0 to INTTC3) 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 (INTTC0 to INTTC3) 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 (INTTC0 to INTTC3) 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 16M bytes (the upper eight bits of the 32 bits are not valid).

Three micro DMA transfer modes are supported: 1-byte transfer, 2-byte (one-word) transfer, and 4-byte transfer. After a transfer in any mode, the transfer source / destination addresses are 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 3.4.2 (4) “Transfer Mode Register”. As the transfer counter is a 16-bit counter, micro DMA processing can be set for up to 65536 times per interrupt source.(The micro DMA processing count is maximized when the transfer counter initial value is set to 0000H.)

Micro DMA processing can be started by the 24 interrupts shown in the micro DMA start vectors of Table 3.4.1 and by the micro DMA soft start, making a total of 25 interrupts.
Figure 3.4.2 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, trandfer source/transfer destination addresses both even-numberd values).

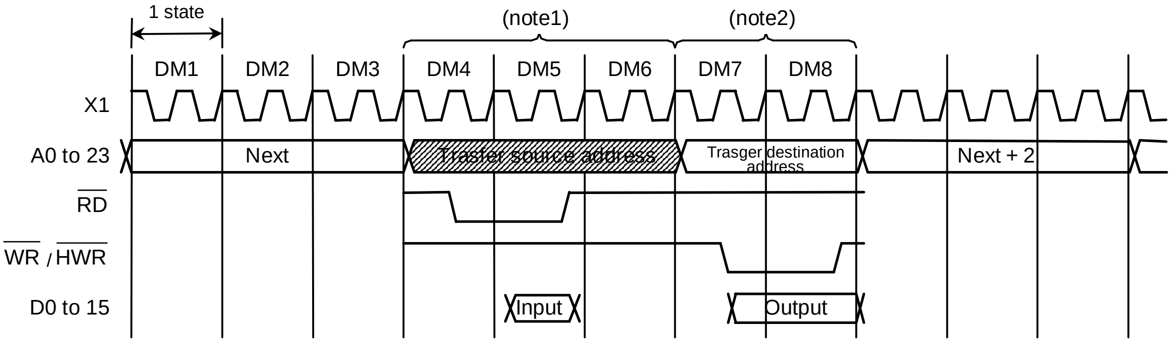


Figure 3.4.2 Timing for micro DMA cycle

States 1~3: 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 4~5: Micro DMA read cycle

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

States 7~8: Micro DMA write cycle

(note1): 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.

(note2): 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, TMP91C824 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.)

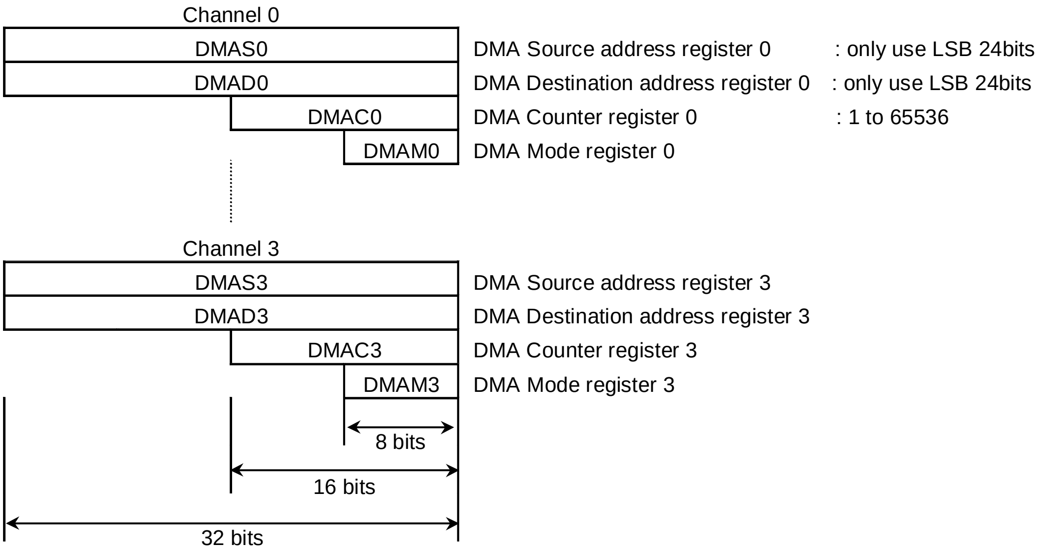
When writing again “1” to the DMAR register, check whether the bit is “0” before writing “1”.

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.

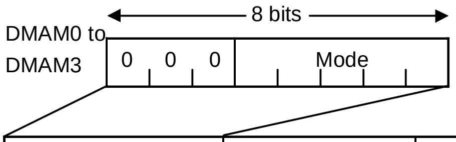
Symbol	NAME	Address	7	6	5	4	3	2	1	0
DMAR	DMA Request Register	89h (no RMW)					DMA Request			
							DMAR3	DMAR2	DMAR1	DMAR0
							R/W			
							0	0	0	0

(3) Transfer control registers

The transfer source address and the transfer destination address are set in the following registers in CPU. Data setting for these registers is done by an “LDC cr,r” instruction.


Channel 0
DMAS0
DMAD0
DMAC0
DMAM0
DMA Source address register 0 : only use LSB 24bits
DMA Destination address register 0 : only use LSB 24bits
DMA Counter register 0 : 1 to 65536
DMA Mode register 0
Channel 3
DMAS3
DMAD3
DMAC3
DMAM3
DMA Source address register 3
DMA Destination address register 3
DMA Counter register 3
DMA Mode register 3
8 bits
16 bits
32 bits

(4) Detailed description of the Transfer Mode Register



(note): When setting a value in this register, write 0 to the upper 3 bits.

			Number of Transfer Bytes	Mode Description	Number of Execution States	Minimum Execution Time @ fc = 16 MHz
000 (fixed)	000	00	Byte transfer	Transfer Destination Address INC Mode ··························I/O to memory (DMADn+) ← (DMASn) DMACn ← DMACn – 1 If DMACn = 0, then INTTCn is generated.	8 states	1000 ns
		01	Word transfer			
		10	4-byte transfer		12 sates	1500 ns
	001	00	Byte transfer	Transfer Destination Address DEC Mode ··························I/O to memory (DMADn–) ← (DMASn) DMACn ← DMACn – 1 If DMACn = 0, then INTTCn is generated.	8 states	1000 ns
		01	Word transfer			
		10	4-byte transfer		12 sates	1500 ns
	010	00	Byte transfer	Transfer Source Address INC Mode ··························Memory to I/O (DMADn) ← (DMASn+) DMACn ← DMACn – 1 If DMACn = 0, then INTTCn is generated.	8 states	1000 ns
		01	Word transfer			
		10	4-byte transfer		12 sates	1500 ns
	011	00	Byte transfer	Transfer Source Address DEC Mode ··························Memory to I/O (DMADn) ← (DMASn–) DMACn ← DMACn – 1 If DMACn = 0, then INTTCn is generated.	8 states	1000 ns
		01	Word transfer			
		10	4-byte transfer		12 sates	1500 ns
	100	00	Byte transfer	Fixed Address Mode ······························· I/O to I/O (DMADn) ← (DMASn–) DMACn ← DMACn – 1 If DMACn = 0, then INTTCn is generated.	8 states	1000 ns
		01	Word transfer			
		10	4-byte transfer		12 sates	1500 ns
	101	00	Counter Mode ··········· For counting number of times interrupt is generated DMASn ← DMASn + 1 DMACn ← DMACn – 1 If DMACn = 0, then INTTCn is generated.		5 sates	625 ns

(note1): "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)

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

(note2): Execution time is under the condition of:

16bit bus width(both translation and destination address area) / 0 wait /
fc = 16MHz / selected high frequency mode (fc x 1)

(note3): Do not use an undefined code for the transfer mode register except for the defined codes listed in the above table.

3.4.3 Interrupt controller operation

The block diagram in Figure 3.4.3 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 36 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 (write 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. INTE0AD or INTE12). 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 (NMI pin interrupts and Watch dog 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 simulateous interrupts and its vector address to the CPU. The CPU compares the priority value <IFF[2:0]> 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 <IFF[2:0]>. 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<IFF[2:0]>.

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 3.4.1), 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.

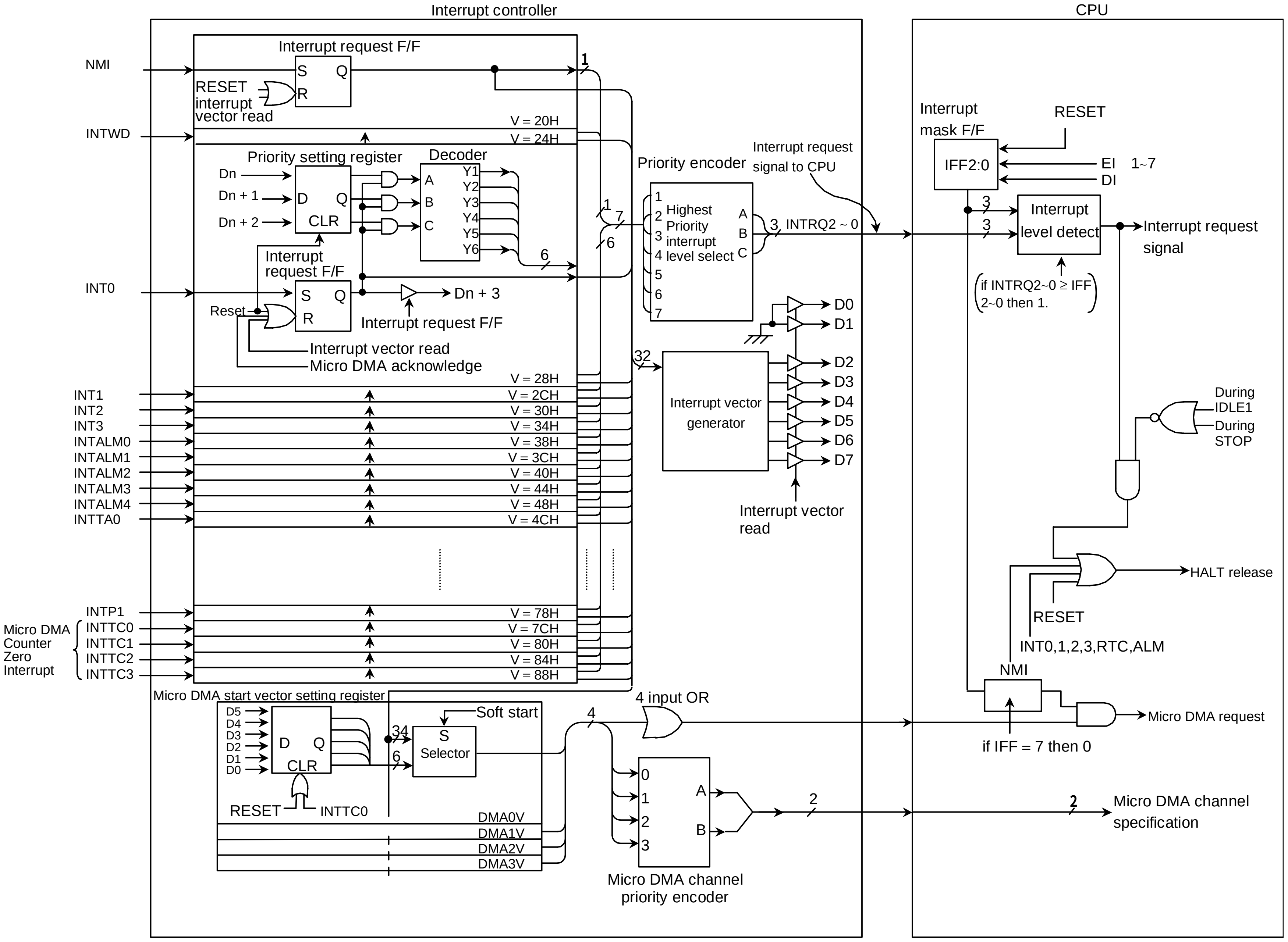
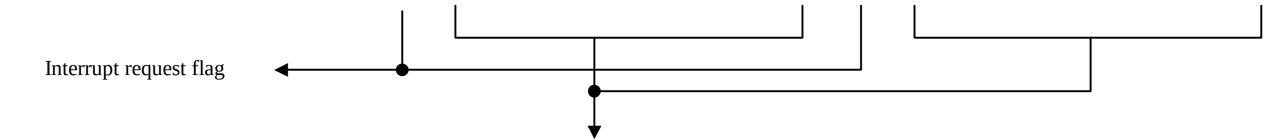


Figure 3.4.3 Block Diagram of Interrupt Controller

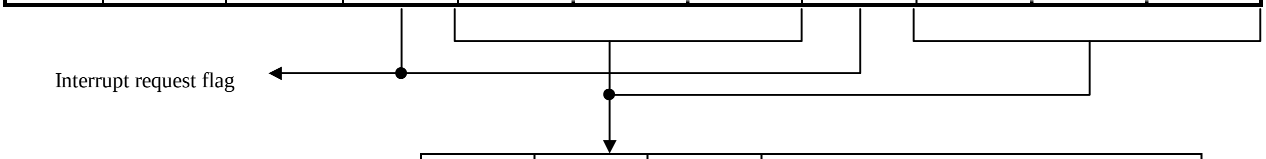
(1) Interrupt level setting registers

Symbol	NAME	Address	7	6	5	4	3	2	1	0
INTE0AD	INT0 & INTAD Enable	90h	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE12	INT1 & INT2 Enable	91h	INT2				INT1			
			I2C	I2M2	I2M1	I2M0	I1C	I1M2	I1M1	I1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE3 ALM4	INT3& INTALM 4Enable	92h	INTALM4				INT3			
			IA4C	IA4M2	IA4M1	IA4M0	I3C	I3M2	I3M1	I3M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTEALM 01	INTALM 0 & INTALM 1 Enable	93h	INTALM1				INTALM0			
			IA1C	IA1M2	IA1M1	IA1M0	IA0C	IA0M2	IA0M1	IA0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTEALM 23	INTALM 2 & INTALM 3 Enable	94h	INTALM3				INTALM2			
			IA3C	IA3M2	IA3M1	IA3M0	IA2C	IA2M2	IA2M1	IA2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETA01	INTTA0 & INTTA1 Enable	95h	INTTA1(TMRA1)				INTTA0(TMRA0)			
			ITA1C	ITA1M2	ITA1M1	ITA1M0	ITA0C	ITA0M2	ITA0M1	ITA0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETA23	INTTA2 & INTTA3 Enable	96h	INTTA3(TMRA3)				INTTA2(TMRA2)			
			ITA3C	ITA3M2	ITA3M1	ITA3M0	ITA2C	ITA2M2	ITA2M1	ITA2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTERTC	INTRTC enable	97h					INTRTC			
							IRC	IRM2	IRM1	IRM0
							R	R/W		
							0	0	0	0



lxxM2	lxxM1	lxxM0	Function (Write)
0	0	0	Disables interrupt requests
0	0	1	Sets interrupt priority level to 1
0	1	0	Sets interrupt priority level to 2
0	1	1	Sets interrupt priority level to 3
1	0	0	Sets interrupt priority level to 4
1	0	1	Sets interrupt priority level to 5
1	1	0	Sets interrupt priority level to 6
1	1	1	Disables interrupt requests

Symbol	NAME	Address	7	6	5	4	3	2	1	0
INTES0	Interrupt Enable Serial 0	98H	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTES1	INTRX1 & INTTX1 Enable	99H	INTTX1				INTRX1			
			ITXT1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTES2	INTES2 Enable	9AH					INTS2			
							IS2C	IS2M2	IS2M1	IS2M0
							R	R/W		
							0	0	0	0
INTETC01	INTTC0 & INTTC1 Enable	9BH	INTTC1				INTTC0			
			ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC23	INTTC2 & INTTC3 Enable	9CH	INTTC3				INTTC2			
			ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTEP01	INTP0 & INTP1 Enable	9DH	INTP1				INTP0			
			IP1C	IP1M2	IP1M1	IP1M0	IP0C	IP0M2	IP0M1	IP0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0



lxxM2	lxxM1	lxxM0	Function (Write)
0	0	0	Disables interrupt requests
0	0	1	Sets interrupt priority level to 1
0	1	0	Sets interrupt priority level to 2
0	1	1	Sets interrupt priority level to 3
1	0	0	Sets interrupt priority level to 4
1	0	1	Sets interrupt priority level to 5
1	1	0	Sets interrupt priority level to 6
1	1	1	Disables interrupt requests

(2) External interrupt control

Symbol	NAME	Address	7	6	5	4	3	2	1	0
IIMC	Interrupt Input Mode control	8CH (no RMW)		-	I3EDGE	I2EDGE	I1EDGE	I0EDGE	I0LE	NMIREE
			W							
			0	0	0	0	0	0	0	0
			Always write”0”	Always write”0”	INT3EDGE 0: Rising 1: Falling	INT2EDGE 0: Rising 1: Falling	INT1EDGE 0: Rising 1: Falling	INT0EDGE 0: Rising 1: Falling	INT0 mode 0: Edge 1: Level	1: Operates even on rising / falling edge of $\overline{\text{NMI}}$

INT0 level Enable

0	Rising edge detect INT
1	“H” level INT

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

0	INT request generation at falling edge
1	INT 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 3.4 1, to the register INTCLR.

For example, to clear the interrupt flag INT0, perform the following register operation after execution of the DI instruction.

INTCLR ← 0AH : Clears interrupt request flag INT0.

Symbol	NAME	Address	7	6	5	4	3	2	1	0
INTCLR	Interrupt Clear Control	88H (no RMW)			CLRV5	CLRV4	CLRV3	CLRV2	CLRV1	CLRV0
			W							
					0	0	0	0	0	0
			Interrupt Vector							

(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 channel, the channel with the lowest number has a higher priority.

Accordingly, if the same vector is set in the micro DMA start vector registers of two channels, the interrupt generated in the channel with the lower number is executed until 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
DMA0V	DMA0 Start Vector	80H			DMA0 Start Vector					
					DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
					R/W					
					0	0	0	0	0	0
DMA1V	DMA1 Start Vector	81H			DMA1 Start Vector					
					DMA1V5	DMA1V4	DMA1V3	DMA0V2	DMA1V1	DMA1V0
					R/W					
					0	0	0	0	0	0
DMA2V	DMA2 Start Vector	82H			DMA2 Start Vector					
					DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
					R/W					
					0	0	0	0	0	0
DMA3V	DMA3 Start Vector	83H			DMA3 Start Vector					
					DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
					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
DMAR	DMA Software Request Register	89H					DMAR3	DMAR2	DMAR1	DMAR0
							R/W	R/W	R/W	R/W
							0	0	0	0
							1: DMA Software request			
DMAB	DMA Burst Register	8AH					DMAB3	DMAB2	DMAB1	DMAB0
							R/W			
							0	0	0	0

(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 avobe plogram, place instructions that clear interrupt request flags after a DI instruction.

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. 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 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 → L)

INTRX: Instruction which read the Receive Buffer

3.5 Port Functions

The TMP91C824 features 56 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 3.5.1 lists the functions of each port pin. Table 3.5.2 lists I/O registers and their specifications.

Table 3.5.1 Port functions

(R: PU= with programmable pull-up resistor / U= with pull-up resistor)

Port name	Pin name	Number of pins	Direction	R	Direction Setting unit	Pin name for built-in function
Port 1	P10 to P17	8	I/O	–	Bit	D8 to D15
Port 2	P20 to P27	8	Output	–	(Fixed)	A16 to A23
Port 5	P54	1	I/O	PU	Bit	$\overline{\text{BUSRO}}$
	P55	1	I/O	PU	Bit	$\overline{\text{BUSAK}}$
	P56	1	I/O	PU	Bit	$\overline{\text{WAIT}}$
Port 6	P60	1	Output	–	(Fixed)	/CS0
	P61	1	Output	–	(Fixed)	/CS1
	P62	1	Output	–	(Fixed)	/CS2,/CS2A
	P63	1	Output	–	(Fixed)	/CS3
	P64	1	Output	–	(Fixed)	EA24,/CS2B
	P65	1	Output	–	(Fixed)	EA25,/CS2C
	P66	1	Output	–	(Fixed)	/CS2D
	P67	1	Output	–	(Fixed)	/CS2E
Port 7	P70	1	I/O	–	Bit	SCK,OPTRX0
	P71	1	I/O	PU	Bit	SO/SDA,OPTTX0
	P72	1	I/O	PU	Bit	SI/SCL
Port 8	P80 to P87	8	Input	–	(Fixed)	AN0 to AN7, $\overline{\text{ADTRG}}$ (P83)
Port B	PB0	1	I/O	–	Bit	TA0IN
	PB1	1	I/O	–	Bit	TA1OUT
	PB2	1	I/O	–	Bit	TA3OUT
	PB3	1	I/O	–	Bit	INT0
	PB4	1	I/O	–	Bit	INT1
	PB5	1	I/O	–	Bit	INT2
	PB6	1	I/O	–	Bit	INT3
Port C	PC0	1	I/O	–	Bit	TXD0
	PC1	1	I/O	–	Bit	RXD0
	PC2	1	I/O	–	Bit	SCLK0/ $\overline{\text{CTS0}}$
	PC3	1	I/O	–	Bit	TXD1
	PC4	1	I/O	–	Bit	RXD1
	PC5	1	I/O	–	Bit	SCLK1/ $\overline{\text{CTS1}}$
PortD	PD5	1	Output	–	(Fixed)	SCOUT
	PD6	1	Output	–	(Fixed)	ALARM,/MLDALM
	PD7	1	Output	–	(Fixed)	MLDALM
Port Z	PZ2	1	I/O	PU	Bit	$\overline{\text{HWR}}$
	PZ3	1	I/O	PU	Bit	R/$\overline{\text{W}}$

Table 3.5.2 I/O Registers and Specifications (1/2) X: Don’t care

Port	Pin name	Specification	I/O register			
			Pn	PnCR	PnFC	PnFC2
Port 1 (note1)	P10 to P17	Input port	X	0	None	None
		Output port	X	1		
		D8 to D15 bus	X	X		
Port 2	P20 to P27	Output port	X	None	0	
		A16 to A23 output	X		1	
Port 5	P54 to P56	Input port (Without PU)	0	0	0	
		Input port (with PU)	1	0	0	
		Output port	X	1	0	
	P54	$\overline{\text{BUSRQ}}$ input (Without PU)	0	0	1	
		$\overline{\text{BUSRQ}}$ input (With PU)	1	0	1	
	P55	$\overline{\text{BUSAK}}$ output	X	1	1	
	P56	$\overline{\text{WAIT}}$ input (Without PU)	0	0	None	
		$\overline{\text{WAIT}}$ input (With PU)	1	0		
Port 6	P60 to P64	Output port	X	None	0	0
	P60	/CS0 output	X		1	None
	P61	/CS1 output	X		1	
	P62	/CS2 output	X		1	0
		/CS2A output	X		X	1
	P63	/CS3 output	X		1	None
	P64	EA24 output	X		1	0
		/CS2B output	X		X	1
	P65	EA25 output	X		1	0
		/CS2C output	X		X	1
	P66	/CS2D output	X		0	1
	P67	/CS2E output	X		0	1
Port 7	P70 to P72	Input port (without PU)	0	0	0	0
		Input port (With PU)	1	0	0	0
		Output port	X	1	0	0
	P70	SCK input	X	0	0	0
		SCK output	X	1	1	0
		OPTRX0 input (note2)	1	0	X	1
	P71	SDA input	X	0	0	0
		SDA output (note3)	X	1	1	0
		SO output	X	1	1	0
		OPTTX0 output (note2)	1	1	X	1
	P72	SI input	X	0	0	0
		SCL input	X	0	0	0
		SCL output (note3)	X	1	1	0

Table 3.5.3 I/O Registers and Specifications (2/2) X: Don’t care

Port	Pin name	Specification	I/O register			
			Pn	PnCR	PnFC	PnFC2
Port 8	P80 to P87	Input port	X	None		None
		AN0 to 7 input (note4)	X			
	P83	$\overline{\text{ADTRG}}$ input (note5)	X			
Port B	PB0 to PB6	Input port	X	0	0	
		Output port	X	1	0	
	PB0	TA0IN input	X	0	None	
	PB1	TA1OUT output	X	1	1	
	PB2	TA3OUT output	X	1	1	
	PB3	INT0 input	X	0	1	
	PB4	INT1 input	X	0	1	
	PB5	INT2 input	X	0	1	
	PB6	INT3 input	X	0	1	
Port C	PC0 to PC5	Input port	X	0	0	
		Output port	X	1	0	
	PC0	TXD0 output (Note2)	1	1	1	
	PC1	RXD0 input (Note2) (Note6)	1	0	None	
	PC2	SCLK0 input (Note2)	1	0	0	
		SCLK0 output (Note2)	1	1	1	
		$\overline{\text{CTS0}}$ input (Note2)	1	0	0	
	PC3	TXD1 output (Note2)	1	1	1	
	PC4	RXD1 input (Note2)	1	0	None	
	PC5	SCLK1 input (Note2)	1	0	0	
		SCLK1 output (Note2)	1	1	1	
		$\overline{\text{CTS1}}$ input (Note2)	1	0	0	
Port D	PD5 to PD7	Output port	X	None	0	
	PD5	SCOUT output	X		1	
	PD6	/ALARM output	1		1	
		/MLDALM output	0		1	
	PD7	MLDALM output	X		1	
Port Z	PZ2 to PZ3	Input port (Without PU)	0	0	0	
		Input port (with PU)	1	0	0	
		Output port	X	1	0	
	PZ2	$\overline{\text{HWR}}$ output	X	1	1	
	PZ3	R/$\overline{\text{W}}$ output	X	1	1	

(note1): PORT1 is only use for PORT or DATA bus(D8 to D15) by setting AM1 and AM0 pins.

(note2): As for input ports of SIO1 and SIO2: (OPTRX0,OPTTX0,TXD0,TRX0,SCCLK0,/CTS0, TXD1,TRX1,SCCLK1,/CTS1), logical selection for output data or input data is determined by the output latch register Pn of each port.

(note3): In case using P71 and P72 for SDA and SCL as open-drain ports, set to P7ODE<ODEP71:ODEP72>.

(note4): In case using P80 to P87 for analog input ports of A/D converter, set to ADMOD1<ADCH2:ADCH1:ADCH0>.

(note5): In case using P83 for $\overline{\text{ADTRG}}$ input port, set to ADMOD1<ADTRGE>.

(note6): In case using PC1 for RXD0 port, set “1” to P7FC2<P70FC>.

After Reset, the port pins listed below function as general-purpose I/O port pins.

Resetting sets I/O pins, which can be programmed for either input or output to be input ports pins. Setting the port pins for internal function use must be done in software.

Note about bus release and programmable pull-up I/O port pins

When the bus is released (i.e. when $\overline{\text{BUSAK}}$ = 0), the output buffers for D0 to D15, A0 to A23, and the control signals (RD, WR, HWR, R/W and CS0 to CS3, EA24, 25/CS2A to 2E) are off and are set to High-Impedance.

However, the output of built-in programmable pull-up resistors are kept before the bus is released. These programmable pull-up resistors can be selected ON/OFF by programmable when they are used as the input ports.

When they are used as output ports, they cannot be turned ON/OFF in software.

Table 3.5.4 shows the pin states after the bus has been released.

Table 3.5.4 Pin states (after bus release)

Pin name	The pin state (when the bus is released)	
	Port mode	Function mode
D0-D7		Become high-impedance(Hz).
D8-D15(P10-P17)	The state is not changed. (do not become to high impedance (Hz).)	↑
A0-A15		First sets all bits to high, then sets them to High-impedance(Hz).
A16-23(P20-P27)	The state is not changed. (do not become to high impedance (Hz).)	↑
/RD. /WR		↑
PZ2 (/HWR), PZ3 (R/W),	The state is not changed. (do not become to high impedance (Hz).)	First sets all bits to high, then the output buffer is OFF. The programmable pull up resistor is ON irrespective of the output latch.
P60 (/CS0),P61 (/CS1), P62 (/CS2,/CS2A), P63 (/CS3),	↑	First sets all bits to high, then sets them to high-impedance(Hz).

3.5.1 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. Resetting , the control register P1CR 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 an address data bus (D8 to 15).

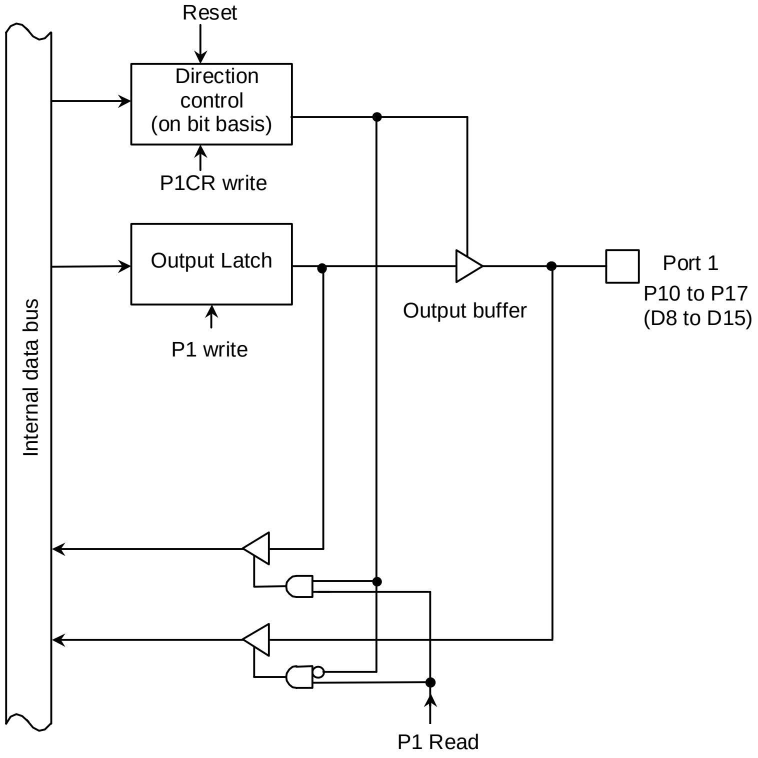


Figure 3.5.1 Port 1

3.5.2 Port 2 (P20 to P27)

Port 2 is an 8-bit output port. In addition to functioning as a output port, Port 2 can also function as an address bus (A16 to A23).

Each bit can be set individually for address bus using the function register P2FC. Resetting sets all bits of the function register P2FC to 1 and sets Port 2 to address bus.

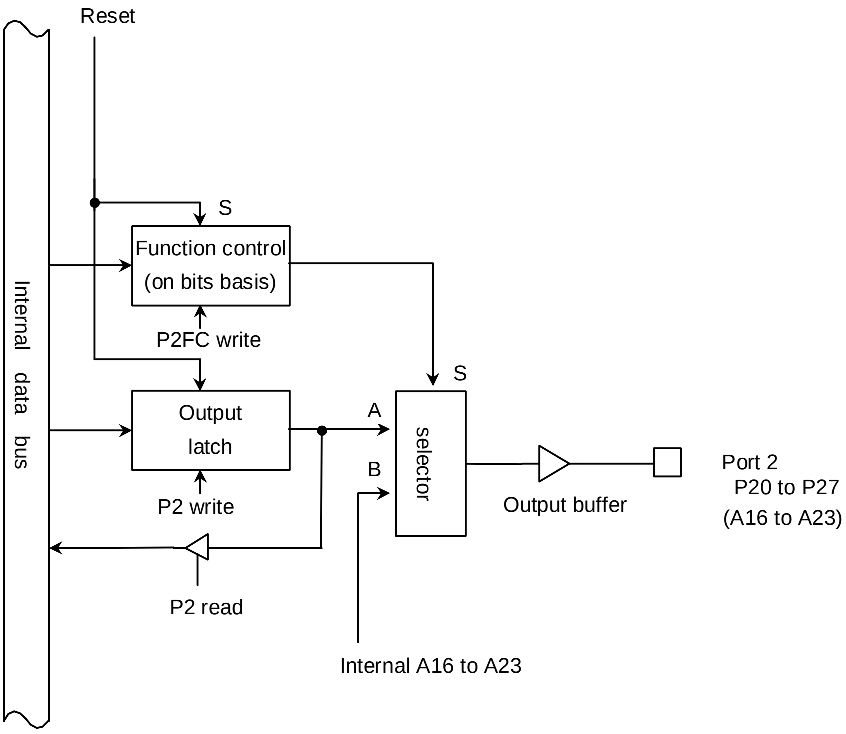


Figure 3.5.2 Port 2

Port 1 Register

		7	6	5	4	3	2	1	0
P1 (0001H)	bit Symbol	P17	P16	P15	P14	P13	P12	P11	P10
	Read/Write	R/W							
	After Reset	Input mode (Output latch register is cleared to 0.)							

Port 1 Control Register

		7	6	5	4	3	2	1	0
P1CR (0004H)	bit Symbol	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
	Read/Write	W							
	After Reset	0	0	0	0	0	0	0	0
	Function	0: IN 1: OUT							

Port 1 I/O setting
0: Input
1: Output

Port 2 Register

		7	6	5	4	3	2	1	0
P2 (0006H)	bit Symbol	P27	P26	P25	P24	P23	P22	P21	P20
	Read/Write	R/W							
	After Reset	Output latch register is set to “1”							

Port 2 Function Register

		7	6	5	4	3	2	1	0
P2FC (0009H)	bit Symbol	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
	Read/Write	W							
	After Reset	1	1	1	1	1	1	1	1
	Function	0: Port 1: Address bus (A23 to A16)							

(note): Read-modify-write is prohibited for P1CR and P2FC.

Figure 3.5.3 Registers for Ports 1 and 2

3.5.3 Port 5 (P54 to P56)

Port 5 is an 5-bit general-purpose I/O port. I/O is set using control register P5CR and P5FC. Resetting resets all bits of the output latch P5 to "1", the control register P5CR and the function register P5FC to "0" and sets P54 to P56 to input mode with pull-up register.

In addition to functioning as a general-purpose I/O port, Port 5 also functions as I/O for the CPU's control / status signal.

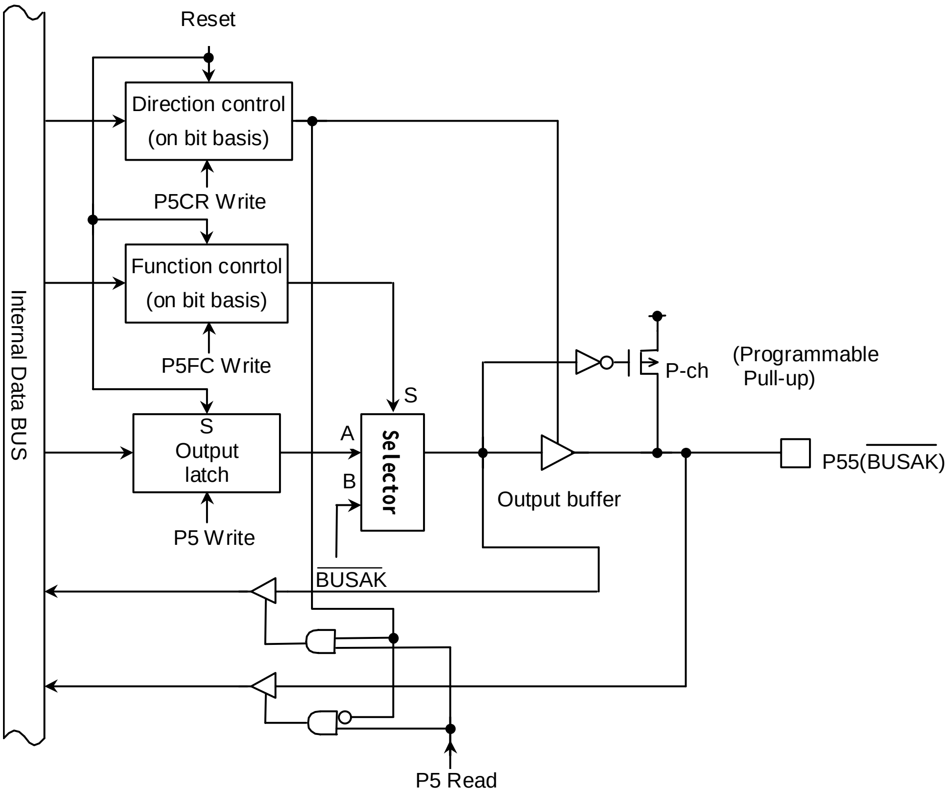


Figure 3.5.4 Port 5 (P55)

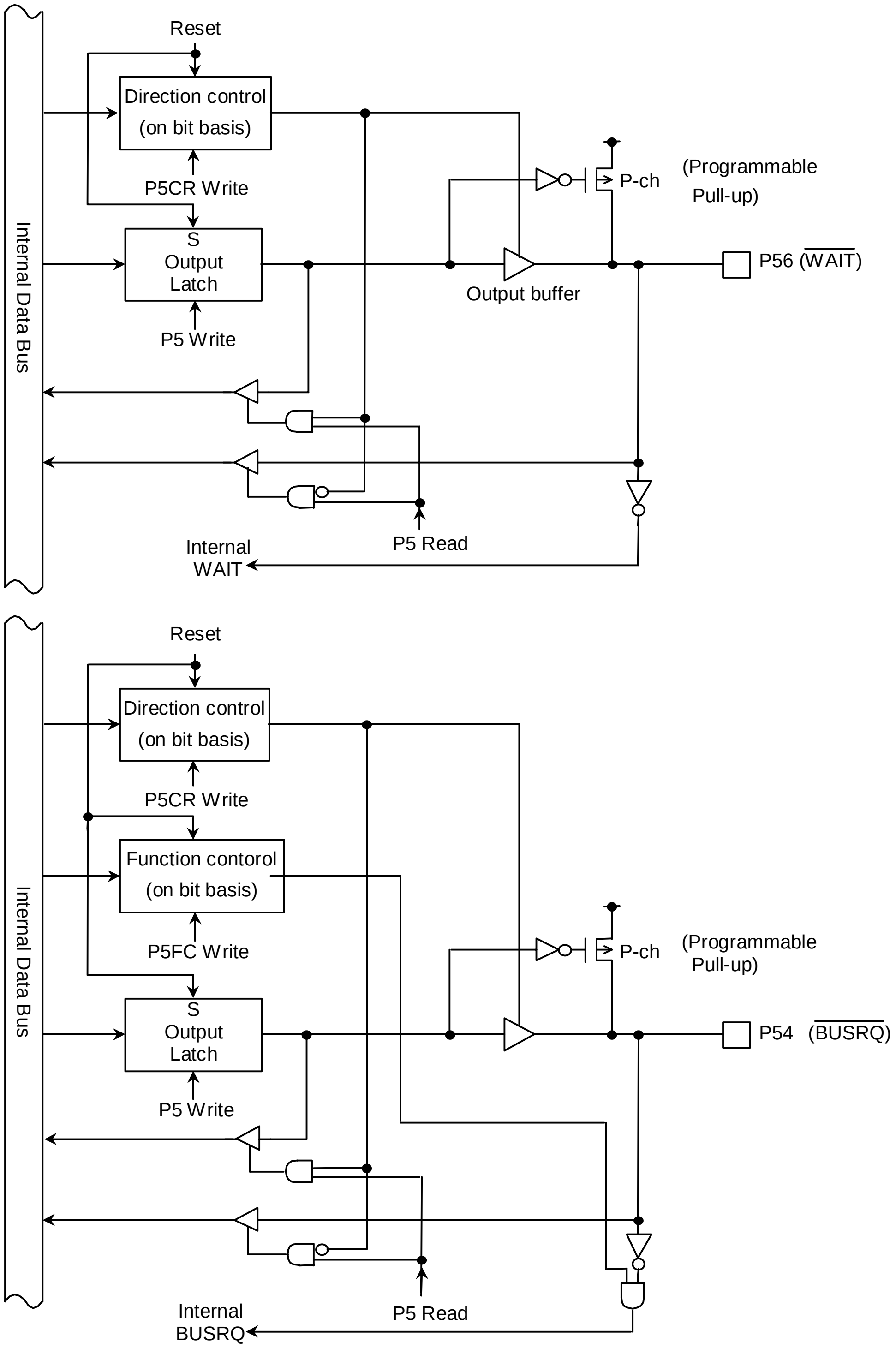


Figure 3.5.5 Port 5 (P56,P54)

Port 5 Register

		7	6	5	4	3	2	1	0
P5 (000DH)	bit Symbol		P56	P55	P54				
	Read/Write	R/W							
	After reset	Input mode (With Pull-up)							
			1	1	1				

Port 5 Control Register

		7	6	5	4	3	2	1	0
P5CR (000AH)	bit Symbol		P56C	P55C	P54C				
	Read/Write	W							
	After reset		0	0	0				
			0: IN 1: OUT						

I/O setting

0	Input
1	Output

Port 5 function register

		7	6	5	4	3	2	1	0
P5FC (000BH)	bit Symbol			P55F	P54F				
	Read/Write	W							
	After reset			0	0				
	Function			0: PORT 1: $\overline{\text{BUSAK}}$	0: PORT 1:$\overline{\text{BUSRQ}}$				

(note1): Read-modify-write is prohibited for register P5CR,P5FC.
(note2): When port5 is used in the input mode, P5 register controls the built-in pull-up resistor. Read-modify-write is prohibited in the input mode or the I/O mode. Setting the built-in pull-up resistor may be depended on the States of the input pin.
(note3): When P56 pin is used as a /WAIT pin ,set P5CR<P56C> to "0" and Chip Select/WAIT control register <BnW2:0> to "010"

Figure 3.5.6 Registers for Port 5

3.5.4 Port6 (P60 to P67)

Port60 to 67 are 8bit output ports. Resetting sets output latch of P62 to “0” and output latchs of P60 to P61,P63 to P67 to “1”.

Port6 also function as chip-select output (/CS0 to /CS3), extend address output(EA24).

Writing “1” in the corresponding bit of P6FC, P6FC2 enables the respective functions.

Resetting resets the P6FC, P6FC2 to “0”, and sets all bits to output ports.

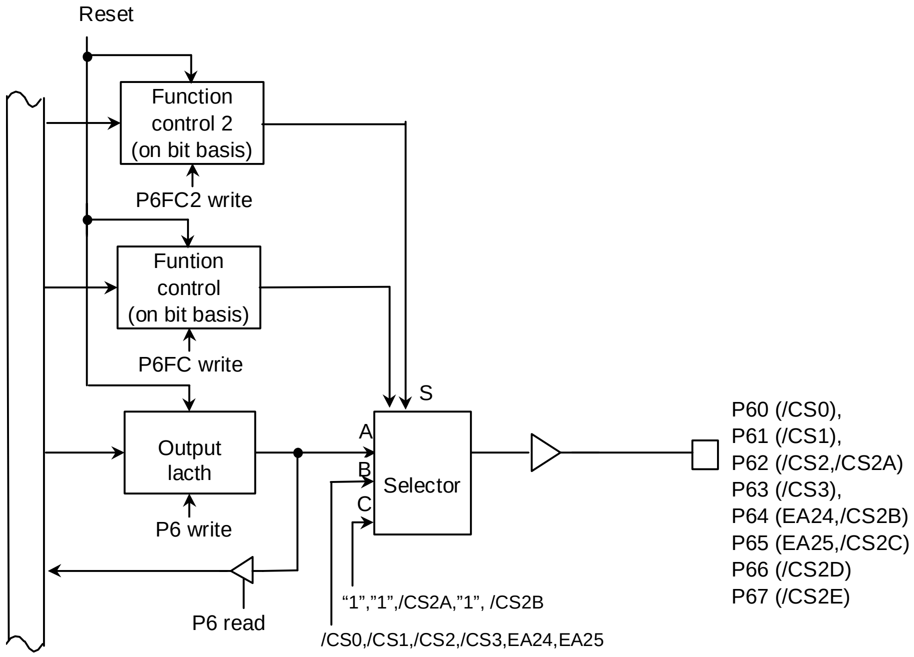


Figure 3.5.7 Port 6

Port 6 Register

		7	6	5	4	3	2	1	0
P6 (0012H)	bit Symbol	P67	P66	P65	P64	P63	P62	P61	P60
	Read/Write	R/W							
	After reset	1	1	1	1	1	0	1	1

Port 6 Function Register

		7	6	5	4	3	2	1	0
P6FC (0015H)	bit Symbol			P65F	P64F	P63F	P62F	P61F	P60F
	Read/Write			W					
	After reset			0					
	Function	Always write “0”		0:PORT 1:EA25	0: PORT 1: EA24	0: PORT 1: /CS3	0: PORT 1: /CS2	0: PORT 1: /CS1	0: PORT 1: /CS0

Port 6 Function Register 2

		7	6	5	4	3	2	1	0
P6FC2 (001BH)	bit Symbol	P67F2	P66F2	P65F2	P64F2		P62F2		
	Read/Write	W					W		
	After reset	0					0		
	Function	0: <P67F> 1: /CS2E	0: <P66F> 1: /CS2D	0: <P65F> 1: /CS2C	0: <P64F> 1: /CS2B	Always write “0”	0: <P62F> 1: /CS2A	Always write “0”	

(note): Read-modify-write is prohibited for P6FC and P6FC2 .

Figure 3.5.8 Register for Port 6

3.5.5 Port7 (P70 to P77)

Port 7 is an 8-bit general-purpose I/O port. I/O can be set on bit basis using the control register. Resetting sets Port 7 to input port and all bits of output latch to”1”.

In addition to functioning as a general-purpose I/O port, Port 7 also functions as follows.

1. Input/output function for serial bus interface(SCK,SO/SDA.SI/SCL)
2. Input/output function for IrDA (OPTRX0,OPTTX0)

Writing “1” in the corresponding bit of P7FC, P7FC2 enables the respective functions.

Resetting resets the P7FC, P7FC2 to “0”, and sets all bits to input ports.

(1) Port70 (SCK, OPTRX0)

Port70 is a general-purpose I/O port. It is also used as SCK(clock signal for SIO mode)and OPTRX0 (receive input for IrDA mode of SIO0).

Used as OPTRX0, it is possible to logical-invert by P7<P70>=”0”.

For PortC1, RXD0 or OPTRX0 is used P7FC2<P70F2>.

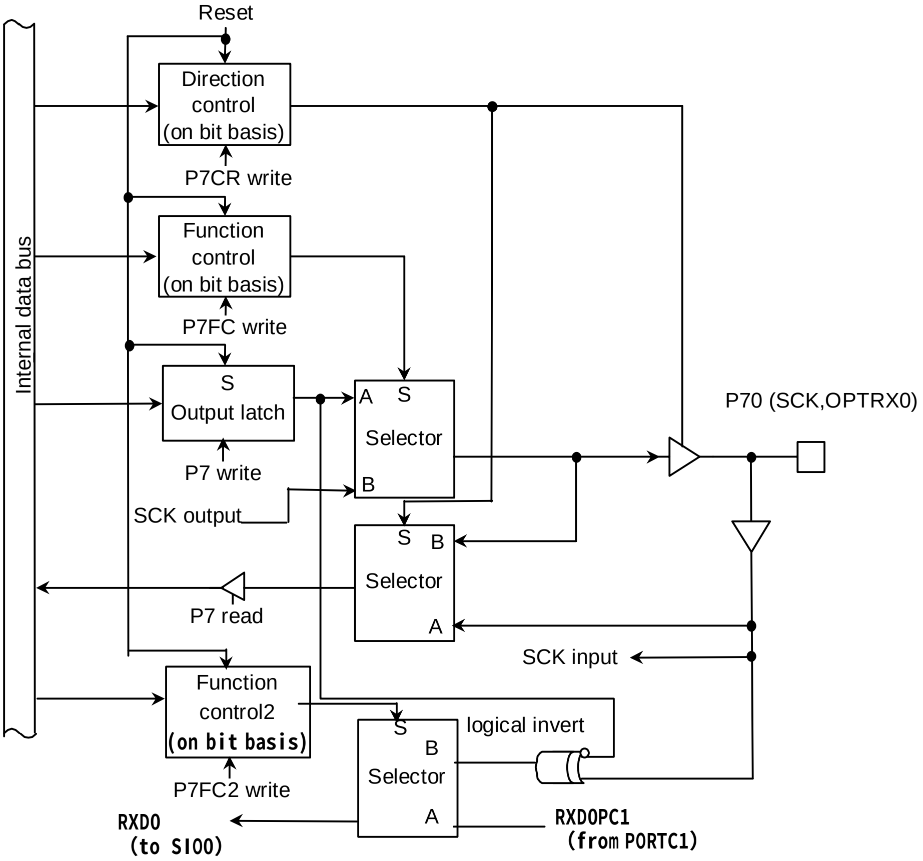


Figure 3.5.9 Port 70

(2) Port71 (SO/SDA/OPTTX0)

Port71 is a general-purpose I/O port. It is also used as SDA (data input for I^{2}C mode), SO (data output for SIO mode) for serial bus interface and OPTTX0 (transmit output for IrDA mode of SIO0).

Used as OPTTX0, it is possible to logical-invert by P7<P71>=”0”.

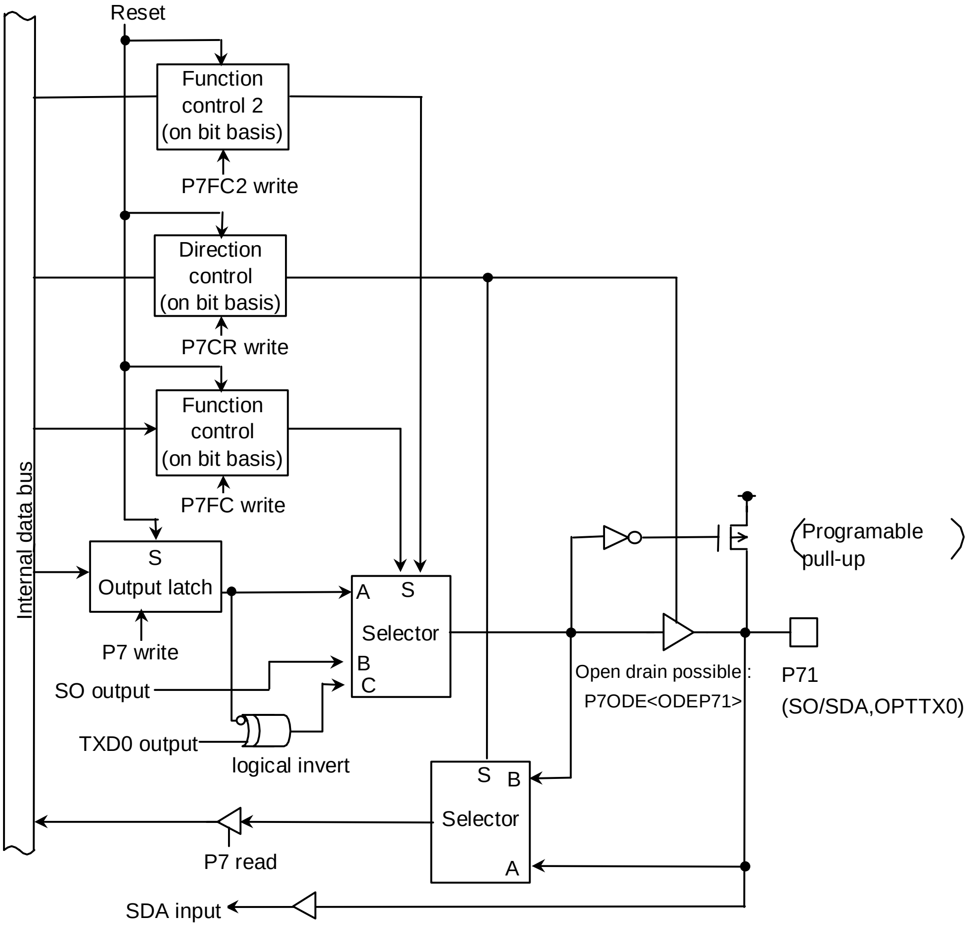


Figure 3.5.10 Port 71

(3) Port 72 (SI/SCL,/HRESET)

Port72 is a general-purpose I/O port. It is also used as SI (data input for SIO mode), SCL (clock input/output for I^{2}C mode) for serial bus interface and input for release hard-protect.

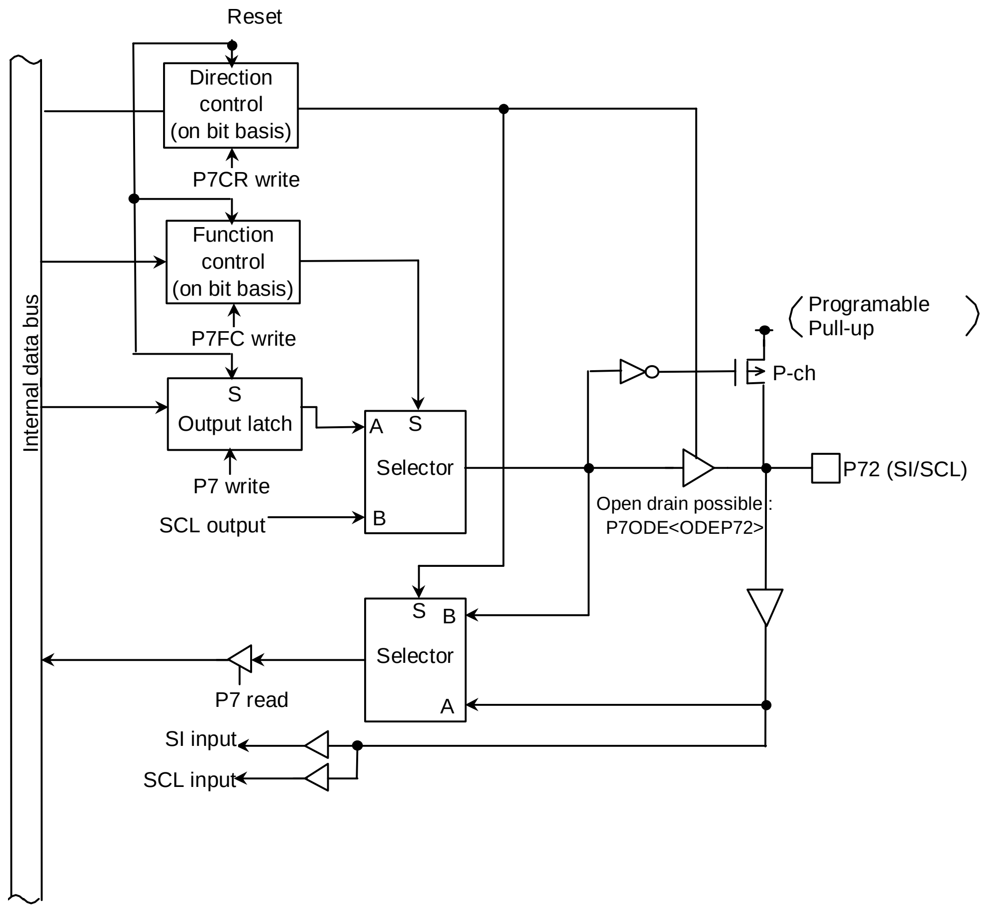


Figure 3.5.11 Port 72

Port 7 Register

		7	6	5	4	3	2	1	0
P7 (0013H)	bit Symbol						P72	P71	P70
	Read/Write	R/W							
	After reset	Input mode							
							1	1	1

Port 7 Control Register

		7	6	5	4	3	2	1	0
P7CR (0016H)	bit Symbol						P72C	P71C	P70C
	Read/Write	W							
	After reset						0	0	0
		0: IN 1: OUT							

Port 7 Function Register

		7	6	5	4	3	2	1	0
P7FC (0017H)	bit Symbol						P72F	P71F	P70F
	Read/Write	W							
	After reset	0							
	Function						0: PORT 1: SCL output	0: PORT 1: SDA/SO output	0: PORT 1: SCK output

Port 7 Function Register 2

		7	6	5	4	3	2	1	0
P7FC2 (001CH)	bit Symbol						P72F2	P71F2	P70F2
	Read/Write			W					
	After reset			0					
	Function						Always write to ‘0’	0: <P71F> 1: OPTTX0	SIO0 RXD Pin select 0: RXD0(PC1) 1:OPTRX0 (P70)

Port 7 ODE Register

		7	6	5	4	3	2	1	0
P7ODE (001FH)	bit Symbol						ODEP72	ODEP71	
	Read/Write								
	After reset						0	0	
	Function						0: 3-STATE 1: Open Drain		

(note): Read-modify-write is prohibited for P7CR,P7FC, P7FC2 and P7ODE.

Figure 3.5.12 Register for Port 7

3.5.6 Port 8 (P80 to P87)

Port 8 is an 8-bit input port and can also be used as the analog input pins for the internal A/D converter. P83 can also be used as ADTRG pin for the A/D converter.

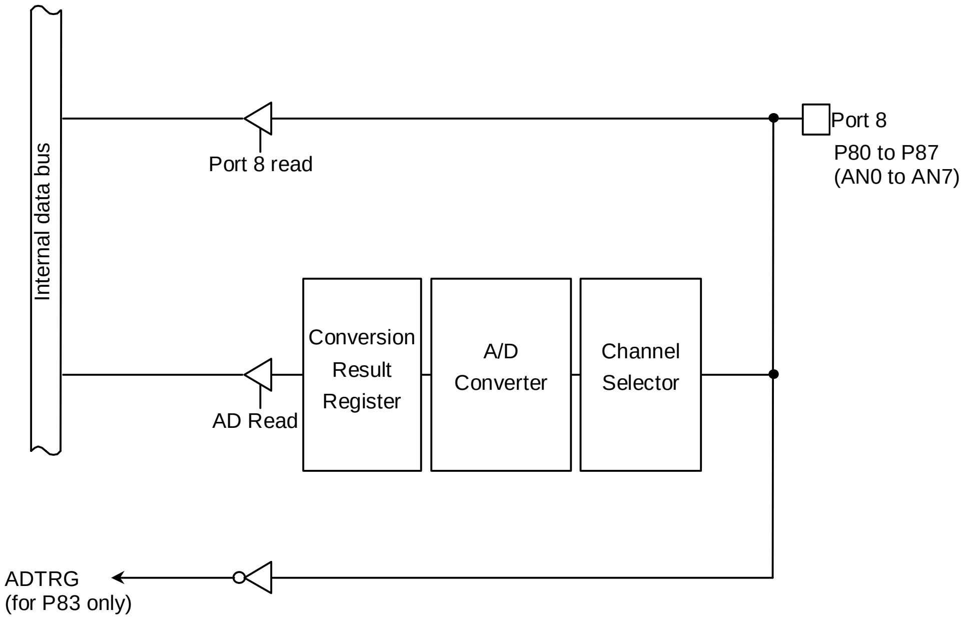


Figure 3.5.13 Port 8

Port 8 Register

		7	6	5	4	3	2	1	0
P8 (0018H)	Bit Symbol	P87	P86	P85	P84	P83	P82	P81	P80
	Read/Write	R							
	After reset	Input mode							

(note): The input channel selection of A/D Converter and the permission of ADTRG input are set by A/D Converter mode register ADMOD1.

Figure 3.5.14 Register for Port 8

Port B (PB0 to PB6)

Port B0 to PB6 is a 7-bit general-purpose I/O port. Each bit can be set individually for input or output. Resetting sets Port B to be an input port.

In addition to functioning as a general-purpose I/O port, Port B0 has clock input terminal TA0IN of 8 bits timer 0, and port B1, B2 each has facility of 8 bits timer listing TA1OUT, TA3OUT terminal. And, port B3 to B6 has each external interruption input facility of INT0 to INT3. Edge selection of external interruption is establishes by IIMC register in the interrupt controller.

Timer output function and external interrupt function can be enabled by writing “1” to the corresponding bits in the Port B Function Register (PBFC). Resetting resets all bits of the registers PBCR and PBFC to “0”, and sets all bits to be input ports.

(1) PB0 to PB2

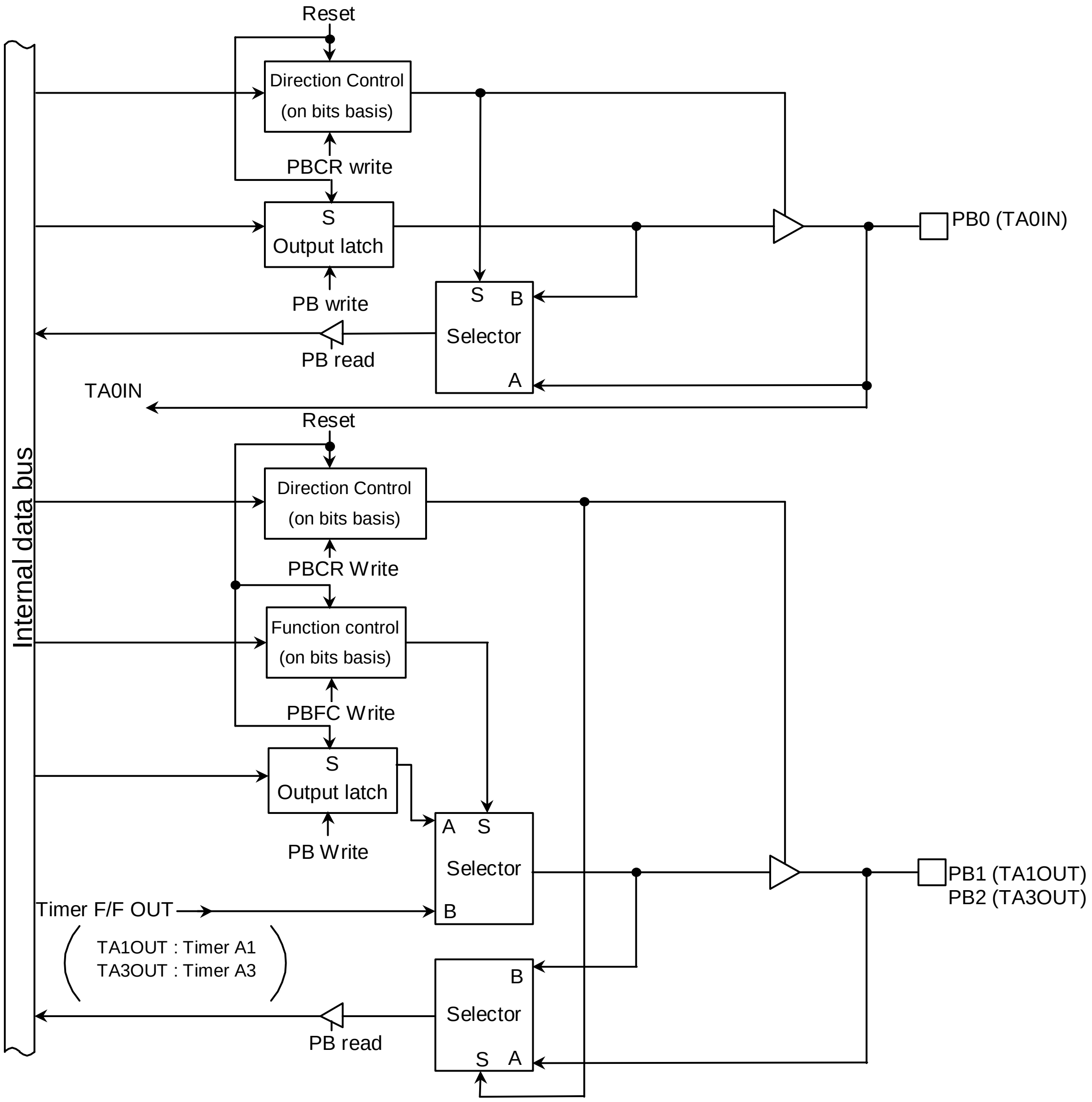


Figure 3.5.15 Port B0 to B2

(2) PB3 (INT0), PB4 (INT1)-PB6 (INT3)

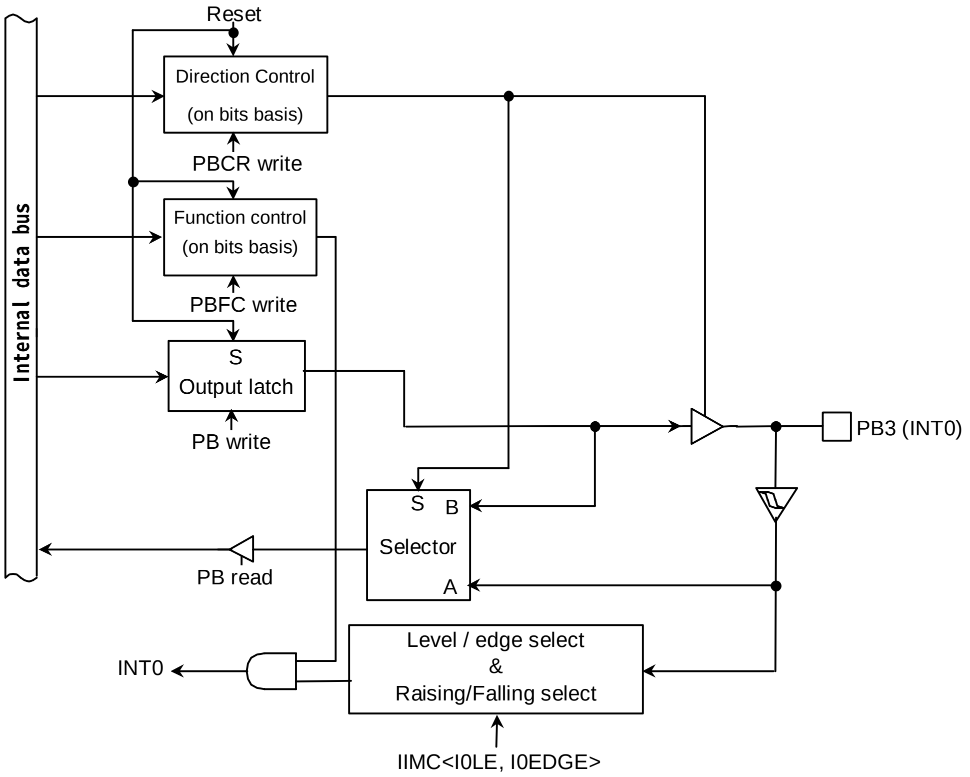


Figure 3.5.16 Port B3

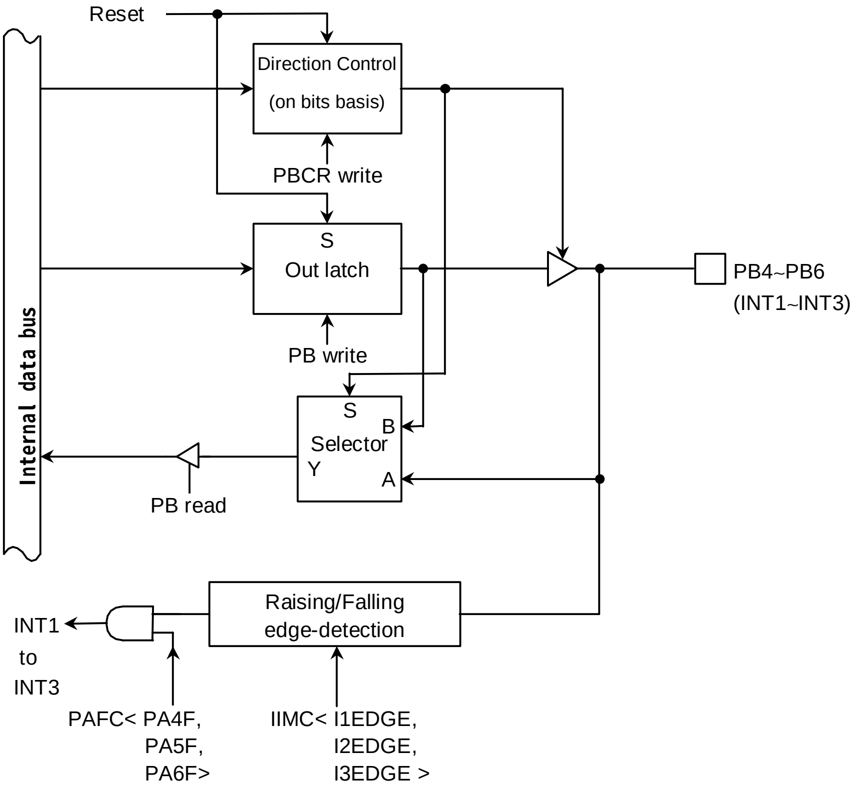


Figure 3.5.17 Port B4 to B6

Port B Register

		7	6	5	4	3	2	1	0
PB (0022H)	bit Symbol		PB6	PB5	PB4	PB3	PB2	PB1	PB0
	Read/Write		R/W						
	After Reset		Input Mode						
			1	1	1	1	1	1	1

Port B Control Register

		7	6	5	4	3	2	1	0
PBCR (0024H)	bit Symbol		PB6C	PB5C	PB4C	PB3C	PB2C	PB1C	PB0C
	Read/Write		W						
	After Reset		0						
			0: IN 1: OUT						

Port B Function Register

		7	6	5	4	3	2	1	0
PBFC (0025H)	bit Symbol		PB6F	PB5F	PB4F	PB3F	PB2F	PB1F	
	Read/Write		W						
	After Reset		0						
	Function		0: PORT 1: INT3	0: PORT 1: INT2	0: PORT 1: INT1	0: PORT 1: INT0	0: PORT 1: TA3OUT	0: PORT 1: TA1OUT	

(note1): Read-Modify-Write is prohibited for the registers PBCR and PBFC.

(note2): PB0/TA0IN 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 3.5.18 Register for Port B

3.5.7 Port C (PC0 to PC5)

Port C0 to C5 are 6-bit general-purpose I/O ports. Each bit can be set individually for input or output. Resetting sets PC0 to PC5 to be an input ports. It also sets all bits of the output latch register to "1".

In addition to functioning as general-purpose I/O port pins, PC0 to PC5 can also function as the I/O for serial channels 0 and 1. A pin can be enabled for I/O by writing "1" to the corresponding bit of the Port C Function Register (PCFC).

Resetting resets all bits of the registers PCCR and PCFC to 0 and sets all pins to be input ports .

(1) Port C0, C3 (TXD0/TXD1)

As well as functioning as I/O port pins, port C0 and C3 can also function as serial channel TXD output pins. In case of use TXD0/TXD1, it is possible to logical invert by setting the register PC<PC0,PC3>.

And port C0 to C3 have a programmable open drain function which can be controled by the register PCODE<ODEPC0, ODEPC3>.

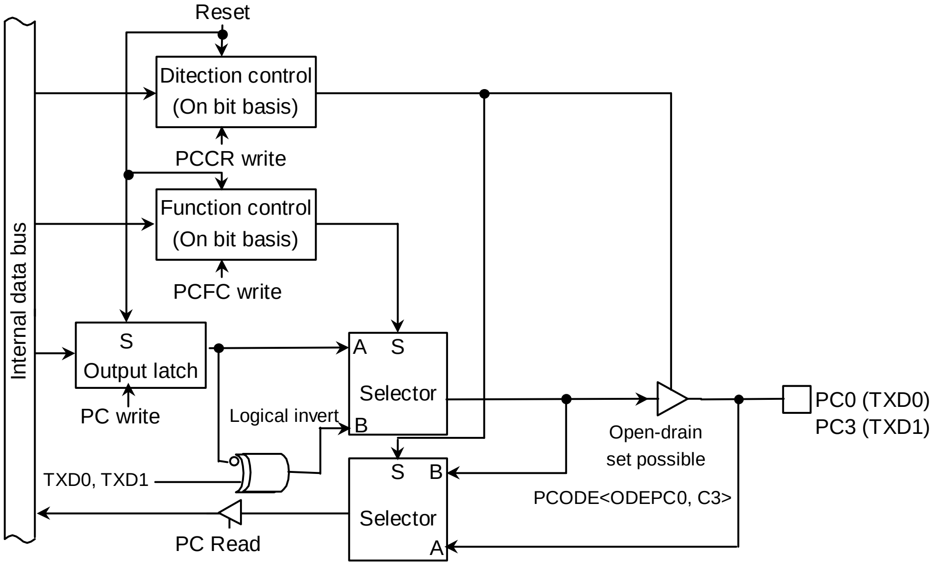


Figure 3.5.19 Port C0 and C3

(2) Port C1, C4 (RXD0, 1)

Port C1 and C4 are I/O port pins and can also is used as RXD input for the serial channels. In case of use RXD0/RXD1, it is possible to logical invert by setting the register PC<PC1,PC4>.

And input data of SIO0 can be select from RXD/PC1 pin or OPTRX0/P70 by setting the register PCFC2<P70F2>.

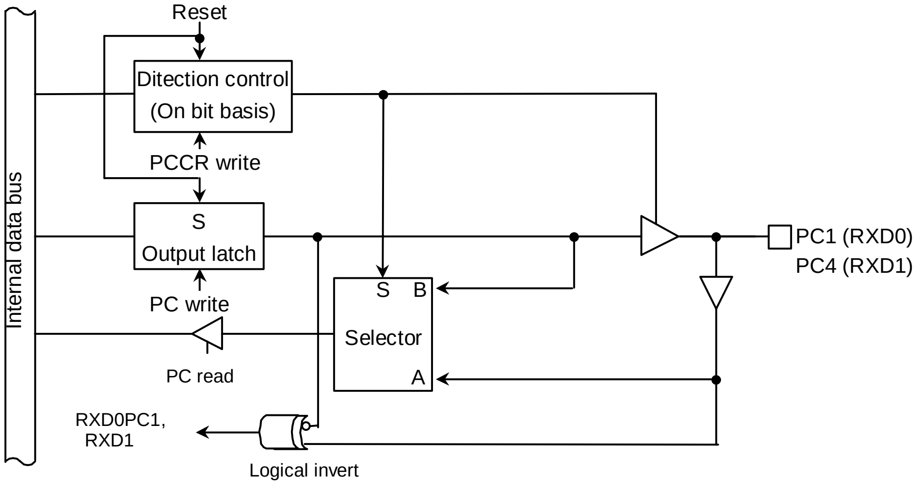


Figure 3.5.20 Port C1 and C4

(3) Port C2(/CTS0,SCLK0),C5(/CST1,SCLK1)

Port C2 and C4 are I/O port pins and can also is used as /CTS input or SCLK input/output for the serial channels. In case of use /CTS,SCLK, it is possible to logical invert by setting the register PC<PC2,PC5>.

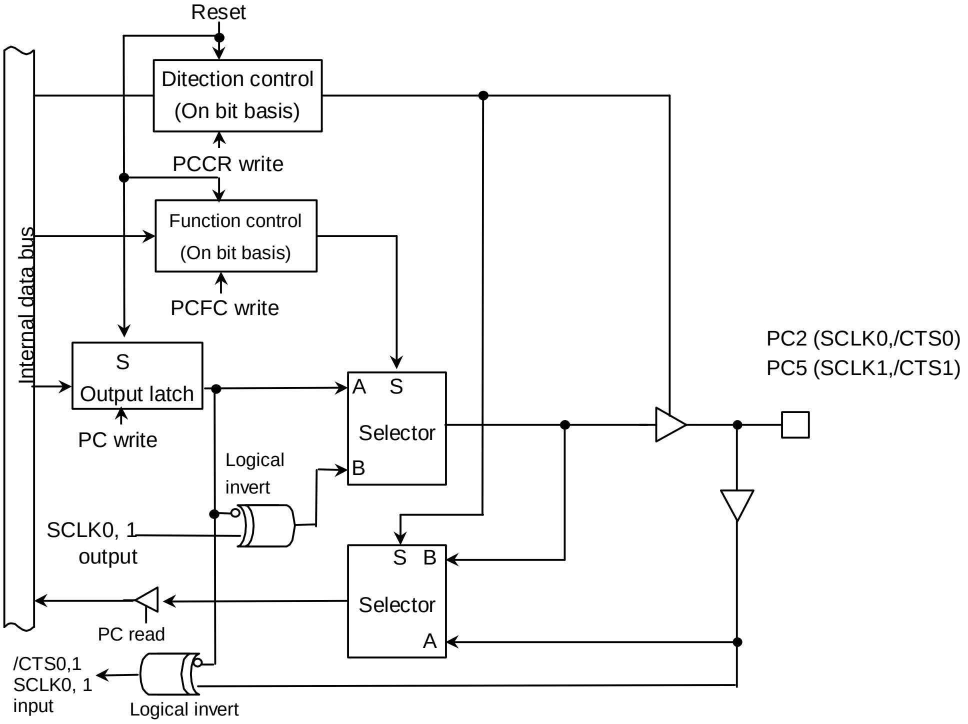


Figure 3.5.21 Port C2 and C5

Port C Register

PC
(0023H)

	7	6	5	4	3	2	1	0
bit Symbol			PC5	PC4	PC3	PC2	PC1	PC0
Read/Write			R/W					
After Reset			Input mode					
			1	1	1	1	1	1

Port C Control Register

PCCR
(0026H)

	7	6	5	4	3	2	1	0
bit Symbol			PC5C	PC4C	PC3C	PC2C	PC1C	PC0C
Read/Write			W					
After Reset			0	0	0	0	0	0
			0: IN 1: OUT					

Port C Functon Register

PCFC
(0027H)

	7	6	5	4	3	2	1	0
bit Symbol			PC5F		PC3F	PC2F		PC0F
Read/Write			W		W	W		W
After Reset			0		0	0		0
Function			0: PORT 1: SCLK1 Output		0: PORT 1: TXD1	0: PORT 1: SCLK0 Out put		0: PORT 1: TXD0

Port C ODE Register

PCODE
(0028H)

	7	6	5	4	3	2	1	0
bit Symbol					ODEPC3			ODEPC0
Read/Write					W			W
After Reset					0			0
Function					TXD1 0: CMOS 1: Open Drain			TXD0 0: CMOS 1: Open Drain

(note1): Read-Modify-Write is prohibited for the registers PCCR, PCFC and PCODE.
(note2): PC1/RXD0, PC4/RXD1 pins do not have a register changing PORT/FUNCTION. For example, when it is used as an input port, the input signal is inputted to SIO as the cereal receive data.

Figure 3.5.22 Register for Port C

3.5.8 Port D (PD0 to PD7)

Port D is an 8-bit output port. Resetting sets the output latch PD to “1”, and PD5 to PD7 pin output “1”.

In addition to functioning as output port, Port D also function as output pin for output pin for internal clock (SCOUT), output pin for RTC alarm (/ALARM) and output pin for melody/alarm generator (MLDALM,/MLDALM). Above setting is used the function register PDFC.

Only PD6 has two output functions which /ALARM and /MLDALM. This selection is used PD<PD6>. Resetting resets the function register PDFC to “0”, and sets all ports to output ports.

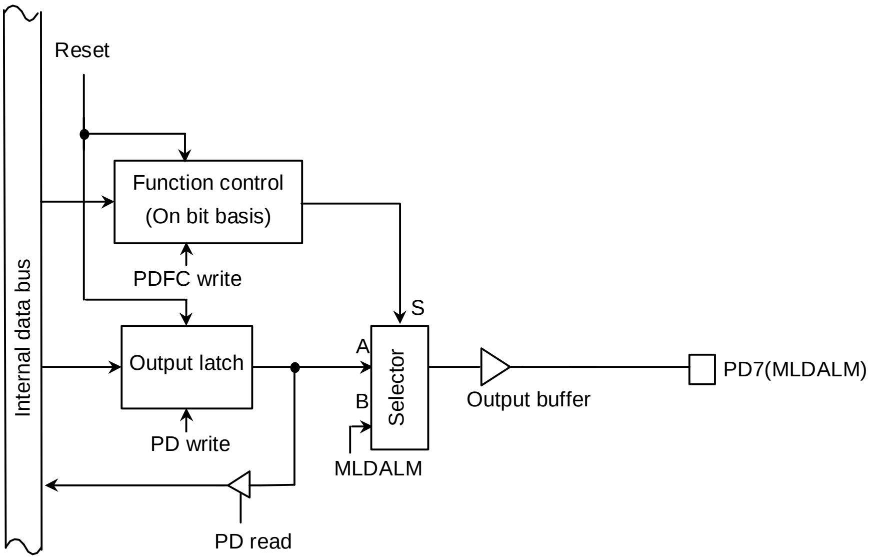


Figure 3.5.23 Port D

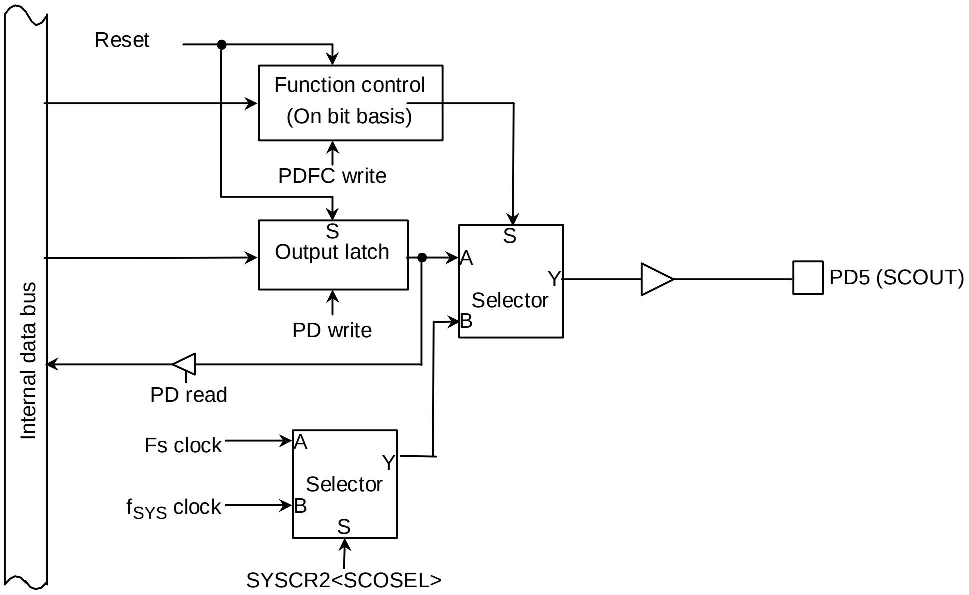


Figure 3.5.24 Port D

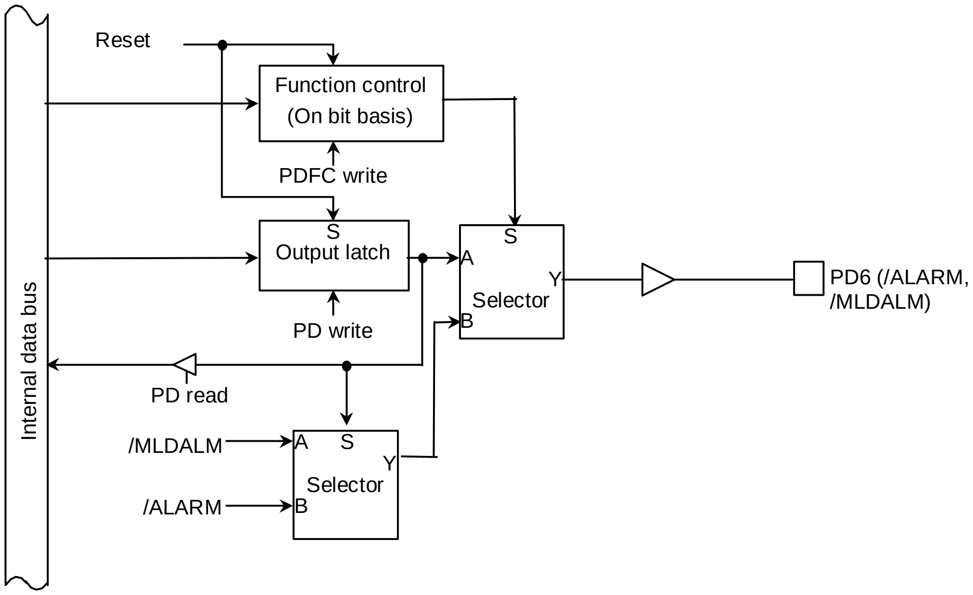


Figure 3.5.25 Port D

Port D register

PD (0029H)		7	6	5	4	3	2	1	0
	bit Symbol	PD7	PD6	PD5					
	Read/Write	R/W							
	After Reset	1	1	1					

Port D function register

PDFC (002AH)		7	6	5	4	3	2	1	0
	bit Symbol	PD7F	PD6F	PD5F					
	Read/Write	W							
	After Reset	0							
	Function	0: PORT 1: MLDALM	0: PORT 1: /ALARM @<PD6>=1 1: /MLDALM @<PD6>=0	0: PORT 1: SCOUT					

(note): Read-Modify-Write is prohibited for the registers PDFC.

Figure 3.5.26 Register for Port D

3.5.9 Port Z(PZ2 to PZ3)

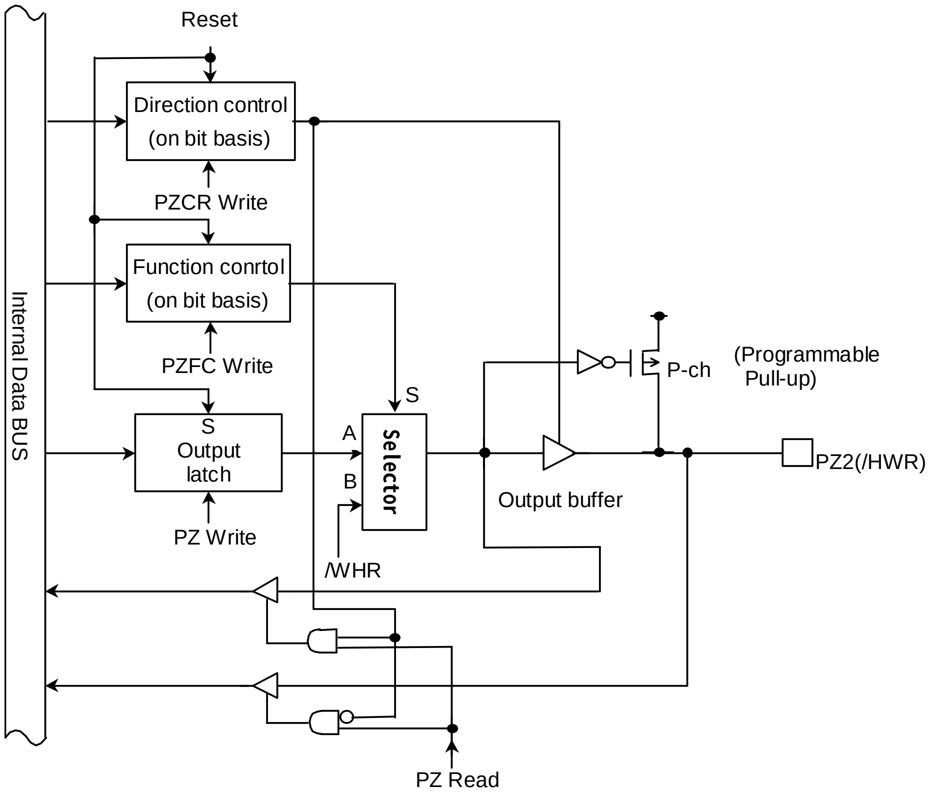
Port Z is the 2-bit general-purpose I/O port. I/O is set using control register PZCR and PZFC. Resetting resets all bits of the output latch PZ to “1”.

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

Resetting initializes PZ2 and PZ3 pins to input mode with pull-up register.

When the PZ<RDE> register clearing to “0”,outputs the $\overline{\text{RD}}$ strobe (used for the peused static RAM) of the $\overline{\text{RD}}$ pin even when the internal addressed.

If the <RDE> remains “1”, the $\overline{\text{RD}}$ strobe signal is output only when the external address are is accessed.



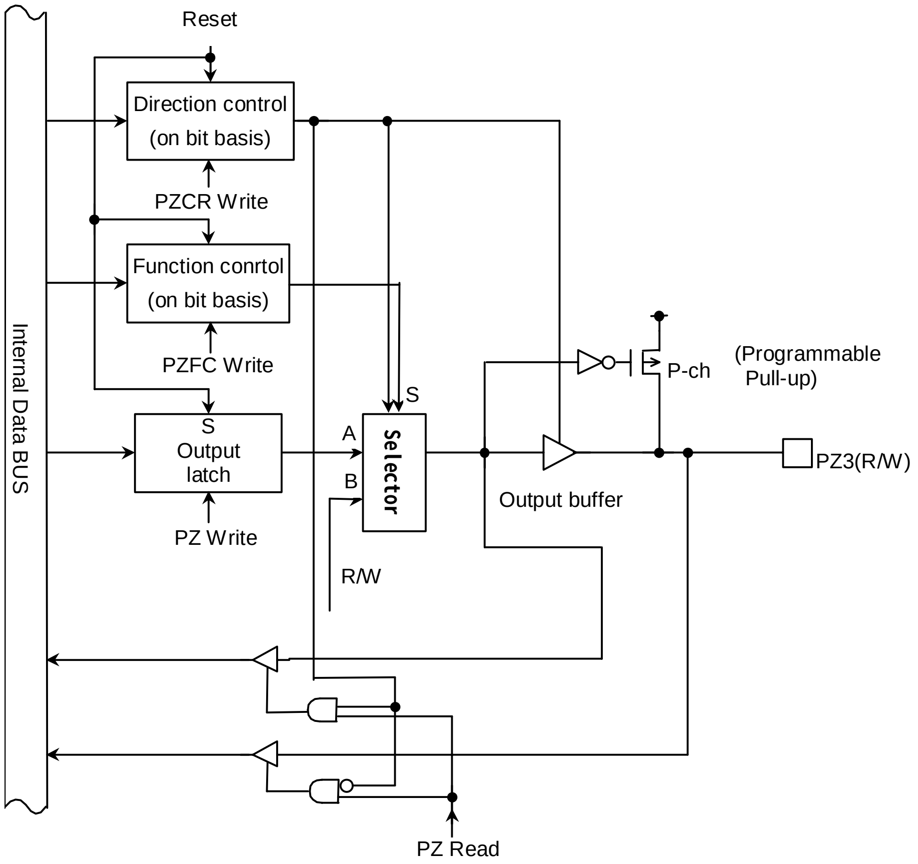


Figure 3.5.27 Port Z (PZ2,PZ3)

Port Z Register

		7	6	5	4	3	2	1	0
PZ (007DH)	bit Symbol					PZ3	PZ2		RDE
	Read/Write	R/W							
	After reset	Input mode (pulled-up)							
						1	1		1

Port Z Control Register

		7	6	5	4	3	2	1	0
PZCR (007EH)	bit Symbol					PZ3C	PZ2C		
	Read/Write	W							
	After reset					0	0		
						0: IN	1: OUT		

Port Z Function Register

		7	6	5	4	3	2	1	0
PZFC (007FH)	Bit Symbol					PZ3F	PZ2F		
	Read/Write	W							
	After reset					0	0		
	Function	(Note) Always fixed to “0”				0: PORT 1: R/W	0: PORT 1:/HWR		

/HWR setting

PZFC<PZ2F>	1
PZCR<PZ2C>	1

Note 1: Read-Modify-write is prohibited for registers PZCR and PZFC.

Note 2: When Port Z is used in Input Mode, the PZ register controls the built-in pull-up resistor. Read-Modify-Write is prohibited in Input Mode or I/O Mode. Setting the built-in pull-up resistor may be depended on the states of the input pin.

Figure 3.5.28 Port Register for Port Z

3.6 Chip Select/Wait Controller

On the TM91C824, 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 /CS0 to /CS3 (which can also function as port pins P60 to P63) are the respective output pins for the areas CS0 to CS3. When the CPU specifies an address in one of these areas, the corresponding /CS0 to /CS3 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 6 Function Register P6FC must be set.

/CS2A to /CS2E (CS pin except /CS0 to /CS3) are made by MMU.

These pins is /CS pin that area and BANK value is fixed without concern in setting of CS/WAIT controller.

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/Disable status the data bus width and the number of waits for each address area.

The input pin controlling these states is the bus wait request pin ($\overline{\text{WAIT}}$).

3.6.1 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 /CS0 to /CS3 pin outputs the chip select signal and the bus cycle operates in accordance with the settings in chip select/wait control register B0CS to B3CS. (See 3.6.2, Chip Select/Wait Control Registers.)

(1) Memory Start Address Registers

Figure 3.6.1 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 eight 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. Figure 3.6.2 shows the relationship between the start address and the start address register value.

Memory Start Address Registers (for areas CS0 to CS3)

MSAR0 (00C8H) / MSAR1 (00CAH)
MSAR2 (00CCH) / MSAR3 (00CEH)

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

→ Sets start addresses for areas CS0 to CS3.

Figure 3.6.1 Memory Start Address Register

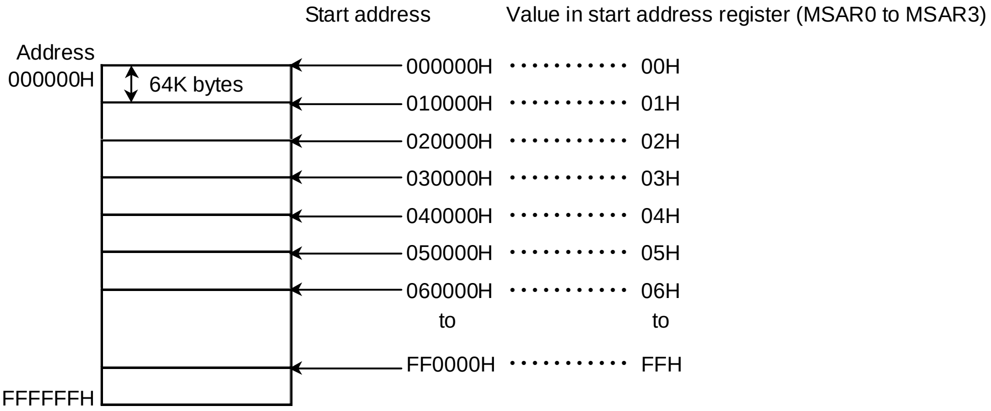


Figure 3.6.2 Relationship between Start Address and Start Address Register value

(2) Memory Address Mask Registers

Figure 3.6.3 shows the Memory Address Mask Registers. Memory address mask registers MAMR0 to MAMR3 are used to set the size of the CS0 to CS3 areas by specifying a mask for each bit of the start address set in memory start address registers MAMR0 to MAMR3. 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.

Memory address mask register (for CS0 area)

MAMR0
(00C9H)

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

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

Memory address mask register (CS1)

MAMR1
(00CBH)

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

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

Memory address mask register (CS2, CS3)

MAMR2 / MAMR3
(00CDH) / (00CFH)

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

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

Figure 3.6.3 Memory Address mask Registers

(3) Setting Memory Start Addresses and Address Areas

Figure 3.6.4 show an example of specifying a 64K-byte address area starting from 010000H using the CS0 areas.
Set “01H” in memory start address register MSAR0<S23 to S16>(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 to V8>sets the area size This example sets “07H” in MAMR0 to specify a 64K-byte area.

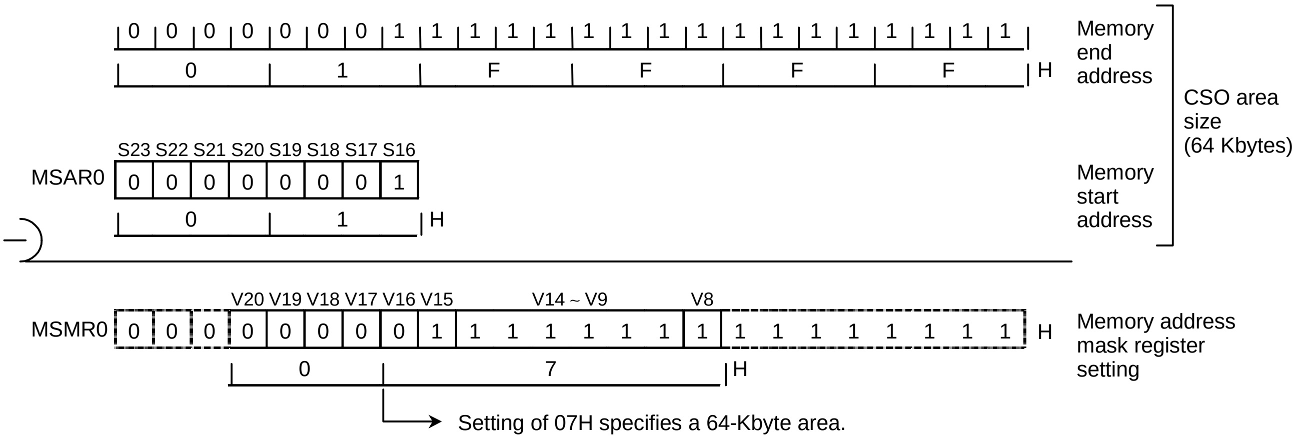

0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
0 1 F F F F H
Memory end address
CSO area size (64 Kbytes)
S23 S22 S21 S20 S19 S18 S17 S16
MSAR0 0 0 0 0 0 0 0 1
0 1 H
Memory start address
V20 V19 V18 V17 V16 V15 V14 ~ V9 V8
MSMR0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 H
0 7 H
Memory address mask register setting
Setting of 07H specifies a 64-Kbyte area.

Figure 3.6.4 Example showing how to set the CS0 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”.this disabling the CS0, CS1 and CS3 areas. However, as B2CS<B2M> to “0” and B2CS<B2E> to “1”, CS2 is enabled from 000FE0H-000FFFH to 003000H-FFFFFFH in TMP91C824. Also, the bus width and number of waits specified in BEXCS are used for accessing addresses outside the specified CS0 to CS3 area. (See 3.6.2, Chip Select/Wait Control Registers.)

(4) Address Area Size Specification

Table 3.6.1 shows the relationship between CS area and area size.△ Indicates areas that cannot be set by memory start address register and address mask register combinations. When setting an area size using a combination indicated by △ , set the start address mask register in the desired steps starting from 000000H.

If the CS2 area is set to 16M-bytes 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:

① Valid start addresses

000000H
128K bytes
020000H
128K bytes
040000H
128K bytes
060000H
⋮

Any of these addresses may be set as the start address.

② Invalid start addresses

000000H
64K bytes ← This is not an integer multiple of the desired area size setting. Hence, none of these addresses can be set as the start address.
010000H
128K bytes
030000H
128K bytes
050000H
⋮

Table 3.6.1 Valid area sizes for each CS area

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

(note): △ Indicates areas that cannot be set by memory start address register and address mask register combinations.

3.6.2 Chip Select/Wait Control Registers

Figure 3.6.5 lists the Chip Select/Wait Control Registers.

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

Chip Select/Wait Control Register

		7	6	5	4	3	2	1	0
B0CS (00C0H)	Bit symbol	B0E		B0OM1	B0OM0	B0BUS	B0W2	B0W1	B0W0
	Read/Write	W		W					
Read-Modify-Write instructions are prohibited.	After Reset	0		0	0	0	0	0	0
	Function	0: Disable 1: Enable		Chip Select output waveform selection 00: For ROM/SRAM 01: 10: Don't care 11:		Data bus width 0: 16 bits 1: 8 bits	Number of Waits 000: 2 waits 001: 1 wait 010: 1 wait + N 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					
Read-Modify-Write instructions are prohibited.	After Reset	0		0	0	0	0	0	0
	Function	0: Disable 1: Enable		Chip Select output waveform selection 00: For ROM/SRAM 01: 10: Don't care 11:		Data bus width 0: 16 bits 1: 8 bits	Number of Waits 000: 2 waits 001: 1 wait 010: 1 wait + N 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							
Read-Modify-Write instructions are prohibited.	After Reset	1	0	0	0	0	0	0	0
	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: Don't care 11:		Data bus width 0: 16 bits 1: 8 bits	Number of waits 000: 2 waits 001: 1 wait 010: 1 wait + N 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					
Read-Modify-Write instructions are prohibited.	After Reset	0		0	0	0	0	0	0
	Functions	0: Disable 1: Enable		Chip Select output waveform selection 00: For ROM/SRAM 01: 10: Don't care 11:		Data bus width 0: 16 bits 1: 8 bits	Number of waits 000: 2 waits 001: 1 wait 010: 1 wait + N 011: 0 waits	100: reserved 101: 3 waits 110: 4 waits 111: 8 waits	
BEXCS (00C7H)	Bit Symbol					BEXBUS	BEXW2	BEXW1	BEXW0
	Read/Write					0			
Read-Modify-Write instructions are prohibited.	After Reset					0	0	0	0
	Functions					Data bus width 0: 16 bits 1: 8 bits	Number of Waits 000: 2 waits 001: 1 wait 010: 1 wait + N 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

(1) Master Enable bits

Bit 7 (<B0E>, <B1E>, <B2E> or <B3E>) of a chip select/wait control register is the master bit which 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 enabled (sets to “1”) <B2E>. This enables area CS2 only.

(2) Data bus width selection

Bit 3 (<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 3.6.2.

Table 3.6.2 Dynamic bus sizing

Operand Data Bus Width	Operand Start Address	Memory Data Bus Width	CPU Address	CPU Data	
				D15 to D8	D7 to D0
8 bits	2n + 0 (Even number)	8 bits	2n + 0	xxxxx	b7 ~ b0
		16 bits	2n + 0	xxxxx	b7 ~ b0
	2n + 1 (Odd number)	8 bits	2n + 1	xxxxx	b7 ~ b0
		16 bits	2n + 1	b7 ~ b0	xxxxx
16 bits	2n + 0 (Even number)	8 bits	2n + 0	xxxxx	b7 ~ b0
			2n + 1	xxxxx	b15 ~ b8
		16 bits	2n + 0	b15 ~ b8	b7 ~ b0
	2n + 1 (Odd number)	8 bits	2n + 1	xxxxx	b7 ~ b0
			2n + 2	xxxxx	b15 ~ b8
		16 bits	2n + 1	b7 ~ b0	xxxxx
			2n + 2	xxxxx	b15 ~ b8
32 bits	2n + 0 (Even number)	8 bits	2n + 0	xxxxx	b7 ~ b0
			2n + 1	xxxxx	b15 ~ b8
			2n + 2	xxxxx	b23 ~ b16
			2n + 3	xxxxx	b31 ~ b24
		16 bits	2n + 0	b15 ~ b8	b7 – b0
			2n + 2	b31 ~ b24	b23 – b16
	2n + 1 (Odd number)	8 bits	2n + 1	xxxxx	b7 ~ b0
			2n + 2	xxxxx	b15 ~ b8
			2n + 3	xxxxx	b23 ~ b16
			2n + 4	xxxxx	b31 ~ b24
		16 bits	2n + 1	b7 ~ b0	xxxxx
			2n + 2	b23 ~ b16	b15 ~ b8
			2n + 4	xxxxx	b31 ~ 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 to B0W2>, <B1W0 to B1W2>, <B2W0 to B2W2>, <B3W0 to B3W2>, <BEXW0 to BEXW2>) 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.

Table 3.6.3 Wait operation settings

<BxW2 ~ BxW0>	No. of Waits	Wait Operation
000	2WAIT	Inserts a wait of 2 states, irrespective of the $\overline{\text{WAIT}}$ pin state.
001	1WAIT	Inserts a wait of 1 state, irrespective of the $\overline{\text{WAIT}}$ pin state.
010	1WAIT + N	Samples the state of the $\overline{\text{WAIT}}$ pin after inserting a wait of one state. If the $\overline{\text{WAIT}}$ pin is Low, the waits continue and the bus cycle is extended until the pin goes high.
011	0WAIT	Ends the bus cycle without a wait, regardless of the $\overline{\text{WAIT}}$ pin state.
100	Reserved	Invalid setting
101	3WAIT	Inserts a wait of 3 state, irrespective of the $\overline{\text{WAIT}}$ pin state.
110	4WAIT	Inserts a wait of 4 state, irrespective of the $\overline{\text{WAIT}}$ pin state.
111	8WAIT	Inserts a wait of 8 state, irrespective of the $\overline{\text{WAIT}}$ pin state.

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 which 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> (bit 6 of the chip select/wait control register for CS2) to “0” designates the 16-Mbyte area 000FE0H-000FFFH, 003000H-FFFFFFH 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 (i.e. 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-M bytes 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:

① Set the Memory Start Address Registers MSAR0 to MSAR3.
Set the start addresses for CS0 to CS3.

② Set the Memory Address Mask Registers MAMR0 to MAMR3.
Set the sizes of CS0 to CS3.

③ 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.

The CS0 to S3 pins can also function as pins P60 to P63. To output a Chip Select signal using one of these pins, set the corresponding bit in the Port 6 Function Register P6FC to "1". If a CS0 "to S3 address is specified which is actually an internal I/O and RAM area address, the CPU accesses the internal address area and no Chip Select signal is output on any of the /CS0 to /CS3 pins.

Setting 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 set to 0.

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

3.6.3 Connecting external memory

Figure 3.6.6 shows an example of how to connect external memory to the TMP91C824. In this example the ROM is connected using a 16-bit bus. The RAM and I/O are connected using an 8-bit bus.

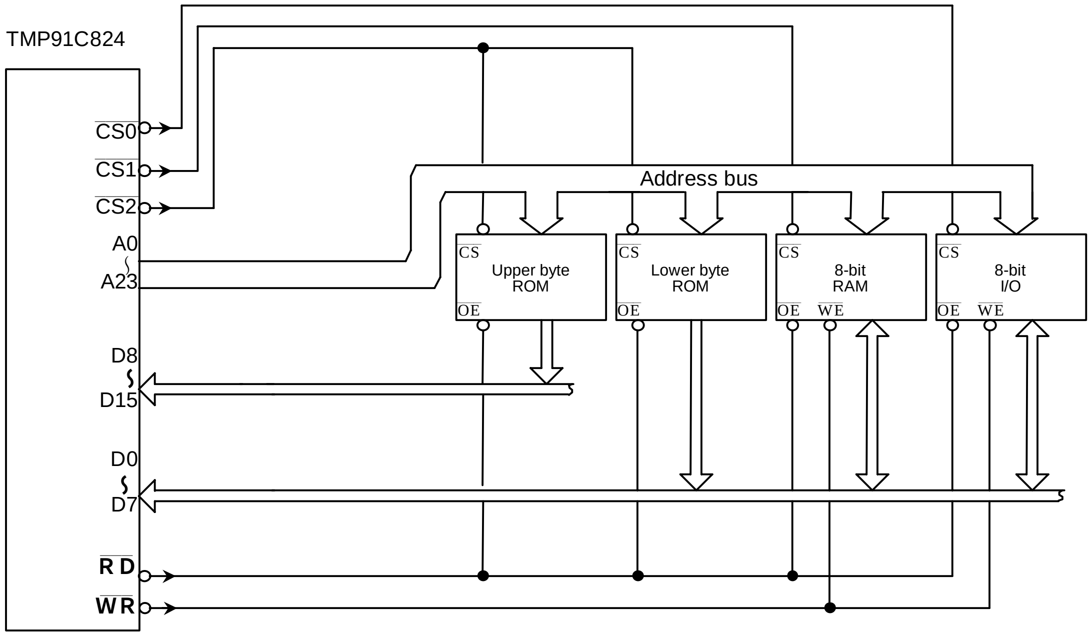


Figure 3.6.6 Example of external memory connection
(ROM uses 16-bit bus; RAM and I/O use 8-bit bus.)

A Reset clears all bits of the Port 6 Control Register P6CR and the Port 6 Function Register P6FC to “0” and disables output of the CS signal. To output the CS signal, the appropriate bit must be set to “1”.

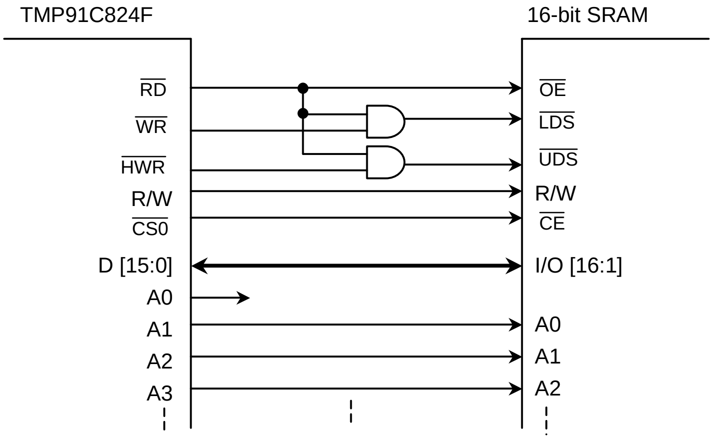


Figure 3.6.7 Example of external memory connection (RAM and I/O use 16-bit bus)

3.7 8-bit Timers (TMRA)

The TMP91C824 features 4 channel(TMRA0 to TMRA3) built-in 8-bit timers.
These timers are paired into 2 modules: TMRA01 and TMRA23. 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 3.7.1 to Figure 3.7.2 Show block diagrams for TMRA01 and TMRA23.
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-flop condition are controlled by 5bytes registers.
We call control registers SFRs: Special Function Registers.
Each of the two modules (TMRA01 and TMRA23) can be operated independently. All modules operate in the same manner; hence only the operation of TMRA01 is explained here.
The contents of this chapter are as follows.

Table 3.7.1 Registers and pins for each module

	Module	TMRA01	TMRA23
External pin	Input pin for external clock	TA0IN (shared with PB0)	None
	Output pin for timer flip-flop	TA1OUT (shared with PB1)	TA3OUT (shared with PB2)
SFR (address)	Timer run register	TA01RUN (0100H)	TA23RUN (0108H)
	Timer register	TA0REG (0102H) TA1REG (0103H)	TA2REG (010AH) TA3REG (010BH)
	Timer mode register	TA01MOD (0104H)	TA23MOD (010CH)
	Timer flip-flop control register	TA1FFCR (0105H)	TA3FFCR (010DH)

3.7.1 Block diagrams

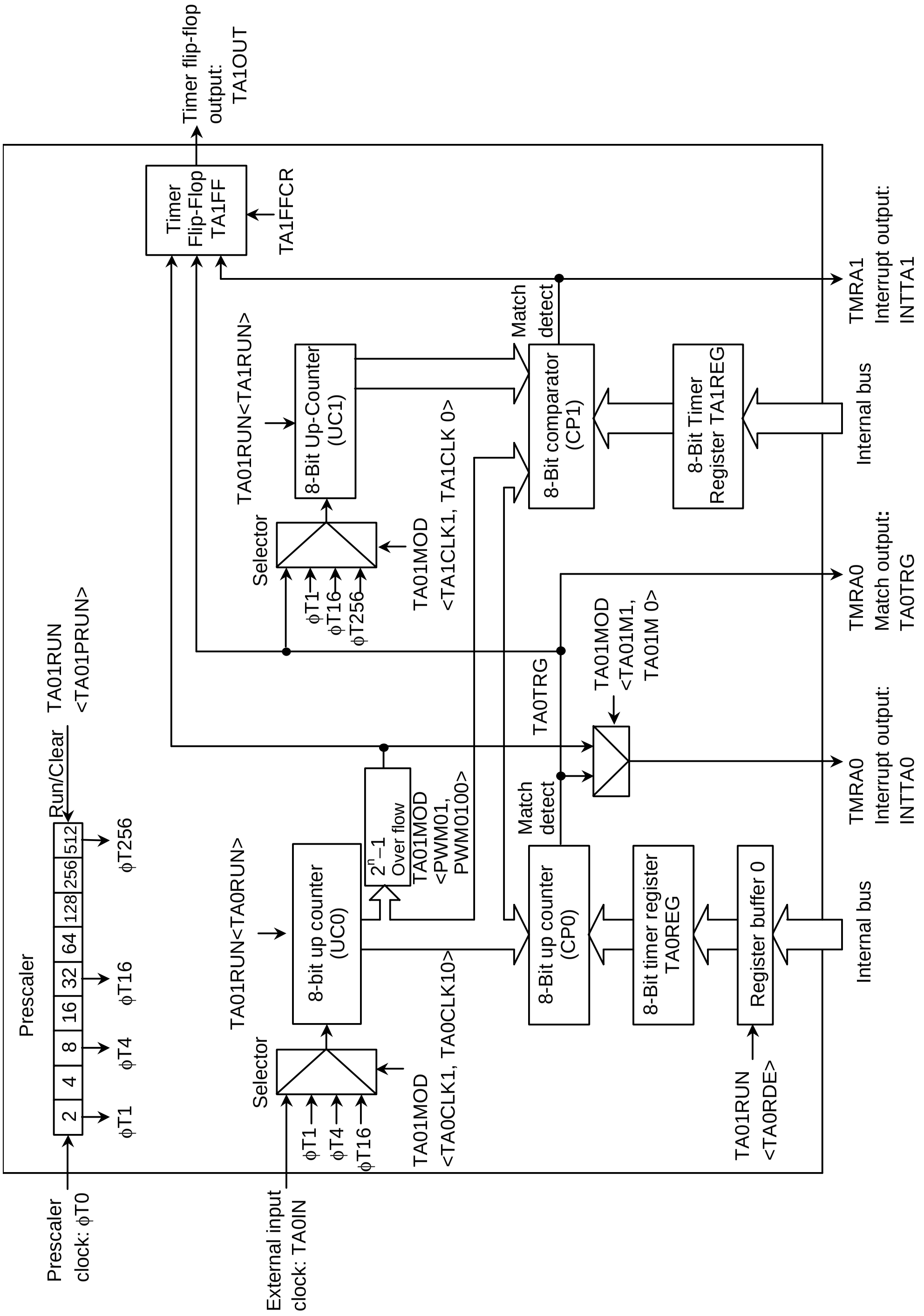


Figure 3.7.1 TMRA01 block diagram

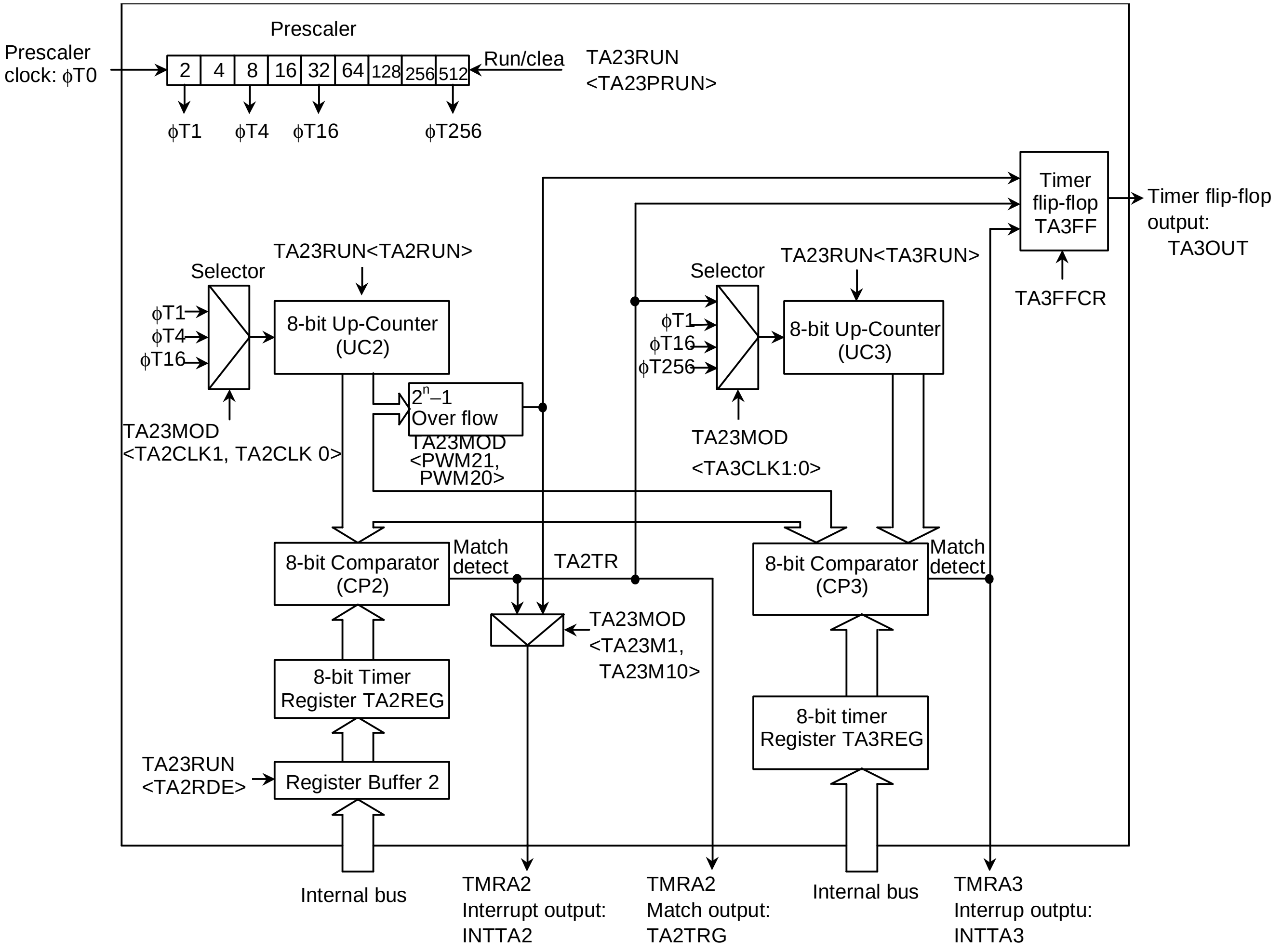


Figure 3.7.2 TMRA23 block diagram

3.7.2 Operation of each circuit

(1) Prescalers

A 9-bit prescaler generates the input clock to TMRA01.
The “φT0” as the input clock to prescaler is a clock divided by 4 which selected using the Prescaler Clock Selection Register SYSCR0<PRCK1,PRCK0>.
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 3.7 (2) shows the various prescaler output clock resolutions.

Table 3.7.2 Prescaler output clock resolution

@fc = 16 MHz, fs = 32.768 kHz

System Clock Selection <SYSCK>	Prescaler Clock Selection <PRCK1,PRCK0>	Gear Value <GEAR2~GEAR0>	Prescaler Output Clock Resolution			
			φT1	φT4	φT16	φT256
1 (fs)	00 (f_{FPH})	XXX	fs/2^3 (244 μs)	fs/2^5 (977 μs)	fs/2^7 (3.9 μs)	fs/2^{11} (62.5 μs)
0 (fc)		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)
		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)
		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)
		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 ($fc_{/16\ CLOCK}$)	XXX	fc/2^7 (8.0 μs)	fc/2^9 (32 μs)	fc/2^{11} (128 μs)	fc/2^{15} (2048 μs)

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 φT1, φT4 or φT16. The clock setting is specified by the value set in TA01MOD<TA01CLK1,TA01CLK0>.
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 φT1, φT16 or φT256, or the comparator output (the match detection signal) from TMRA0.

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.

(3) Timer registers (TA0REG and TA1REG)

These are 8-bit registers which can be used to set a time interval. When the value set in the timer register TA0REG 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 TA0REG 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 <TA0RDE> = "0" and enabled if <TA0RDE> = "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 <TA0RDE> to "0", disabling the double buffer. To use the double buffer, write data to the timer register, set <TA0RDE> to "1", and write the following data to the register buffer. Figure 3.7.3 show the configuration of TA0REG.

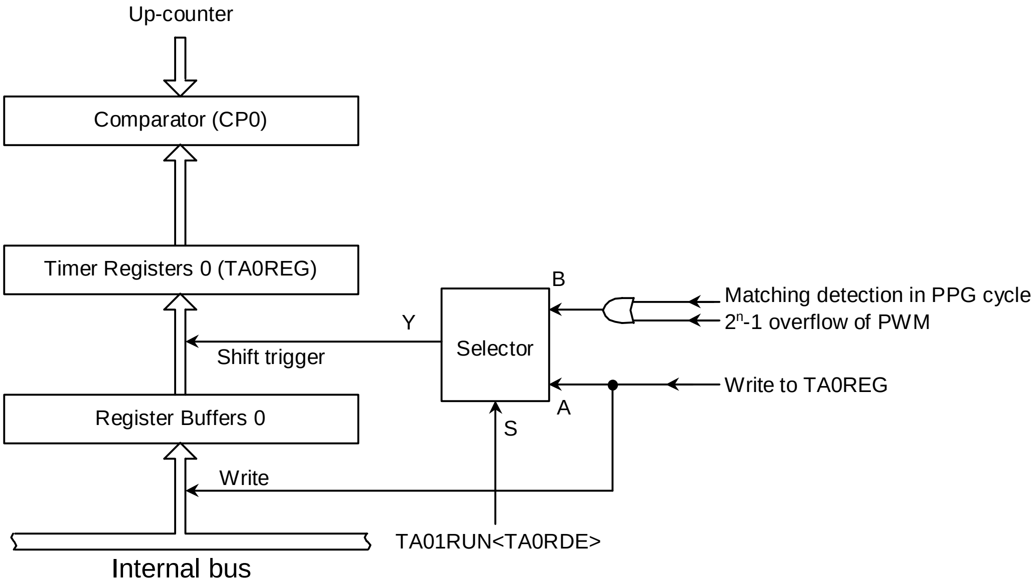


Figure 3.7.3 Configuration of TA0REG

(note): The same memory address is allocated to the timer register and the register buffer. When <TA0RDE> = 0, the same value is written to the register buffer and the timer register; when <TA0RDE> = 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

All these registers are write only and cannot be read.

(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<TAFF1IE> in the Timer Flip-Flop Control Register.

A Reset clears the value of TA1FF1 to “0”.
Writing “01” or ”10” to TA1FFCR<TAFF1C[1:0]> 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 PB1). When this pin is used as the timer output, the timer flip-flop should be set beforehand using the Port B Function Register PBCR,PBFC.

3.7.3 SFRs

TMRA01 Run Register

		7	6	5	4	3	2	1	0
TA01RUN (0100H)	Bit symbol	TA0RDE				I2TA01	TA01PRUN	TA1RUN	TA0RUN
	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	Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		

TA0REG double buffer control

0	Disable
1	Enable

Timer Run/Stop control

0	Stop & Clear
1	Run (count up)

I2TA01	: Operation in IDLE2 Mode
TA01PRUN	: Run prescaler
TA1RUN	: Run Timer 1
TA0RUN	: Run Timer 0

(note): The values of bits 4,5,6 of TA01RUN are undefined when read.

TMRA23 Run Register

		7	6	5	4	3	2	1	0
TA23RUN (0108H)	Bit symbol	TA2RDE				I2TA23	TA23PRUN	TA3RUN	TA2RUN
	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	Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		

TA2REG double buffer control

0	Disable
1	Enable

Timer Run/Stop control

0	Stop & Clear
1	Run (count up)

I2TA23	: Operation in IDLE2 Mode
TA23PRUN	: Run prescaler
TA3RUN	: Run Timer 3
TA2RUN	: Run Timer 2

(note): The values of bits 4,5,6 of TA23RUN are undefined when read.

Figure 3.7.4 TMRA Registers

TMRA01 Mode Register

TA01MOD (0104H)		7	6	5	4	3	2	1	0
	Bit symbol	TA01M1	TA01M0	PWM01	PWM00	TA1CLK1	TA1CLK0	TA0CLK1	TA0CLK0
	Read/Write	R/W							
	After Reset	0	0	0	0	0	0	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		PWM cycle 00: reserved 01: 2^6-1 10: 2^7-1 11: 2^8-1		Source clock for TMRA1 00: TA0TRG 01: φT1 10: φT16 11: φT256		Source clock for TMRA0 00: TA0IN pin 01: φT1 10: φT4 11: φT16	

TMRA0 source clock selection

00	TA0IN (external input)
01	φT1 (prescaler)
10	φT4 (prescaler)
11	φT16 (prescaler)

TMRA1 source clock selection

	TA01MOD <TA01M1~TA01M0>≠01	TA01MOD <TA01M1~TA01M 0>=01
00	Comparator output from TMRA0	Overflow output from TMRA0 (16-Bit Timer Mode)
01	φT1	
10	φT16	
11	φT 256	

PWM cycle selection

00	reserved
01	(2^6-1) × clock source
10	(2^7-1) × clock source
11	(2^8-1) × clock source

TMRA01 operation mode selection

00	Two 8-bit timers
01	16-bit timer
10	8-bit PPG
11	8-bit PWM (TMRA0) + 8-bit timer (TMRA1)

Figure 3.7.5 TMRA registers

TMRA23 Mode Register

		7	6	5	4	3	2	1	0
TA23MOD (010CH)	Bit Symbol	TA23M1	TA23M0	PWM21	PWM20	TA3CLK1	TA3CLK0	TA2CLK1	TA2CLK0
	Read/Write	R/W							
	After Reset	0	0	0	0	0	0	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		PWM cycle 00: reserved 01: 2^6-1 10: 2^7-1 11: 2^8-1		TMRA3 clock for TMRA3 00: TA2TRG 01: φT1 10: φT16 11: φT256		TMRA2 clock for TMRA2 00: reserved 01: φT1 10: φT4 11: φT16	

TMRA2 source clock selection

00	Do not set
01	φT1 (prescaler)
10	φT4 (prescaler)
11	φT16 (prescaler)

TMRA3 source clock selection

	TA23MOD <TA23M1~TA23M0>≠01	TA23MOD <TA23M1~TA23M0>=01
00	Comparator output from TMRA2	Overflow output from TMRA2 (16-Bit Timer Mode)
01	φT1	
10	φT16	
11	φT256	

PWM cycle selection

00	reserved
01	(2^6-1) × clock source
10	(2^7-1) × clock source
11	(2^8-1) × clock source

TMRA23 operation mode selection

00	Two 8-bit timers
01	16-bit timer
10	8-bit PPG
11	8-bit PWM (TMRA2) + 8-bit timer (TMRA3)

Figure 3.7.6 TMRA registers

TMRA1 Flip-Flop Control Register

TA1FFCR
(0105H)

Read-Modify-Write instructions are prohibited.

	7	6	5	4	3	2	1	0
Bit symbol					TAFF1C1	TAFF1C0	TAFF1IE	TAFF1IS
Read/Write					R/W		R/W	
After Reset					1	1	0	0
Function					00: Invert TA1FF 01: Set TA1FF 10: Clear TA1FF 11: Don’t care		TA1FF Control for inversion 0: Disable 1: Enable	TA1FF Inversion select 0: TMRA0 1: TMRA1

Inverse signal for Timer Flop-Flop 1 (TA1FF)
(Don’t care except in 8-Bit Timer Mode)

0	Inversion by TMRA0
1	Inversion by TMRA1

Inversion of TA1FF

0	Disabled
1	Enabled

Control of TA1FF

00	Inverts the value of TA1FF
01	Sets TA1FF to “1”
10	Clears TA1FF to “0”
11	Don’t care

Figure 3.7.7 TMRA registers

TMRA3 Flip-Flop Control Register

		7	6	5	4	3	2	1	0
TA3FFCR (010DH)	Bit symbol					TAFF3C1	TAFF3C0	TAFF3IE	TAFF3IS
	Read/Write					R/W		R/W	
	After Reset					1	1	0	0
Read-Modify-Write instructions are prohibited.	Function					00: Invert TA3FF 01: Set TA3FF 10: Clear TA3FF 11: Don't care		TA3FF Control for inversion 0: Disable 1: Enable	TA3FF Inversion select 0: TMRA2 1: TMRA3

Inverse signal for Timer Flip-Flop 3 (TA3FF)
(Don't care except in 8-Bit Timer Mode)

0	Inversion by TMRA2
1	Inversion by TMRA3

Inversion of TA3FF

0	Disabled
1	Enabled

Control of TA3FF

00	Inverts the value of TA3FF
01	Sets TA3FF to "1"
10	Clears TA3FF to "0"
11	Don't care

Figure 3.7.8 TMRA registers

3.7.4 Operation in each mode

(1) 8-Bit Timer Mode

Both TMRA0 and TMRA1 can be used independently as 8-bit interval timers. Setting its function or counter data for TMRA0 and TMRA1 after stop these registers.

① 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 μseconds at fc = 16 MHz, set each register as follows:

∗ Clock state
System clock: High frequency (fc)
Prescaler clock: f_{FPH}

		MSB							LSB	
		7	6	5	4	3	2	1	0	
TA01RUN	←	–	–	X	X	–	–	0	–	Stop TMRA1 and clear it to 0.
TA01MOD	←	0	0	X	X	1	0	X	X	Select 8-Bit Timer Mode and select φT1 (0.5 μs at fc = 16 MHz) as the input clock.
TA1REG	←	0	0	1	0	1	0	0	0	Set TA1REG to 20 μs ÷ φT1 = 40 = 28H
INTETA01	←	X	1	0	1	–	–	–	–	Enable INTTA1 and set it to Level 5.
TA01RUN	←	–	X	X	X	–	1	1	–	Start TMRA1 counting.

(note): X = Don’t care; “–” = No change

Select the input clock using in Table 3.7.2.

(note): The input clocks for TMRA0 and TMRA1 are different from as follows.
TMRA0: TA0IN input, φT1, φT4 or φT16
TMRA1: Match output of TMRA0,φT1, φT16, φT256

② 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.

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

		7	6	5	4	3	2	1	0	
TA01RUN	←	–	X	X	X	–	–	0	–	Stop TMRA1 and clear it to 0.
TA01MOD	←	0	0	X	X	0	1	–	–	Select 8-Bit Timer Mode and select ϕT1 (0.5 μs at fc = 16 MHz) as the input clock.
TA1REG	←	0	0	0	0	0	0	1	1	Set the timer register to 3.0 μs ÷ ϕT1 ÷ 2 = 3
TA1FFCR	←	X	X	X	X	1	0	1	1	Clear TA1FF to “0” and set it to invert on the match detects signal from TMRA1.
PBCR	←	X	–	–	–	–	–	1	–	Set PB1 to function as the TA1OUT pin.
PBFC	←	X	–	–	–	–	–	1	X	
TA01RUN	←	–	X	X	X	–	1	1	–	Start TMRA1 counting.

(note): X = Don’t care; “–” = No change

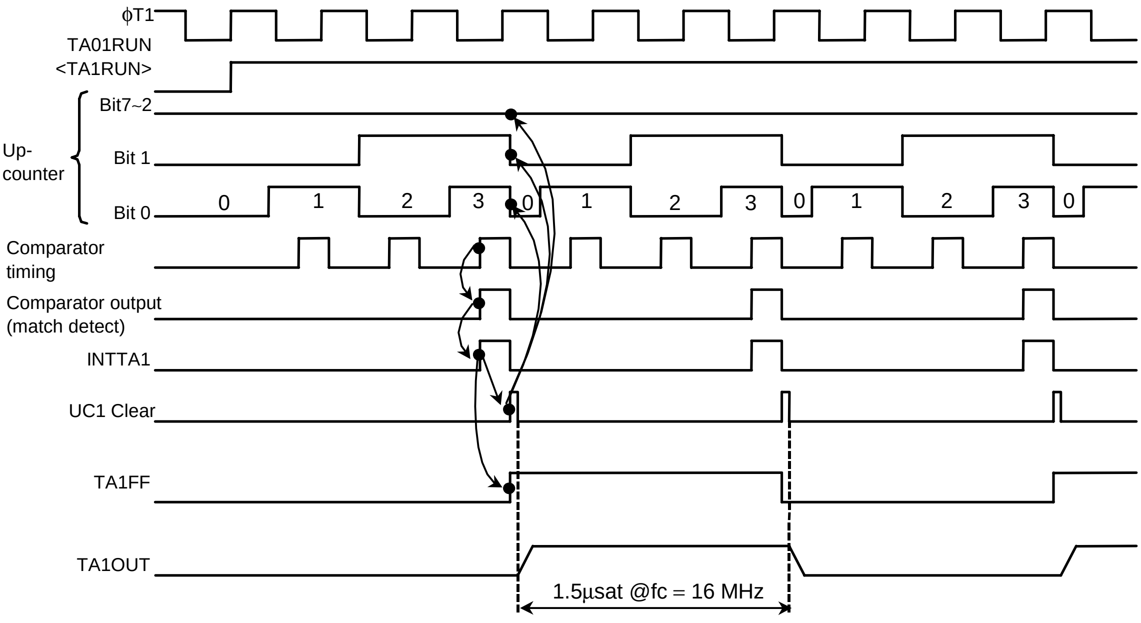


Figure 3.7.9 Square wave output timing chart (50% Duty)

③ 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.

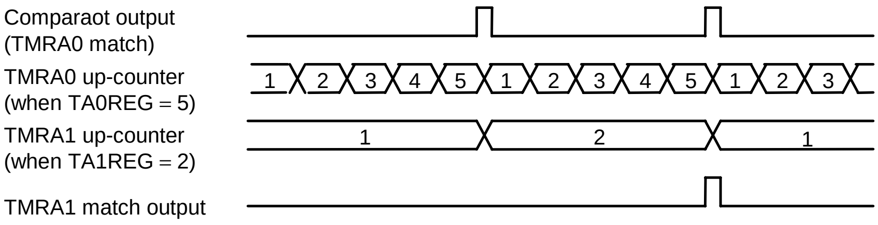


Figure 3.7.10 TMRA1 count up on signal from TMRA0

(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,TA01M0> 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<TA01CLK1,TA01CLK0>. Table 3.7.2 shows the relationship between the timer (interrupt) cycle and the input clock selection.

LSB 8-bit set to TA0REG and MSB 8-bit is for 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 seconds 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}

If ϕT16 (8.0 μs at 16 MHz) is used as the input clock for counting, set the following value in the registers: 0.5 sec / 8.0 μsec = 62500 = F424H; 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

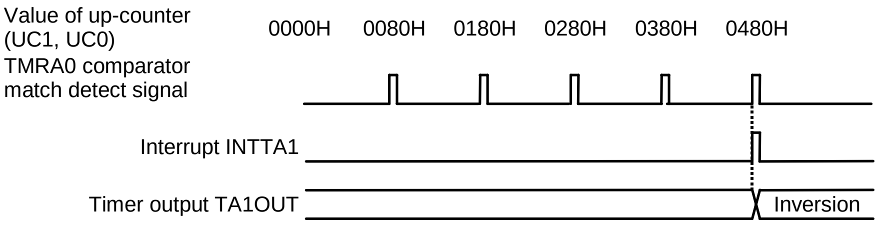


Figure 3.7.11 Timer output by 16-Bit Timer Mode

(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 (concurrent with P71).

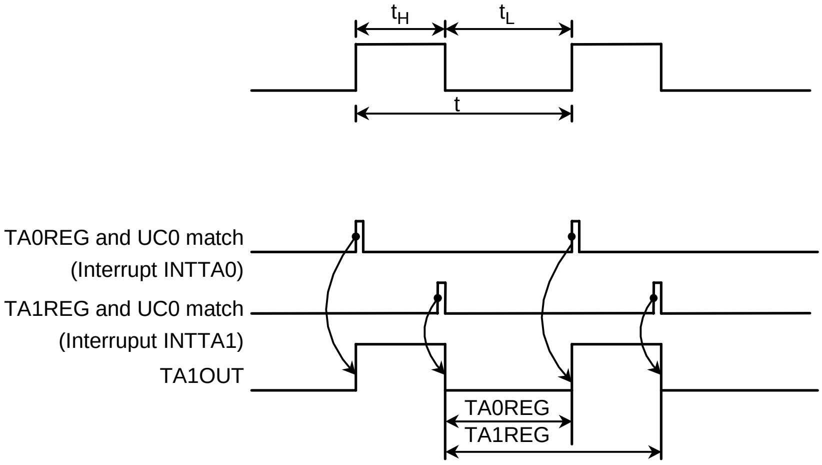


Figure 3.7.12 8 bit PPG output waveforms

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 TA0REG 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 3.7.13 shows a block diagram representing this mode.

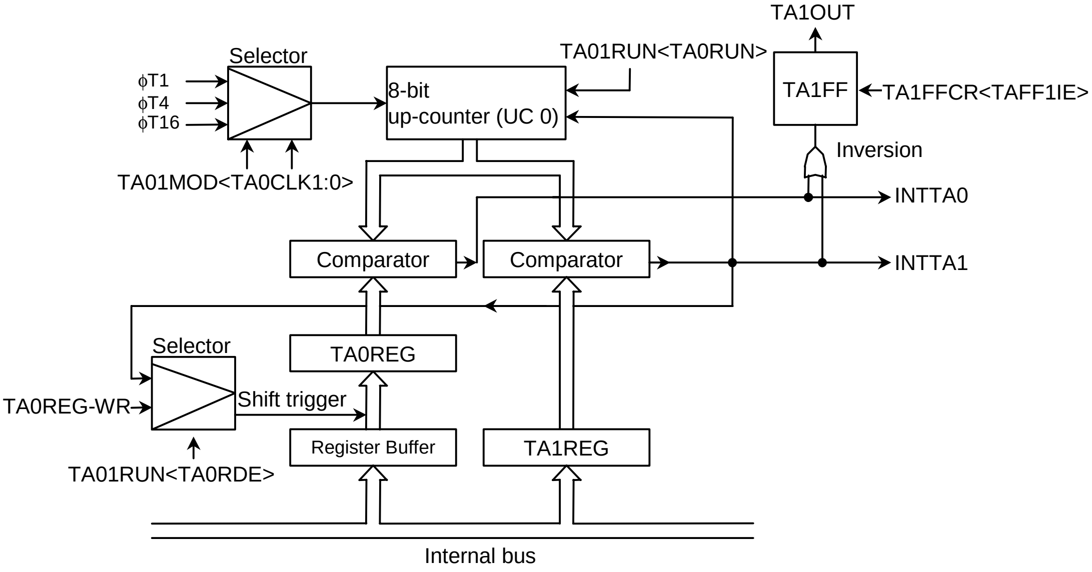


Figure 3.7.13 Block diagram of 8-Bit PPG Output Mode

If the TA0REG double buffer is enabled in this mode, the value of the register buffer will be shifted into TA0REG each time TA1REG matches UC0.
Use of the double buffer facilitates the handling of low-duty waves (when duty is varied).

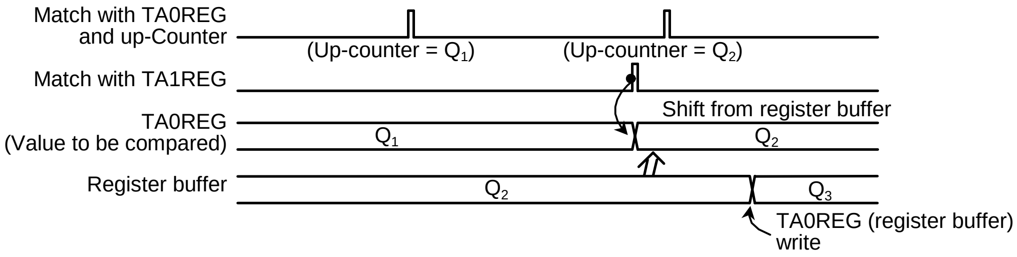
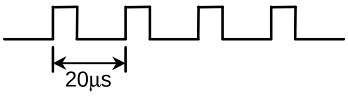


Figure 3.7.14 Operation of register buffer

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



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

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

To obtain a frequency of 50 kHz, the pulse cycle t should be: t = 1/50 kHz = 20 μ sec

ϕT1 = 0.5 μsec (at 16 MHz);

20 μsec / 0.5 μsec = 40

Therefore set TA1REG to 40 (28H)

The duty is to be set to 1/4: t × 1/4 = 20 μsec × 1/4 = 5 μsec

5 μsec / 0.5 μsec = 10

Therefore, set TA0REG = 10 = 0AH.

		7	6	5	4	3	2	1	0	
TA01RUN	←	0	X	X	X	–	0	0	0	Stop TMRA0 and TMRA01 and clear it to “0”.
TA01MOD	←	1	0	X	X	X	X	0	1	Set the 8-bit PPG mode, and select ϕT1 as input clock.
TA0REG	←	0	0	0	0	1	0	1	0	Write 0AH
TA1REG	←	0	0	1	0	1	0	0	0	Write 28H
TA1FFCR	←	X	X	X	X	0	1	1	X	Set TA1FF, enabling both inversion and the double buffer.
										Writing “10” provides negative logic pulse.
PBCR	←	X	–	–	–	–	–	1	–	Set PB1 as the TA1OUT pin.
PBFC	←	X	–	–	–	–	–	1	X	
TA01RUN	←	1	X	X	X	–	1	1	1	Start TMRA0 and TMRA01 counting.

(note): 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 to PWM00>). 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 TA0REG < value set for 2^n-1 counter overflow
Value set in TA0REG $\neq$ 0

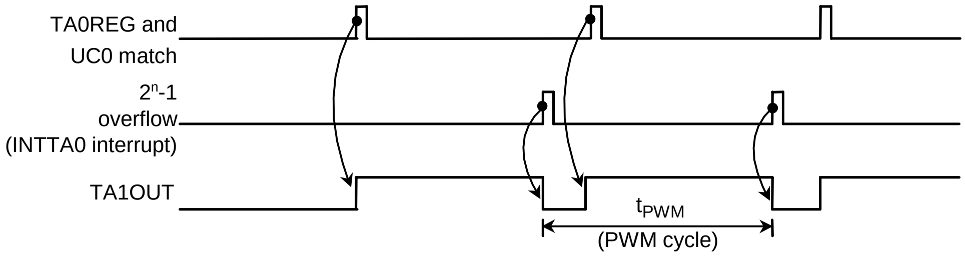


Figure 3.7.15 8-bit PWM waveforms

Figure 3.7.16 shows a block diagram representing this mode.

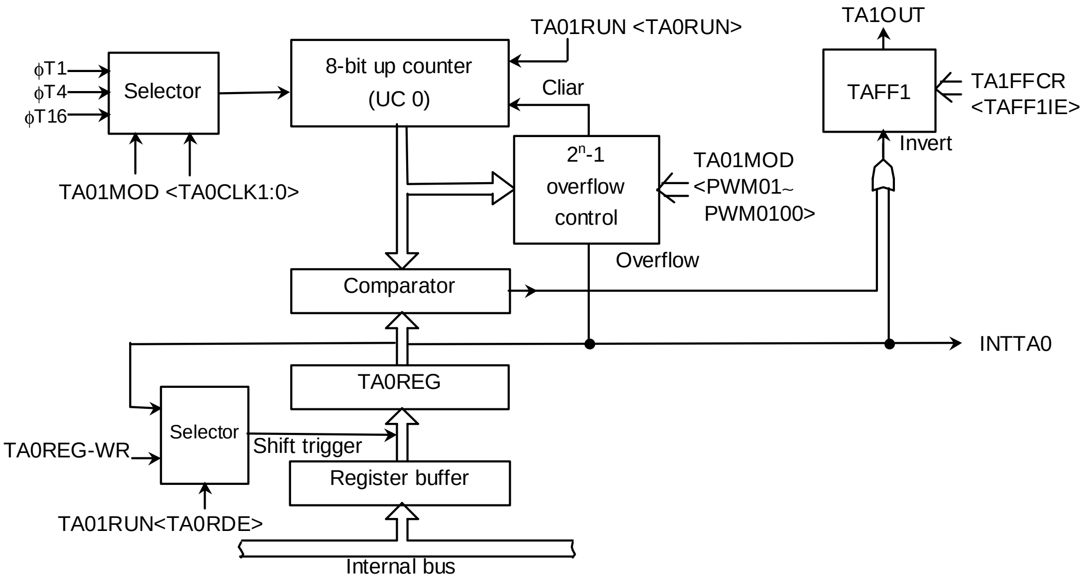


Figure 3.7.16 Block diagram of 8-Bit PWM Mode

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.

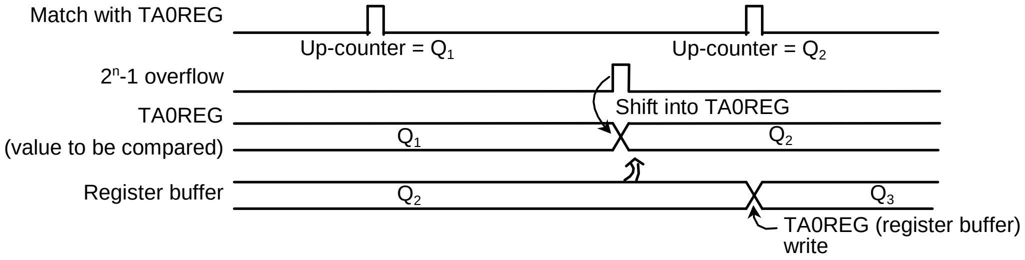
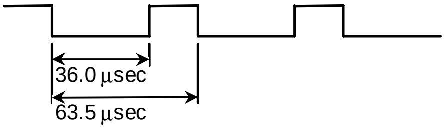


Figure 3.7.17 Register buffer operation

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-μs PWM cycle by setting φT1 to 0.5 μsec (at fc = 16 MHz):

63.5 μsec / 0.5 μsec = $127 = 2^n - 1$

Therefore n should be set to 7.

Since the low-level period is 36.0 μsec when φT1 = 0.5 μsec,

set the following value for TA0REG:

36.0 μsec / 0.5 μsec = 72 = 48H

		MSB							LSB	
		7	6	5	4	3	2	1	0	
TA01RUN	←	–	X	X	X	–	–	–	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 φT1 as the input clock.
TA0REG	←	0	1	0	0	1	0	0	0	Write 48H.
TA1FFCR	←	X	X	X	X	1	0	1	X	Clear TA1FF to 0, enable the inversion and double buffer.
PBCR	←	X	–	–	–	–	–	1	–	Set PB1 and the TA1OUT pin.
PBFC	←	X	–	–	–	–	–	1	X	
TA01RUN	←	1	X	X	X	–	1	-	1	Start TMRA0 counting.

(note): X = Don’t care; “–” = No change

Table 3.7.3 PWM cycle

@fc = 16 MHz, fs = 32.768 kHz

Select System Clock <SYSCK>	Select Prescaler Clock <PRCK1~PRCK0>	Gear Value <GEAR2~GEAR0>	PWM cycle								
			$2^6 – 1$			$2^7 – 1$			$2^8 – 1$		
			φT1	φT4	φT16	φT1	φT4	φT16	φT1	φT4	φT16
1 (fs)	00 (f$_{FPH}$)	XXX	15.4 ms	61.5 ms	246 ms	31.0 ms	124 ms	496 ms	62.3 ms	249 ms	996 ms
0 (fc)		000 (fc)	31.5 μs	126 μs	504 μs	63.5 μs	254 m	1016 μs	127.5 μs	510 μs	2040 μs
		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
		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

XXX: Don't care

(5) Settings for each mode

Table 3.7.4 shows he SFR settings for each mode.

Table 3.7.4 Timer mode setting registers

Register name	TA01MOD				TA1FFCR
<Bit Symbol>	<TA01M1:TA01M 0>	<PWM01:00>	<TA1CLK1:0>	<TA0CLK1:0>	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)	–
8-bit PWM × 1 channel	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

(note): "–" = Don't care

3.8 External memory extension function (MMU)

This is MMU function which can expand program / data area to 106M byte by having 4 local area.

Address pins to external memory are 2 extended address bus pins (EA24,EA25) and 8 extended chip select pins (/CS2A to /CS2E) in addition to 24 address bus pins (A0～A23) which are common specification of TLCS-900 and 4 chip select pins (/CS0 ～ /CS3) output from CS/WAIT controller.

The feature and the recommendation setting method of two types are shown below.

In addition, AH in the table is the value which number address 23-16 displayed as hex .

Purpose	Item	(A): For standard extended memory	(B): For many pieces extended memory
Program-ROM	Maximum memory size	COMMON2 2MB+14MB (16MB×1pcs)	
	Used local area, BANK number	LOCAL2(AH=C0-DF: 2MB ×7BANK)	
	Setting CS/WAIT	Set up AH=C0-FF to CS2	Set up AH=80-FF to CS2
	Used /CS pin	/CS2	/CS2A
Data-ROM	Maximum memory size	64MB(64MB×1pcs)	64MB(16MB×4pcs)
	Used local area, BANK number	LOCAL3(AH=80-BF: 4MB × 16BANK)	LOCAL3(AH=80-BF:4MB× 24BANK)
	Setting CS/WAIT	Set up AH=80-BF to CS3	Set up AH=80-FF to CS2
	Used /CS pins	/CS3,EA24,EA25	/CS2B,/CS2C,/CS2D,/CS2E
Option Program-ROM	Maximum memory size	COMMON1 2MB+14MB(16MB×1pcs)	
	Used local area, BANK number	LOCAL1(AH=40-5F: 2MB ×7BANK))	
	Setting CS/WAIT	Set up AH=40-7F to CS1	
	Used /CS pin	/CS1	
Data-RAM	Maximum memory size	COMMON0 1MB+7MB(8MB×1pcs)	
	Used local area, BANK number	LOCAL0(AH=10-1F: 1MB ×7BANK))	
	Setting CS/WAIT	Set up AH=00-3F to CS0	Set up AH=00-1F to CS3
	Used /CS pin	/CS0	/CS3
Extended memory -1	Maximum memory size	COMMON0 α	2MB(2MB×1pcs)
	Used local area, BANK number	Overlapped Data-RAM	None
	Setting CS/WAIT	Set up AH=00-3F to CS0	Set up AH=20-3F to CS0
	Used /CS pin	/CS0	/CS0
Total Memory Size		16M+64M+16M+8M=104Mbyte	16M+(16M+16M+16M+16M) +16M+8M+2M=106Mbyte

3.8.1 Recommendable memory map

The recommendation logic address memory map at the time of varieties extension memory correspondence is shown in Figure 3.8.1.1. And, a physical-address map is shown in Figure 3.8.1.2.

However, when memory area is less than 16M bytes and is not expanded, please refer to section of CS/WAIT controller. Setting of register in MMU is not necessary.

Since it is being fixed, the address of a local-area cannot be changed.

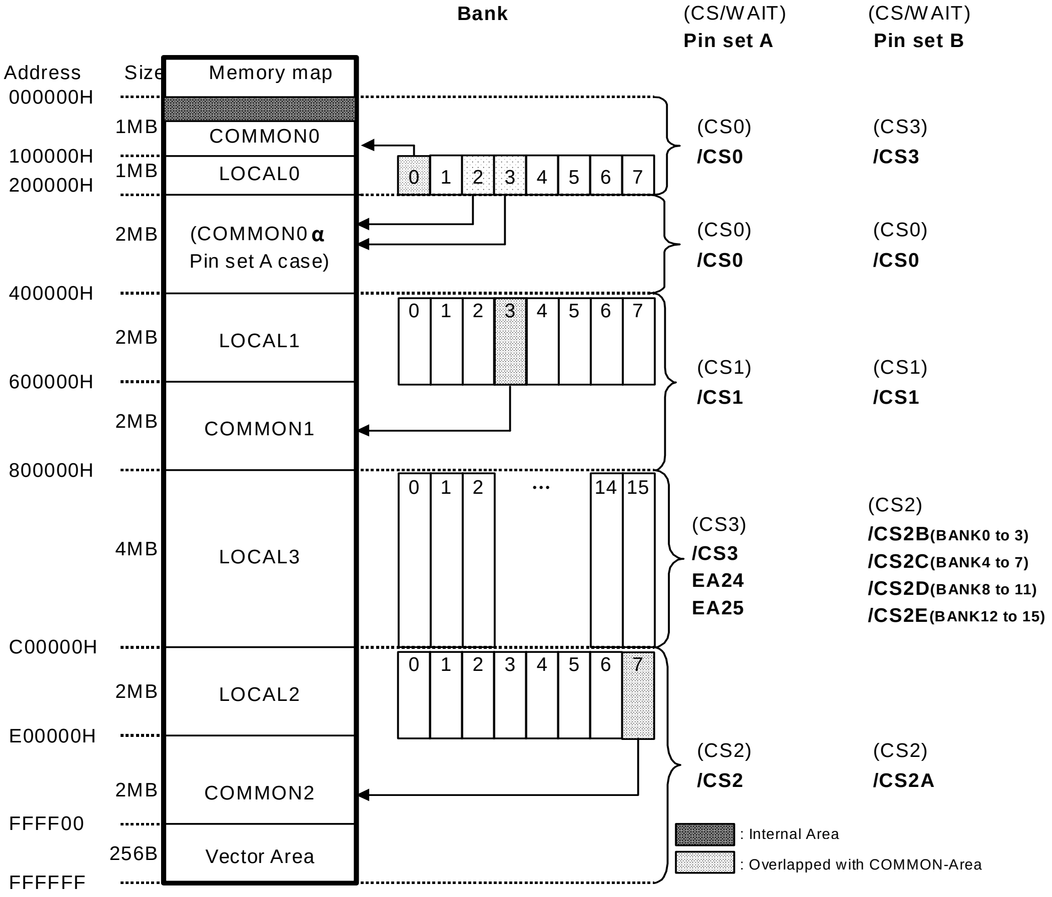


Figure 3.8.1.1 Logical address map

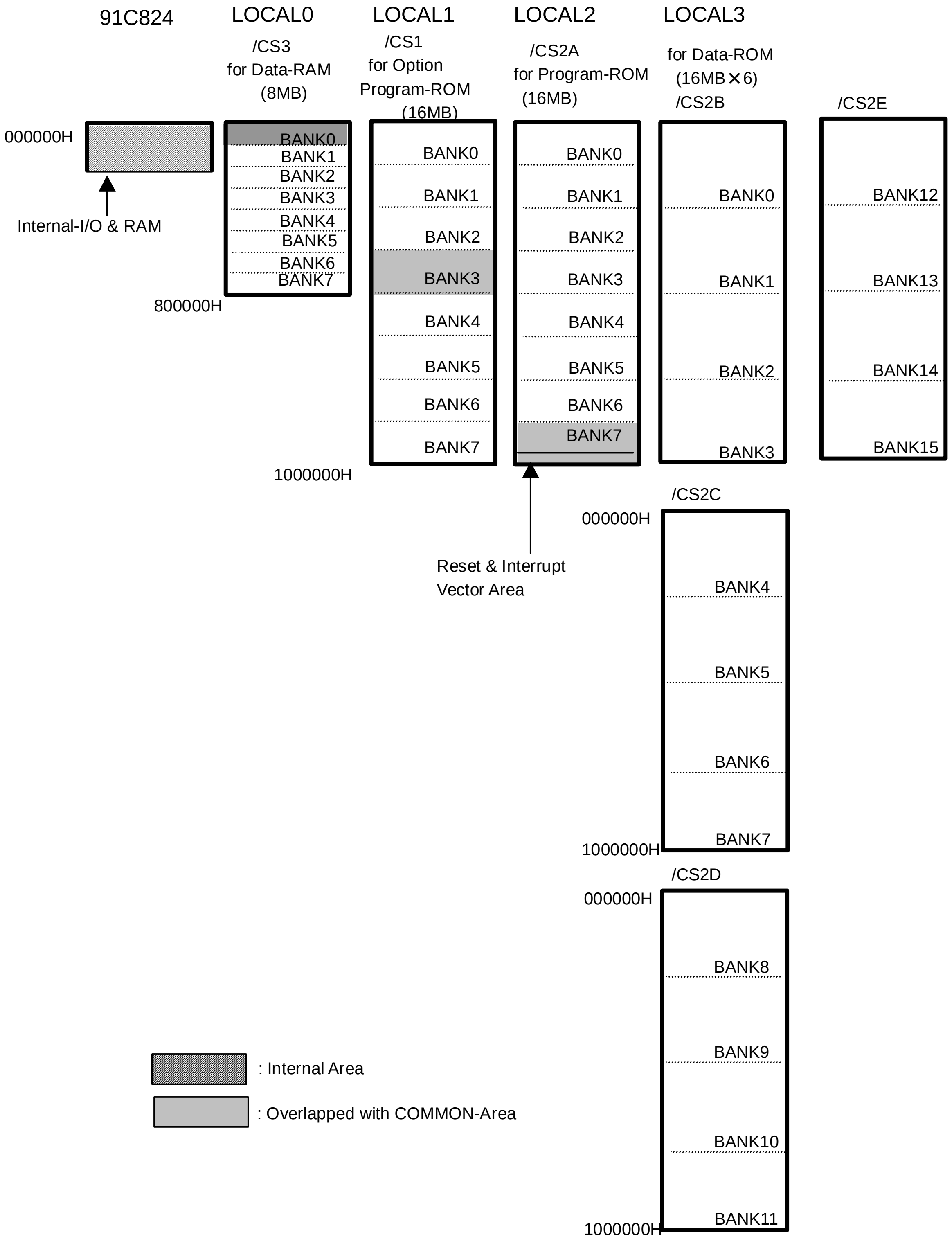
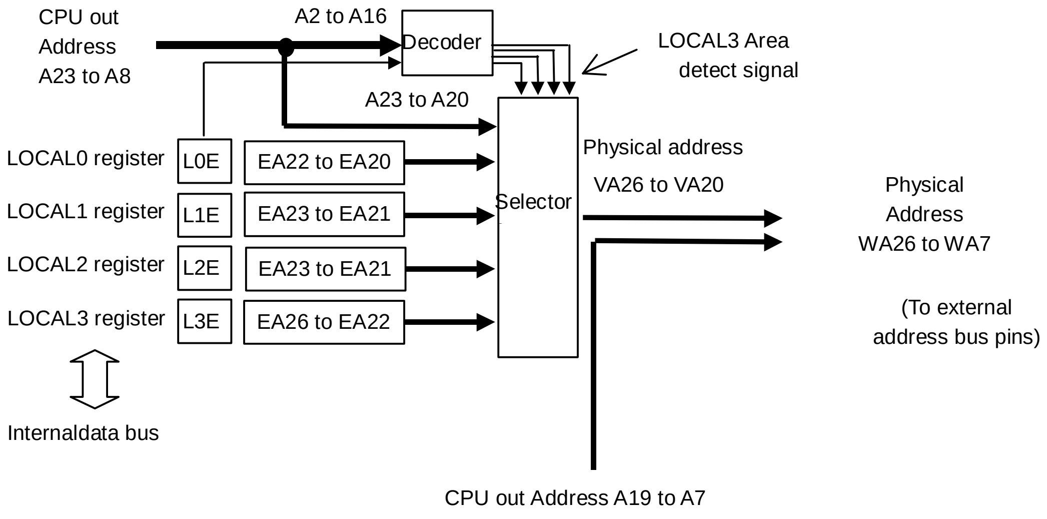


Figure 3.8.1.2 Physical address map (Type B)

3.8.2 Block diagram



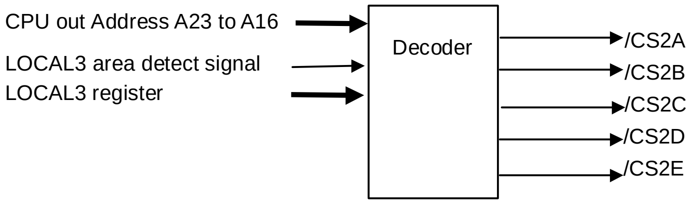


Figure 3.8.2.1 Block diagram of MMU

3.8.3 Control registers

LOCAL0 register

LOCAL0
(0350H)

	7	6	5	4	3	2	1	0
bit Symbol	L0E					L0EA22	L0EA21	L0EA20
Read/Write	R/W					R/W		
After reset	0					0	0	0
Function	Use BANK for LOCAL0 0: not use 1: use					Setting BANK number for LOCAL0		

LOCAL1 register

LOCAL1
(0351H)

	7	6	5	4	3	2	1	0
bit Symbol	L1E					L1EA23	L1EA22	L1EA21
Read/Write	R/W					R/W		
After reset	0					0	0	0
Function	Use BANK for LOCAL1 0: not use 1: use					Setting BANK number for LOCAL1		

LOCAL2 register

LOCAL2
(0352H)

	7	6	5	4	3	2	1	0
bit Symbol	L2E					L2EA23	L2EA22	L2EA21
Read/Write	R/W					R/W		
After reset	0					0	0	0
Function	Use BANK for LOCAL2 0: disable 1: enable					Setting BANK number for LOCAL2		

LOCAL3 register

LOCAL3
(0353H)

	7	6	5	4	3	2	1	0
bit Symbol	L3E			*L3EA26	L3EA25	L3EA24	L3EA23	L3EA22
Read/Write	R/W			R/W	R/W	R/W	R/W	R/W
After reset	0			0	0	0	0	0
Function	Use BANK for LOCAL3 0: disable 1: enable			01000 to 01011 /CS2D 01100 to 01111 : /CS2E 00000 to 00011 /CS2B 00100 to 00111 /CS2C 10000 to 11111 : Set prohibition				

(Note) In case of this TMP91C824, because most upper address bit of physical address is EA25, most upper address bit of BANK register is meaningless. 4-bits of upper 5-bits address means 16-BANKs.

3.8.4 Operational description

Set up bank value and bank use in bank setting-register of each local area of LOCAL register in common area. Moreover, in that case, a combination pin is set up and mapping is simultaneously set up by the CS/WAIT controller. When CPU outputs logical address of the local area, MMU outputs physical address to the outside address bus pin according to value of bank setting-register. Access of external memory becomes possible therefore.

Please do not use as bank that overlaps with another bank since this common area overlaps with either of eight banks of local area on the physical map.

Example program is as next page follows

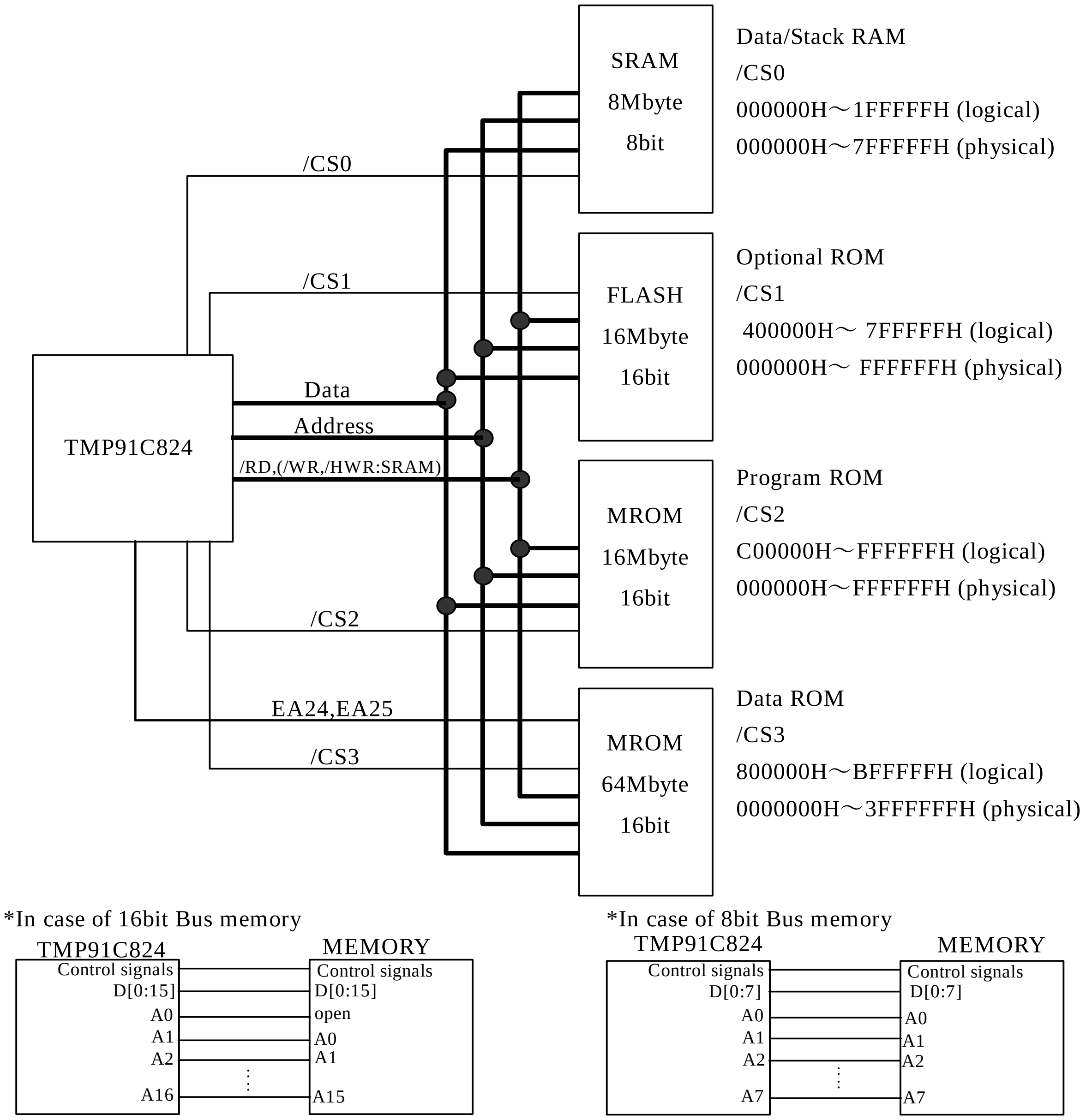


Figure 3.8.4.1 H/W Setting Example

At Figure 3.8.4.1, it shows example of connection TMP91C824 and some memories: Program ROM:MROM,16Mbyte, Data ROM:MROM,64Mbyte, Data RAM:SRAM,8Mbyte, 8bit bus, Option ROM:Flash,16Mbyte.

In case of 16bit bus memory connection, it need to shift 1bit address bus from TMP91C824 and 8bit bus case, direct connection address bus from TMP91C824.

In that figure, Logical address and physical address are shown. And each memory allot each chip select signal, RAM:/CS0, FLASH_ROM:/CS1, Program MROM:/CS2, Data MROM:/CS3. In case of this example, as Data MROM is 64Mbyte, this MROM connect to EA24 and EA25.

Initial condition after reset, because TMP91C824 access from CS2 area, CS2 area allot to Program ROM. It can set free setting except Program ROM.

```
;Initial Setting
;CS0
        LD      (MSAR0),00H     ; Logical address area: 000000H～1FFFFFH
        LD      (MAMR0),FFH     ; Logical address size: 2Mbyte
        LD      (B0CS),89H      ; Condition: 8bit,1wait (8MB, SRAM)
;CS1
        LD      (MSAR1),40H     ; Logical address area: 400000H～7FFFFFH
        LD      (MAMR1),7FH     ; Logical address size: 4Mbyte
        LD      (B1CS),80H      ; Condition: 16bit,2wait (16Mbyte, Flash ROM)
;CS2
        LD      (MSAR2),C0H     ; Logical address area: C00000H～FFFFFFH
        LD      (MAMR2),7FH     ; Logical address size: 4Mbyte
        LD      (B2CS),C3H      ; Condition: 16bit,0wait (16Mbyte, MROM)
;CS3
        LD      (MSAR3),80H     ; Logical address area: 800000H～BFFFFFH
        LD      (MAMR3),FFH     ; Logical address size: 4Mbyte
        LD      (B3CS),85H      ; Condition: 16bit,3wait (64Mbyte, MROM)
;CSX
        LD      (BEXCS),00H     ; Other : 16bit,2wait (don't care)
;Port
        LD      (P6FC),3FH      ; /CS0～/CS3,EA24,EA25 :port6 setting
～
```

Figure 3.8.4.2 Bank Operation S/W Example1

Secondly, it shows example of initial setting at Figure 3.8.4.2.

Because /CS0 connect to RAM: 8bit bus, 8Mbyte, it need to set 8bit bus. At this example, it set 1-wait setting. In the same way /CS1 set to 16bit bus and 2-wait, /CS2 set 16bit bus and 0-wait, /CS3 set 16bit bus and 3-wait.

By CS/WAIT controller, each chip selection signal's memory size, don't set actual connect memory size, need to set that logical address size: fitting to each local area. Actual physical address is set by each area's BANK register setting.

CSEX setting of CS/WAIT controller is except above CS0～CS3's setting. This program example isn't used CSEX setting.

Finally pin condition is set. PORT60～65 set to /CS0,1,2,3,EA24,EA25.

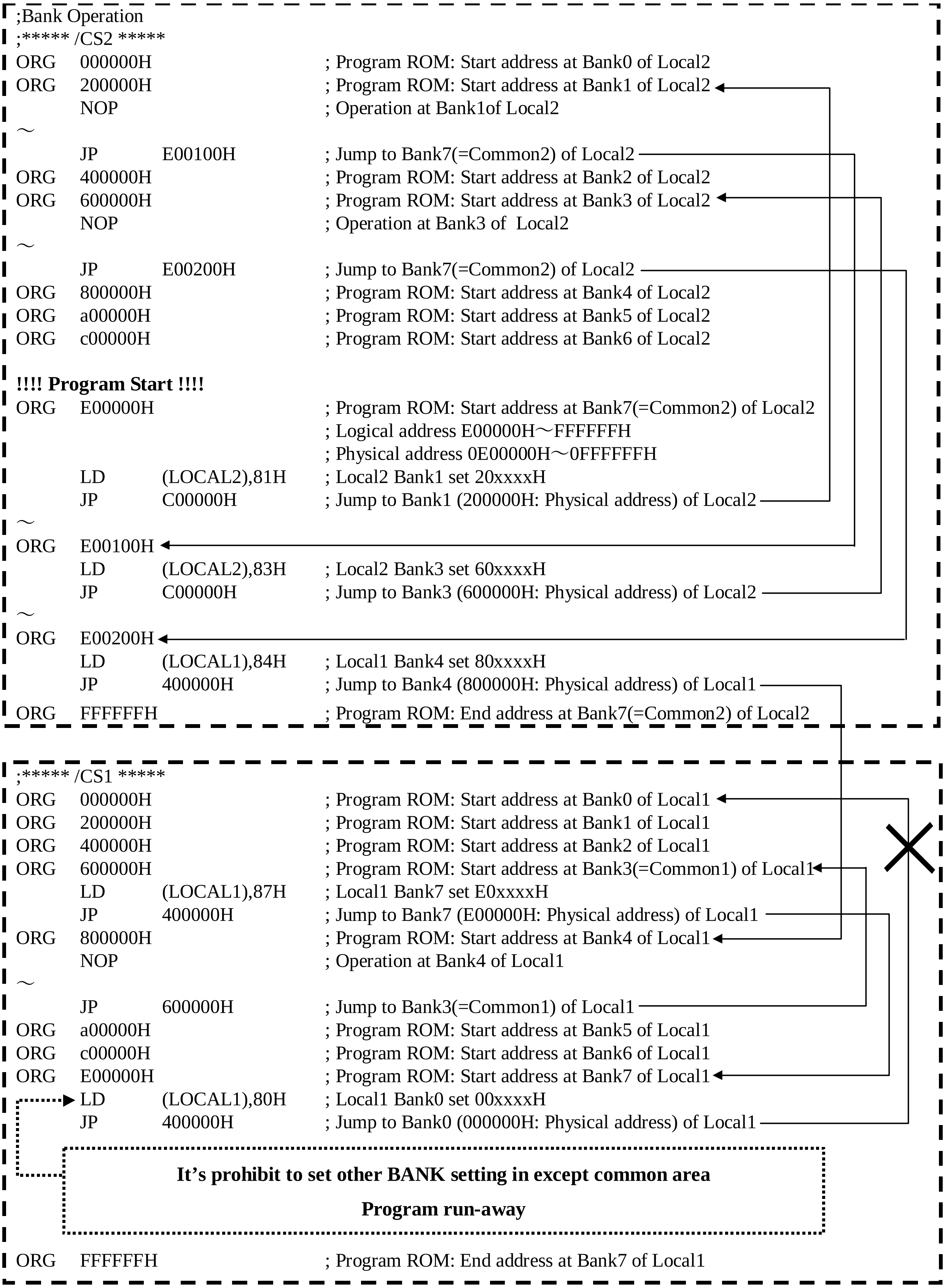
```
;Bank Operation
;***** /CS2 *****
ORG    000000H                      ; Program ROM: Start address at Bank0 of Local2
ORG    200000H                      ; Program ROM: Start address at Bank1 of Local2
ORG    400000H                      ; Program ROM: Start address at Bank2 of Local2
ORG    600000H                      ; Program ROM: Start address at Bank3 of Local2
ORG    800000H                      ; Program ROM: Start address at Bank4 of Local2
ORG    a00000H                      ; Program ROM: Start address at Bank5 of Local2
ORG    c00000H                      ; Program ROM: Start address at Bank6 of Local2

ORG    E00000H                      ; Program ROM: Start address at Bank7(=Common2) of Local2
                                    ; Logical address E00000H～FFFFFFH
                                    ; Physical address 0E00000H～0FFFFFFH
       LD       (LOCAL3),85H        ; Local3 Bank5 set 14xxxxH
       LDW      HL,(800000H)        ; Load data (5555H) form  Bank5 (140000H: Physical address)
                                                                   of Local3 (/CS3)
       LD       (LOCAL3),88H        ; Local3 Bank8 set 20xxxxH
       LDW      BC,(800000H)        ; Load data (AAAAH) form  Bank8 (200000H: Physical address)
                                                                   of Local3 (/CS3)
～
ORG    FFFFFFH                      ; Program ROM: End address at Bank7(=Common2) of Local2
```

```
;***** /CS3 *****
ORG    0000000H                     ; Data ROM: Start address at Bank0 of Local3
ORG    0400000H                     ; Data ROM: Start address at Bank1 of Local3
ORG    0800000H                     ; Data ROM: Start address at Bank2 of Local3
ORG    0C00000H                     ; Data ROM: Start address at Bank3 of Local3
ORG    1000000H                     ; Data ROM: Start address at Bank4 of Local3
ORG    1400000H                     ; Data ROM: Start address at Bank5 of Local3
       dw       5555H
～
ORG    1800000H                     ; Data ROM: Start address at Bank6 of Local3
ORG    1C00000H                     ; Data ROM: Start address at Bank7 of Local3
ORG    2000000H                     ; Data ROM: Start address at Bank8 of Local3
       dw       AAAAH
～
ORG    2400000H                     ; Data ROM: Start address at Bank9 of Local3
ORG    2800000H                     ; Data ROM: Start address at Bank10 of Local3
ORG    2C00000H                     ; Data ROM: Start address at Bank11 of Local3
ORG    3000000H                     ; Data ROM: Start address at Bank12 of Local3
ORG    3400000H                     ; Data ROM: Start address at Bank13 of Local3
ORG    3800000H                     ; Data ROM: Start address at Bank14 of Local3
ORG    3C00000H                     ; Data ROM: Start address at Bank15 of Local3
ORG    3FFFFFFH                     ; Data ROM: End address at Bank15 of Local3
```

Figure 3.8.4.3 Bank Operation S/W Example2

Here shows example of data access between one BANK and other BANK. Figure 3.8.4.3 is one software example. A dot line square area shows one memory and each dot line square shows /CS2's Program ROM and /CS3's Data ROM. Program start from E00000H address, firstly, write to BANK register of LOCAL3 area upper 5-bit address of access point.

In case of this TMP91C824, because most upper address bit of physical address is EA25, most upper address bit of BANK register is meaningless. 4-bits of upper 5-bits address means 16-BANKs. After setting BANK5, accessing 800000～BFFFFFH address: logical local3 address, actually access to physical 1400000～1700000H address.



```
;Bank Operation
;***** /CS2 *****
ORG    000000H                       ; Program ROM: Start address at Bank0 of Local2
ORG    200000H                       ; Program ROM: Start address at Bank1 of Local2
       NOP                           ; Operation at Bank1of Local2
～
       JP          E00100H           ; Jump to Bank7(=Common2) of Local2
ORG    400000H                       ; Program ROM: Start address at Bank2 of Local2
ORG    600000H                       ; Program ROM: Start address at Bank3 of Local2
       NOP                           ; Operation at Bank3 of  Local2
～
       JP          E00200H           ; Jump to Bank7(=Common2) of Local2
ORG    800000H                       ; Program ROM: Start address at Bank4 of Local2
ORG    a00000H                       ; Program ROM: Start address at Bank5 of Local2
ORG    c00000H                       ; Program ROM: Start address at Bank6 of Local2

!!!! Program Start !!!!
ORG    E00000H                       ; Program ROM: Start address at Bank7(=Common2) of Local2
                                     ; Logical address E00000H～FFFFFFH
                                     ; Physical address 0E00000H～0FFFFFFH
       LD          (LOCAL2),81H      ; Local2 Bank1 set 20xxxxH
       JP          C00000H           ; Jump to Bank1 (200000H: Physical address) of Local2
～
ORG    E00100H
       LD          (LOCAL2),83H      ; Local2 Bank3 set 60xxxxH
       JP          C00000H           ; Jump to Bank3 (600000H: Physical address) of Local2
～
ORG    E00200H
       LD          (LOCAL1),84H      ; Local1 Bank4 set 80xxxxH
       JP          400000H           ; Jump to Bank4 (800000H: Physical address) of Local1
ORG    FFFFFFH                       ; Program ROM: End address at Bank7(=Common2) of Local2

;***** /CS1 *****
ORG    000000H                       ; Program ROM: Start address at Bank0 of Local1
ORG    200000H                       ; Program ROM: Start address at Bank1 of Local1
ORG    400000H                       ; Program ROM: Start address at Bank2 of Local1
ORG    600000H                       ; Program ROM: Start address at Bank3(=Common1) of Local1
       LD          (LOCAL1),87H      ; Local1 Bank7 set E0xxxxH
       JP          400000H           ; Jump to Bank7 (E00000H: Physical address) of Local1
ORG    800000H                       ; Program ROM: Start address at Bank4 of Local1
       NOP                           ; Operation at Bank4 of Local1
～
       JP          600000H           ; Jump to Bank3(=Common1) of Local1
ORG    a00000H                       ; Program ROM: Start address at Bank5 of Local1
ORG    c00000H                       ; Program ROM: Start address at Bank6 of Local1
ORG    E00000H                       ; Program ROM: Start address at Bank7 of Local1
       LD          (LOCAL1),80H      ; Local1 Bank0 set 00xxxxH
       JP          400000H           ; Jump to Bank0 (000000H: Physical address) of Local1
ORG    FFFFFFH                       ; Program ROM: End address at Bank7 of Local1
```

It's prohibit to set other BANK setting in except common area

Program run-away

Figure 3.8.4.4 Bank Operation S/W Exapmle3

At Figure 3.8.4.4, it shows example of program jump.

In the same way with before example, two dot line squares show each /CS2's program ROM and /CS1's option ROM. Program start from E00000H common address, firstly, write to BANK register of LOCAL2 area upper 3-bit address of jumping point.

After setting BANK1, jumping C00000～DFFFFFH address: logical local2 address, actually jump to physical 2000000～3FFFFFH address. When return to common area, it can only jump to E00000～FFFFFFH without writing to BANK register of LOCAL2 area.

By a way of setting of BANK register, the setting that BANK address and common address conflict with is possible. When two kinds or more logical addresses to show common area exist, management of BANK is confused. We recommends not to use The BANK setting, BANK address and common address conflict with.

When it jump to one memory from other different memory, it can set same as the last time setting. It needs to write to BANK register of LOCAL1 area upper 3-bit address of jumping point. After setting BANK4, jumping 400000～5FFFFFH address: logical local1 address, actually jump to physical 8000000～9FFFFFH address.

It is a mark paid attention to here, it needs to go by way of common area by all means when moves from a bank to a bank. In other words, it must write to BANK register only in common area and It is prohibit to write the BANK register in BANK area. If it modify the BANK register's data in BANK area, program run-away.

3.9 Serial Channels

TMP91C824 includes 2 serial I/O channels. 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.
• UART Mode	Mode 1:	7-bit data
	Mode 2:	8-bit data
	Mode 3:	9-bit data

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

Figure 3.9 2, 3 are block diagrams for each channel.

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.

Table 3.9.1 Differences between Channels 0 to 1

	Channel 0	Channel 1
Pin Name	TXD0 (PC0) RXD0 (PC1) $\overline{\text{CTS0}}$ /SCLK0 (PC2)	TXD1 (PC3) RXD1 (PC4) $\overline{\text{CTS1}}$ /SCLK1 (PC5)
IrDA Mode	Yes	No

This chapter contains the following sections:

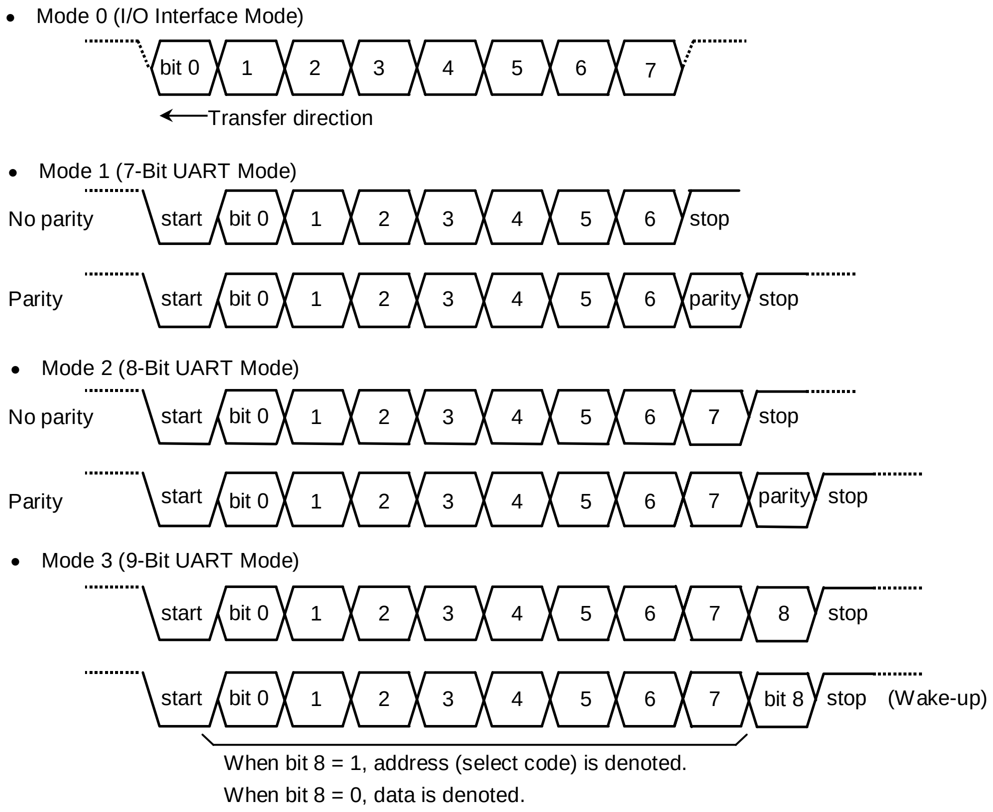

• Mode 0 (I/O Interface Mode)
bit 0 1 2 3 4 5 6 7
Transfer direction
• Mode 1 (7-Bit UART Mode)
No parity
start bit 0 1 2 3 4 5 6 stop
Parity
start bit 0 1 2 3 4 5 6 parity stop
• Mode 2 (8-Bit UART Mode)
No parity
start bit 0 1 2 3 4 5 6 7 stop
Parity
start bit 0 1 2 3 4 5 6 7 parity stop
• Mode 3 (9-Bit UART Mode)
start bit 0 1 2 3 4 5 6 7 8 stop
start bit 0 1 2 3 4 5 6 7 bit 8 stop (Wake-up)
When bit 8 = 1, address (select code) is denoted.
When bit 8 = 0, data is denoted.

Figure 3.9.1 Data formats

3.9.1 Block diagrams

Figure 3.9.2 is a block diagram representing Serial Channel 0.

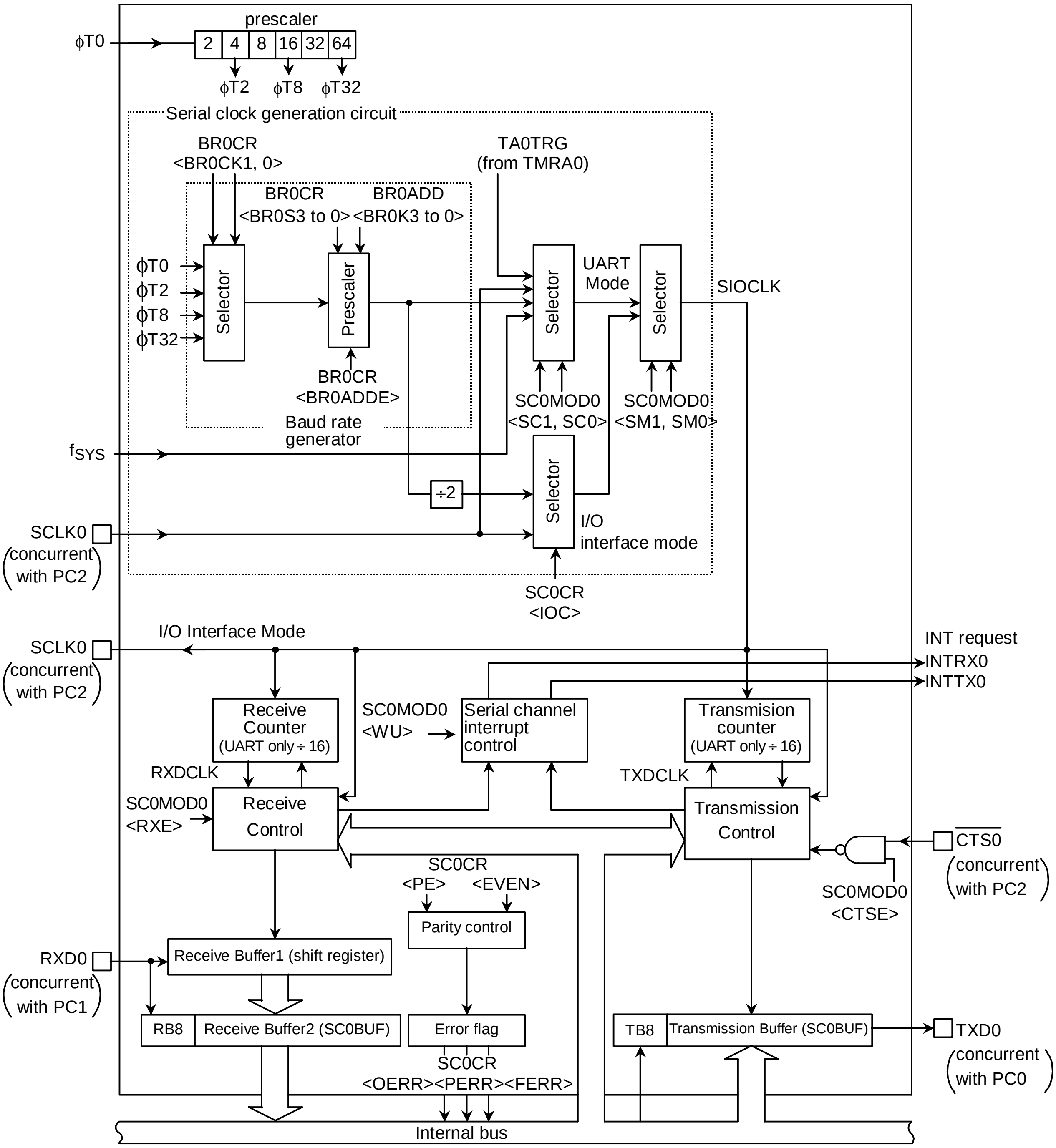


Figure 3.9.2 Block diagram of the Serial Channel 0 (SIO0)

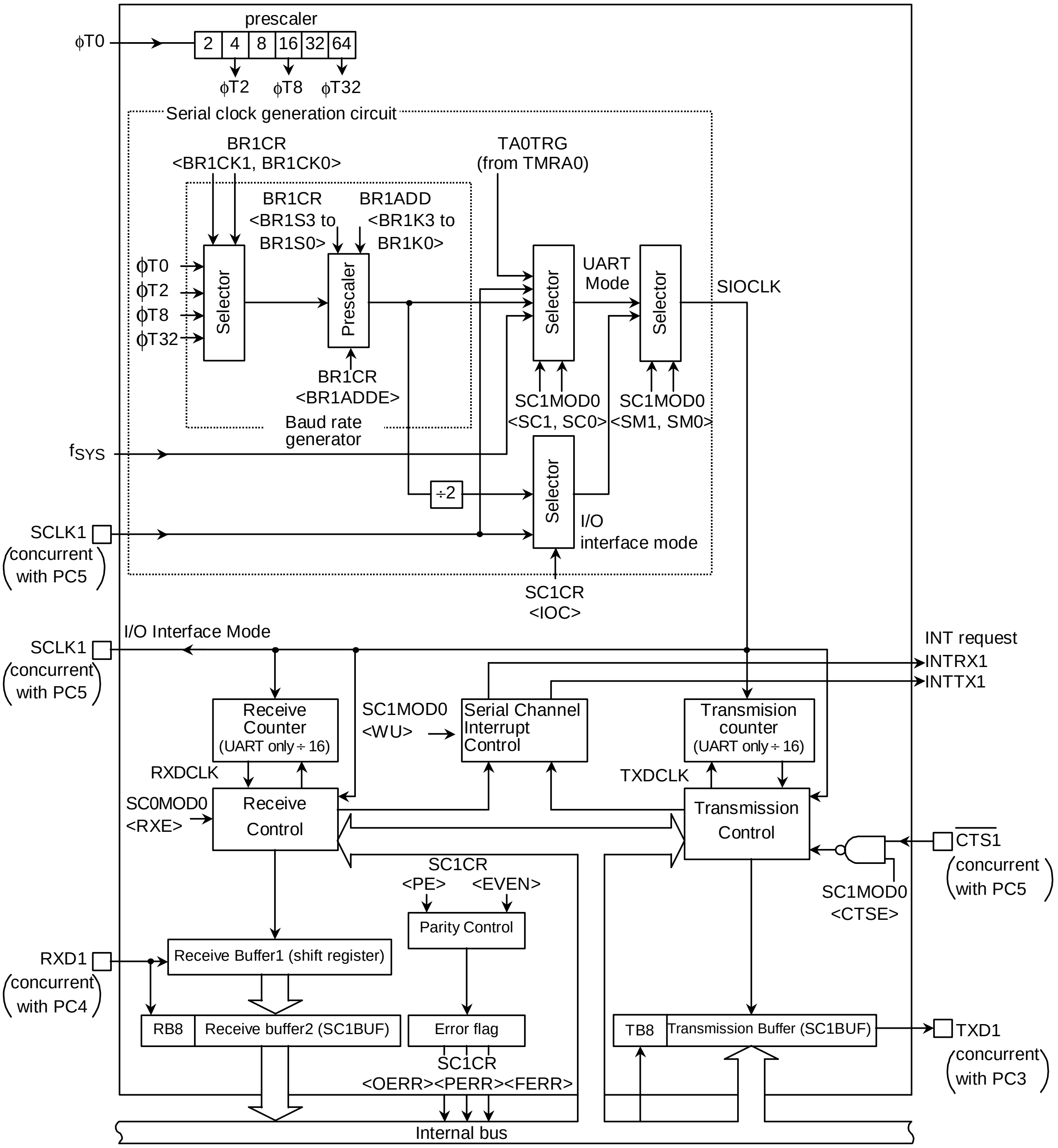


Figure 3.9.3 Block diagram of the Serial Channel 1(SIO1)

3.9.2 Operation of each circuit

(1) Prescaler

There is a 6-bit prescaler for generating a clock to SIO0. The clock selected using SYSCR<PRCK1:PRCK0> is divided by 4 and input to the prescaler as φT0. The prescaler can be run by selecting the baud rate generator as the serial transfer clock.

Table 3.9.2 shows prescaler clock resolution into the baud rate generator.

Table 3.9.2 Prescaler Clock Resolution to Baud Rate Generator

Select System Clock <SYSCK>	Select Prescaler Clock <PRCK1 to PRCK0>	Gear Value <GEAR2 to GEAR0>	Prescaler Output Clock Resolution			
			φT0	φT2	φT8	φT32
1 (fs)	00 (f_{FPH})	XXX	$fs/2^2$	$fs/2^4$	$fs/2^6$	$fs/2^8$
0 (fc)		000 (fc)	$fc/2^2$	$fc/2^4$	$fc/2^6$	$fc/2^8$
		001 ($fc/2$)	$fc/2^3$	$fc/2^5$	$fc/2^7$	$fc/2^9$
		010 ($fc/4$)	$fc/2^4$	$fc/2^6$	$fc/2^8$	$fc/2^{10}$
		011 ($fc/8$)	$fc/2^5$	$fc/2^7$	$fc/2^9$	$fc/2^{11}$
		100 ($fc/16$)	$fc/2^6$	$fc/2^8$	$fc/2^{10}$	$fc/2^{12}$
	10 ($fc/16$ clock)	XXX	–	$fc/2^8$	$fc/2^{10}$	$fc/2^{12}$

(note): X = Don’t care; “–” = Cannot be used

The Baud Rate Generator selects between 4 clock inputs : φT0, φT2, φT8, and φT32 among the prescaler outputs.

(2) Baud rate generator

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

The input clock to the baud rate generator, ϕT0, ϕT2, ϕT8 or ϕT32, is generated by the 6-bit prescaler which is shared by the timers. One of these input clocks is selected using the BR0CR<BR0CK1 to BR0CK0> 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 to BR0S0> and BR0ADD<BR0K3 to BR0K0>.

- In UART Mode

(1) When BR0CR<BR0ADDE> = 0

The settings BR0ADD<BR0K3 to BR0K0> are ignored. The baud rate generator divides the selected prescaler clock by N, which is set in BR0CK<BR0S3 to BR0S0>. (N = 1, 2, 3 ⋯ 16)

(2) When BR0CR<BR0ADDE> = 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 to BR0S0> (N = 2, 3 ⋯ 15) and the value of K set in BR0ADD<BR0K3 to R0K0> (K = 1, 2, 3 ⋯ 15)

Note: If N = 1 or N = 16, the N + (16 – K) / 16 division function is disabled. Set BR0CR<BR0ADDE> 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

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

- In I/O Interface Mode

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

- Integer divider (N divider)

For example, when the source clock frequency (fc) = 12.288 MHz, the input clock frequency = ϕT2 (fc/16), the frequency divider N (BR0CR<BR0S3 to BR0S0>) = 5, and BR0CR<BR0ADDE> = 0, the baud rate in UART Mode is as follows:

$*$ Clock state
System clock: High frequency (fc)
Clock gear: 1 (fc)
Prescaler clock: System clock

$$\text{Baud Rate} = \frac{\text{fc}/16}{5} \div 16$$

$$= 12.288 \times 10^6 \div 16 \div 5 \div 16 = 9600 \text{ (bps)}$$

Note: The N + (16 – K) / 16 division function is disabled and setting BR0ADD<BR0K3 to BR0K0> is invalid.

- N+(16-K)/16 divider (UART Mode only)

Accordingly, when the source clock frequency (fc) = 4.8 MHz, the input clock frequency = ϕT0 , the frequency divider N (BR0CR<BR0S3 to BR0S0>) = 7, K (BR0ADD<BR0K3 to BR0K0>) = 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: System clock

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

$$= 4.8 \times 10^6 \div 4 \div (7+13/16) \div 16 = 9600 \text{ (bps)}$$

Table 3.9.3, Table 3.9.4 show examples of UART Mode transfer rates.

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

- In UART Mode

  Baud rate = external clock input frequency ÷ 16

  It is necessary to satisfy (external clock input cycle)> = fc / 4

- In I/O Interface Mode

  Baud rate = external clock input frequency

  It is necessary to satisfy (external clock input cycle) >=16 / fc

Table 3.9.3 Transfer rate selection

(when baud rate generator Is used and BR0CR <BR0ADDE> = 0)

Unit (kbps)

fc [MHz]	Input Clock Frequency Divider	φT0	φT2	φT8	φT32
9.830400	2	76.800	19.200	4.800	1.200
	4	38.400	9.600	2.400	0.600
	8	19.200	4.800	1.200	0.300
	0	9.600	2.400	0.600	0.150
12.288000	5	38.400	9.600	2.400	0.600
	A	19.200	4.800	1.200	0.300
14.745600	2	115.200			
	3	76.800	19.200	4.800	1200
	6	38.400	9.600	2.400	0.600
	C	19.200	4.800	1.200	0.300

(note1): Transfer rates in I/O Interface Mode are eight times faster than the values given above.

(note2): The values in this table are calculated for when fc is selected as the system clock, the clock gear is set for fc/1 and the system clock is the prescaler clock input f_{FPH}.

Table 3.9.4 UART baud rate selection

(When TMRA0 with input Clock φT1 is used)

Unit (kbps)

fc	12.288	12	9.8304	8	6.144
TA0REG0	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

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

$$\text{Transfer rate} = \frac{\text{Clock frequency determined by SYSCR0<PRCK1, PRCK0>}}{\text{TA0REG} \times \underbrace{8}_{\text{(when TMRA0 (input clock } \phi\text{T1) is used)}} \times 16}$$

(note1): The TMRA0 match detect signal cannot be used as the transfer clock in I/O Interface Mode.

(note2): The values in this table are calculated for when fc is selected as the system clock, the clock gear is set for fc/1 and the system clock is the prescaler clock input f_{FPH}.

(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 SC0CR<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 SC0CR<IOC> = 1, the rising edge or falling edge will be detected according to the setting of the SC0CR<SCLKS> register to generate the basic clock.

- In UART Mode

The SC0MOD0 <SC1, SC0> setting determines whether the baud rate generator clock, the internal system clock fSYS, the match detect 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 which 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 is taken to be 0.

(5) Receiving control

- In I/O Interface Mode

In SCLK Output Mode with the setting SC0CR<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 SCLK0 input, according to the SC0CR<SCLKS> setting.

- In UART Mode

The receiving control block has circuit which 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.

SC0CR<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 which 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.

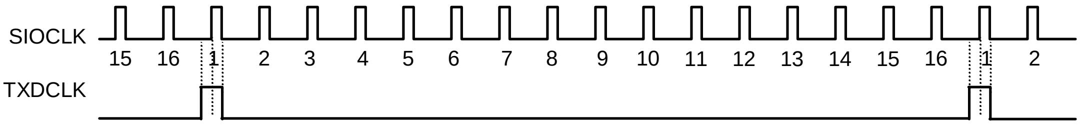


Figure 3.9.4 Generation of the transmission clock

(8) Transmission controller

- In I/O Interface Mode

In SCLK Output Mode with the setting SC0CR<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 SC0CR<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 SC0CR<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, 1 each has a CTS pin. Use of this pin allows data can be sent in units of one frame; thus, Overrun errors can be avoided. The handshake functions is enabled or disabled by the SC0MOD <CTSE> setting.

When the CTS0 pin foes High on completion of the current data send, data transmission is halted until the CTS0 pin foes Low again. However, the INTTX0 Interrupt is generated, it requests the next data send to the CPU. The next data is written in the Transmission Buffer and data sending is halted.

Though there is no RTS pin, a handshake function can be easily configured by setting any port assigned to be the RTS function. The RTS should be output "High" to request send data halt after data receive is completed by software in the RXD interrupt routine.

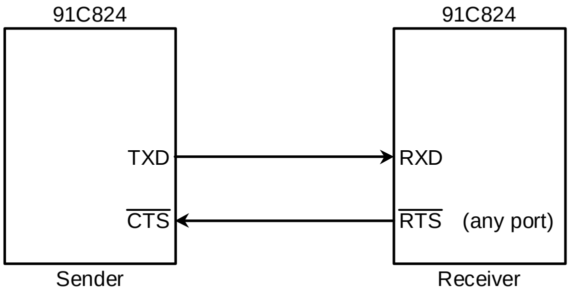
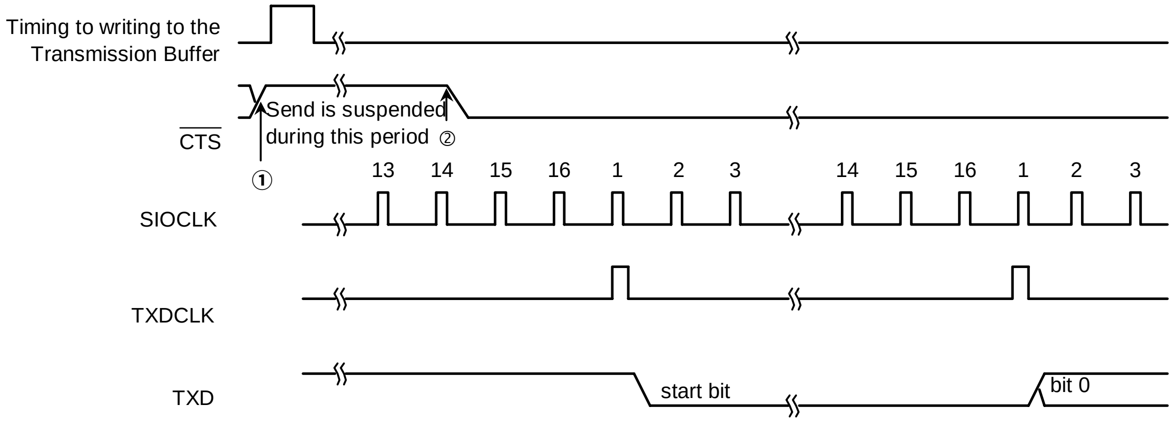


Figure 3.9.5 Handshake function


Timing to writing to the Transmission Buffer
$\overline{\text{CTS}}$
Send is suspended during this period
①
②
SIOCLK
13
14
15
16
1
2
3
14
15
16
1
2
3
TXDCLK
TXD
start bit
bit 0

(note1): If the $\overline{\text{CTS}}$ signal goes High during transmission, no more data will be sent after completion of the current transmission.

(note2): Transmission starts on the first falling edge of the TXDCLK clock after the $\overline{\text{CTS}}$ signal has fallen.

Figure 3.9.6 $\overline{\text{CTS}}$ (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 SC0CR<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 SC0CR<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 SC0BUF. The data is transmitted after the parity bit has been stored in SC0BUF<TB7> in 7-Bit UART Mode or in SC0MOD0<TB8> in 8-Bit UART Mode. SC0CR<PE> and SC0CR<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 <OERR>=1
           then
           4) Set to disable receiving (write ‘0’ to SC0MOD0<RXE>)
           5) Wait to terminate current frame
           6) Read receiving buffer
           7) Read error flag
           8) Set to enable receiving (write ‘1’ to SC0MOD0<RXE>)
           9) Request to transmitt again
10) 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

① In UART Mode

Receiving

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

Transmitting

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

② I/O interface

Transmission Interrupt timing	SCLK Output Mode	Immediately after rise of last SCLK signal. (See figure 3.9 19.)
	SCLK Input Mode	Immediately after rise of last SCLK signal Rising Mode, or immediately after fall in Falling Mode. (See figure 3.9 20.)
Receiving Interrupt timing	SCLK Output Mode	Timing used to transfer received to data Receive Buffer 2 (SC0BUF) (i.e. immediately after last SCLK). (See figure 3.9 21.)
	SCLK Input Mode	Timing used to transfer received data to Receive Buffer 2 (SC0BUF) (i.e. immediately after last SCLK). (See figure 3.9 22.)

3.9.3 SFR

		7	6	5	4	3	2	1	0
SC0MOD0 (0202H)	Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
	Read/Write	R/W							
	After Reset	0	0	0	0	0	0	0	0
	Function	Transfer data bit 8	Hand shake 0: CTS disable 1: CTS enable	Receive function 0: Receive disable 1: Receive enable	Wake up function 0: disable 1: enable	Serial Transmission Mode 00: I/O interface Mode 01: 7-bit UART Mode 10: 8-bit UART Mode 11: 9-bit UART Mode		Serial transmission clock (UART) 00: TMRA0 trigger 01: Baud rate generator 10: Internal clock f_{SYS} 11: External clcok (SCLK0 input)	

Serial transmission clock source (UART)

00	Timer TMRA0 match detect signal
01	Baud rate generator
10	Internal clock f_{SYS}
11	External clock (SCLK0 input)

Note: The clock selection for the I/O interface mode is controlled by the serial bontrol register (SC0CR).

Serial Transmission Mode

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

Wake-up function

	9-Bit UART	Other Modes
0	Interrupt generated when data is received	Don't care
1	Interrupt generated only when RB8 = 1	

Receiving Function

0	Receive disabled
1	Receive enabled

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

0	Disabled (always transferable)
1	Enabled

Transmission data bit 8

Figure 3.9.7 Serial Mode Control Register (SIO0, SC0MOD0)

SC0MOD0
(020AH)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After Reset	0	0	0	0	0	0	0	0
Function	Transfer data bit 8	Hand shake 0: CTS disable 1: CTS enable	Receive function 0: Receive disable 1: Receive enable	Wake up function 0: disable 1: enable	Serial Transmission Mode 00: I/O interface Mode 01: 7-bit UART Mode 10: 8-bit UART Mode 11: 9-bit UART Mode		Serial transmission clock (UART) 00: TMRA0 trigger 01: Baud rate generator 10: Internal clock f_{SYS} 11: External clcok (SCLK0 input)	

Serial transmission clock source (for UART)

00	Timer TMRA0 match detect signal
01	Baud rate generator
10	Internal clock f_{SYS}
11	External clock (SCLK1 input)

Serial Transmission Mode

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

Wake-up function

	9-Bit UART	Other Modes
0	Interrupt generated when data is received	Don’t care
1	Interrupt generated only when RB8 = 1	

Receiving function

0	Receive disabled
1	Receive enabled

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

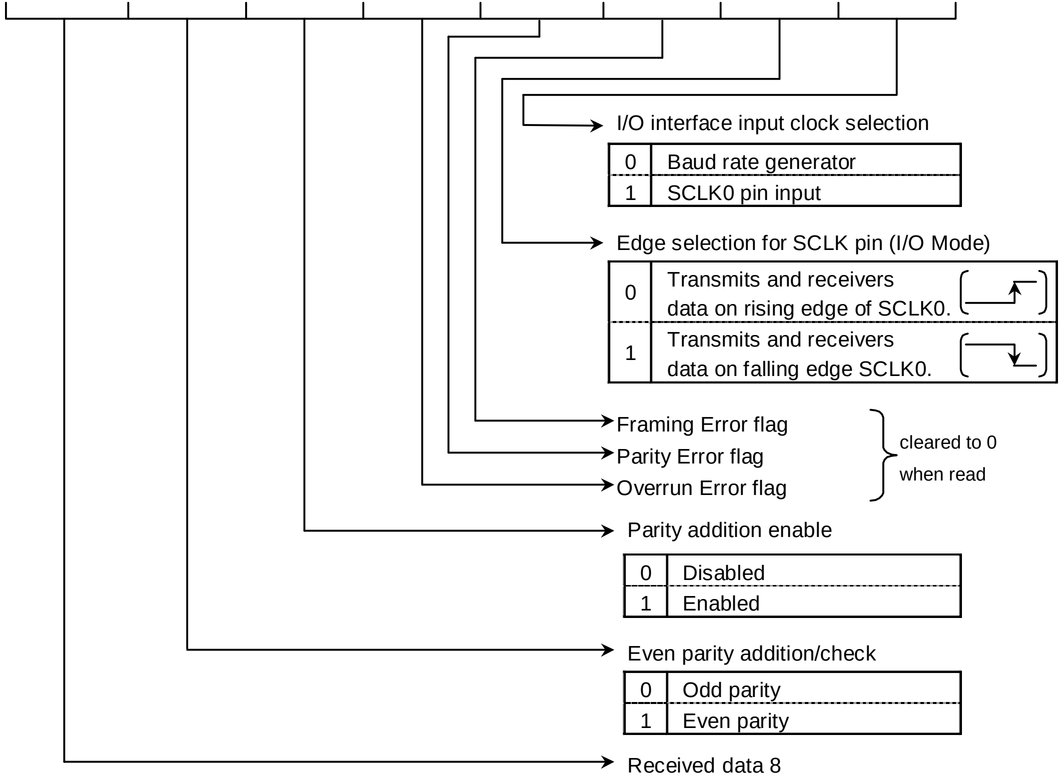
0	Disabled (always transferable)
1	Enabled

Transmission data bit 8

Figure 3.9.8 Serial Mode Control Register (SIO1, SC1MOD0)

SC0CR
(0201H)

	7	6	5	4	3	2	1	0
bit Symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R(cleared to 0 when read)			R/W	
After Reset		0	0	0	0	0	0	0
Function	Received data bit 8	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: SCLK0 pin input



I/O interface input clock selection

0	Baud rate generator
1	SCLK0 pin input

Edge selection for SCLK pin (I/O Mode)

0	Transmits and receivers data on rising edge of SCLK0.
1	Transmits and receivers data on falling edge SCLK0.

Framing Error flag
Parity Error flag
Overrun Error flag } cleared to 0 when read

Parity addition enable

0	Disabled
1	Enabled

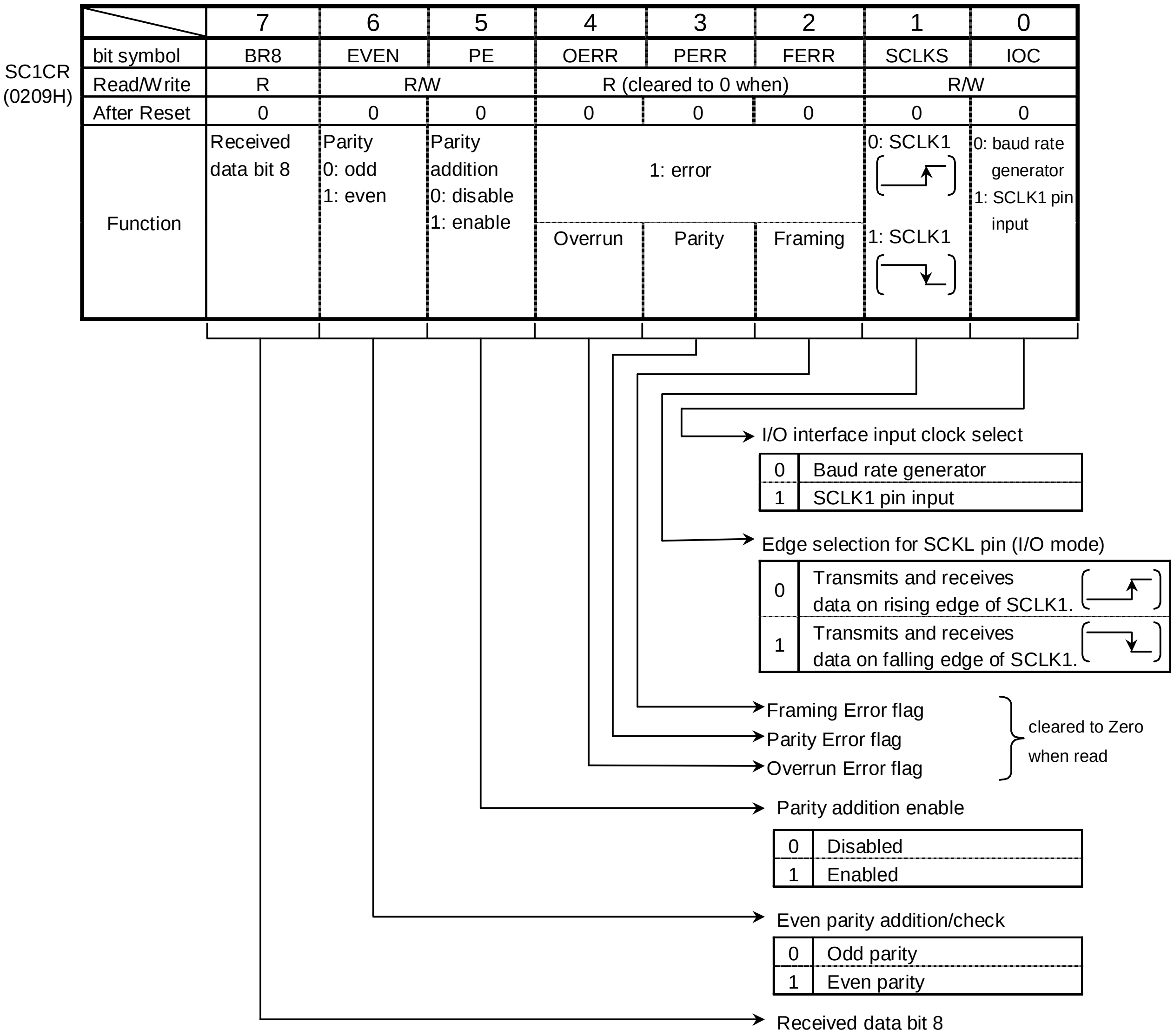
Even parity addition/check

0	Odd parity
1	Even parity

Received data 8

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

Figure 3.9.9 Serial Control Register (SIO0, SC0CR)



SC1CR (0209H)

	7	6	5	4	3	2	1	0
bit symbol	BR8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (cleared to 0 when)			R/W	
After Reset	0	0	0	0	0	0	0	0
Function	Received data bit 8	Parity 0: odd 1: even	Parity addition 0: disable 1: enable	1: error Overrun	Parity	Framing	0: SCLK1 1: SCLK1	0: baud rate generator 1: SCLK1 pin input

I/O interface input clock select

0	Baud rate generator
1	SCLK1 pin input

Edge selection for SCKL pin (I/O mode)

0	Transmits and receives data on rising edge of SCLK1.
1	Transmits and receives data on falling edge of SCLK1.

Parity addition enable

0	Disabled
1	Enabled

Even parity addition/check

0	Odd parity
1	Even parity

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

Figure 3.9.10 Serial Control Register (SIO1, SC1CR)

BR0CR
(0203H)

	7	6	5	4	3	2	1	0
Bit symbol		BR0ADDE	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
Read/Write	R/W							
After Reset	0	0	0	0	0	0	0	0
Function	Always write “0”	+(16−K)/16 division 0: Disable 1: Enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Setting of the Divided frequency			

+(16−K)/16 division enable

0	Disable
1	Enable

Setting the input clock of baud rate generator

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

BR0ADD
(0204H)

	7	6	5	4	3	2	1	0
bit Symbol					BR0K3	BR0K2	BR0K1	BR0K0
Read/Write					R/W			
After reset					0	0	0	0
Function					Sets frequency divisor “K” (divided by N + (16-K) / 16)			

Sets baud rate generator frequency divisor

	BR0CR<BR0ADDE> = 1		BR0CR<BR0ADDE> = 0

BR0CR
<BR0S3:0>
BR0ADD
<BR0K3:0>
	0000 (N = 16)
	or
	0001 (N = 1)
		0010 (N = 2)
		to
		1111 (N = 15)
			0001 (N = 1) (UART only)
			to
			1111(N = 15)
			0000(N = 16)

0000	Disable	Disable	Divided by N

0001(K = 1)
to
1111(K = 15)
	Disable	Divided by
		N + (16-K) /16	

(note1): The baud rate generator can be set “1” when UART mode and disable + (16 − K)/16 division function. Don’t use in I/O interface mode.

(note2): Set BR0CR <BR0ADDE> to “1” after setting K (K=1 to 15) to BR0ADD<BR0K3 to 0> when + (16 − K)/16 division function is used. However, don’t use + (16 − K)/16 division function when BR0CR<BR0S3 to 0>=”0000” or “0001”(N=16 or 1).

(note3): + (16 − K)/16 division function is possible to use in only UART mode.
Set BR0CR <BR0ADDE> to “0” and disable + (16 − K)/16 division function in I/O interface mode.

Figure 3.9.11 Baud rate generator control (SIO0, BR0CR, BR0ADD)

BR1CR
(020BH)

	7	6	5	4	3	2	1	0
bit Symbol	–	BR1ADDE	BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Always write “0”	+(16–K)/16 division 0: Disable 1: Enable	00: φT0 01: φT2 10: φT8 11:φT32		Divided Frequency setting			

+(16 - K) / 16 division enable

0	Disabled
1	Enabled

Input clock selection for baud rate generator

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

BR1ADD
(020CH)

	7	6	5	4	3	2	1	0
Bit symbol					BR1K3	BR1K2	BR1K1	BR1K0
Read/Write					R/W			
After Reset					0	0	0	0
Function					Set frequency divisor K (divided by N + (16 – K)/16)			

Baud rate generator frequency divisor setting

	BR1CR<BR1ADDE> = 1		BR1CR<BR1ADDE> = 0

BR0CR <BR1S3 to BR1S0>
BR1ADD <BR1K3 to BR1K0>
	0000(N = 16)
	or
	0001(N = 1)
		0010(N = 2)
		to
		1111(N = 15)
			0001(N = 1) (UART only)
			to
			1111(N = 15)
			0000(N = 16)

0000	Disable	Disable	Divided by N

0001(K = 1)
to
1111(K = 15)
	Disable	Disabled by
		N + (16 – K) / 16	

(note1): The baud rate generator can be set “1” when UART mode and disable + (16 – K)/16 division function. Don’t use in I/O interface mode.

(note2): Set BR1CR <BR1ADDE> to “1” after setting K (K=1 to 15) to BR1ADD<BR1K3 to 0> when + (16 – K)/16 division function is used. However, don’t use + (16 – K)/16 division function when BR1CR<BR1S3 to 0>=”0000” or “0001”(N=16 or 1).

(note3): + (16 – K)/16 division function is possible to use in only UART mode.
Set BR1CR <BR1ADDE> to “0” and disable + (16 – K)/16 division function in I/O interface mode.

Figure 3.9.12 Baud rate generator control (SIO1, BR1CR, BR1ADD)

SC0BUF
(0200H)

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

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

(note): Prohibit read modify write for SC0BUF.

Figure 3.9.13 Serial Transmission/Receiving Buffer Registers (SIO0, SC0BUF)

SC0MOD1
(0205H)

	7	6	5	4	3	2	1	0
Bit symbol	I2S0	FDPX0						
Read/Write	R/W	R/W						
After Reset	0	0						
Function	IDLE2 0: Stop 1: Run	duplex 0: half 1: full						

Figure 3.9.14 Serial Mode Control Register 1 (SIO0, SC0MOD1)

SC1BUF
(0208H)

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

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

(note): Prohibit read modify write for SC1BUF.

Figure 3.9.15 Serial Transmission/Receiving Buffer Registers (SIO1, SC1BUF)

SC1MOD1
(020DH)

	7	6	5	4	3	2	1	0
bit Symbol	I2S0	FDPX0						
Read/Write	R/W	R/W						
After Reset	0	0						
Function	IDLE2 0: Stop 1: Run	duplex 0: half 1: full						

Figure 3.9.16 Serial Mode Control Register 1 (SIO1, SC1MOD1)

3.9.4 Operation in 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.

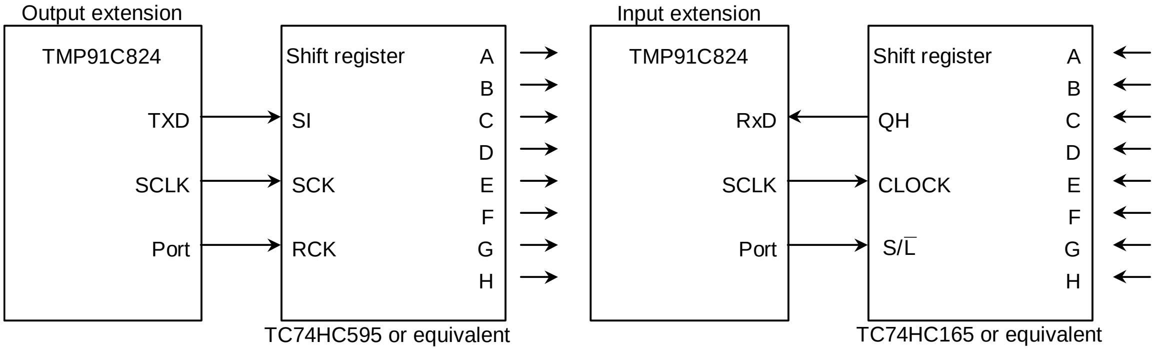


Figure 3.9.17 SCLK Output Mode connection example

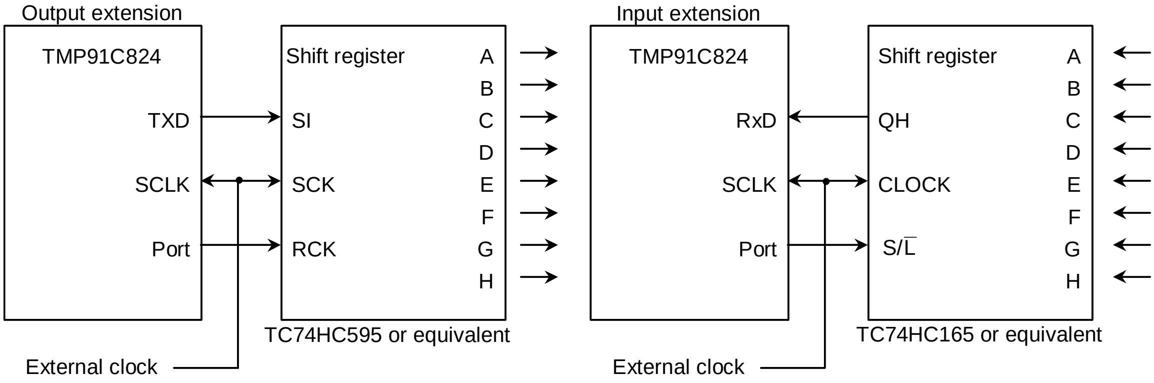


Figure 3.9.18 SCLK Input Mode Connection example

① 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 output, INTES0 <ITX0C> will be set to generate the INTTX0 interrupt.

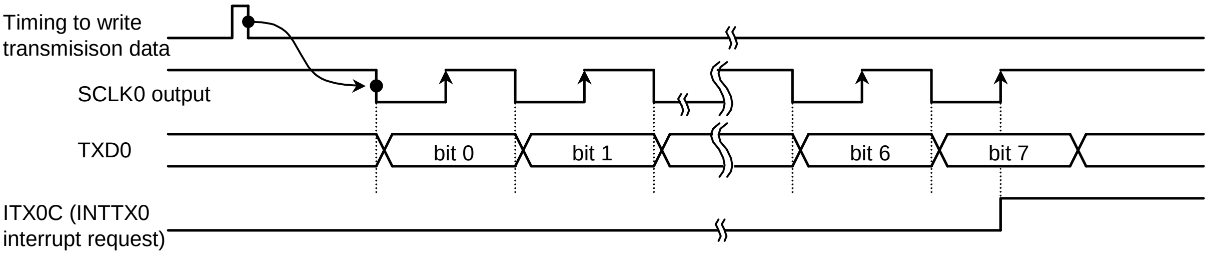


Figure 3.9.19 Transmitting Operation in I/O Interface Mode (SCLK0 Output Mode)

In SCLK Input Mode, 8-bit data is output on 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 output, INTES0 <ITX0C> will be set to generate INTTX0 interrupt.

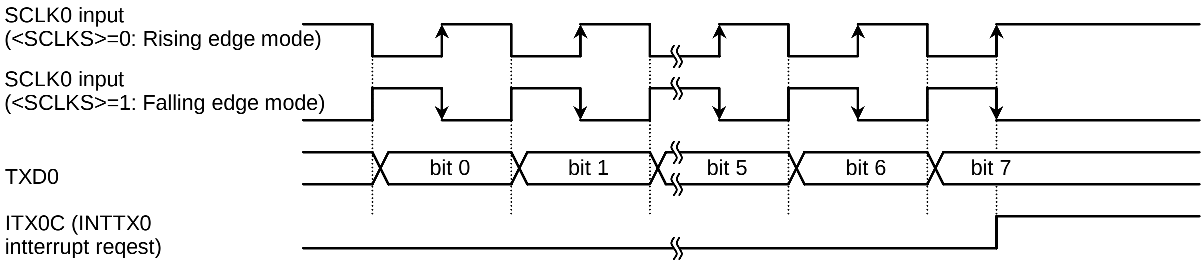


Figure 3.9.20 Transmitting Operation in I/O Interface Mode (SCLK0 Input Mode)

② 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 INTES0<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 INTES0<IRX0C> will be set to generate INTRX0 interrupt.
The outputting for the first SCLK0 starts by setting SC0MOD0<RXE>to 1.

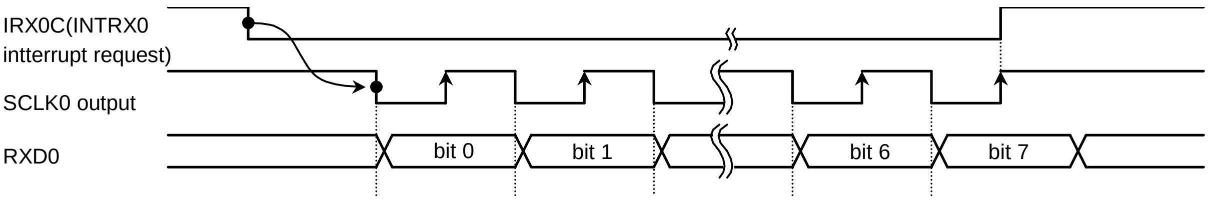


Figure 3.9.21 Receiving operation in I/O Interface Mode (SCLK0 Output Mode)

In SCLK input mode, the data is shifted to Receiving Buffer 1 when the SCLK input becomes active after the receive Interrupt flag INTES0 <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 INTES0 <IRX0C> will be set again to be generate INTRX0 interrupt.

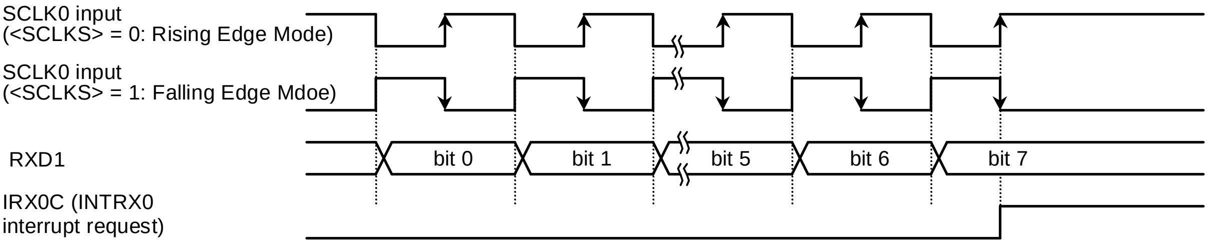


Figure 3.9.22 Receiving Operation in I/O interface Mode (SCLK0 Input Mode)

(note): The system must be put in the Receive Enable state (SCMOD0<RXE> = 1) before data can be received.

③ 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

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	0	1	0	0	0	0	Set the INTRX0 level to 0.
PCCR	–	–	–	–	–	1	0	1	Set PC0, PC1 and PC2 to function as the TXD0, RXD0 and SCLK0 pins respectively.
PCFC	–	–	–	–	–	1	–	1	
SC0MOD0	0	0	0	0	0	0	0	0	Select I/O Interface Mode.
SC0MOD1	1	1	0	0	0	0	0	0	Select 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	1	1	0	0	1	1	Baud rate = 9600 bps
SC0MOD0	0	0	1	0	0	0	0	0	Enable receiving
SC0BUF	*	*	*	*	*	*	*	*	Set the transmit data and start.

INTTX0 interrupt routine

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

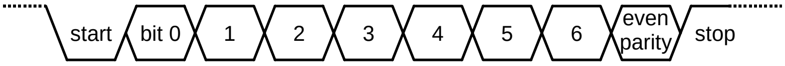
(note): X = Don't care; "–" = No change

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

7-Bit UART Mode is selected by setting Serial Channel Mode Register SC0MOD0<SM1, SM0> 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 SC0CR<PE> bit; 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 transmitting data of the following format, the control registers should be set as described below. This explanation applies to Channel 0.



← Transmission direction (transmission rate: 2400 bps at fc = 12.288 MHz)

* Clock state
System clock: High frequency (fc)
Clock gear: 1 (fc)
Prescaler clock: System clock

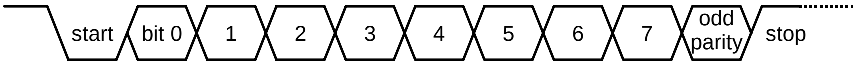
		7	6	5	4	3	2	1	0	
PCCR	←	–	–	–	–	–	–	–	1	Set PC0 to function as the TXD0 pin.
PCFC	←	–	–	–	–	–	–	–	1	
SC0MOD	←	X	0	–	X	0	1	0	1	Select 7-Bit UART Mode.
SC0CR	←	X	1	1	X	X	X	0	0	Add even parity.
BR0CR	←	0	0	1	0	0	1	0	1	Set the transfer rate to 2400 bps.
INTES0	←	1	1	0	0	–	–	–	–	Enable the INTTX0 interrupt and set it to Interrupt Level 4.
SC0BUF	←	*	*	*	*	*	*	*	*	Set data for transmission.

(note): X = Don't care; "–" = No change

(3) Mode 2 (8-Bit UART Mode)

8-Bit UART Mode is selected by setting SC0MOD0<SM1, SM0> 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.



← Transmission direction (transmission rate: 9600 bps at fc = 12.288 MHz)

* Clock state
System clock: High frequency (fc)
Clock gear: 1 (fc)
Prescaler clock: System clock

Main settings

```
                7  6  5  4  3  2  1  0
PCCR      ←  –  –  –  –  –  –  0  –        Set PC1 to function as the RXD0 pin.
SC0MOD    ←  –  0  1  X  1  0  0  1        Enable receiving in 8-Bit UART Mode.
SC0CR     ←  X  0  1  X  X  X  0  0        Add even parity.
BR0CR     ←  0  0  0  1  0  1  0  1        Set the transfer rate to 9600 bps.
INTES0    ←  –  –  –  –  1  1  0  0        Enable the INTTX0 interrupt and set it to Interrupt Level 4.
```

Interrupt processing

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

(note): X = Don’t care; “–” = No change

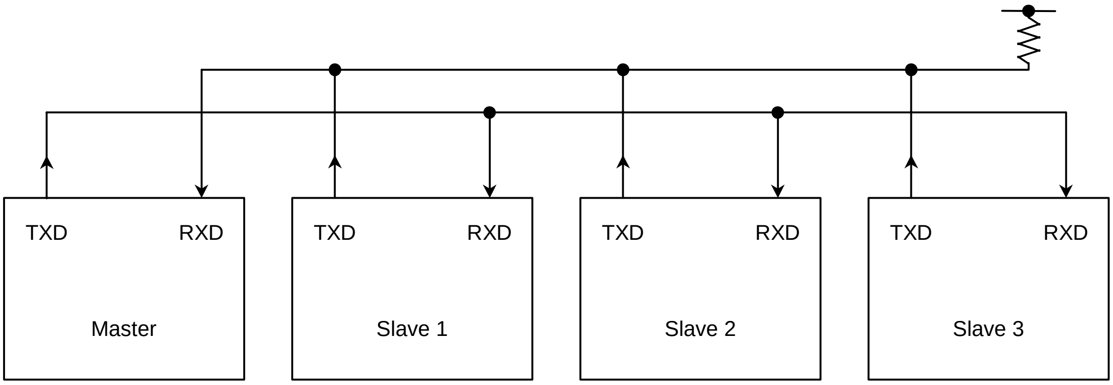
(4) Mode 3 (9-Bit UART Mode)

9-Bit UART Mode is selected by setting SC0MOD0<SM1, SM0> 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.

Wake-up function

In 9-Bit UART Mode, the wake-up function for slave controllers is enabled by setting SC0MOD0<WU> to 1. The interrupt INTRX0 occurs only when<RB8> = 1.



(note): The TXD pin of each slave controller must be in Open-Drain Output Mode.

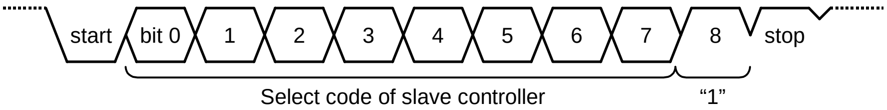
Figure 3.9.23 Serial Link using Wake-up function

Protocol

① Select 9-Bit UART Mode on the master and slave controllers.

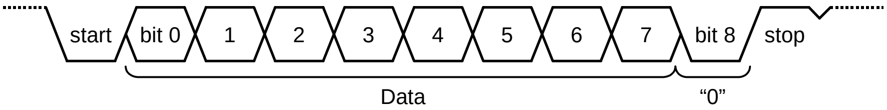
② Set the SC0MOD0<WU> bit on each slave controller to 1 to enable data receiving.

③ The master controller transmits one-frame data including the 8-bit select code for the slave controllers. The MSB (bit 8)<TB8> is set to 1.



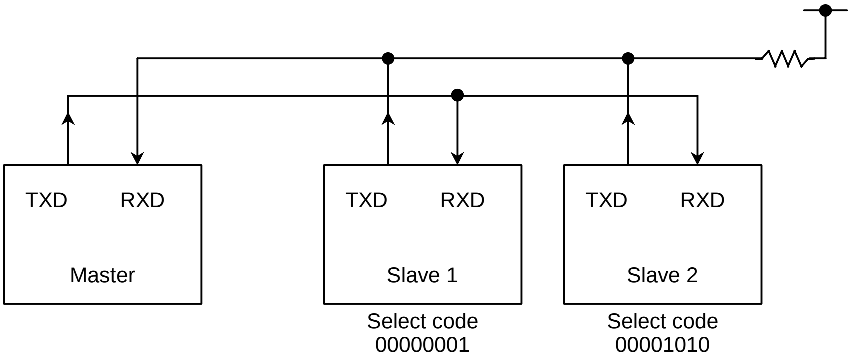
④ Each slave controller receives the above frame. Each controller checks the above select code against its own select code. The controller whose code matches clears its WU bit to 0.

⑤ The master controller transmits data to the specified slave controller whose SC0MOD<WU> bit is cleared to 0. The MSB (bit 8) <TB8> is cleared to 0.



⑥ The other slave controllers (whose <WU> bits remain at 1) ignore the received data because their MSBs (bit 8 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.

Setting example: To link two slave controllers serially with the master controller using the internal clock f_{SYS} as the transfer clock.



Since Serial Channels 0 and 1 operate in exactly the same way, Channel 0 only is used for the purposes of this explanation.

- Setting the master controller

Main

PCCR	← − − − − − − 0 1	Set PC0 and PC1 to function as the TXD0 and RXD0 pins respectively.
PCFC	← − − − − − − X 1	
INTES0	← 1 1 0 0 1 1 0 1	Enable the INTTX0 interrupt and set it to Interrupt Level 4. Enable the INTRX0 interrupt and set it to Interrupt Level 5.
SC0MOD0	← 1 0 1 0 1 1 1 0	Set f_{SYS} as the transmission clock for 9-Bit UART Mode.
SC0BUF	← 0 0 0 0 0 0 0 1	Set the select code for slave controller 1.

INTTX0 interrupt

SC0MOD0	← 0 − − − − − − −	Set TB8 to 0.
SC0BUF	← * * * * * * * *	Set data for transmission.

- Setting the slave controller

Main

P9CR	← − − − − − − 0 1	Set PC1 to RXD and PC0 to TXD0(open-drain output).
P9FC	← − − − − − − X 1	
PCODE	← X X X X − X X 1	
INTES0	← 1 1 0 1 1 1 1 0	Enable INTRX0 and INTTX0.
SC0MOD0	← 0 0 1 1 1 1 1 0	Set <WU> to 1 in 9-Bit UART Transmission Mode using f_{SYS} as the transfer clock.

INTRX0 interrupt

Acc ← SC0BUF
if Acc = select code
Then SC0MOD0 ← - - - 0 - - - - Clear <WU> to 0.

3.9.5 Support for IrDA

SIO0 includes support for the IrDA 1.0 infrared data communication specification. Figure 3.9.24 shows the block diagram.

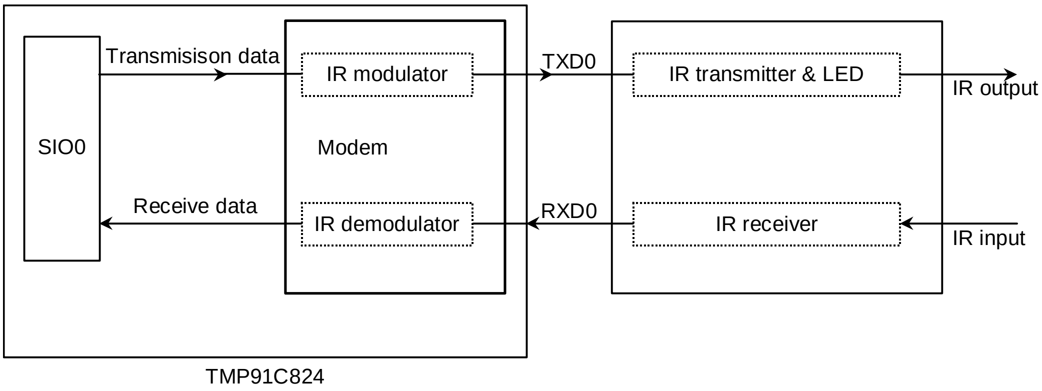


Figure 3.9.24 IrDA block diagram

(1) Modulation of the transmission data

When the transfer data is 0, the modem outputs 1 to TXD0 pin with either 3/16 or 1/16 times for width of baud-rate. The pulse width is selected by the SIRCR<PLSEL>. When the transfer data is 1, the modem outputs 0.

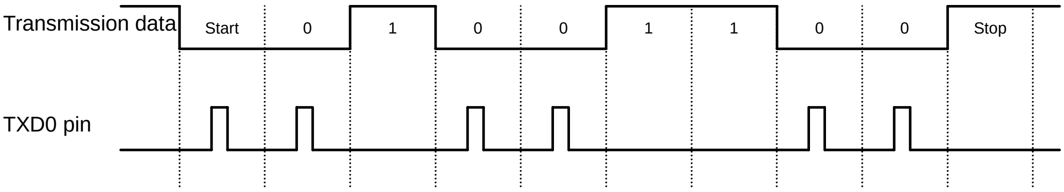


Figure 3.9.25 Modulation example of transfer data

(2) Modulation of the receive data

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

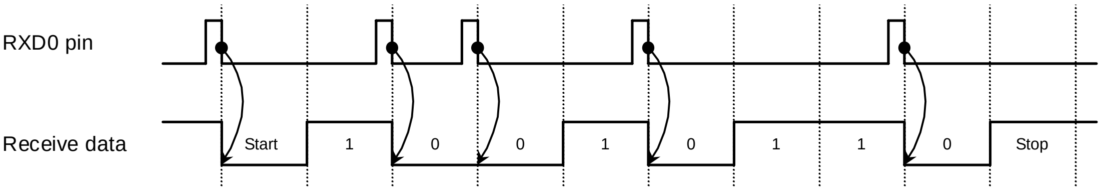


Figure 3.9.26 Demodulation example of receive data

(3) Data format

The data format is fixed as follows:

- Data length: 8-bit
- Parity bits: none
- Stop bits: 1

Any other setting don’t guarantee the normal operation.

(4) SFR

Figure 3.9.27 shows the control register SIRCR. Set the data SIRCR during SIO0 is inhibited (Both TXEN and RXEN of this register should be set to 0).

Any changing for this register during transmission or receiving operation don’t guarantee the normal operation.

The following example describes how to set this register:

```
1) SIO setting             ; Set the SIO to UART Mode.
        ↓
2) LD (SIRCR), 07H         ; Set the receive data pulse width to 16×.
3) LD (SIRCR), 37H         ; TXEN, RXEN Enable the Transmission and receiving of SIO.
        ↓
4) Start transmission      ; The modem operates as follows:
   and receiving for SIO0  • SIO0 starts transmitting.
                           • IR receiver starts receiving.
```

(5) Notes

1) Baud rate generator for IrDA

To generate baud-rate for IrDA, use baud-rate generator in SIO0 by setting “01” to SC0MOD0<SC1:0>. To use another source (TA0TRG,f_{SYS} and SCLK0-input) are not allowed.

2) As the IrDA 1.0 physical layer specification, the data transfer speed and infra-red pulse width is specified

Table 3.9.5 Baud rate and pulse width specifications

Baud Rate	Modulation	Rate Tolerance (% of rate)	Pulse Width (minimum)	Pulse Width (typical)	Pulse width (maximum)
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	±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

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

The TMP91C824F has the function selects the pulse width on the transmission either 3/16 or 1/16. But 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 x 1/16.

As the same reason, + (16-k)/16 division function in the baud rate generator of SIO0 can not 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 3.9.6 shows Baud rate and pulse width for (16 – k) / 16 division function.

Table 3.9.7 Baud rate and pulse width for (16 – k) / 16 division function

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	×	○	○	○	○	○
T × 1/16	–	–	×	○	○	○

○: Can be used (16-k)/16 division function
×: Can not be used (16-k)/16 division function
–: Can not be set to 1/16 pulse width

SIRCR
(0207H)

	7	6	5	4	3	2	1	0
Bit symbol	PLSEL	RXSEL	TXEN	RXEN	SIRWD3	SIRWD2	SIRWD1	SIRWD0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Select transmit pulse width 0: 3/16 1: 1/16	Receive data 0: "H" pulse 1: "L" pulse	Transmit 0: disable 1: enable	Receive 0: disable 1: enable	Select receive pulse width Set effective pulse width for equal or more than 2x × x (value + 1) Can be set : 1 to 4 Can not be set : 0, 15			

Select receive pulse width

Formula: Effective pulse width ≥ 2x × x (value + 1)

x = 1/fFPH

0000	Cannot be set
0001	Equal or more than 4x+100nS
to	
1110	Equal or more than 30x+100nS
1111	Can not be set

Receive operation

0	Disabled
1	Enabled

Transmit operation

0	Disabled
1	Enabled

Select transmit pulse width

0	3/16
1	1/16

Figure 3.9.27 IrDA Control Register

3.10 Serial Bus Interface (SBI)

The TMP91C824F has a 1-channel serial bus interface which employs a clocked-synchronous 8-bit SIO mode and an I^{2}C bus mode.

The serial bus interface is connected to an external device through P71 (SDA) and P72 (SCL) in the I^{2}C bus mode; and through P70 (SCK), P71 (SO), P72 (SI) in the clocked-synchronous 8-bit SIO mode.
Each pin is specified as follows.

	P7ODE <ODE72, ODE71>	P7CR <P72C, P71C, P70C>	P7FC <P72F, P71F, P70F>
I^{2}C Bus Mode	11	11X	11X
Clocked Synchronous 8-Bit SIO Mode	XX	011 010	X11

X: Don’t care

3.10.1 Configuration

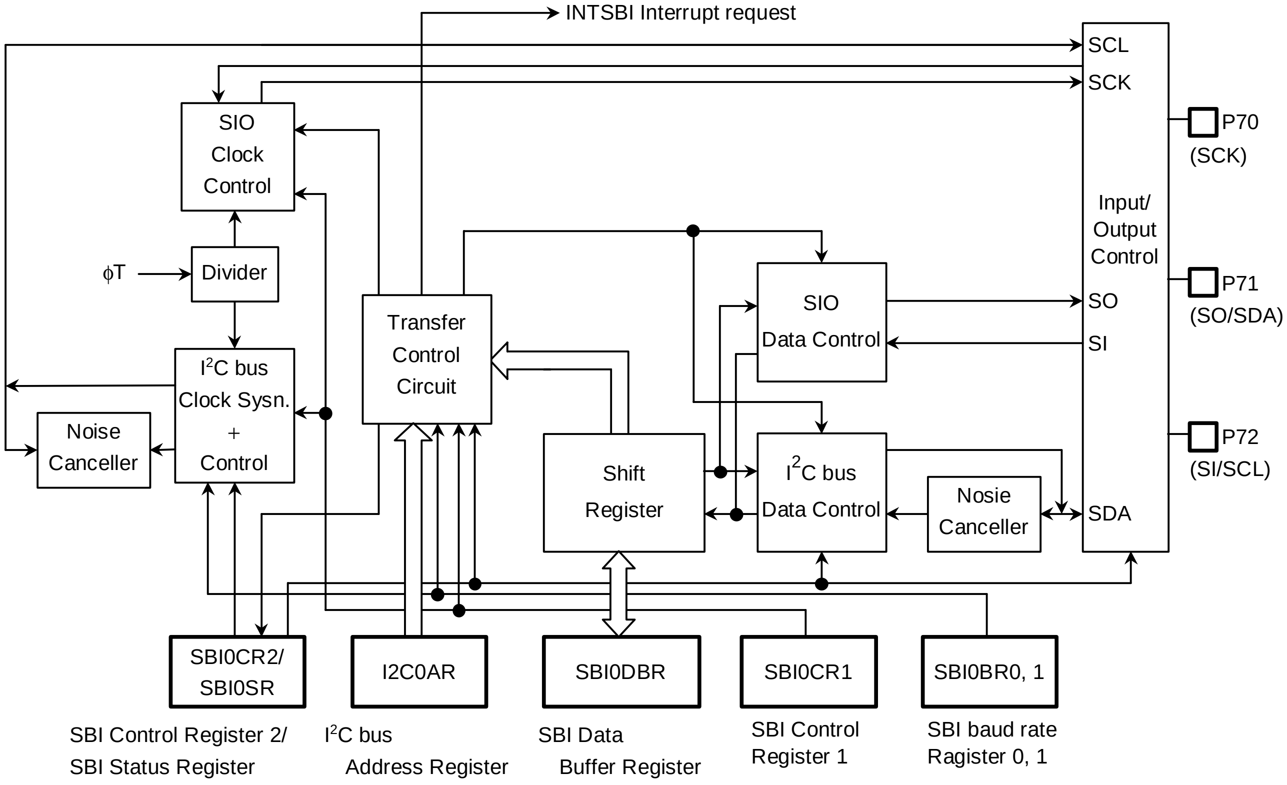


Figure 3.10.1 Serial Bus Interface (SBI)

3.10.2 Serial Bus Interface (SBI) Control

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

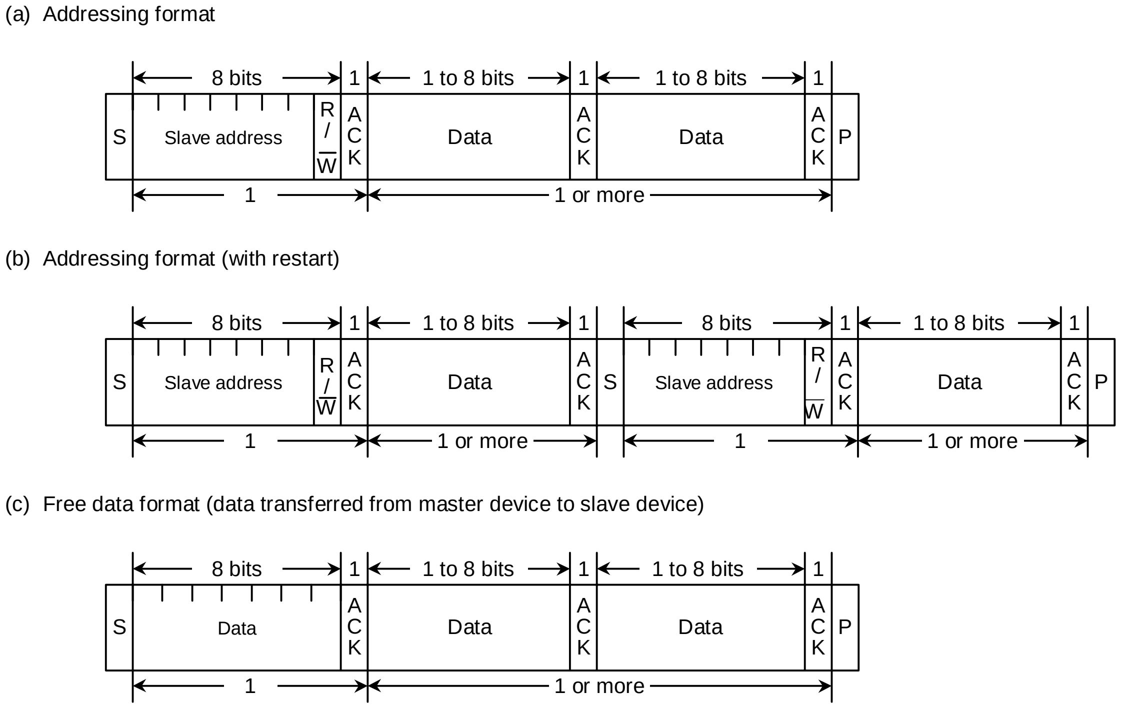
- Serial bus interface control register 1 (SBI0CR1)
- Serial bus interface control register 2 (SBI0CR2)
- Serial bus interface data buffer register (SBI0DBR)
- I^{2}C bus address register (I2C0AR)
- Serial bus interface status register (SBI0SR)
- Serial bus interface baud rate register 0 (SBI0BR0)
-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

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

Refer to Section “3.10.4 I2C bus Mode Control” and “3.10.7 Clocked-synchronous 8-bit SIO Mode Control”.

3.10.3 The Data Formats in the I^{2}C Bus Mode

The data formats in the I^{2}C bus mode is shown below.


(a) Addressing format

(b) Addressing format (with restart)

(c) Free data format (data transferred from master device to slave device)

Note: S: Start condition

R/$\overline{W}$: Direction bit

ACK: Acknowledge bit

P: Stop condition

Figure 3.10.2 Data format in the I^{2}C Bus Mode

3.10.4 I^2C Bus Mode Control

The following registers are used to control and monitor the operation status when using the serial bus interface (SBI) in the I^2C bus mode.

Seirial Bus Interface Conrol Register 1

SBI0CR1
(0240H)

Prohibit
Read-
modify-write

	7	6	5	4	3	2	1	0
Bit symbol	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0/ SWRMON
Read/Write	W			R/W		W		R/W
After Reset	0	0	0	0		0	0	0
Function	Number of transferred bits (Note 1)			Acknowledge mode specification 0: Not generate 1: Generate		Internal serial clock selection and software reset monitor (Note 2)		

Internal serial clock selection <SCK2 : SCK 0> @ write

000	n = 5	400	kHz
001	n = 6	222	kHz
010	n = 7	118	kHz
011	n = 8	60.6	kHz
100	n = 9	30.8	kHz
101	n = 10	15.5	kHz
110	n = 11	7.78	kHz
111		(reserved)	

System clock: fc
Clock gear: fc/1
fc = 16 MHz
(internal SCL output)

$$\text{fscl} = \frac{\text{fc}}{2^n + 8} \ [\text{Hz}]$$

Software reset state monitor <SWRMON> @ read

0	During software reset
1	Not during soft ware reset

Acknowledge mode specification

0	Not generate clock pulse for acknowledge signal
1	Generate clock pulse for acknowledge signal

Number of bits transferred

<BC2 to BC0>	<ACK> = 0		<ACK> = 1	
	Number of clock pulses	Bits	Number of clock pulses	Bits
000	8	8	9	8
001	1	1	2	1
010	2	2	3	2
011	3	3	4	3
100	4	4	5	4
101	5	5	6	5
110	6	6	7	6
111	7	7	8	7

Note 1: Set the <BC2 to 0> to “000” before switching to a clock-synchronous 8-bit SIO mode.
Note 2: For the frequency of the SCL line clock, see 3.10.5 (3) Serial clock.

Figure 3.10.3 Registers for the I^2C Bus Mode

Serial Bus Interface Control Register 2

		7	6	5	4	3	2	1	0
SBI0CR2 (0243H)	Bit symbol	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
	Read/Write	W				W (Note 1)		W (Note 1)	
	After Reset	0	0	0	1	0	0	0	0
Prohibit Read-modify-write	Function	Master/Slave selection	Transmitter/ Receiver selection	Start/Stop generation	Cancel INTSBI interrupt request	Serial bus interface operating mode selection (note 2) 00: Port Mode 01: SIO Mode 10: I^{2}C Bus Mode 11: (reserved)		Software reset generate write "10" and "01", then an internal reset signal is generated.	

Serial bus interface operating mode selection (Note 2)

00	Port Mode (Serial Bus Interface output disabled)
01	Clocked Synchronous 8-Bit SIO Mode
10	I^{2}C Bus Mode
11	(reserved)

INTSBI interrupt request

0	−
1	Cancel interrupt request

Start/Stop generation

0	Generates the stop condition
1	Generates the start condition

Transmitter/Receiver selection

0	Receiver
1	Transmitter

Master/Slave selection

0	Slave
1	Master

Note1: Reading this register function as SBI0SR register.

Note2: Switch a mode to port mode after confirming that the bus is free.
Switch a mode between I^{2}C bus mode and clock-synchronous 8-bit SIO mode after confirming that input signals via port are high-level.

Figure 3.10.4 Registers for the I^{2}C Bus Mode

Serial Bus Interface Status Register

		7	6	5	4	3	2	1	0
SBI0SR (0243H)	bit Symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
	Read/Write	R							
	After reset	0	0	0	1	0	0	0	0
Prohibit Read- modify-write	Function	Master/ Slave status monitor	Transmitter/ Receiver status monitor	I^2C bus status monitor	INTSBI2 interrupt request monitor	Arbitration lost detection monitor 0: – 1: Detected	Slave address match detection monitor 0: – 1: Detected	GENERAL CALL detection monitor 0: – 1: Detected	Last received bit monitor 0: 0 1:1

Last received bit monitor

0	Last received bit was 0
1	Last received bit was 1

GENERAL CALL detection monitor

0	–
1	GENERAL CALL detected

Slave address match detection monitor

0	–
1	Slave address match or GENERAL CALL detected

Arbitration lost detection monitor

0	–
1	Arbitration lost

INTSBI interrupt request monitor

0	Interrupt requested
1	Interrupt canceled

I^2C bus status monitor

0	Free
1	Busy

Transmitter / Receiver status monitor

0	Receiver
1	Transmitter

Master / Slave status monitor

0	slave
1	Master

Note1: Writing in this register functions as SBI0CR2.

Figure 3.10.5 Registers for the I^2C Bus Mode

Serial Bus Interface Baud Rate Regster 0

SBI0BR0
(0244H)

	7	6	5	4	3	2	1	0
bit Symbol		I2SBI0						
Read/Write		R/W						
After Reset		0						
Function	Allways '0' write	IDLE2 0: Stop 1: Run						

Operation during IDLE 2 Mode

0	Stop
1	Operation

Serial Bus Interface Baud Rate Register 1

SBI0BR1
(0245H)

	7	6	5	4	3	2	1	0
Bit symbol	P4EN							
Read/Write	R/W							
After Reset	0							
Function	Internal clock 0: Stop 1: Operate							

Baud rate clock control

0	Stop
1	Operate

Sirial Bus Interface Data Buffer Register

SBI0DBR
(0241H)

Prohibit Read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (received)/W (transfer)							
After Reset	Undefined							

Note: When writing transmitted data, start from the MSB (bit 7).

I^{2}C Bus Address Register

I2C0AR
(0242H)

Prohibit Read-modify-write

	7	6	5	4	3	2	1	0
bit Symbol	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
Read/Write	W							
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

Address recognition mode specification

0	Slave address recognition
1	Non slave address recognition

Figure 3.10.6 Registers for the I^{2}C Bus Mode

3.10.5 Control in I^{2}C Bus Mode

(1) Acknowledge Mode Specification

Set the SBI0CR1<ACK> to 1 for operation in the acknowledge mode. The TMP91C824F generates an additional clock pulse for an Acknowledge signal when operating in Master Mode, it counts a clock pulse for an acknowledge signal when operating in the slave 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 TMP91C824F does not generate a clock pulse for the Acknowledge signal when operating in the Master Mode, and it does not count a clock pulse as an Acknowledge signal when operating in Slave Mode.

(2) Number of transfer bits

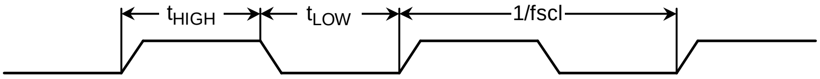
The SBI0CR1<BC2 to BC0> is used to select a number of bits for next transmitting and receiving data.

Since the <BC2 to BC0> is cleared to 000 as a start condition, a slave address and direction bit transmission are executed in 8 bits. Other than these, the <BC2 to 0> retains a specified value.

(3) Serial clock

① Clock source

The SBI0CR1 <SCK2 to SCK0> is used to select a maximum transfer frequency outputted on the SCL pin in Master Mode.



$t_{LOW} = 2^n/f_{SBI}$

$t_{HIGH} = 2^n/f_{SBI} + 8/f_{SBI}$

$fscl = 1/(t_{Low} + t_{HIGH})$

$= \dfrac{f_{SBI}}{2^n + 8}$

SBI0CR1 <SCK2~SCK0>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

Figure 3.10.7 Clock Source

② Clock synchronization

In the I^{2}C 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 TMP91C824F 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.

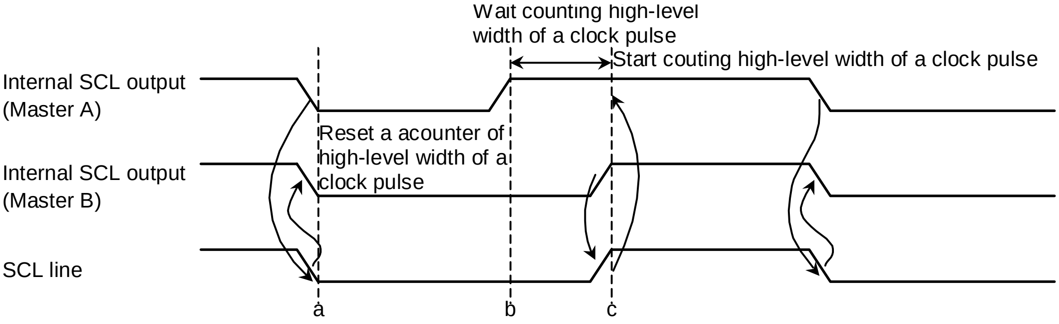

Wait counting high-level width of a clock pulse
Start couting high-level width of a clock pulse
Internal SCL output (Master A)
Reset a acounter of high-level width of a clock pulse
Internal SCL output (Master B)
SCL line
a
b
c

Figure 3.10.8 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 TMP91C824F is used as a slave device, set the slave address <SA6 to SA0> 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 TMP91C824F 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 TMP91C824F as a transmitter. Clear the <TRX> to "0" for operation as a receiver. When data with an addressing format is transferred in Slave Mode, when a slave address with the same value that an I2C0AR or a 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 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 it is "0". When an Acknowledge signal is not returned, the current condition is maintained.
The <TRX> is cleared to "0" by the hardware after a stop condition on the I^2C bus is detected or arbitration is lost.

(7) Start/Stop condition generation

When the SBI0SR<BB> is "0", 8-bit data 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.

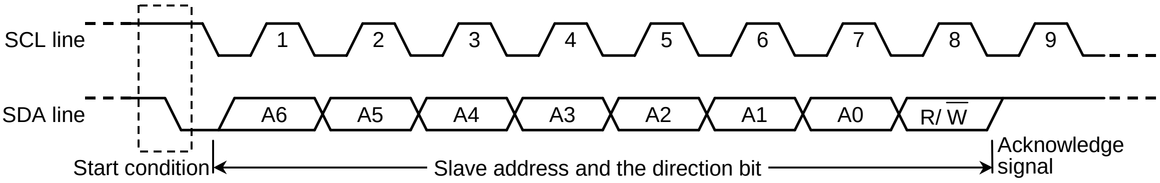


Figure 3.10.9 Start condition generation and slave address generation

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.

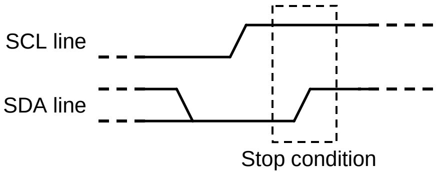


Figure 3.10.10 Stop condition generation

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.

And about generation of stop condition in master mode, there are some limitation point. Please refer to " 3.10.6(4) Stop condition generation ".

(8) Interrupt service requests and interrupt cancellation

When a serial bus interface interrupt request (INTS2) 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 (<ALS> = 0), <PIN> is cleared to "0" when the received slave address is the same as the value set at the I2C0AR or when a GENERAL CALL is received (all 8-bit data are "0" after a start condition). Although SBI0CR2<PIN> can be set to "1" by the program, the <PIN> is not clear it to "0" when it is written "0".

(9) Serial bus interface operation mode selection

SBI0CR2<SBIM1 to SBIM0> is used to specify the serial bus interface operation mode. Set SBI0CR2<SBIM1 to SBIM0> to "10" when the device is to be used in I^{2}C Bus Mode.

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^{2}C 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 I^{2}C 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 which 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.

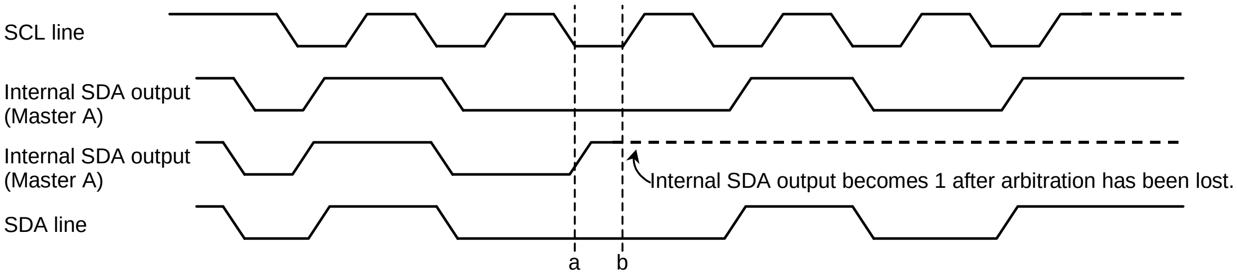


Figure 3.10.11 Arbitration Lost

The TMP91C824F 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 "00" and the mode is switched to Slave Receiver Mode.

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

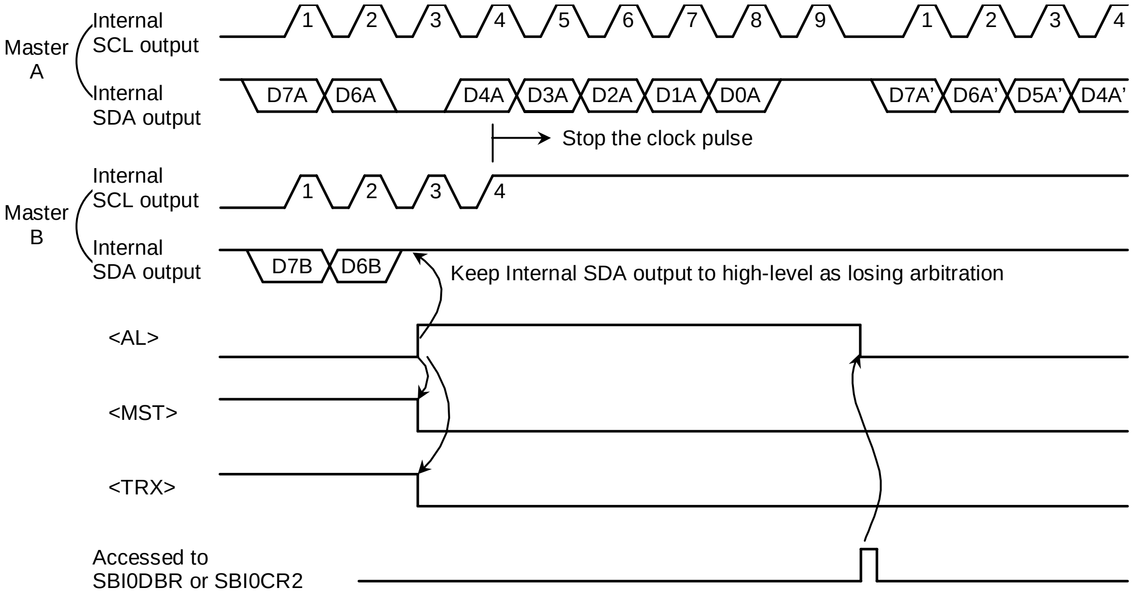


Figure 3.10.12 Example of when TMP91CW12 is a Master Device B
(D7A = D7B, D6A = D6B)

(11) Slave address match detection monitor

SBI0SR<AAS> is set to "1" in Slave Mode, in Address Recognition Mode (i.e. 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 INTS2 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,SWRST0> to “10” and “01”. This initializes the SBI circuit internally. All command registers and status registers are initialized as well.

SBI0CR2<SWRST1, SWRST0> is automatically cleared to “00” after the SBI circuit has been initialized.

(15) Serial Bus Interface Data Buffer Register (SBI0DBR)

The received data can be read and transferred data can be written by reading or writing the SBI0DBR.

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

(16) I^{2}CBUS Address Register (I2C0AR)

I2C0AR<SA6 to SA0> is used to set the slave address when the TMP91C824F functions as a slave device.

The slave address output 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 <ALS> = ”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. Therefor, setting <I2SBI0> is necessary before the HALT instruction is executed.

3.10.6 Data Transfer in I^{2}C Bus Mode

(1) Device initialization

Set the SBI0BR1<P4EN>, SBI0CR1<ACK,SCK2 to SCK0>, Set SBI0BR1 to "1" and clear bits 7 to 5 and 3 in the SBI0CR1 to "0".
Set a slave address <SA6 to SA0> 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 to SBIM 0>.

(2) Start condition and slave address generation

① 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 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.

② 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 INTS2 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".

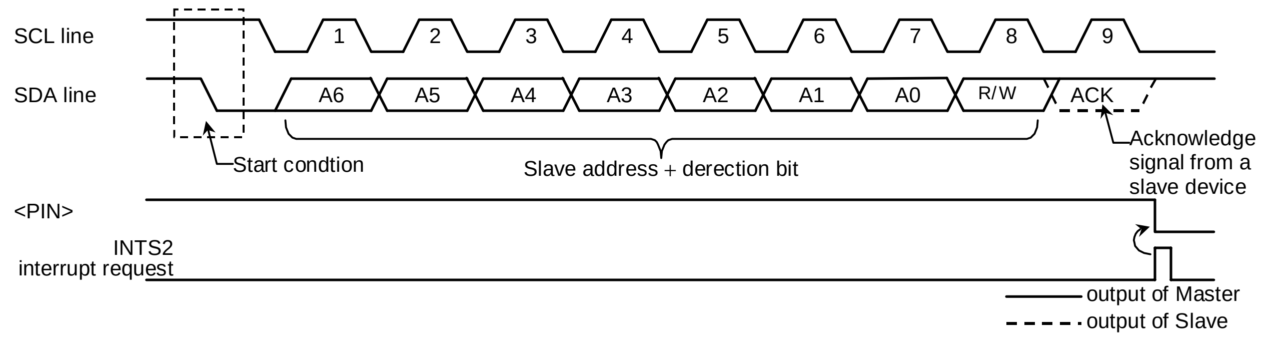


Figure 3.10.13 Start Condition Generation and Slave Address Transfer

(3) 1-word Data Transfer

Check the <MST> by the INTS2 interrupt process after the 1-word data transfer is completed, and determine whether the mode is a master or slave.

① 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 3.10.6 (4)) and terminate data transfer.
When the <LRB> is "0", the receiver is 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 to BC0> <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 INTS2 interrupt request occurs. 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.

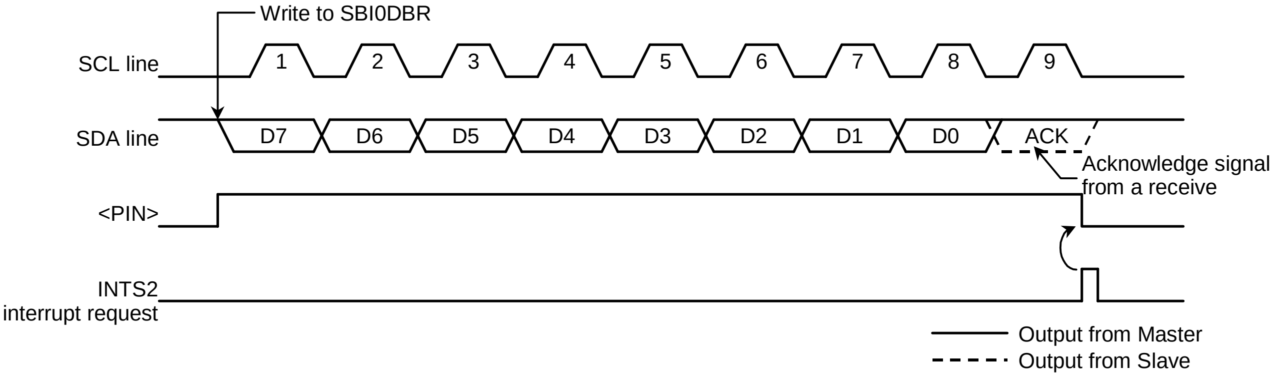


Figure 3.10.14 Example in which <BC2 to BC0> = "000" and <ACK> = "1" in Transmitter Mode

When the **<TRX>** is “0” (Receiver mode)

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 <BC2 to BC0> <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”. The TMP91C824F outputs a serial clock pulse to the SCL to transfer new 1-word of data and sets the SDA pin to “0”, When the acknowledge signal is set to Low-level at the final bit.

An INTS2 interrupt request then occurs and the <PIN> becomes “0”, Then the TMP91C824F pulls down the SCL pin to the Low-level. The TMP91C824F outputs a clock pulse for 1-word of data transfer and the acknowledge signal each time that received data is read from the SBI0DBR.

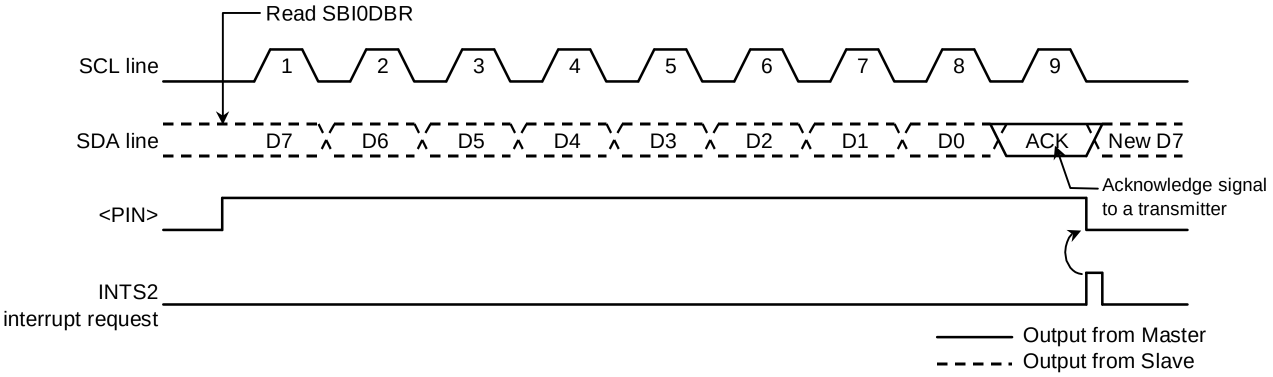


Figure 3.10.15 Example of when <BC2 to 0> = “000”, <ACK> = “1” in Receiver Mode

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 <BC2 to BC0> to “001” and read the data. The TMP91C824F 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 TMP91C824F generates a stop condition (see Section 3.10.6 (4)) and terminates data transfer.

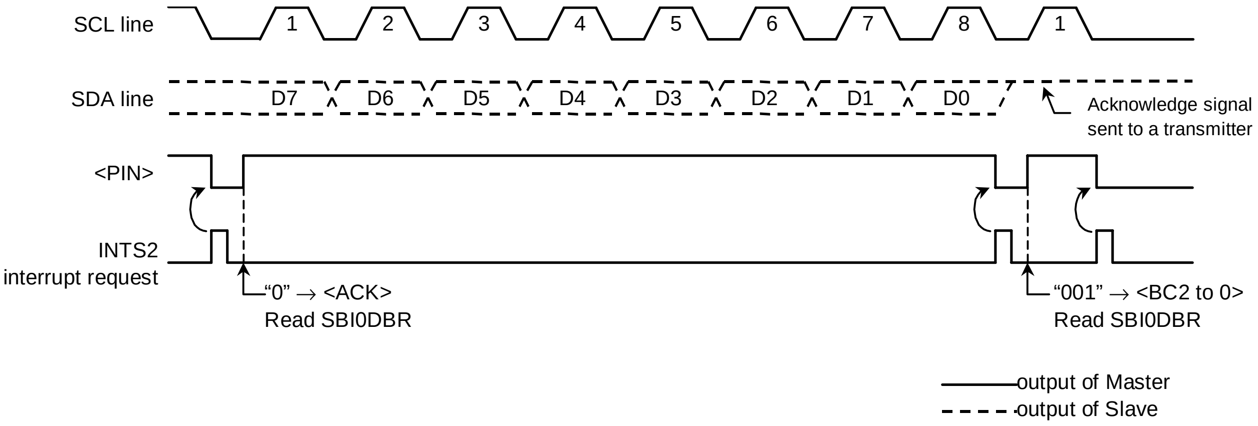


Figure 3.10.16 Termination of data Transfer in Master Receiver Mode

② If <MST> = 0 (Slave Mode)

In the slave mode, an INTS2 interrupt request occurs when the TMP91C824F 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 TMP91C824F operates in a slave mode if it losing arbitration. An INTS2 interrupt request occurs when a word data transfer terminates after losing arbitration. When an INTS2 interrupt request occurs 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 t_{LOW} time.

In the slave mode the TMP91C824F operates either in normal slave mode or in slave mode after losing arbitration.

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

Table 3.10.1 Operation in the Slave Mode

<TRX>	<AL>	<AAS>	<AD0>	Conditions	Process
1	1	1	0	The TMP91C824F 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 a word in <BC2 to BC0> and write the transmitted data to SBI0DBR
	0	1	0	In Salve Receiver Mode the TMP91C824F receives a slave address for which the value of the direction bit sent from the master is "1".	
		0	0	In Salve Transmitter Mode a single word of is transmitted. Set <BC2 to BC0> to the number of bits in a word.	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, cleat <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.
0	1	1	1/0	The TMP91C824F 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".
		0	0	The TMP91C824F loses arbitration when transmitting a slave address or data and terminates word data transfer.	
	0	1	1/0	In Slave Receiver Mode the TMP91C824F 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 TMP91C824F terminates receiving word data.	Set <BC2 to BC0> to the number of bits in a word and read the received data from SBI0DBR.

(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 TMP91CW12 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.

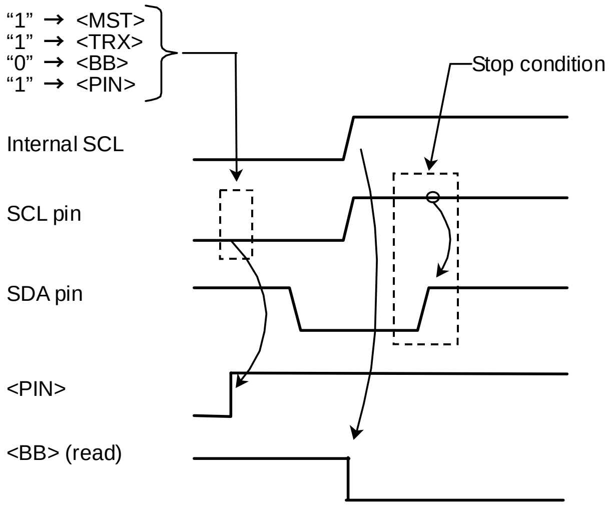


Figure 3.10.17 Stop condition generation (Single-master)

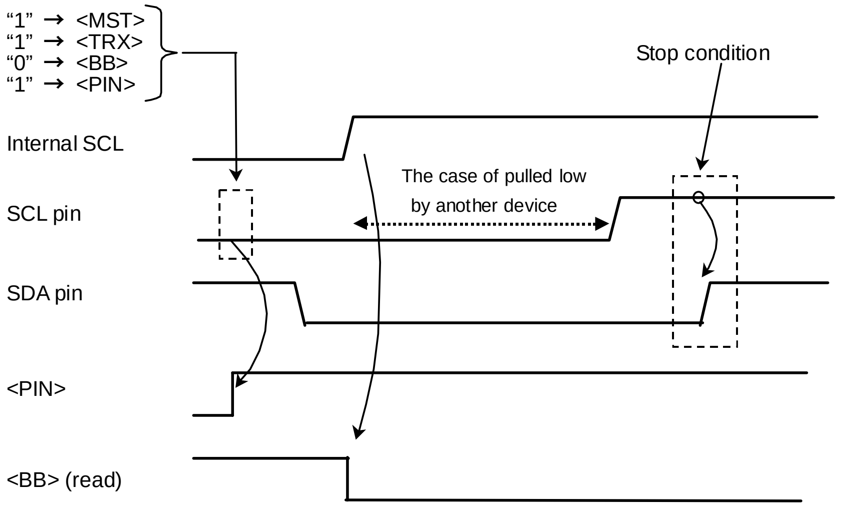


Figure 3.10.18 condition generation (Multi-master)

(5) Restart

Restart is used to change the direction of data transfer between a master device and a slave device during transferring data. The following explains how to restart when the TMP91C824F is in the master mode.
Clear "0" to the SBI0CR2<MST,TRX,BB>, and set "1" to the <PIN> and release the bus. The SDA line remains the High-level and the SCL pin is released. Since a stop condition is not generated on a bus, a bus is assumed to be in a busy state from other devices. Check the SBI0SR<BB> is "0" and SCL terminal level is "1" to check that the TMP91C824F is released. Check the <LRB> 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 a bus stays in a free state, generate a start condition with procedure 3.10.6 (2). In order to meet setup time 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.

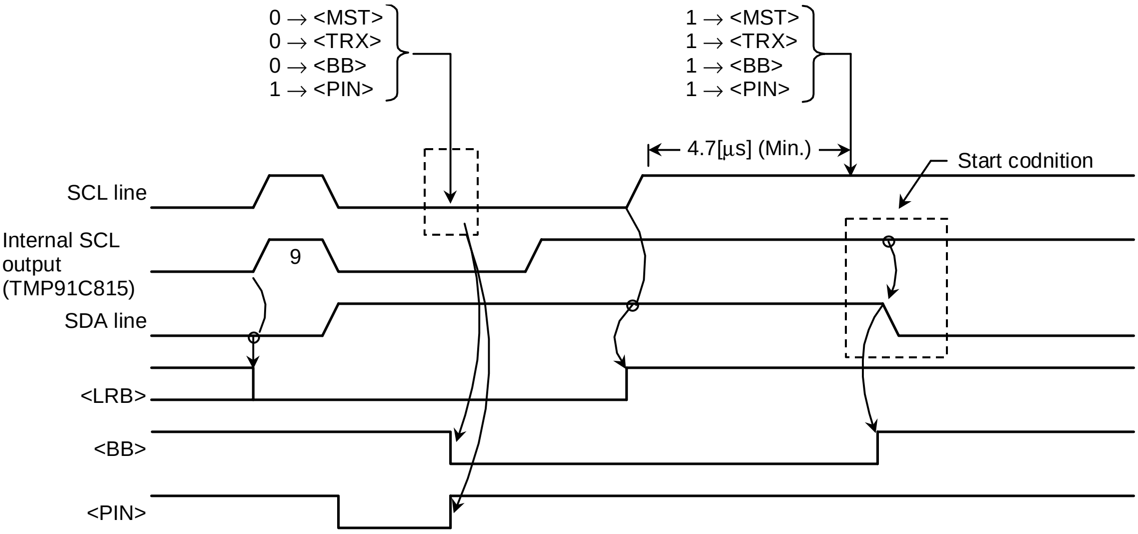


Figure 3.10.19 Timing diagram for TMP91C824F Restart

3.10.7 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.

Serial Bus Interface Control Register 1

SBI0CR1 (0240H)

Prohibit Read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	SIOS	SIOINH	SIOM1	SIOM0		SCK2	SCK1	SCK0/ SWRMON
Read/Write	W					W		R/W
After Reset	0	0	0	0		0	0	0
Function	Transfer start 0: stop 1: start	Continue/ abort transfer 0: Continue transfer 1: Abort transfer	Transfer mode select 00: Transmit Mode 01: (reserved) 10: Transmit/Receive Mode 11: Receive Mode			Serial clock selection and Reset monitor		

Serial clock selection <SCK2 to 0> @ write

000	n = 4	1 MHz
001	n = 5	500 kHz
010	n = 6	250 kHz
011	n = 7	125 kHz
100	n = 8	62.5 kHz
101	n = 9	31.25 kHz
110	n = 10	1.625 kHz
111	–	External mode

(System clcok: fc
Clock gear: fc/1
fc = 16 MHz
(output to SCK pin)
fscl = $\frac{fc}{2^n}$ [Hz])

(Input from SCK terminal)

Software Reset state monitor <SWRMON> @ read

0	Software Reset in progress
1	Software Reset not in progress

Transfer mode selection

00	8-Bit Transmit Mode
01	(reserved)
10	8-Bit Transmit / Received Mode
11	8-Bit Received Mode

Continue/Abort transfer

0	Continue transfer
1	Abort transfer (automatically cleared after transfer aborted)

Transfer start/stop

0	Stopped
1	Started

Note: Set the tranfer mode and the serial clock after setting <SIOS> to "0" and <SIOINH> to "1".

Serial Bus interface Data Buffer Register

SBI0DBR (0241H)

Prohibit Read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
Read/Write	R (receiver) / W (transfer)							
After Reset	Undefined							

Figure 3.10.20 Register for the SIO Mode

Serial Bus Interface Control Register 2

SBI0CR2
(0243H)

Prohibit Read-modify-write

		7	6	5	4	3	2	1	0
SBI0CR2 (0243H)	Bit symbol					SBIM1	SBIM0		
	Read/Write					W			
	After Reset					0	0		
Prohibit Read-modify-write	Function					Serial bus interface operation mode selection 00: Port mode 01: SIO mode 10: I^{2}C bus mode 11: (reserved)			

Serial bus interface operation mode selection

00	Port Mode (serial bus interface output disabled)
01	Clocked-Synchronous 8-Bit SIO Mode
10	I^{2}C Bus Mode
11	(reserved)

Note: Set the SBI0CR1<BC2 to 0> “000” before switching to a clocked-synchronous 8-bit SIO mode.

Serial Bus Interface Status Register

		7	6	5	4	3	2	1	0
SBI0SR (0243H)	bit Symbol					SIOF	SEF		
	Read/Write					R			
	After reset					0	0		
	Function					Serial transfer operation status monitor	Shift operation status monitor		

Shift operation status monitor

0	Shift operation terminated
1	Shift operation in progress

Serial transfer operating status monitor

0	Transfer terminated
1	Transfer in progress

Figure 3.10.21 Registers for the SIO Mode

Serial Bus Interface Baud Rate Register 0

		7	6	5	4	3	2	1	0
SBI0BR0 (0244H)	Bit symbol		I2SBI0						
	Read/Write		R/W						
	After Reset		0						
	Function	Allways '0' write	IDLE2 0: STOP 1: RUN						

Operation in IDLE 2 Mode

0	Stop
1	Operate

Serial Bus Interface Baud Rate Register 1

		7	6	5	4	3	2	1	0
SBI0BR1 (0245H)	Bit symbol	P4EN							
	Read/Write	R/W							
	After Reset	0							
	Function	Internal clock 0: Stop 1: Operate							

Baud rate clock control

0	Stop
1	Operate

Figure 3.10.22 Registers for the SIO Mode

(1) Serial Clock

① Clock source

SBI0CR1<SCK2 to SCK0> 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.

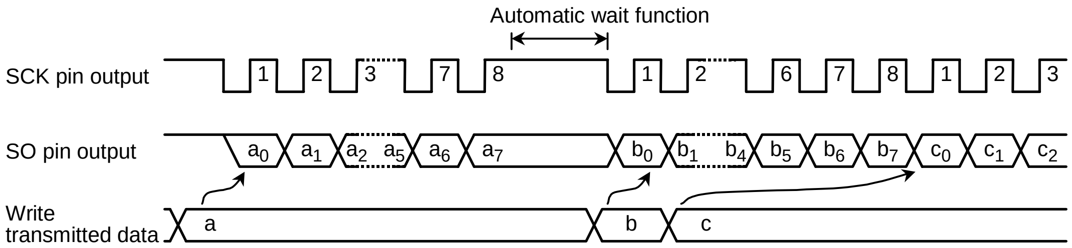


Figure 3.10.23 Automatic-wait Function

External clock (<SCK2 to SCK0> = "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).

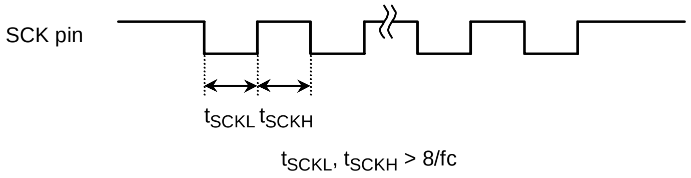


Figure 3.10.24 Maximum data transfer frequency when external clock input used

② Shift edge

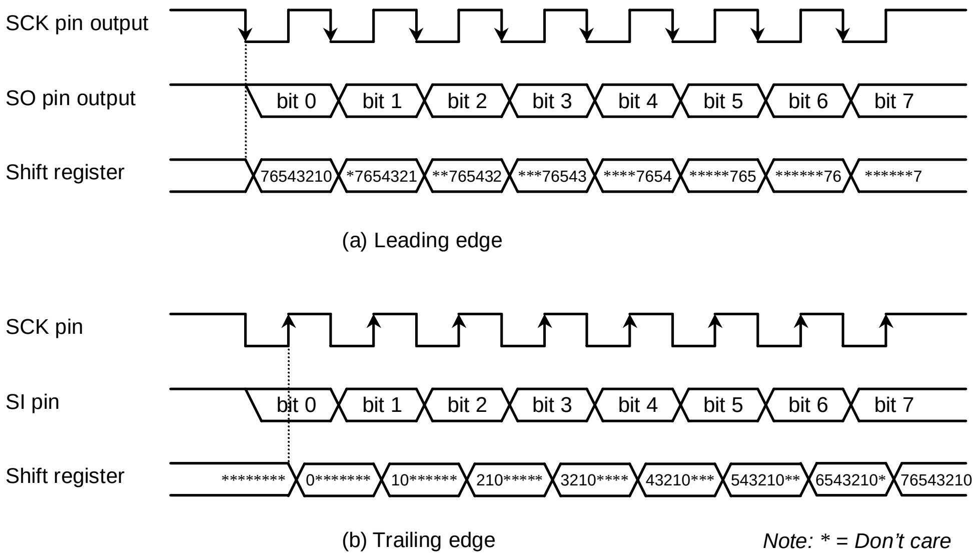
Data is transmitted on the leading edge of the clock and received on the trailing edge.

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).



(a) Leading edge

(b) Trailing edge

*Note: * = Don't care*

Figure 3.10.25 Shift edge

(2) Transfer modes

The SBI0CR1<SIOM1 to SIOM0> is used to select a transmit, receive or transmit / receive mode.

① 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 INTS2 (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.

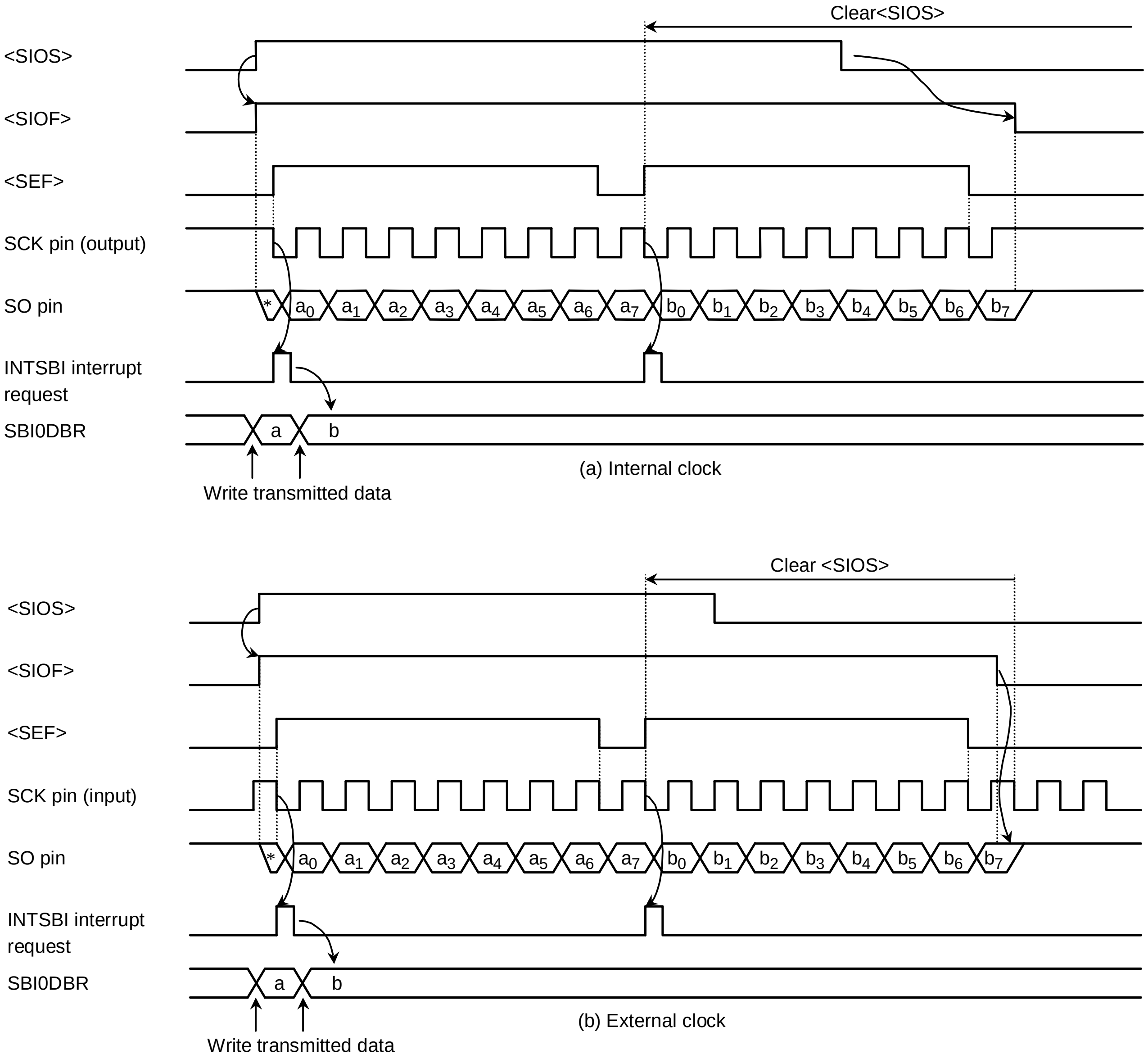
When the external clock is used, data should be written to SBI0DBR before new data is shifted. The transfer speed is determined by the maximum delay time between the time when an interrupt request is generated and the time when data is written to SBI0DBR by the interrupt service program.

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> (bit 3 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 SBI0SR<SIOS> to “0” before new data is shifted; otherwise, dummy data is transmitted and operation ends.

Example: Program to stop data transmission (when an external clock is used)



(a) Internal clock

(b) External clock

Figure 3.10.26 Transfer Mode

```
STEST1:  BIT      SEF, (SBI0SR)           ; If <SEF> = 1 then loop
         JR NZ,  STEST1
STEST2:  BIT      0, (P6)                 ; If SCK ]= 0 then loop
         JR Z, STEST2
         LD       (SBI0CR1), 00000111B    ; <SIOS> ← 0
```

② 8-Bit Receive Mode

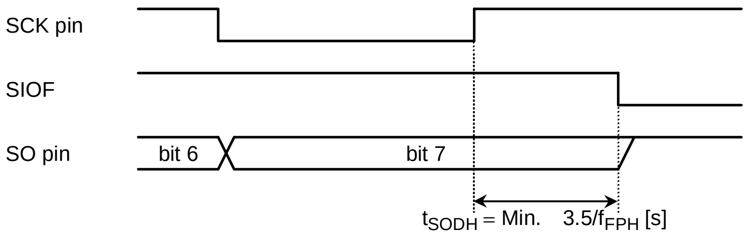


Figure 3.10.27 Transmitted data hold time at end of transmission

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 INTS2 (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 SBI0DBR 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 SBI0SR<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.

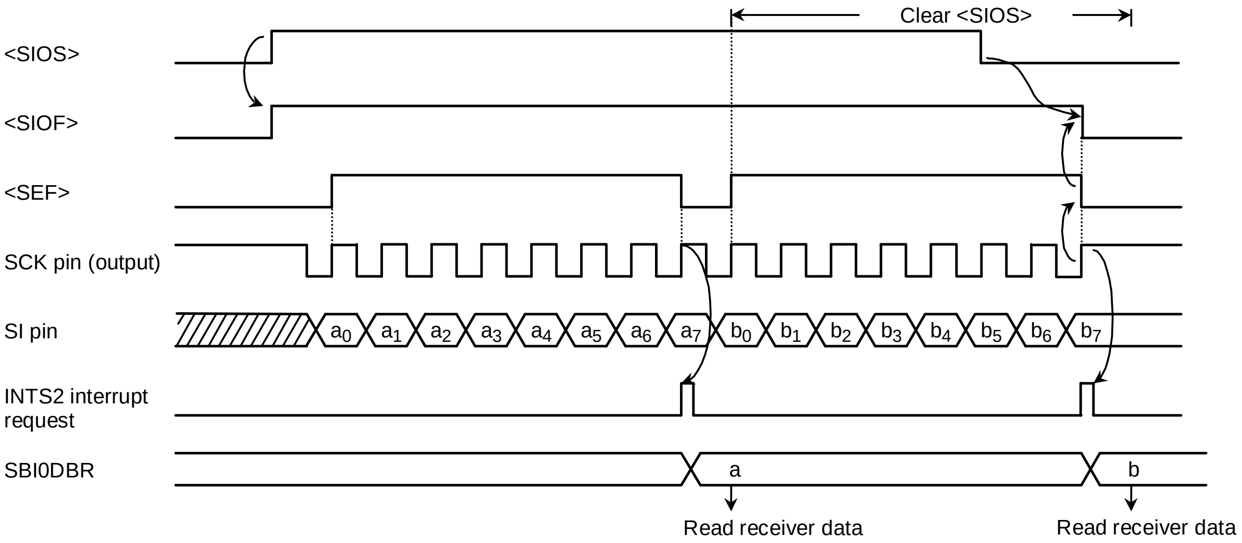


Figure 3.10.28 Receiver Mode (example: Internal clock)

③ 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 SBI0CR<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 INTS2 interrupt request is generated. The interrupt service program reads the received data from the data buffer register and writes the data which 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 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/receiving data ends when <SIOS> is cleared to "0" by the INTS2 interrupt service program or when SBI0CR1<SIOINH> 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 to be sensed. <SIOF> is set to "0" when transmitting/receiving has been completed. When <SIOINH> is set to 1, data transmitting/receiving stops. <SIOF> is then cleared to 0.

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.

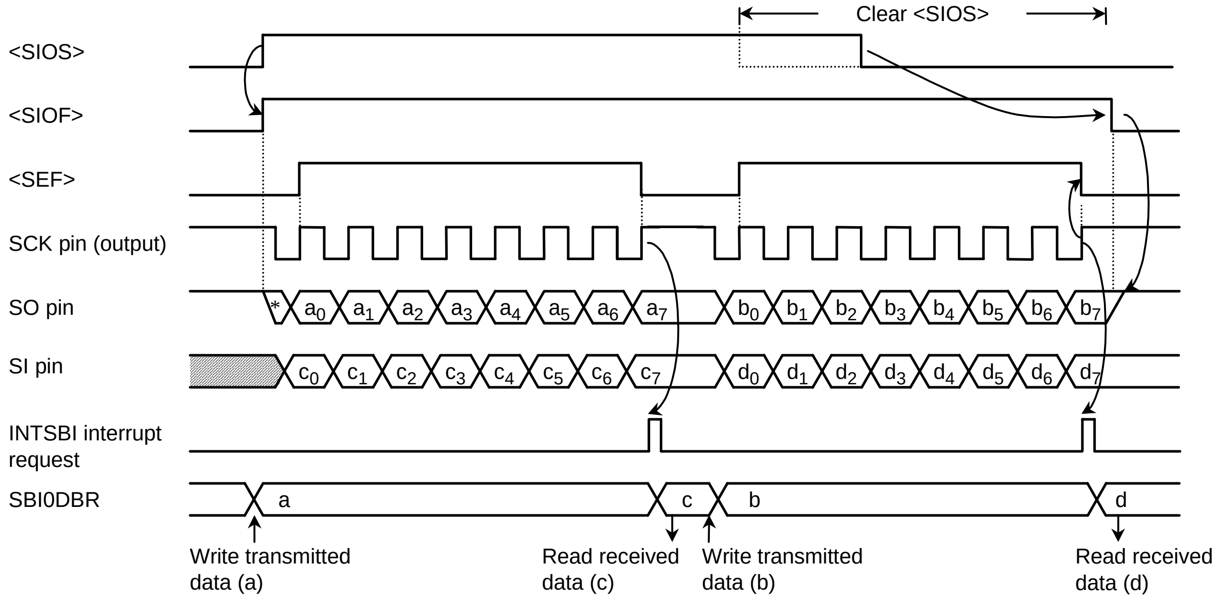

Clear <SIOS>
<SIOS>
<SIOF>
<SEF>
SCK pin (output)
SO pin
SI pin
INTSBI interrupt request
SBI0DBR
*
a_0 a_1 a_2 a_3 a_4 a_5 a_6 a_7
b_0 b_1 b_2 b_3 b_4 b_5 b_6 b_7
c_0 c_1 c_2 c_3 c_4 c_5 c_6 c_7
d_0 d_1 d_2 d_3 d_4 d_5 d_6 d_7
a
c
b
d
Write transmitted data (a)
Read received data (c)
Write transmitted data (b)
Read received data (d)

Figure 3.10.29 Transmit/Received Mode (example using internal clock)

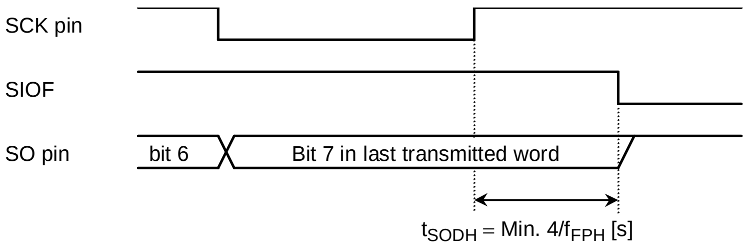

SCK pin
SIOF
SO pin
bit 6
Bit 7 in last transmitted word
t_{SODH} = Min. $4/f_{FPH}$ [s]

Figure 3.10.30 Transmitted data hold time at end of transmit/receive

3.11 Analog/Digital Converter

The TMP91C824 incorporates a 10-bit successive approximation-type analog/digital converter (A/D converter) with 8-channel analog input.

Figure 3.11.1 is a block diagram of the A/D converter. The 8-channel analog input pins (AN0 to AN7) are shared with the input-only port 8 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 timings the system may enter a stand-by mode even though the internal comparator is still enabled. Therefore be sure to check that A/D converter operations are halted before a HALT instruction is executed.

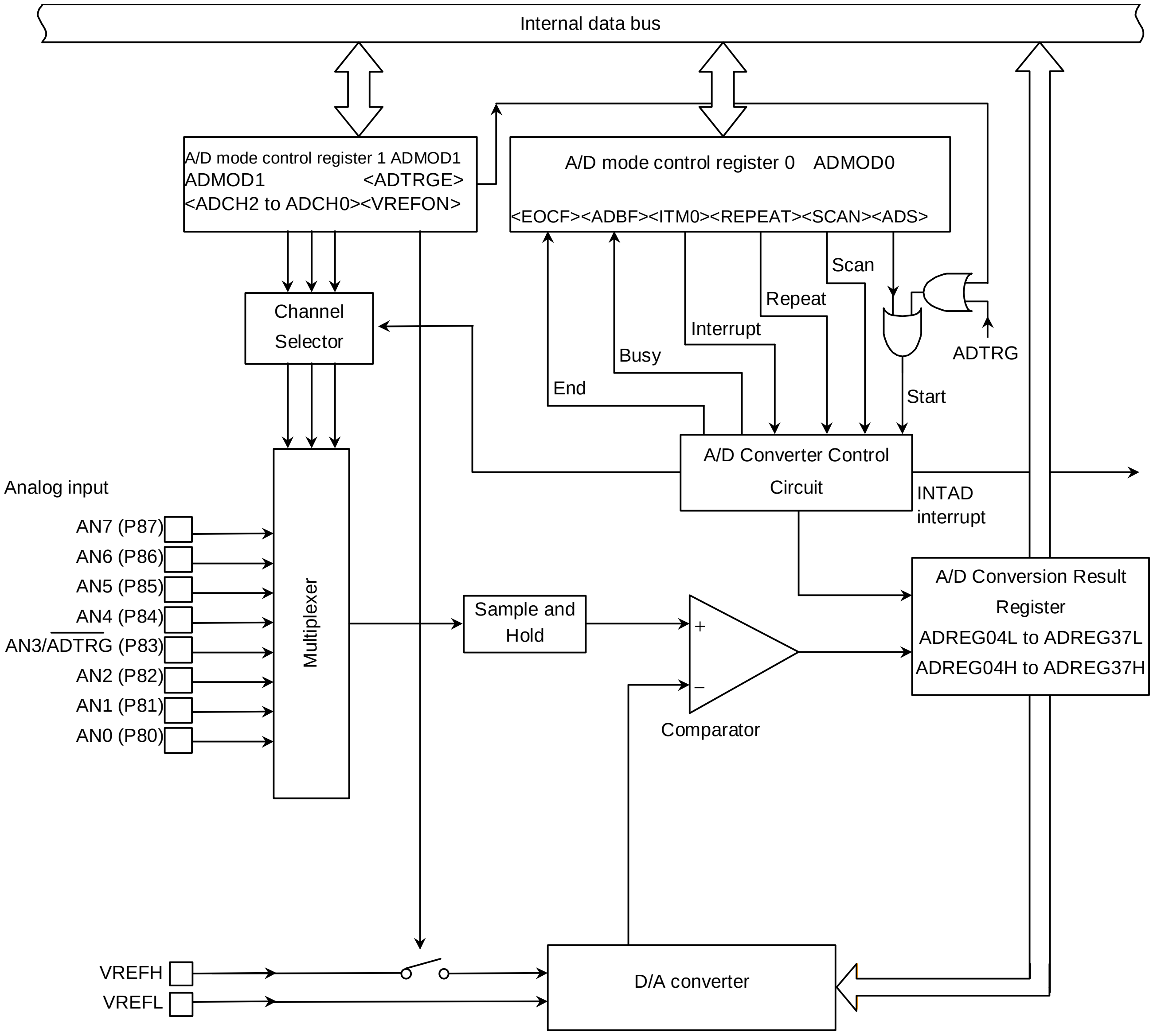


Figure 3.11.1 Block diagram of A/D converter

3.11.1 Analog/Digital converter registers

The A/D converter is controlled by the two A/D mode control registers: ADMOD0 and ADMOD1. The A/D conversion results are stored in 8 kinds of A/D conversion data Upper and Lower registers: ADREG04H/L, ADREG15H/L, ADREG26H/L and ADREG37H/L.

Figure 3.11.2 shows the registers related to the A/D converter.

A/D Mode Control Register 0

ADMOD0
(02B0H)

	7	6	5	4	3	2	1	0
Bit symbol	EOCF	ADBF	–	–	ITM0	REPEAT	SCAN	ADS
Read/Write	R		R/W					
After Reset	0	0	0	0	0	0	0	0
Function	A/D Conversion End flag 0: conversion in progress 1: conversion complete	A/D Conversion Busy flag 0: conversion stopped 1: Conversion in progress	Note: Always fixed to 0	Note: Always fixed to 0	Interrupt specification in conversion channel fixed repeat mode 0: every conversion 1: every fourth conversion	Repeat mode specification 0: Single Conversion 1: Repeat Conversion Mode	Scan mode specification 0: Conversion Channel Fixed Mode 1: Conversion Channel Scan Mode	A/D conversion start 0: Don’t care 1: start conversion Always 0 when read

A/D conversion start

0	Don’t care
1	Start A/D conversion

(note): Always read as 0.

A/D scan mode setting

0	A/D Conversion Channel Fixed Mode
1	A/D Conversion Channel Scan Mode

A/D repeat mode setting

0	A/D Single Conversion Mode
1	A/D Repeat Conversion Mode

Specify A/D conversion interrupt for Channel Fixed Repeat Conversion Mode

	Channel Fixed Repeat Conversion Mode <SCAN> = “0”, <REPEAT> = “1”
0	Generates interrupt every conversion.
1	Generates interrupt every fourth conversion.

A/D Conversion Busy flag

0	A/D conversion stopped
1	A/D conversion in progress

A/D Conversion End flag

0	Before or during A/D conversion
1	A/D conversion complete

Figure 3.11.2 A/D Converter Related Register

A/D Mode Control Register 1

ADMOD1
(02B1H)

	7	6	5	4	3	2	1	0
Bit symbol	VREFON	I2AD			ADTRGE	ADCH2	ADCH1	ADCH0
Read/Write	R/W	R/W			R/W			
After Reset	0	0			0	0	0	0
Function	VREF application control 0: OFF 1: ON	IDLE2 0: Stop 1: Operate			A/D external trigger start control 0: disable 1: enable	Analog input channel selection		

Analog input channel selection

<SCAN> <ADCH2, ADCH1, ADCH0>	0 (channel fixed)	1 (channel scanned)
000	AN0	AN0
001	AN1	AN0 → AN1
010	AN2	AN0 → AN1 → AN2
011(Note)	AN3	AN0 → AN1 → AN2 → AN3
100	AN4	AN4
101	AN5	AN4 → AN5
110	AN6	AN4 → AN5 → AN6
111	AN7	AN4 → AN5 → AN6 → AN7

A/D conversion start control by external trigger ($\overline{\text{ADTRG}}$ input)

0	Disabled
1	Enabled

IDLE2 control

0	Stopped
1	In operation

Control of application of reference voltage to A/D converter

0	OFF
1	ON

Before starting conversion (before writing “1” to ADMOD0 <ADS>), set the <VREFON> bit to “1”.

(note): As pin AN3 also functions as the $\overline{\text{ADTRG}}$ input pin, do not set < ADCH2: 0> = “011” when using $\overline{\text{ADTRG}}$ with < ADTRGE> =“0”.

Figure 3.11.3 A/D Converter related registers

A/D Conversion Data Low Register 0/4

		7	6	5	4	3	2	1	0
ADREG04L (02A0H)	Bit symbol	ADR01	ADR00						ADR0RF
	Read/Write	R							R
	After Reset	Undefined							0
	Function	Stores lower 2-bits of A/D conversion result							A/D Conversion Data Storage flag 1: Conversion result stored

A/D Conversion Data Upper Register 0/4

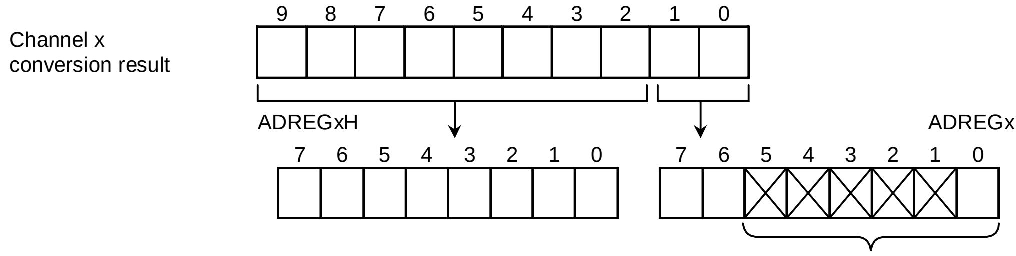
		7	6	5	4	3	2	1	0
ADREG04H (02A1H)	Bit symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
	Read/Write	R							
	After Reset	Undefined							
	Function	Stores upper 8-bits A/D conversion result.							

A/D Conversion Data Lower Register 1/5

		7	6	5	4	3	2	1	0
ADREG15L (02A2H)	Bit symbol	ADR11	ADR10						ADR1RF
	Read/Write	R							R
	After Reset	Undefined							0
	Function	stores lower 2-bits of A/D conversion result							A/D Conversion Result flag 1: Conversion result stored

A/D Conversion Data Upper Register 1/5

		7	6	5	4	3	2	1	0
ADREG15H (02A3H)	Bit symbol	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
	Read/Write	R							
	After Reset	Undefined							
	Function	Stores upper 8-bits of A/D conversion result.							



- Bits 5-1 are always read as “1”.
- Bit 0 is the A/D conversion data storage flag <ADRxRF>. When the A/D conversion result is stored, the flag is set to “1”. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to “0”.

Figure 3.11.4 A/D Converter related registers

A/D Conversion Result Lower Register 2/6

		7	6	5	4	3	2	1	0
ADREG26L (02A4H)	Bit symbol	ADR21	ADR20						ADR2RF
	Read/Write	R							R
	After Reset	Undefined							0
	Function	Stores lower 2-bits of A/D conversion result.							A/D conversion data storage flag 1: Conversion result stored

A/D Conversion Data upper Register 2/6

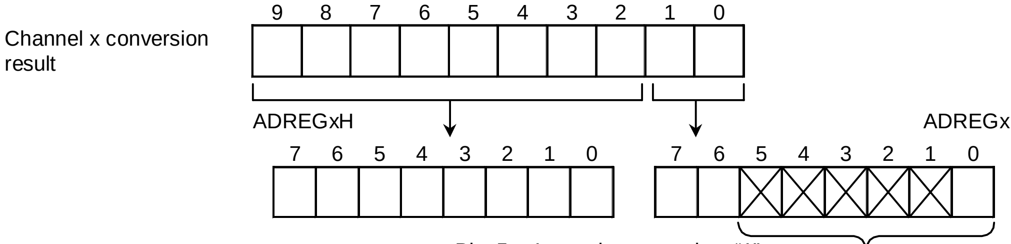
		7	6	5	4	3	2	1	0
ADREG26H (02A5H)	Bit symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
	Read/Write	R							
	After Reset	Undefined							
	Function	Stores upper 8-bits of A/D conversion result.							

A/D Conversion Data Lower Register 3/7

		7	6	5	4	3	2	1	0
ADREG37L (02A6H)	Bit symbol	ADR31	ADR30						ADR3RF
	Read/Write	R							R
	After Reset	Undefined							0
	Function	Stores lower 2-bits of AD conversion result.							AD Conversion Data Storage flag 1: conversion result stored

A/D Conversion Result Upper Register 3/7

		7	6	5	4	3	2	1	0
ADREG37H (02A7H)	Bit symbol	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
	Read/Write	R							
	After Reset	Undefined							
	Function	Stores upper 8-bits of A/D conversion result.							



- Bits 5 to1 are always read as “1”.
- Bit 0 is the A/D conversion data storage flag <ADRxRF>. When the A/D conversion result is stored, the flag is set to “1”. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to “0”.

Figure 3.11.5 A/D Converter related registers

3.11.2 Description of operation

(1) Analog reference voltage

A High-level analog reference voltage is applied to the VREFH pin; a low-level analog reference voltage is applied to the VREFL pin. To perform A/D conversion, the reference voltage as the difference between VREFH and VREFL, 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 VREFH and VREFL, write “0” to ADMOD1<VREFON> in A/D Mode Control Register 1. To start A/D conversion in the OFF state, first write “1” to ADMOD1<VREFON>, wait 3 μs 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 A/D converter.

- In Analog Input Channel Fixed Mode (A/D MOD0<SCAN> = “0”)
  Setting ADMOD1<ADCH2 to ADCH0> selects one of the input pins AN0 to AN7 as the input channel.
- In Analog Input Channel Scan Mode (ADMOD0<SCAN> = “1”)
  Setting ADMOD1<ADCH2 to ADCH0> selects one of the 8 scan modes.

Table 3.11.1 illustrates analog input channel selection in each operation mode.

After Reset, ADMOD0<SCAN> = ”0” and ADMOD1<ADCH2 to ADCH0> = ”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.

Table 3.11.1 Analog input channel selection

<ADCH2~0>	Channel fixed <SCAN> = “0”	Channel scan <SCAN> = “1”
000	AN0	AN0
001	AN1	AN0 → AN1
010	AN2	AN0 → AN1 → AN2
011	AN3	AN0 → AN1 → AN2 → AN3
100	AN4	AN4
101	AN5	AN4 → AN5
110	AN6	AN4 → AN5 → AN6
111	AN7	AN4 → AN5 → AN6 → AN7

(3) Starting A/D Conversion

To start A/D conversion, write “1” to ADMOD0<ADS> in A/D Mode Control Register 0, or ADMOD1<ADTRGE> in A/D Mode Control Register 1 and input falling edge on $\overline{\text{ADTRG}}$ pin. When A/D conversion starts, the A/D Conversion Busy flag ADMOD0<ADBF> will be set to “1”, indicating that A/D conversion is in progress.
Writing “1” to ADMOD0<ADS> during A/D conversion restarts conversion. At that time, to determine whether the A/D conversion results have been preserved, check the value of the conversion data storage flag ADREGxL<ADRxRF>.
During A/D conversion, a falling edge input on the $\overline{\text{ADTRG}}$ pin will be ignored.

(4) A/D conversion modes and the A/D Conversion End interrupt

The 4 A/D conversion modes are:

- Channel Fixed Single Conversion Mode
- Channel Scan Single Conversion Mode
- Channel Fixed Repeat Conversion Mode
- Channel Scan Repeat Conversion Mode

The ADMOD0<REPET> and ADMOD0<SCAN> settings in A/D Mode Control Register 0 determine the A/D mode setting.

Completion of A/D conversion triggers an INTAD A/D Conversion End interrupt request. Also, ADMOD0<EOCF> will be set to “1” to indicate that A/D conversion has been completed.

① Channel Fixed Single Conversion Mode

Setting ADMOD0<REPET> 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.

② Channel Scan Single Conversion Mode

Setting ADMOD0<REPET> 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.

③ Channel Fixed Repeat Conversion Mode

Setting ADMOD0<REPET> 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, ADMOD0<EOCF> is set to “1” and ADMOD0<ADBF> is not cleared to “0” but held “1”. INTAD interrupt request generation timing is determined by the setting of ADMOD0<ITM0>.

Setting <ITM0> to “0” generates an interrupt request every time an A/D conversion is completed.

Setting <ITM0> to “1” generates an interrupt request on completion of every fourth conversion.

④ Channel Scan Repeat Conversion Mode

Setting ADMOD0<REPET> 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 (i.e. in cases ③ and ④), write a “0” to ADMOD0<REPET>. 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 A/D converter even when A/D conversion is still in progress. In repeat conversion modes (i.e. in cases ③ and ④), when the halt is released, conversion restarts from the beginning. In single conversion modes (i.e. in cases ① and ②), conversion does not restart when the halt is released (the converter remains stopped).

Table 3.11.2 shows the relationship between the A/D conversion modes and interrupt requests .

Table 3.11.2 Relationship between A/D Conversion modes and Interrupt requests

Mode	Interrupt Request Generation	ADMOD0		
		<ITM0>	<REPEAT>	<SCAN>
Channel Fixed Single Conversion Mode	After completion of conversion	X	0	0
Channel Scan Single Conversion Mode	After completion of scan conversion	X	0	1
Channel Fixed Repeat Conversion Mode	Every conversion	0	1	0
	Every forth conversion	1		
Channel Scan Repeat Conversion Mode	After completion of every scan conversion	X	1	1

X: Don’t care

(5) A/D conversion time

84 states (10.5 μs @ f_{FPH} = 16MHz) are required for the A/D conversion for one channel.

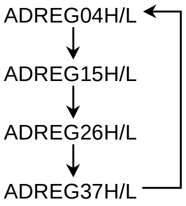
(6) Storing and reading the results of A/D conversion

The A/D Conversion Data Upper and Lower Registers (ADREG04H/L to ADREG37H/L) store the A/D conversion results. (ADREG04H/L to ADRG37H/L are read-only registers.)

In Channel Fixed Repeat Conversion Mode, the conversion results are stored successively in registers ADREG04H/L to ADRG37H/L. In other modes, the AN0 and AN4, AN1 and AN5, AN2 and AN6, and AN3 and AN7 conversion results are stored in ADREG04H/L, ADREG15H/L, ADREG26H/L and ADREG37H/L respectively.

Table 3.11.3 shows the correspondence between the analog input channels and the registers which are used to hold the results of A/D conversion.

Table 3.11.3 Correspondence Between Analog Input Channels and A/D Conversion Result Registers

Analog input channel (Port A)	A/D Conversion Result Register	
	Conversion modes other than at right	Channel fixed repeat conversion mode (every 4th conversion)
AN0	ADREG04H/L	 ADREG04H/L ADREG15H/L ADREG26H/L ADREG37H/L
AN1	ADREG15H/L	
AN2	ADREG26H/L	
AN3	ADREG37H/L	
AN4	ADREG04H/L	
AN5	ADREG15H/L	
AN6	ADREG26H/L	
AN7	ADREG37H/L	

<ADRxRF>, bit “0” of the A/D conversion data lower register, is used as the A/D conversion data storage flag. The storage flag indicates whether the A/D conversion result register has been read or not. When a conversion result is stored in the A/D conversion result register, the flag is set to “1”. When either of the A/D conversion result registers (ADREGxH or ADREGxL) is read, the flag is cleared to “0”.

Reading the A/D conversion result also clears the A/D Conversion End flag ADMOD0<EOCF> to “0”.

Setting example:

① Convert the analog input voltage on the AN3 pin and write the result, to memory address 0800H using the A/D interrupt (INTAD) processing routine.

Main routine:

		7	6	5	4	3	2	1	0	
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 be the analog input channel.
ADMOD0	←	X	X	0	0	0	0	0	1	Start conversion in Channel Fixed Single Conversion Mode.

Interrupt routine processing example:

WA	← ADREG37	Read value of ADREG37L and ADREG37H into 16-bit general-purpose register WA.
WA	> > 6	Shift contents read into WA six times to right and zero-fill upper bits.
(0800H)	← WA	Write contents of WA to memory address 0800H.

② This example repeatedly converts the analog input voltages on the three pins AN0, AN1 and AN2, using Channel Scan Repeat Conversion Mode.

INTE0AD	←	X	0	0	0	-	-	-	-	Disable INTAD.
ADMOD1	←	1	X	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.

(note): X = Don’t care; “–” = No change

3.12 Watch Dog Timer (runaway detection timer)

The TMP91C824 features a watch dog timer for detecting runaway.

The watch dog 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 watch dog timer detects a malfunction, it generates a non-maskable interrupt INTWD to notify the CPU. Connecting the watch dog timer output to the Reset pin internally forces a reset.

3.12.1 Configuration

Figure 3.12.1 is a block diagram of he watchdog timer (WDT).

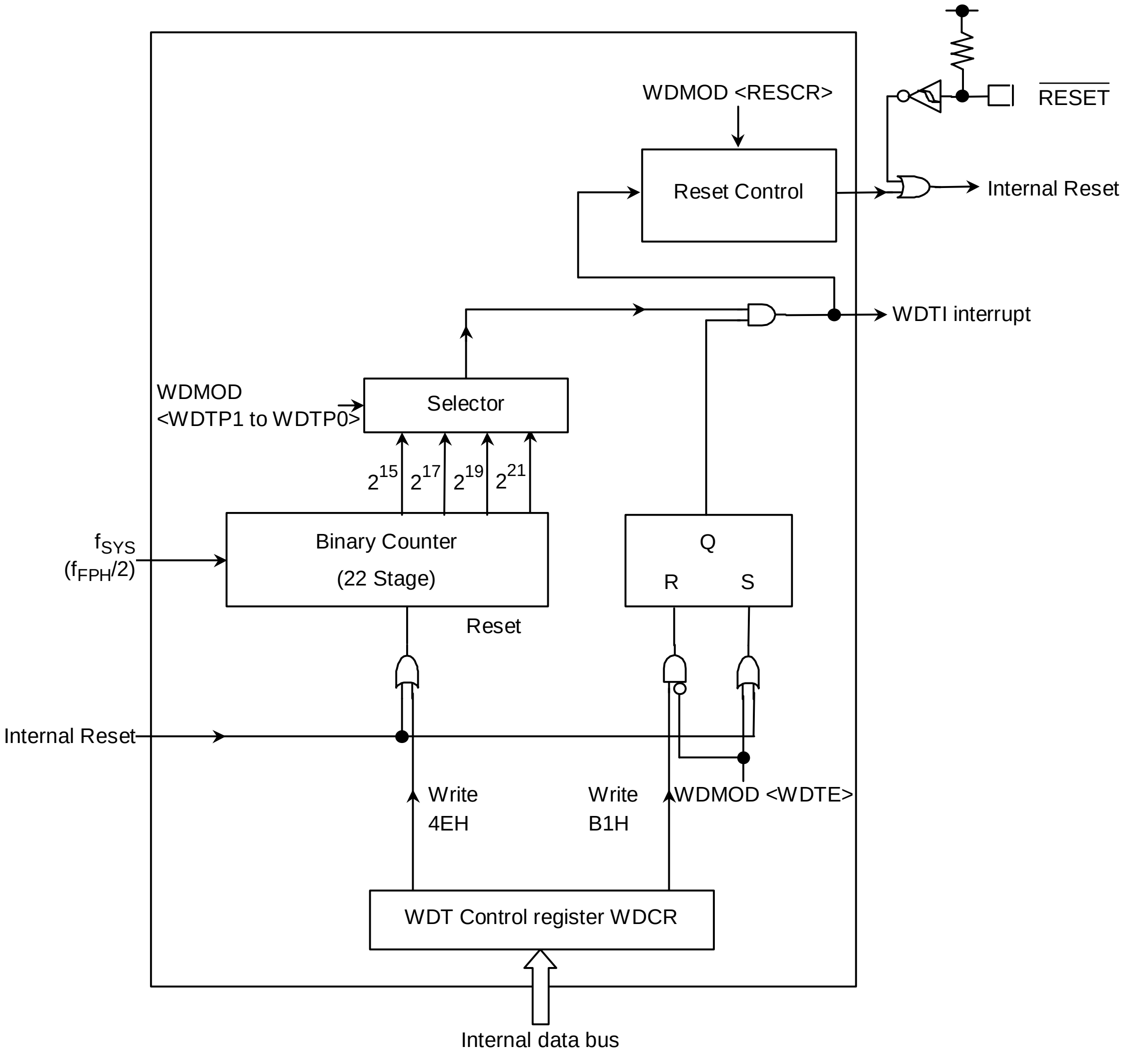


Figure 3.12.1 Block diagram of watch dog timer

The watch dog timer consists of a 22-stage binary counter which uses the system clock (f_{SYS}) as the input clock. The binary counter can output f_{SYS}/215, f_{SYS}/217, f_{SYS}/219 and f_{SYS}/221. Selecting one of the outputs using WDMOD<WDTP1,WDTP0> generates a Watchdog interrupt and outputs watchdog timer out when an overflow occurs as shown in Figure 3.12.2.

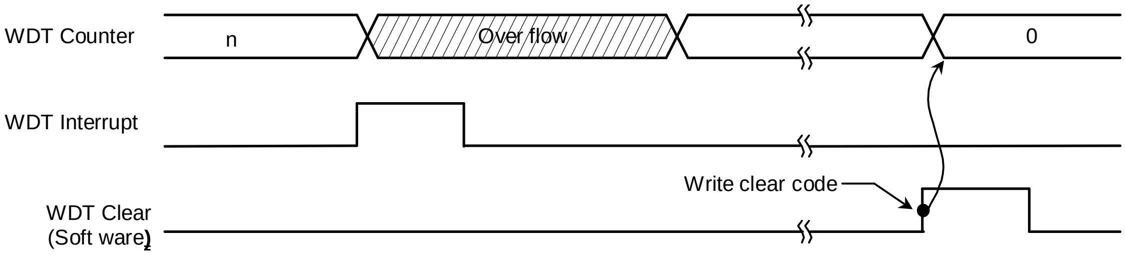


Figure 3.12.2 Normal mode

The runaway detection result can also be connected to the Reset pin internally.
In this case, the reset time will be between 22 and 29 states as shown in Figure 3.12.3.

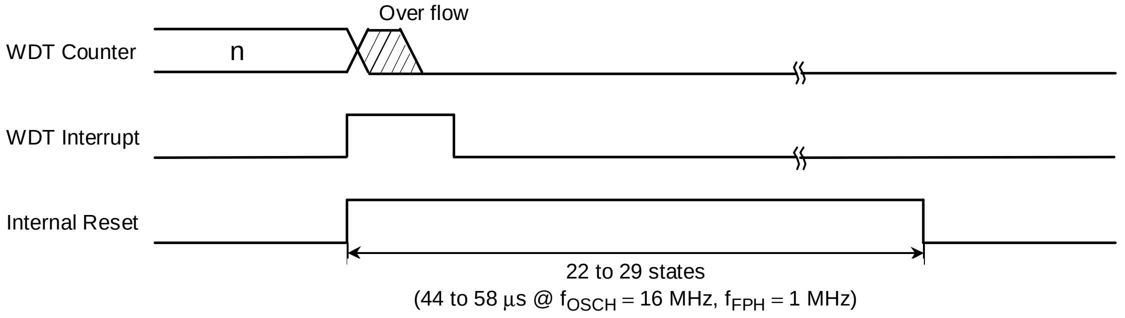


Figure 3.12.3 Reset mode

3.12.2 Control registers

The watchdog timer WDT is controlled by two control registers WDMOD and WDCR.

(1) Watch dog timer Mode Register (WDMOD)

① Setting the detection time for the watch dog timer in <WDTP1,WDTP0>

This 2-bit register is used for setting the watch dog timer interrupt time used when detecting runaway. After Reset, this register is initialized to WDMOD<WDTP1,WDTP0> = “00”.
The detection times for WDT are shown in Figure 3.12.4.

② Watch dog timer Enable/Disable Control Register <WDTE>

After Reset, WDMOD<WDTE> is initialized to “1”, enabling the watch dog timer.
To disable the watch dog timer, it is necessary to set this bit to “0” and to write the disable code (B1H) to the watch dog timer Control Register WDCR. This makes it difficult for the watch dog timer to be disabled by runaway.
However, it is possible to return the watch dog timer from the disabled state to the enabled state merely by setting <WDTE> to “1”.

③ Watch dog timer out reset connection <RESCR>

This register is used to connect the output of the watch dog timer with the RESET terminal internally. Since WDMOD<RESCR>is initialized to “0” on Reset, a Reset by the watch dog timer will not be performed.

(2) Watch dog timer Control Register (WDCR)

This register is used to disable and clear the binary counter for the watch dog timer.

Disable control the watch dog timer can be disabled by clearing WDMOD<WDTE> to “0” and then writing the disable code (B1H) to the WDCR register.

```
WDMOD    ←  0  -  -  -  -  -  X  X      Clear WDMOD<WDTE>to “0”.
WDCR     ←  1  0  1  1  0  0  0  1      Write the disable code (B1H).
```

- Enable control

Set WDMOD<WDTE>to “1”.

- Watch dog timer clear control

To clear the binary counter and cause counting to resume, write the clear code (4EH) to the WDCR register.

```
WDCR     ←  0  1  0  0  1  1  1  0      Write the clear code (4EH).
```

WDMOD
(0300H)

	7	6	5	4	3	2	1	0
Bit symbol	WDTE	WDTP1	WDTP0			I2WDT	RESCR	–
Read/Write	R/W	R/W				R/W		R/W
After Reset	1	0	0			0	0	0
Function	WDT control 1: enable	Select detecting time 00: $2^{15}/f_{SYS}$ 01: $2^{17}/f_{SYS}$ 10: $2^{19}/f_{SYS}$ 11: $2^{21}/f_{SYS}$				IDLE2 0: Stop 1: Operate	1: Internally connects WDL out to the reset pin	Always write “0”

Watch dog timer out control

0	—
1	Connects WDT out to a reset

IDLE2 Control

0	Stop
1	Operation

Watch dog timer detection time @ fc = 16 MHz, fs = 32.768 kHz

SYSCR1 System Clcok Selection <SYSCK>	SYSCR1 Gear Value <GEAR2 to GEAR0>	Watch dog timer Detection Time			
		WDMOD<WDTP1 to WDTP0>			
		00	01	10	11
1 (fs)	XXX	2.0 s	8.0 s	32.0 s	128.0 s
0 (fc)	000 (fc)	4.096 ms	16.384 ms	65.536 ms	262.144 ms
	001 (fc/$_{2}$)	8.192 ms	32.768 ms	131.072 ms	524.288 ms
	010 (fc/$_{4}$)	16.384 ms	65.536 ms	262.144 ms	1.049 s
	011 (fc/$_{8}$)	32.768 ms	131.072 ms	524.288 ms	2.097 s
	100 (fc/$_{16}$)	65.536 ms	262.144 ms	1.049 s	4.194 s

Watch dog timer Enable/Disable control

0	Disabled
1	Enabled

Figure 3.12.4 Watch dog timer mode register

		7	6	5	4	3	2	1	0
WDCR (0301H)	Bit symbol	–							
	Read/Write	W							
	After reset	–							
	Function	B1H: WDT disable code 4EH: WDT clear code							

Disable/Clear WDT

B1H	Disable code
4EH	Clear code
Others	Don’t care

Figure 3.12.5 Watch dog timer control register

3.12.3 Operation

The watch dog timer generates an INTWD interrupt when the detection time set in the WDMOD<WDTP1,WDTP0> has elapsed. The watch dog timer must be cleared "0" by software before an INTWD interrupt will be generated. If the CPU malfunctions (i.e. if runaway occurs) due to causes such as noise, but 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 to the CPU to normal operation by means of an anti-malfunction program.

The watch dog timer works immediately after reset.

The watch dog timer does not operate in IDLE1 or STOP mode,
as the binary counter continues counting during bus release (When $\overline{\text{BUSAK}}$ goes Low).

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: ① Clear the binary counter.

```
WDCR   ←  0 1 0 0 1 1 1 0     Write the clear code (4EH).
```

② Set the watchdog timer detection time to 2^{17} / f_{SYS}.

```
WDMOD ←  1 0 1 - - - X X
```

③ Disable the watchdog timer.

```
WDMOD ←  0 - - - - - X X     Clear WDTE to "0".
WDCR   ←  1 0 1 1 0 0 0 1     Write disable code (B1H).
```

3.13 Real time clock (RTC)

3.13.1 Function description for RTC

1) Clock function (hour , minute , second)
2) Calendar function (month and day , day of the week , and leap year)
3) 24 or 12-hour (AM/PM) clock function
4) ± 30 second adjustment function (by software)
5) Alarm function (Alarm output)
6) Alarm interrupt generate

3.13.2 Block diagram

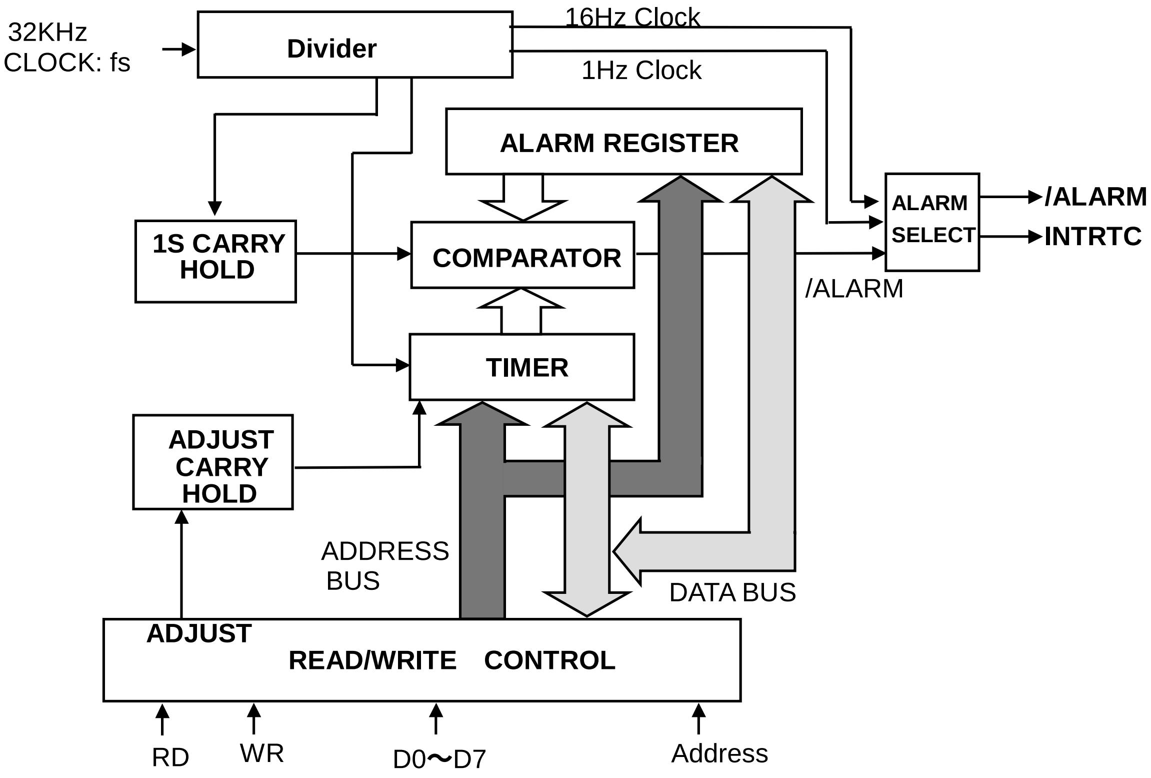


Figure 3.13.1 RTC block diagram

(note1) The Christian era year column:
This product has year column toward only lower two columns. Therefore the next year in 99 works as 00 years. In system to use it, please manage upper two columns with the system side when handle year column in the Christian era.

(note2) Leap year:
A leap year is the year, which is divisible with 4, but the year, which there is exception, and is divisible with 100 is not a leap year. However, the year, which is divisible with 400, is a leap year. But there is not this product for the correspondence to the above exception. Because there are only with the year which is divisible with 4 as a leap year, please cope with the system side if this function is problem.

3.13.1 Control registers

Table 3.13.1 PAGE 0 (Timer function) registers

Symbol	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Function	Read /Write
SECR	0320H		40 s	20 s	10 s	8 s	4 s	2 s	1 s	Second column	R/W
MINR	0321H		40 min	20 min	10 min	8 min	4 min	2 min	1 min	Minute column	R/W
HOURR	0322H			20 H	10 H	8 H	4 H	2 H	1 H	Hour column	R/W
DAYR	0323H						W2	W1	W0	Day of the week column	R/W
DATER	0324H			Day 20	Day 10	Day 8	Day 4	Day 2	Day 1	Day column	R/W
MONTHR	0325H				Oct.	Aug.	Apr.	Feb.	Jan.	Month column	R/W
YEARR	0326H	Year 80	Year 40	Year 20	Year 10	Year 8	Year 4	Year 2	Year 1	Year column (lower two columns)	R/W
PAGER	0327H	INTENA			ADJUST	ENATMR	ENAALM		PAGE	PAGE register	W,R/W
RESTR	0328H	DIS1HZ	DIS16HZ	RSTTMR	RSTALM	Always Write to 0				Reset register	Write only

(note): As for SECR,MINR,HOURR,DAYR,MONTHR,YEAR of PAGE0, current state is read when read it.

Table 3.13.2 PAGE 1(Alarm function) registers

Symbol	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Function	Read /Write
SECR	0320H										R/W
MINR	0321H		40 min	20 min	10 min	8 min	4 min	2 min	1 min	Minute column	R/W
HOURR	0322H			20 H	10 H	8 H	4 H	2 H	1 H	Hour column	R/W
DAYR	0323H						W2	W1	W0	Day of the week column for Alarm	R/W
DATER	0324H			Day 20	Day 10	Day 8	Day 4	Day 2	Day 1	Day column	R/W
MONTHR	0325H								24/12	24-hour clock mode	R/W
YEARR	0326H							Leap 1	Leap 2	Leap – year mode	R/W
PAGER	0327H	INTENA				ENATMR	ENAALM		PAGE	PAGE register	W,R/W
RESTR	0328H	DIS1HZ	DIS16HZ	RSTTMR	RSTALM	Always Write to 0				Reset register	Write only

3.13.2 Detailed explanation of control register

RTC is not initialized by reset.
Therefore, all registers must be initialized at the beginning of the program.

(1) Second column register (for PAGE0 only)

SECR
(0320H)

	7	6	5	4	3	2	1	0
bit Symbol		SE6	SE5	SE4	SE3	SE2	SE1	SE0
Read/Write		R/W						
After reset		Undefined						
Function	"0" is read.	40 sec. column	20 sec. Column	10sec. Column	8 sec. column	4 sec. column	2sec. column	1sec. column

0	0	0	0	0	0	0	0 sec.
0	0	0	0	0	0	1	1 sec.
0	0	0	0	0	1	0	2 sec.
0	0	0	0	1	0	0	4 sec.
0	0	0	0	1	0	1	5 sec.
0	0	0	0	1	1	0	6 sec.
0	0	0	0	1	1	1	7 sec.
0	0	0	1	0	0	0	8 sec.
0	0	0	1	0	0	1	9 sec.
0	0	1	0	0	0	0	10 sec.

0	0	1	1	0	0	1	19 sec.
0	1	0	0	0	0	0	20 sec.

0	1	0	1	0	0	1	29 sec.
0	1	1	0	0	0	0	30 sec.

0	1	1	1	0	0	1	39 sec.
1	0	0	0	0	0	0	40 sec.

1	0	0	1	0	0	1	49 sec.
1	0	1	0	0	0	0	50 sec.

1	0	1	1	0	0	1	59 sec.

(2) Minute column register (for PAGE0/1)

MINR
(0321H)

		7	6	5	4	3	2	1	0
bit Symbol			MI6	MI5	MI4	MI3	MI2	MI1	MI0
Read/Write			R/W						
After reset			Undefined						
Function		"0" is read.	40 min, column	20min, column	10min, column	8 min. column	4 min. column	2 min, column	1min, column

0	0	0	0	0	0	0	0 min.
0	0	0	0	0	0	1	1 min.
0	0	0	0	0	1	0	2 min.
0	0	0	0	0	1	1	3 min.
0	0	0	0	1	0	0	4 min.
0	0	0	0	1	0	1	5 min.
0	0	0	0	1	1	0	6 min.
0	0	0	0	1	1	1	7 min.
0	0	0	1	0	0	0	8 min.
0	0	0	1	0	0	1	9 min.
0	0	1	0	0	0	0	10 min.

0	0	1	1	0	0	1	19 min.
0	1	0	0	0	0	0	20 min.

0	1	0	1	0	0	1	29 min.
0	1	1	0	0	0	0	30 min.

0	1	1	1	0	0	1	39 min.
1	0	0	0	0	0	0	40 min.

1	0	0	1	0	0	1	49 min.
1	0	1	0	0	0	0	50 min.

1	0	1	1	0	0	1	59 min.

(3) Hour column register (for PAGE0/1)

① In case of 24-hour clock mode (MONTHR<MO0>='1') of PAGE1

		7	6	5	4	3	2	1	0
HOURR (0322H)	bit Symbol			HO5	HO4	HO3	HO2	HO1	HO0
	Read/Write			R/W					
	After reset			Undefined					
	Function	"0" is read.		20 hour column	10 hour column	8 hour column	4 hour column	2 hour column	1 hour column

0	0	0	0	0	0	0 o'clock
0	0	0	0	0	1	1 o'clock
0	0	0	0	1	0	2 o'clock

0	0	1	0	0	0	8 o'clock
0	0	1	0	0	1	9 o'clock
0	1	0	0	0	0	10 o'clock

0	1	1	0	0	1	19 o'clock
1	0	0	0	0	0	20 o'clock

1	0	0	0	1	1	23 o'clock

② In case of 12-hour clock mode (MONTHR<MO0>='0') of PAGE1

		7	6	5	4	3	2	1	0
HOURR (0322H)	bit Symbol			HO5	HO4	HO3	HO2	HO1	HO0
	Read/Write			R/W					
	After reset			Undefined					
	Function	"0" is read.		PM/$\overline{\text{AM}}$	10 hour column	8 hour column	4 hour column	2 hour column	1 hour column

0	0	0	0	0	0	0 o'clock (AM)
0	0	0	0	0	1	1 o'clock
0	0	0	0	1	0	2 o'clock
0	0	1	0	0	1	9 o'clock
0	1	0	0	0	0	10 o'clock
0	1	0	0	0	1	11 o'clock
1	0	0	0	0	0	0 o'clock (PM)
1	0	0	0	0	1	1 o'clock

(4) Day of the week column register (for PAGE0/1)

DAYR
(0323H)

	7	6	5	4	3	2	1	0
bit Symbol						WE2	WE1	WE0
Read/Write						R/W		
After reset						Undefined		
Function	"0" is read.					W2	W1	W0

0	0	0	Sunday
0	0	1	Monday
0	1	0	Tuesday
0	1	1	Wednesday
1	0	0	Thursday
1	0	1	Friday
1	1	0	Saturday

(5) Day column register (for PAGE0/1)

DATER
(0324H)

	7	6	5	4	3	2	1	0
bit Symbol			DA5	DA4	DA3	DA2	DA1	DA0
Read/Write			R/W					
After reset			Undefined					
Function	"0" is read.		Day 20	Day 10	Day 8	Day 4	Day 2	Day 1

0	0	0	0	0	0	0
0	0	0	0	0	1	1st day
0	0	0	0	1	0	2nd day
0	0	0	0	1	1	3rd day
0	0	0	1	0	0	4th day

0	0	1	0	0	1	9th day
0	1	0	0	0	0	10th day
0	1	0	0	0	1	11th day

0	1	1	0	0	1	19th day
1	0	0	0	0	0	20th day

1	0	1	0	0	1	29th day
1	1	0	0	0	0	30th day
1	1	0	0	0	1	31st day

(6) Month column register (for PAGE0 only)

		7	6	5	4	3	2	1	0
MONTHR (0325H)	bit Symbol				MO4	MO3	MO2	MO1	MO0
	Read/Write				R/W				
	After reset				Undefined				
	Function	"0" is read.			10 months	8 months	4 months	2 months	1 month

0	0	0	0	1	January
0	0	0	1	0	February
0	0	0	1	1	March
0	0	1	0	0	April
0	0	1	0	1	May
0	0	1	1	0	June
0	0	1	1	1	July
0	1	0	0	0	August
0	1	0	0	1	September
1	0	0	0	0	October
1	0	0	0	1	November
1	0	0	1	0	December

(7) Select 24-hour clock or 12-hour clock (for PAGE1 only)

		7	6	5	4	3	2	1	0
MONTHR (0325H)	bit Symbol								MO0
	Read/Write								R/W
	After reset								Undefined
	Function	"0" is read.							1:24-hour 0:12-hour

(8) Year column register (for PAGE0 only)

		7	6	5	4	3	2	1	0
YEARR (0326H)	bit Symbol	YE7	YE6	YE5	YE4	YE3	YE2	YE1	YE0
	Read/Write	R/W							
	After reset	Undefined							
	Function	80 Years	40 Years	20 Years	10 Years	8 Years	4 Years	2 Years	1 Year

↓

1	0	0	1	1	0	0	1	99 year
0	0	0	0	0	0	0	0	00 year
0	0	0	0	0	0	0	1	01 year
0	0	0	0	0	0	1	0	02 year
0	0	0	0	0	0	1	1	03 year
0	0	0	0	0	1	0	0	04 year
0	0	0	0	0	1	0	1	05 year

(9) Leap-year register (for PAGE1 only)

		7	6	5	4	3	2	1	0
YEARR (0326H)	bit Symbol							LEAP1	LEAP0
	Read/Write							R/W	
	After reset							Undefined	
	Function	"0" is read.						00:leap-year 01: one year after leap-year 10:two years after leap-year 11:three years after leap-year	

↓

0	0	Current year is leap-year
0	1	Present is next year of a leap year
1	0	Present is two years after a leap year
1	1	Present is three years after leap year

(10) PAGE register setting (for PAGE0/1)

PAGER
(0327H)

	7	6	5	4	3	2	1	0
bit Symbol	INTENA			ADJUST	ENATMR	ENAALM		PAGE
Read/Write	R/W			W	R/W			R/W
After reset	0			—	Undefined			Undefined
Function	Note: Interrupt 1:ENABLE 0:DISABLE	"0" is read.		1:ADJUST	TIMER 1:ENABLE 0:DISABLE	ALARM 1:ENABLE 0:DISABLE	"0" is read.	PAGE select

Prohibit Read Modify Write
(note): Set order below.

EX.) Clock setting/Alarm setting
1. Clock/Alarm enable ld (pager),0ch
2. Interrupt enable ld (pager),8ch

0	Select Page0
1	Select Page1

0	—
1	Adjust sec. counter. When set the this bit to "1" the sec. counter become to "0" when the value of sec. counter is 0 – 29. And in case that value of sec. counter is 30-59, min. counter is carried and become sec. counter to "0". (PAGE0 only)

(11) Reset register setting (for PAGE0/1)

RESTR
(0328H)

	7	6	5	4	3	2	1	0
bit Symbol	DIS1HZ	DIS16HZ	RSTTMR	RSTALM	RE3	RE2	RE1	RE0
Read/Write	W							
After reset	Undefined							
Function	0:1HZ	0:16HZ	1:TIMER RESET	1:ALARM RESET	Please fix to "0"			

Prohibit Read Modify Write

0	—
1	Reset alarm register

0	—
1	Timer reset

0	Enable 16Hz clock (/ALARM output , INTRTC)
1	Disable 16Hz clock (/ALARM output , INTRTC)

0	Enable 1Hz clock (/ALARM output , INTRTC)
1	Disable 1Hz clock (/ALARM output , INTRTC)

3.13.5 Operational description

(1) Reading timer data

① There is the case which reads wrong data when carry of the inside counter happens during the operation which timer data reads. Therefore, please read two times with the following way for reading correct data.

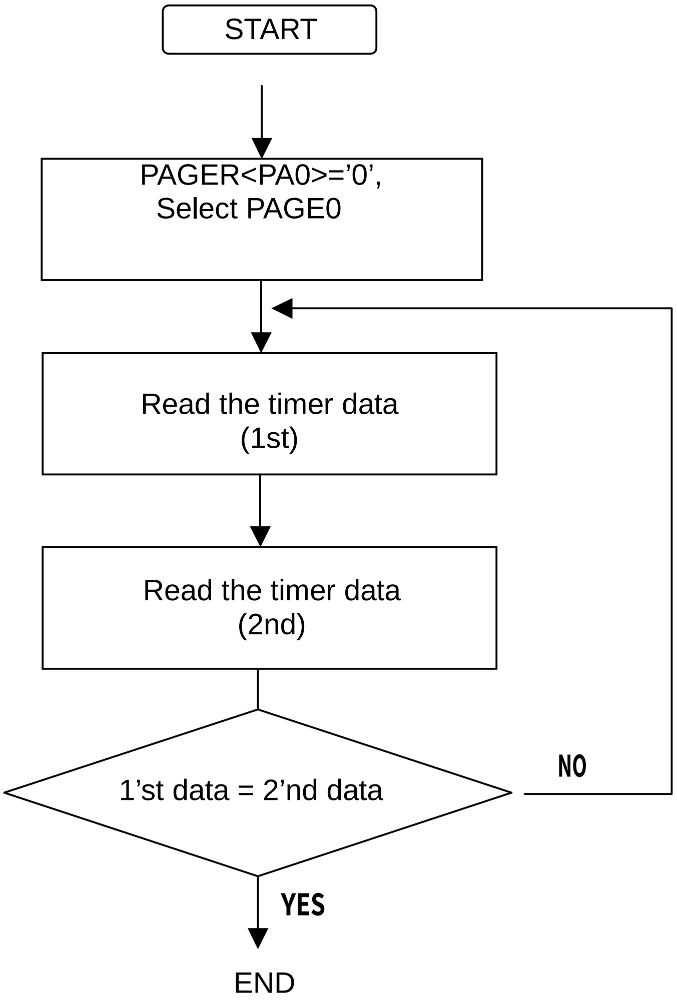


Figure 3.13.2 Flowchart of timer data read

As shown in Figure 3.13.2, confirm the data by reading twice and compare them in case reading timer data. If it happen to take up a digit, the comparing result becomes incorrect. Therefore, It should be read data again.

② Readout of timer data that used /ALARM output

Timer data can be read with rising edge of /ALARM output by detecting /ALARM='1' with interrupt routine of INTRTC of 1 Hz

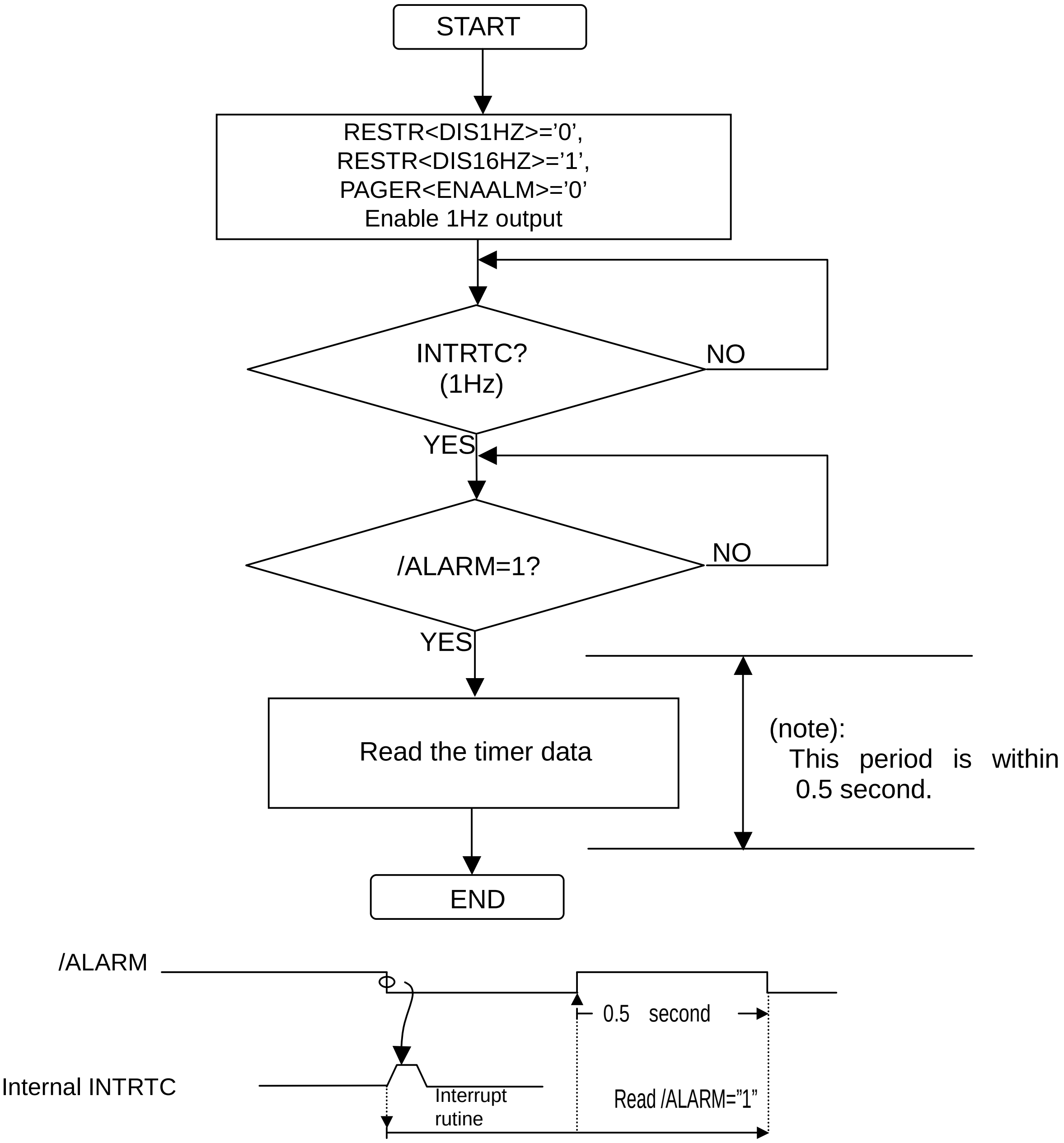


Figure 3.13.3 Read out of the timer table used /ALARM output

The reason why read a timer of RTC after reading PORT in interrupt routine of /ALARM=1 is that carry of RTC timer occurs with rising edge of pulse period of 1 Hz. By reading timer during 0.5second after carry happening, right data (a timer value) can be read.

(2) Writing timer data

When there is carry on the way of write operation , expecting data can not be wrote exactly.

Therefore, in order to write in data exactly please follow the below way.

① Reset for a divider

Inside of RTC, there is 15-stage divider which generates 1Hz clock from 32,768KHz. Carry of a timer is not done for one second when reset this divider. So write in data during this interval.

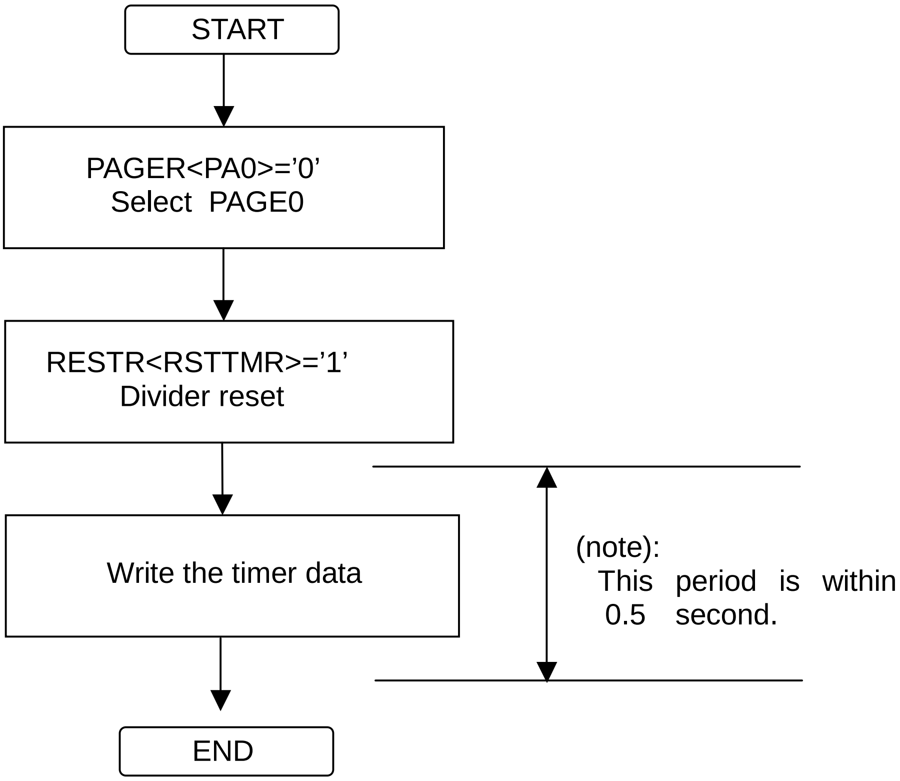


Figure 3.13.4 Flowchart of data write

② Disabling the timer

Carry of a timer is prohibited when write '0' to PAGER<ENATMR> and can prevent malfunction by CLOCK HOLD circuit. During a timer prohibited, CLOCK HOLD circuits holds one sec. carry signal, which is generated from divider. After becoming timer enable state, output the carry signal to timer and revise time and continue operation. However, timer is late when timer-disabling state continues for one second or more. During timer disabling, pay attention with system power is downed. In this case the timer is stopped and time is delayed.

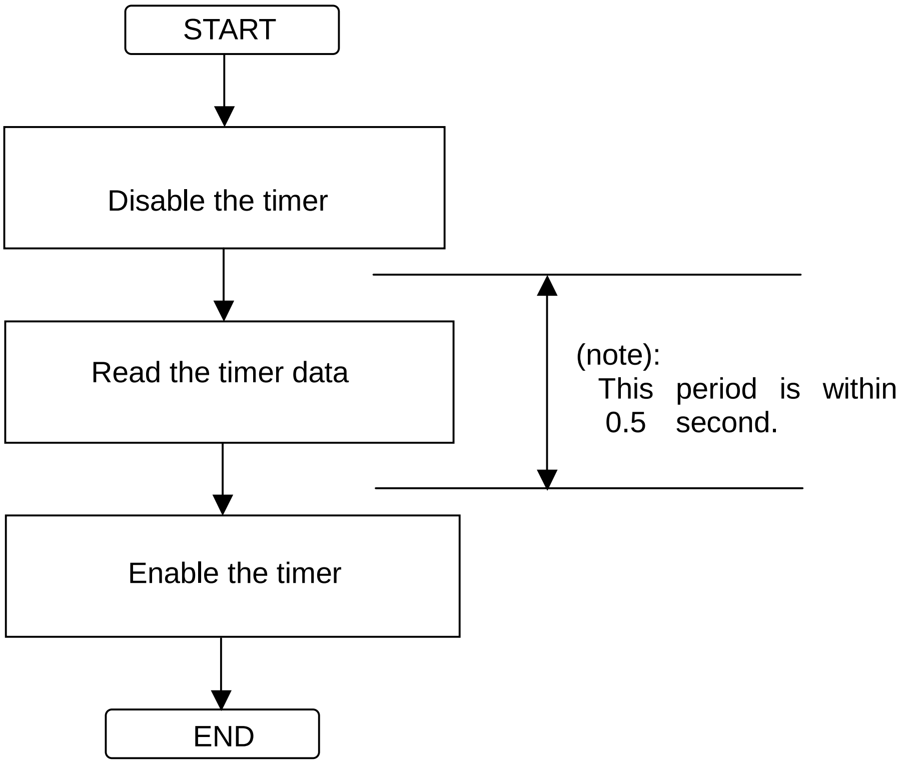


Figure 3.13.5 Flowchart of timer disable

3.13.3 Explanation of the alarm function

It can use alarm function by setting of register of PAGE1 and output either of three signal from /ALARM pin as follows.

INTRTC always output 1-shot pulse detecting falling down edge.

RTC circuit isn’t reset by /RESET signal, it need to init interrupt request flag before setting timer and alarm of RTC.

(1) In accordance of alarm register and the timer, output '0'

(2) Output clock of 1Hz

(3) Output clock of 16Hz

(1) In accordance of alarm register and a timer, output '0'.

When value of a timer of PAGE0 accorded with alarm register of PAGE1 with a state of PAGER<ENAALM>='1', output '0' to /ALARM pin and occur INTRTC.

Follows are ways using alarm.

Initialization of alarm is done by writing in '1' at RESTR<RSTALM>, setting value of all alarm becomes don’t care. In this case, always accorded with value of a timer and occur INTRTC interrupt if PAGER<ENAALM> is '1'.

Setting alarm min., alarm hour, alarm day and alarm the day week is done by writing in data at each register of PAGE1.

When all setting contents accorded, RTC generates INTRTC interrupt, if PAGER<ENAALM> is '1'. However, contents (don't care state) which does not set it up is considered to always accord.

The contents, which set it up once, cannot be returned to don't care state in independence. Initialization of alarm and resetting of alarm register are necessary.

The following is an example program for outputting an alarm from $\overline{\text{ALARM}}$ pin at noon (PM 12:00) every day

```
            LD      (PAGER),09H     ; Alarm disable, setting PAGE1
            LD      (RESTR),D0H     ; Alarm initialize
            LD      (MONTHR),01H    ; 24-hour clock mode
            LD      (HOURR),12H     ; setting 12 o’clock
            LD      (MINR),00H      ; setting 00 min.
                                    ; Set up time 31 µs     Note)
            LD      (PAGER),0CH     ; Alarm enable
(           LD      (PAGER), 8CH    ; Interrupt enable  )
```

When CPU is operated by high frequency oscillation, it may take a maximum of one clock at 32 kHz (about 30 µs) for the time register setting to become valid. In the above example, it is necessary to set 31 µs of set up time between setting the time register and enabling the alarm register.

Note) This set up time is unnecessary under SLOW mode or when you use only internal interruption.

(2) When output clock of 1Hz

RTC outputs clock of 1Hz to /ALARM pin by setting up PAGER<ENAALM>='0', RESTR<DIS1HZ>='0',<DIS16HZ>='1'. And RTC generates INTRTC interrupt by falling edge of the clock.

(3) When output clock of 16Hz

RTC outputs clock of 16Hz to /ALARM pin by setting up PAGER<ENAALM>='0', RESTR<DIS1HZ>='1', <DIS16HZ>='0'. And RTC generates INTRTC interrupt by falling edge of the clock.

3.14 Melody / Alarm generator (MLD)

TMP91C824 incorporates melody function and alarm function, both of which are output from the MLDALM pin. 5 kinds of fixed cycle interrupts are generated by the 15-bit free-run counter, which is used for alarm generator.

Features are as follows.

- Melody generator

The Melody function generates signals of any frequency (4Hz- 5461Hz) based on low-speed clock (32.768KHz) and outputs several signals from the MLDALM pin.

By connecting a loud speaker outside, Melody tone can sound easily.

- Alarm generator

The Alarm function generates 8 kinds of alarm waveform having a modulation frequency (4096Hz) determined by the low-speed clock (32.768KHz). And this waveform is able to invert by setting a value to a register.

By connecting a loud speaker outside, Alarm tone can sound easily.

And also 5 kinds of fixed cycle (1Hz, 2Hz, 64Hz, 512Hz, 8KHz) interrupts are generated by the free-run counter which is used for alarm generator.

This section is constituted as follows.

3.14.1 Block Diagram

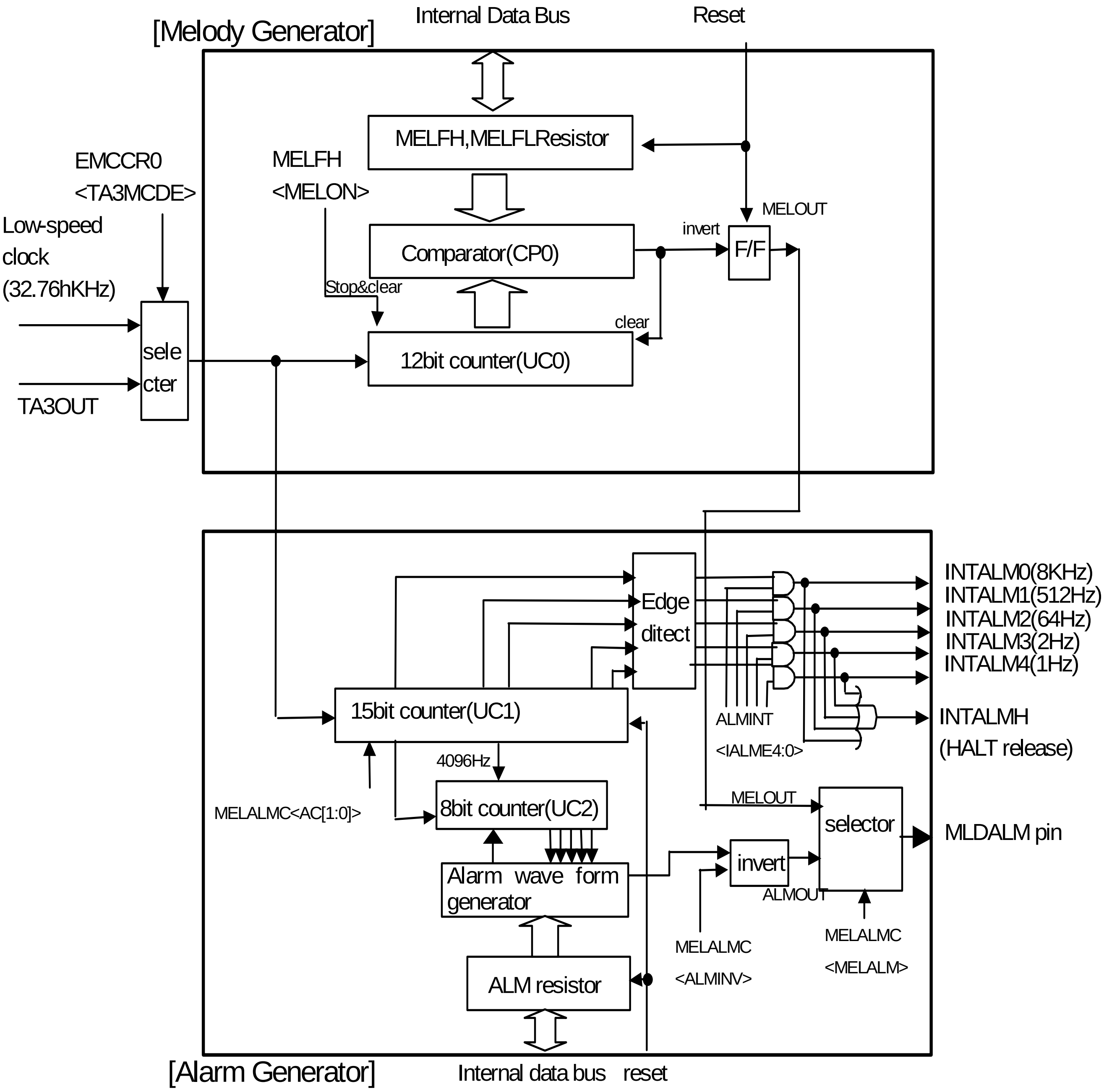


Figure 3.14.1 MLD Block Diagram

3.14.2 Control registers

ALM R register

		7	6	5	4	3	2	1	0
ALM (0330H)	bit Symbol	AL8	AL7	AL6	AL5	AL4	AL3	AL2	AL1
	Read/Write	R/W							
	After reset	0							
	Function	Setting alarm pattern							

MLDALMC register

		7	6	5	4	3	2	1	0
MELALMC (0331H)	bit Symbol	FC1	FC0	ALMINV					MELALM
	Read/Write	R/W		R/W					R/W
	After reset	0		0					0
	Function	Free-run counter control 00: Hold 01: Restart 10: Clear 11: Clear & Start		Alarm Wavefor m invert 1:INVERT	Write “0”				Output Waveform select 0: Alarm 1: Melody

(note1): MELALMEC<FC1> is read always “0”.

(note2): When setting MELALMC register except <FC1:0> during the free-run counter is running , <FC1:0> is kept “01”.

MELFL register

		7	6	5	4	3	2	1	0
MELFL (0332H)	bit Symbol	ML7	ML6	ML5	ML4	ML3	ML2	ML1	ML0
	Read/Write	R/W							
	After reset	0							
	Function	Setting melody frequency (lower 8bit)							

MELFH register

		7	6	5	4	3	2	1	0
MELFH (0333H)	bit Symbol	MELON				ML11	ML10	ML9	ML8
	Read/Write	R/W				R/W			
	After reset	0				0			
	Function	Control melody counter 0: Stop & Clear 1: Start				Setting melody frequency(upper 4bit)			

ALMINT register

		7	6	5	4	3	2	1	0
ALMINT (0334H)	bit Symbol				IALM4E	IALM3E	IALM2E	IALM1E	IALM0E
	Read/Write				R/W				
	After reset				0				
	Function	Write “0”			1:Interrupt enable for INTALM4〜INTALM0				

3.14.3 Operational Description

3.14.3.1 Melody generator

The Melody function generates signals of any frequency (4Hz- 5461Hz) based on low-speed clock (32.768KHz) and outputs the signals from the MLDALM pin.

By connecting a loud speaker outside, Melody tone can sound easily.

(Operation)

At first, MELALMC<MELALM> have to be set as 1 in order to select melody waveform as output waveform from MLDALM. Then melody output frequency has to be set to 12-bit register MELFH, MELFL.

Followings are setting example and calculation of melody output frequency.

(Formula for calculating of melody waveform frequency)

@fs = 32.768[KHz]

melody output waveform f_{MLD}[Hz] = 32768 / (2×N+4)

setting value for melody N = (16384 / f_{MLD}) – 2

(notice: N=1～4095(001H～FFFH) 、 0 is not acceptable)

(Example program)

In case of outputting "La" musical scale (440Hz)

```
LD      (MELALMC),--XXXXX1B     ; select melody waveform
LD      (MELFL),23H             ; N= 16384/440 – 2 = 35.2 = 023H
LD      (MELFH),80H             ; start to generate waveform
```

(Refer to " Basic musical scale setting table")

Scale	Frequency [Hz]	Register value: N
C	264	03CH
D	297	035H
E	330	030H
F	352	02DH
G	396	027H
A	440	023H
B	495	01FH
C	528	01DH

3.14.3.2 Alarm generator

The Alarm function generates 8 kinds of alarm waveform having a modulation frequency 4096Hz determined by the low-speed clock (32.768KHz). And this waveform is reversible by setting a value to a register.

By connecting a loud speaker outside, Alarm tone can sound easily.

5 kinds of fixed cycle (1Hz, 2Hz, 64Hz, 512Hz, 8KHz) interrupts are generated by the free-run counter, which is used for alarm generator.

(Operation)

At first, MELALMC<MELALM> have to be set as 0 in order to select alarm waveform as output waveform from MLDALM. Then alarm pattern have to be set on 8 bit register of ALM. Finally “10” be set on MLDALMC<AC1:0> register, and <ALMINV> be set as invert. By it is setting these values, counter start to generate alarm waveform.

Followings are example program, setting value of alarm pattern and waveform of each setting value.

(Setting value of alarm pattern)

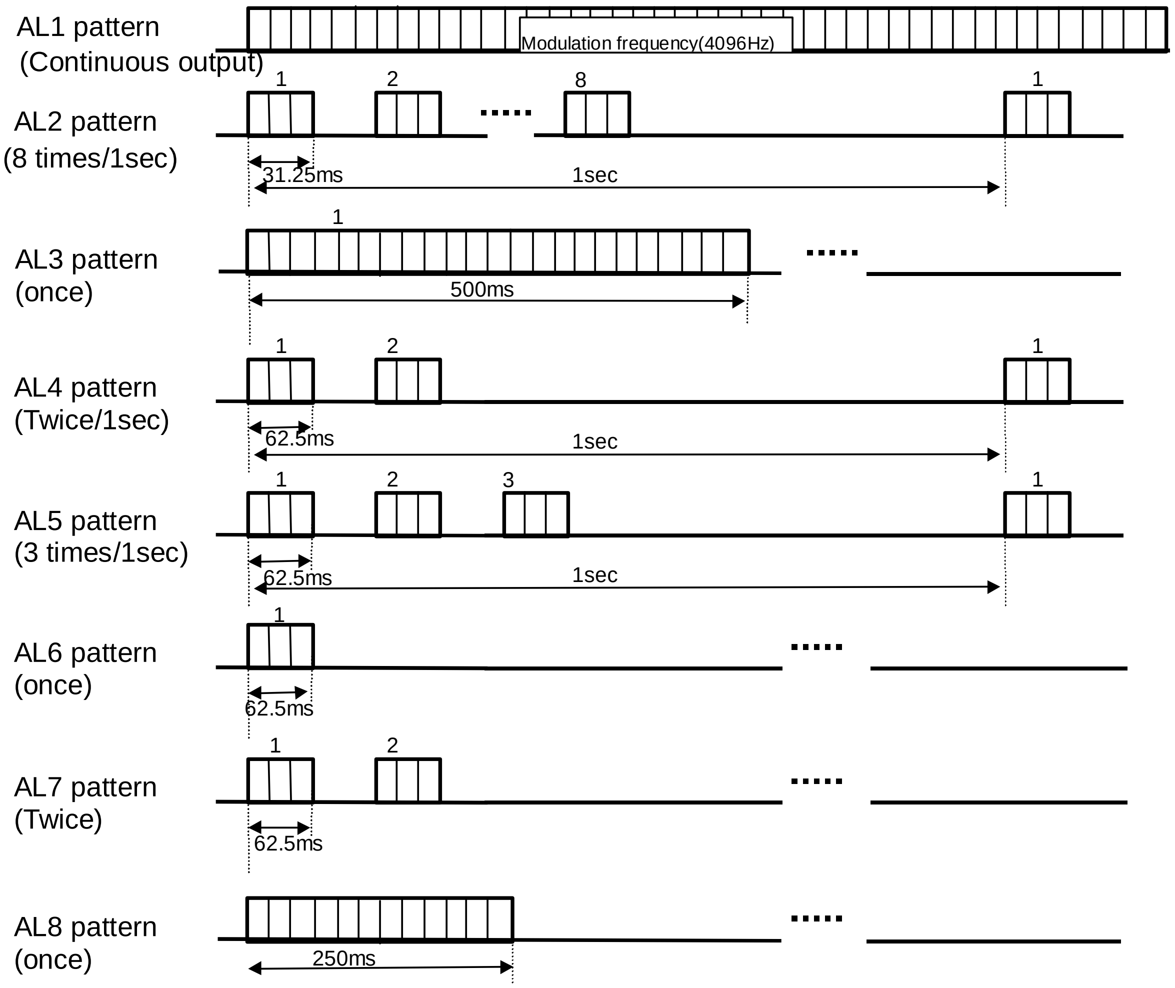
Setting value for ALM register	Alarm waveform
00H	“0” fixed
01H	AL1 pattern
02H	AL2 pattern
04H	AL3 pattern
08H	AL4 pattern
10H	AL5 pattern
20H	AL6pattern
40H	AL7 pattern
80H	AL8 pattern
Other	Undefined (do not set)

(Example program)

In case of outputting AL2 pattern (31.25ms/8 times/1sec)

```
LD      (MELALMC),C0H        ; set output alarm waveform
                             ; free-run counter start
LD      (ALM),02H            ; set AL2 pattern , start
```

Example: Waveform of alarm pattern for each setting value: not invert)



4. Electrical Characteristics

4.1 Absolute Maximum Ratings

Symbol	Parameter	Rating	Unit
Vcc	Power Supply Voltage	–0.5 to 4.0	V
VIN	Input Voltage	–0.5 to Vcc + 0.5	V
IOL	Output Current	2	mA
IOH	Output Current	–2	mA
ΣIOL	Output Current (total)	80	mA
ΣIOH	Output Current (total)	–80	mA
PD	Power Dissipation (Ta = 85°C)	600	mW
TSOLDER	Soldering Temperature (10 s)	260	°C
TSTG	Storage Temperature	–65 to 150	°C
TOPR	Operating Temperature	–40 to 85	°C

4.2 DC Characteristics (1/2)

Parameter		Symbol	Condition		Min.	Typ.	Max.	Unit
Power Supply Voltage (AVCC = DVCC) (AVSS = DVSS = 0 V)		VCC	fc = 2〜33MHz	fs =30to 34kHz	2.7	-	3.6	V
			fc = 2〜10MHz		1.8			
Input Low Voltage	D0 to 15	VIL	Vcc≧2.7V		-0.3	-	0.6	V
			Vcc<2.7V				0.2Vcc	
	P52 to PD7(excep tPB3)	VIL1	Vcc≧2.7V			-	0.3Vcc	
			Vcc<2.7V				0.2Vcc	
	/RESET,/NMI, PB3(INT0)	VIL2	Vcc≧2.7V			-	0.25Vcc	
			Vcc<2.7V				0.15Vcc	
	AM0〜1	VIL3	Vcc≧2.7V			-	0.3	
			Vcc<2.7V				0.3	
	X1	VIL4	Vcc≧2.7V			-	0.2Vcc	
			Vcc<2.7V				0.1Vcc	
Inout High Voltage	D0〜15	VIH	3.6V≧ Vcc> 3.3V		2.4	-	Vcc+0.3	
			3.3V>Vcc≧ 2.7V		2.0			
			Vcc<2.7V		0.7Vcc			
	P52 to PD7(except PB3)	VIH1	Vcc≧2.7V		0.7Vcc	-		
			Vcc<2.7V		0.8Vcc			
	/RESET,/NMI, PB3(INT0)	VIH2	Vcc≧2.7V		0.75Vcc	-		
			Vcc<2.7V		0.85Vcc			
	AM0 to 1	VIH3	Vcc≧2.7V		Vcc-0.3	-		
			Vcc<2.7V		Vcc-0.3			
	X1	VIH4	Vcc≧2.7V		0.8Vcc	-		
			Vcc<2.7V		0.9Vcc			
Output Low Voltage		VOL1	IOL=1.6mA	Vcc≧2.7V	-	-	0.45	V
			IOL=0.4mA	Vcc<2.7V			0.15Vcc	
Output High Voltage		VOH2	IOH=-400uA	Vcc≧2.7V	Vcc-0.3	-	-	
			IOH=-200uA	Vcc<2.7V	0.8Vcc			

(note1): Typical values are for when Ta = 25°C and Vcc = 3.0 V uncles otherwise noted.

4.2 DC Characteristics (2/2)

Parameter	Symbol	Condition	Min	Typ	Max.	Unit
Input Leak Current	ILI	0.0≦VIN≦Vcc	-	0.02	± 5	μA
Output Leak Current	ILO	0.2≦VIN≦ Vcc-0.2	-	0.05	± 10	
Power Down Voltage (at STOP,RAM Back up)	VSTOP	VIL2 = 0.2Vcc, VIH2 = 0.8Vcc	1.8	-	3.6	V
/RESET Pull Up Resister	RRST	3.6V≧Vcc≧2.7V	80	-	400	kΩ
		Vcc=2V±10%	200		1000	
Pin Capacitance	CIO	fc = 1MHz	-	-	10	pF
Schmitt Width /RESET,/NMI, INT0,KI0-7	VTH	Vcc≧2.7V	0.4	0.9	-	V
		Vcc<2.7V	0.3	0.7		
Programmable Pull Up Resister	RKH	3.6V≧Vcc≧2.7V	80	-	400	kΩ
		Vcc=2V±10%	200		1000	
NORMAL (Note2)	Icc	3.6V≧Vcc≧2.7V fc = 33MHz	-	14.0	20.0	mA
IDLE2			-	4.0	6.1	
IDLE1			-	1.2	2.2	
NORMAL (Note2)		Vcc=2V±10% fs=10MHz (Typ.value :Vcc=2.0V)	-	2.6	3.0	mA
IDLE2			-	0.7	1.2	
IDLE1			-	0.2	0.4	
SLOW (Note2)		3.6V≧Vcc≧2.7V fs=32.768kHz	-	17.5	30.5	μA
IDLE2			-	7.0	13.5	
IDLE1			-	5.0	10.0	
SLOW (Note2)		Vcc=2V±10% fs=32.768kHz (Typ. Value : Vcc=2.0V)	-	10.5	13.0	μA
IDLE2			-	4.5	6.5	
IDLE1			-	3.0	4.5	
STOP		3.6V≧Vcc≧1.8V	-	0.2	15	μA

(note1): Typical values are for when Ta = 25°C and Vcc = 3.0 V unless otherwise noted.

(note2): Icc measurement conditions (NORMAL, SLOW):
All functions are operational; output pins are open and input pins are fixed. Data & address bus CL=30pF loaded.

4.3 AC Characteristics

(1) Vcc = 3.0 V ± 10%

No.	Symbol	Parameter	Variable		f_{FPH} = 33 MHz		Unit
			Min	Max	Min	Max	
1	t_{FPH}	f_{FPH} Period (= x)	30.3	31250	30.3		ns
2	t_{AC}	A0 to 23 Vaild → $\overline{RD}$ / $\overline{WR}$ Fall	x – 23		7		ns
3	t_{CAR}	$\overline{RD}$ Rise → A0 to A23 Hold	0.5x –13		2		ns
4	t_{CAW}	$\overline{WR}$ Rise → A0 to A23 Hold	x – 13		17		ns
5	t_{AD}	A0 to A23 Valid → D0 to D15 Input		3.5x – 24		82	ns
6	t_{RD}	$\overline{RD}$ Fall → D0 to D15 Input		2.5x – 24		51	ns
7	t_{RR}	$\overline{RD}$ Low Width	2.5x – 15		60		ns
8	t_{HR}	$\overline{RD}$ Rise → D0 to A15 Hold	0		0		ns
9	t_{WW}	$\overline{WR}$ Low Width	2.0x – 15		45		ns
10	t_{DW}	D0 to D15 Valid → $\overline{WR}$ Rise	1.5x – 35		10		ns
11	t_{WD}	$\overline{WR}$ Rise → D0 to D15 Hold	x – 25		5		ns
12	t_{AW}	A0 to A23 Valid → $\overline{WAIT}$ Input (1WAIT+n)		3.5x – 60		46	ns
13	t_{CW}	$\overline{RD}$ / $\overline{WR}$ Fall → $\overline{WAIT}$ Hold (1WAIT+n)	2.5x + 0		76		ns
14	t_{APH}	A0 to A23 Valid → PORT Input		3.5x – 89		17	ns
15	t_{APH2}	A0 to A23 Valid → PORT Hold	3.5x		106		ns
16	t_{APO}	A0 to A23 Valid → PORT Valid		3.5x + 60		166	ns

AC Measuring Conditions

・Output Level : High = 0.7 Vcc, Low = 0.3 Vcc, CL = 50 pF

・Input Level : High = 0.9 Vcc, Low = 0.1 Vcc

(note): Symbol “ x ” in the above table means the period of clock “ f_{FPH} “, it’s half period of the system clock “ f_{SYS} “ for CPU core. The period of f_{FPH} depends on the clock gear setting or the selection of High / Low oscillator frequency.

(2) Vcc = 2.0 V ± 10%

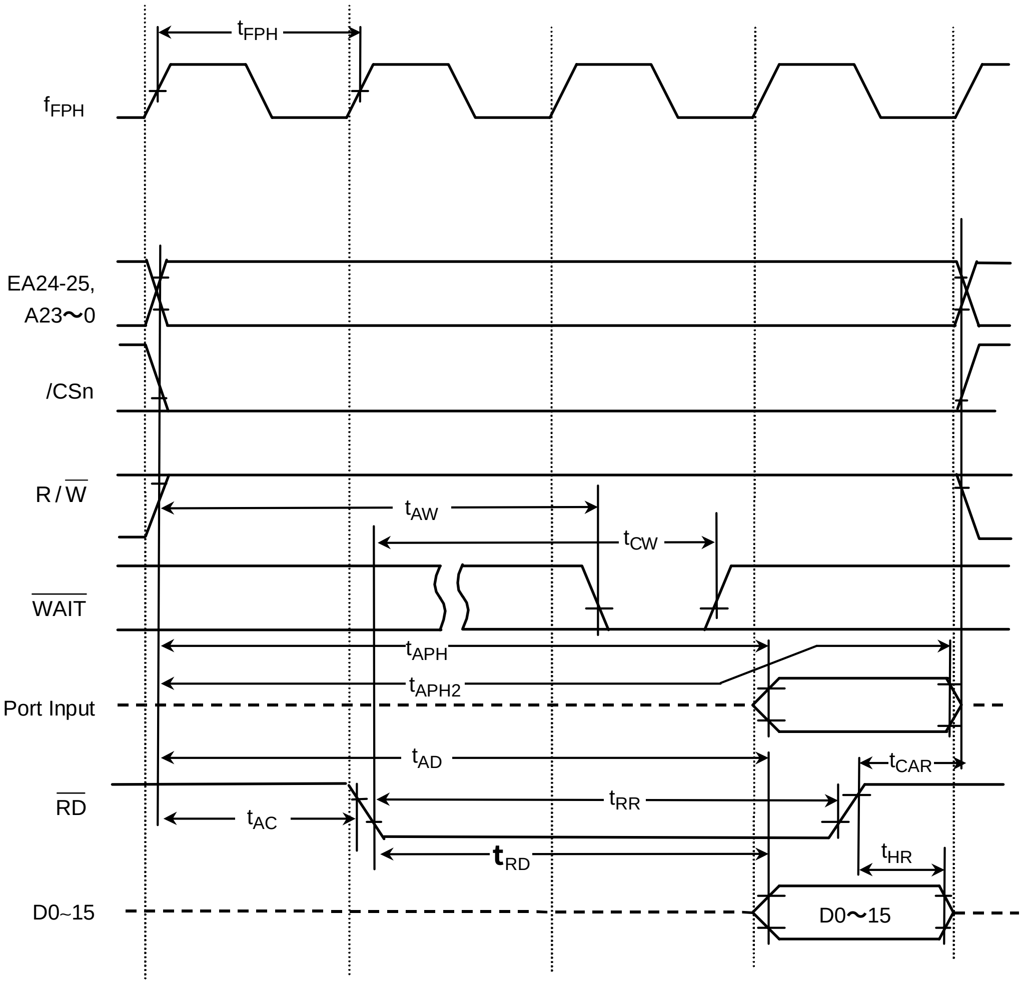
No.	Symbol	Parameter	Variable		10MHz		Unit
			Min	Max	Min	Max	
1	tFPH	f_{FPH} Period (= x)	100	31250	100		ns
2	tAC	A0 to A15 Valid → $\overline{\text{RD}}$ / $\overline{\text{WR}}$ Fall	x -46		54		ns
3	tCAR	$\overline{\text{RD}}$ Rise → A0 to A23 Hold	0.5x - 26		24		ns
4	tCAW	$\overline{\text{WR}}$ Rise → A0 to A23 Hold	x - 26		74		ns
5	tAD	A0 to A23 Valid → $\overline{\text{RD}}$ / $\overline{\text{WR}}$ Fall		3.5x - 48		302	ns
6	tRD	$\overline{\text{RD}}$ Fall → D0 to D15 Input		2.5x - 48		202	ns
7	tRR	$\overline{\text{RD}}$ Low Width	2.5x - 30		220		ns
8	tHR	$\overline{\text{RD}}$ Rise → D0 to D15 Hold	0		0		ns
9	tWW	$\overline{\text{WR}}$ Low Width	2.0x - 30		170		ns
10	tDW	D0 to D15 Valid → $\overline{\text{WR}}$ Rise	1.5x – 70		80		ns
11	tWD	$\overline{\text{WR}}$ Rise → D0 to D15 Hold	x – 50		50		ns
12	tAW	A0 to A23 Valid → $\overline{\text{WAIT}}$ Input (1WAIT +n mode)		3.5x - 120		230	ns
13	tCW	$\overline{\text{RD}}$ / $\overline{\text{WR}}$ Fall → $\overline{\text{WAIT}}$ Hold (1WAIT +n mode)	2.5x + 0		250		ns
14	tAPH	A0 to A23 Valid → PORT Input		3.5x - 50		300	ns
15	tAPH2	A0 to A23 Valid → PORT Hold	3.5x		350		ns
16	tAPO	A0 to A23 Valid → PORT Valid		3.5x + 60		410	ns

AC Measuring Conditions

- Output Level: High = 0.7 V, Low = 0.3 V, CL = 50 pF
- Input Level: High = 0.9 V, Low = 0.1V

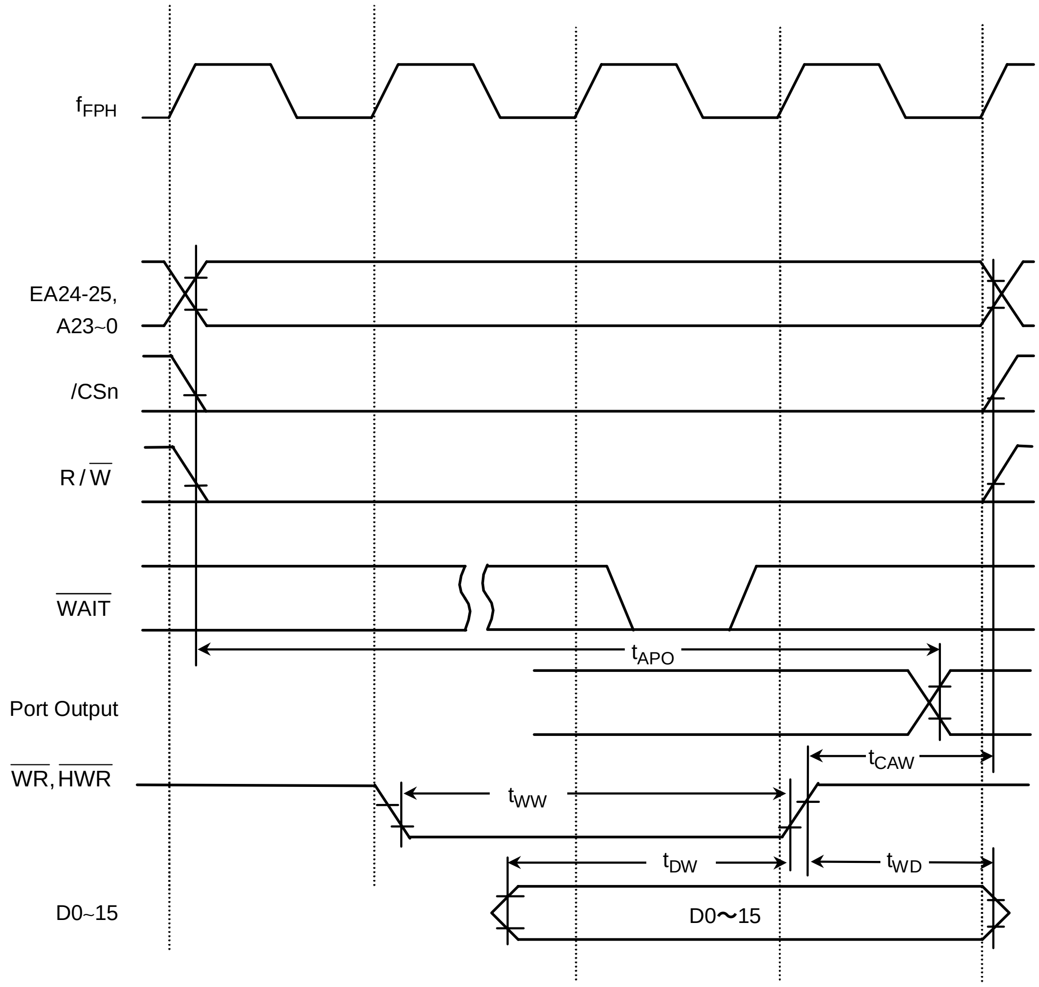
(note): Symbol " x " in the above table means the period of clock " f_{FPH} ", it's half period of the system clock " f_{SYS} " for CPU core. The period of f_{FPH} depends on the clock gear setting or the selection of High / Low oscillator frequency.

(1) Read Cycle



t_{FPH}
f_{FPH}
EA24-25,
A23〜0
/CSn
R / $\overline{W}$
t_{AW}
t_{CW}
$\overline{WAIT}$
t_{APH}
t_{APH2}
Port Input
t_{AD}
t_{CAR}
$\overline{RD}$
t_{RR}
t_{AC}
t_{RD}
t_{HR}
D0~15
D0〜15

(2) Write Cycle



4.4 A/D Conversion Characteristics

AVcc = Vcc, AVss = Vss

Symbol	Parameter	Condition	Min	Typ.	Max	Unit
VREFH	Analog Reference Voltage (+)	V_{CC} = 3 V ± 10%	V_{CC} − 0.2 V	Vcc	Vcc	V
		V_{CC} = 2 V ± 10%	V_{CC}	Vcc	Vcc	
VREFL	Analog Reference Voltage (−)	V_{CC} = 3 V ± 10%	V_{SS}	Vss	Vss + 0.2 V	
		V_{CC} = 2 V ± 10%	V_{SS}	Vss	Vss	
VAIN	Analog Input Voltage Range		V_{REFL}		V_{REFH}	
IREF (VREFL = 0V)	Analog Current for Analog Reference Voltage <VREFON> = 1	V_{CC} = 3 V ± 10%		0.94	1.20	mA
		V_{CC} = 2 V ± 10%		0.65	0.90	
	<VREFON> = 0	V_{CC} = 1.8 V to 3.3 V		0.02	5.0	μA
−	Error (not including quantizing errors)	V_{CC} = 3 V ± 10%		± 1.0	± 4.0	LSB
		V_{CC} = 2 V ± 10%		± 1.0	± 4.0	

(note1): 1 LSB = (VREFH − VREFL)/1024 [V]

(note2): The operation above is guaranteed for $f_{FPH} \geq 4$ MHz.

(note3): The value for I_{CC} includes the current which flows through the AV_{CC} pin.

4.5 Serial Channel Timing (I/O Internal Mode)

(1) SCLK Input Mode

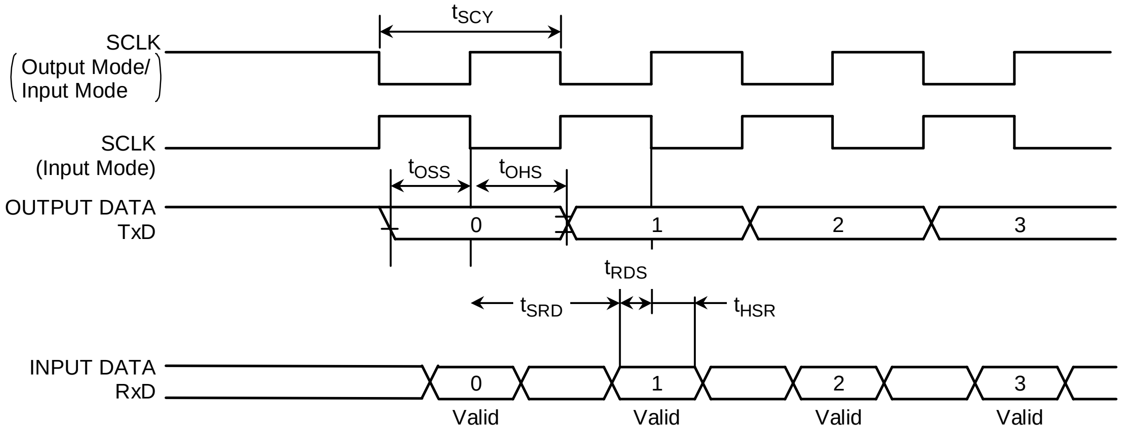
Symbol	Parameter		Variable		10 MHz		27 MHz		Unit
			Min	Max	Min	Max	Min	Max	
T_{SCY}	SCLK Period		16X		1.6		0.59		μs
T_{OSS}	Output Data →SCLK Rising/Falling Edge*	Vcc=3V±10%	$t_{SCY}/2 - 4X - 110$		290		38		ns
		Vcc=2V±10%	$t_{SCY}/2 - 4X - 180$		220		---		ns
T_{OHS}	SCLK Rising/Falling Edge* → Output Data Hold		$t_{SCY}/2 + 2X + 0$		1000		370		ns
T_{HSR}	SCLK Rising/Falling Edge* → Input Data Hold		3X + 10		310		121		ns
T_{SRD}	SCLK Rising/Falling Edge* → Valid Data Input			$t_{SCY} - 0$		1600		592	ns
T_{RDS}	Valid Data Input → SCLK Rising/Falling Edge*		0		0		0		ns

(2) SCLK Output Mode

Symbol	Parameter	Variable		10 MHz		27 MHz		Unit
		Min	Max	Min	Max	Min	Max	
T_{SCY}	SCLK Period	16X	8192X	1.6	819	0.59	303	μs
T_{OSS}	Output Data → SCLK Rising /Falling Edge*	$t_{SCY}/2 - 40$		760		256		ns
T_{OHS}	SCLK Rising/Falling Edge* → Output Data Hold	$t_{SCY}/2 - 40$		760		256		ns
T_{HSR}	SCLK Rising/Falling Edge* → Input Data Hold	0		0		0		ns
T_{SRD}	SCLK Rising/Falling Edge* → Valid Data Input		$t_{SCY} - 1X - 180$		1320		375	ns
T_{RDS}	Valid Data Input → SCLK Rising/Falling Edge*	1X + 180		280		217		ns

(note): SCLK Rinsing/Falling Edge : The rising edge is used in SCLK Rising Mode.
The falling edge is used in SCLK Falling Mode.

27MHz and 10MHz values are calculated from t_{SCY}=16X case.



4.6 Event Counter (TA0IN)

Symbol	Parameter	Variable		10 MHz		27 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{VCK}	Clock Period	8X + 100		900		396		ns
t_{VCKL}	Clock Low Level Width	4X + 40		440		188		ns
t_{VCKH}	Clock High Level Width	4X + 40		440		188		ns

4.7 Interrupt, Capture

(1) $\overline{\text{NMI}}$, INT0 to INT3 Interrupts

Symbol	Parameter	Variable		10 MHz		27 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{INTAL}	$\overline{\text{NMI}}$, INT0 to INT3 Low level width	4X + 40		440		188		ns
t_{INTAH}	$\overline{\text{NMI}}$, INT0 to INT3 High level width	4X + 40		440		188		ns

4.8 SCOUT Pin AC Characteristics

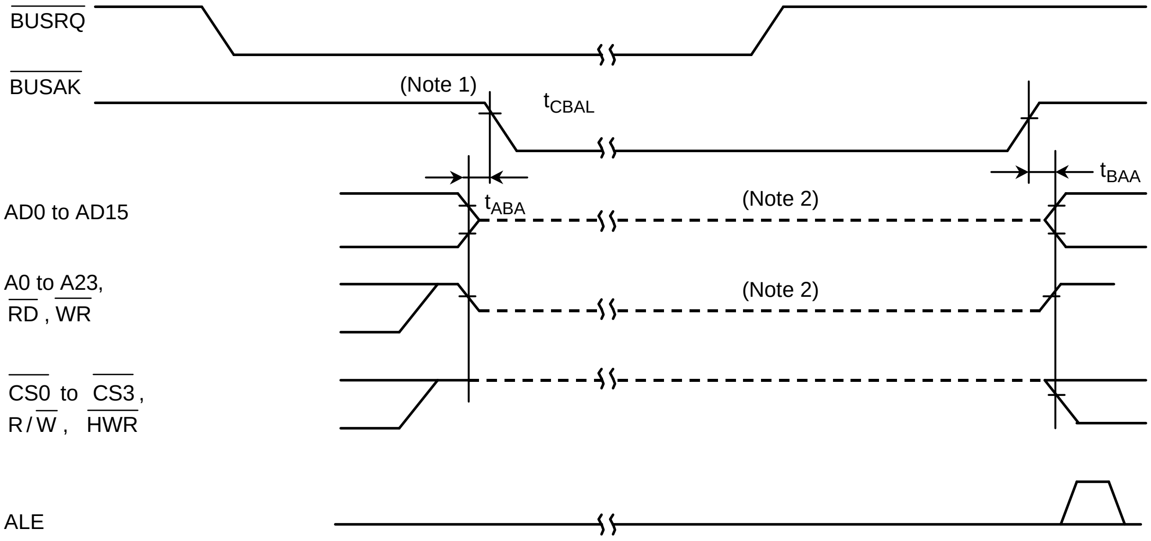
Symbol	Parameter	Variable		4 MHz		16 MHz		Condition	Unit
		Min	Max	Min	Max	Min	Max		
t_{SCH}	Low level Width	0.5T – 10		90		27		Vcc ≧ 2.7 V	ns
		0.5T – 30		70		-		Vcc < 2.7 V	
t_{SCL}	High level Width	0.5T – 10		90		27		Vcc ≧ 2.7 V	ns
		0.5T – 30		70		-		Vcc < 2.7 V	

Note: T = Period of SCOUT

Measuring Conditions

- Output Level: High = 0.7 V, Low = 0.3 V, CL = 10 pF

4.9 Bus Request/Bus Acknowledge



$\overline{\text{BUSRQ}}$
$\overline{\text{BUSAK}}$
(Note 1)
t_{CBAL}
t_{BAA}
AD0 to AD15
t_{ABA}
(Note 2)
A0 to A23, $\overline{\text{RD}}$, $\overline{\text{WR}}$
(Note 2)
$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, R/$\overline{\text{W}}$, $\overline{\text{HWR}}$
ALE

Symbol	Parameter	Variable		f_{FPH} = 4 MHz		f_{FPH} = 16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{ABA}	Output Buffer Off to $\overline{\text{BUSAK}}$ Low	0	80	0	80	0	80	ns
t_{BAA}	$\overline{\text{BUSAK}}$ High to Output Buffer On	0	80	0	80	0	80	ns

Note 1: Even if the $\overline{\text{BUSRQ}}$ Signal foes Low, the bus will not be released while the $\overline{\text{WAIT}}$ signal is Low. The bus will only be released when $\overline{\text{BUSRQ}}$ goes Low while $\overline{\text{WAIT}}$ is High.

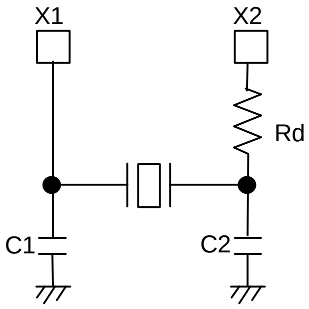
Note 2: This line shows only that the output buffer is in the off state.
It does not indicate that the signal level is fixed.
Just after the bus is released, the signal level set before the bus was released is maintained dynamically by the external capacitance. Therefore, to fix the signal level using 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 Recommended Crystal Oscillation Circuit

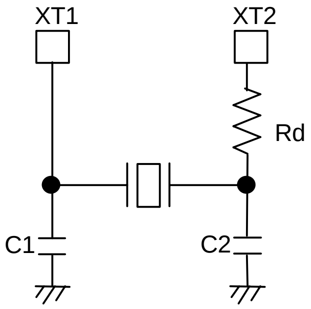
TMP91C824 is evaluated by below oscillator vender. When selecting external parts, make use of this information..

(note): 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 using C1 and C2 value in below table. When designing board, it should design minimum length pattern around oscillator. And we recommend that oscillator evaluation try on your actual using board.

(1) Connection example



High frequency oscillator



Low frequency oscillator

(2) TMP91C824F recommended ceramic oscillator : MURATA co. LTD; JAPAN

Circuit parameter recommended

MCU	Oscillation Frequency [MHZ]	Item of Oscillator	Parameter of elements				Running Condition	
			C1 [pF]	C2 [pF]	Rf [Ω]	Rd [Ω]	Voltage of Power [V]	Tc [℃]
TMP91C824	2.00M	CSTLS2M00G56-B0	(47)	(47)	Open	0	1.8 to 2.2	-40 to +85
	2.50M	CSTLS2M50G56-B0	(47)	(47)	Open	0		
	10.00M	CSTS1000MG03 *CSTLS10M0G53-B0	(15)	(15)	Open	0		
	12.50M	CSA12.5MTZ093 *CSALA12M5T55093-B0	30	30	Open	0		
		CST12.0MTW093 *CSTLA12M5T55093-B0	(30)	(30)	Open	0		

MCU	Oscillation Frequency [MHZ]	Item of Oscillator	Parameter of elements				Running Condition	
			C1 [pF]	C2 [pF]	Rf [Ω]	Rd [Ω]	Voltage of Power [V]	Tc [℃]
TMP91C824	4.00M	CSTS0400MG06 *CSTLS4M00G56-B0	(47)	(47)	Open	0	2.7 to 3.6	-40 to +85
	6.750M	CSTS0675MG06 *CSTLS6M75G56-B0	(47)	(47)	Open	0		
	12.50M	CSA12.5MTZ *CSALA12M5T55-B0	30	30	Open	0		
		CST12.0MTW *CSTLA12M5T55-B0	(30)	(30)	Open	0		
	20.00M	CSALS20M0X53-B0	5	5	Open	0		
		CSTLS20M0X51-B0	(5)	(5)	Open	0		
	27.00M	CSALS27M0X51-B0	Open	Open	10K	0		
	32.00M	CSALA32M0X51-B0	3	3	Open	0		

NOTE: In CST ***type oscillator, Capacitance C1, C2 is built in

* After 2001/06,new products will be made, and the old products(now in production) will not be made in MURATA CO., LTD. (JAPAN)

- 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

5. Table of SFRs

(SFR; special function register)

The SFRs include the I/O ports and peripheral control registers allocated to the 4K bytes 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 Gear

(6) DFM (Clock Doubler)

(7) 8-bit Timer

(8) UART/Serial Channel

(9) I^{2}CBUS/Serial Channel

(10) A/D Converter

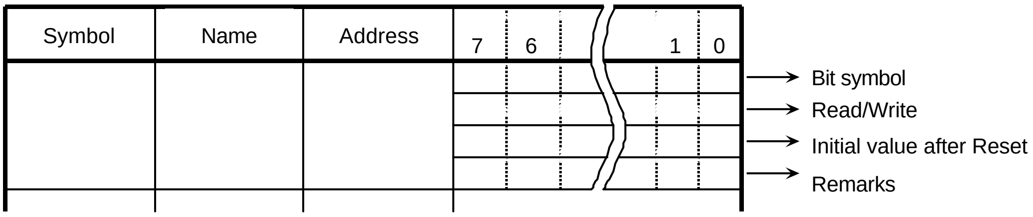
(11) Watchdog Timer

(12) RTC (Real–Time Clock)

(13) Melody/Alarm Generator

(14) MMU

Table layout



Note: “Prohibit RMW” in the table means that you cannot use RMW instructions on these register.

Example: When setting bit0 only of the registerP0CR, the instruction “SET 0, (0002G)” 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 as1)

Prohibit RMW; Read-Modify-Write instructions are prohibited. (The EX, ADD, ADC, BUS, SBC, INC, DEC, AND, OR, XOR, STCF, RES, SET, CHG, TEST, RLC, RRC, RL, RR, SLA, SRA, SLL, SRL, RLD and RRD instruction are read-modify-write instructions.)

Prohibit RMW*; Read-modify-write instructions are prohibited when controlling the pull-up resistor.

Table 5.1 Address map SFRs

[1], [2] PORT

Address	Name
0000H	
1H	P1
2H	
3H	
4H	P1CR
5H	
6H	P2
7H	
8H	
9H	P2FC
AH	P5CR
BH	P5FC
CH	
DH	P5
EH	
FH	

Address	Name
0010H	
1H	
2H	P6
3H	P7
4H	
5H	P6FC
6H	P7CR
7H	P7FC
8H	P8
9H	
AH	
BH	P6FC
CH	P7CR
DH	
EH	
FH	P7ODE

Address	Name
0022H	
1H	
2H	PB
3H	PC
4H	PBCR
5H	PBFC
6H	PCCR
7H	PCFC
8H	PCODE
9H	PD
AH	PDFC
BH	
CH	
DH	
EH	
FH	

Address	Name
0070H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	PZ
EH	PZCR
FH	PZFC

[3] 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	INTE12
2H	INTE3ALM4
3H	INTEALM01
4H	INTEALM23
5H	INTETA01
6H	INTETA23
7H	INTERTC
8H	INTES0
9H	INTES1
AH	INTES2
BH	INTETC01
CH	INTETC23
DH	INTEP01
EH	
FH	

[4] 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

[5], [6] CGEAR,DFM

Address	Name
00E0H	SYSCR0
1H	SYSCR1
2H	SYSCR2
3H	SYSCR0
4H	SYSCR1
5H	SYSCR2
6H	SYSCR3
7H	
8H	DFMCR0
9H	DFMCR1
AH	
BH	
CH	
DH	
EH	
FH	

Note: Do not access to the unnamed addresses, i.e. addresses to which no register has been allocated.

Table 5.2 Address map SFRs

[7] TMRA

Address	Name
0100H	TA01RUN
1H	
2H	TA0REG
3H	TA1REG
4H	TA01MOD
5H	TA01FFCR
6H	
7H	
8H	TA23RUN
9H	
AH	TA2REG
BH	TA3REG
CH	TA23MOD
DH	TA3FFCR
EH	
FH	

[8] UART/SIO

Address	Name
0200H	SC0BUF
1H	SC0CR
2H	SC0MOD0
3H	BR0CR
4H	BR0ADD
5H	SCMOD1
6H	
7H	SIRCR
8H	SC1BUF
9H	SC1CR
AH	SC1MOD0
BH	BR1CR
CH	BR1ADD
DH	SC1MOD1
EH	
FH	

[9] I2CBUS/SIO

Address	Name
0240H	SBI0CR1
1H	SBI0DBR
2H	I2C0AR
3H	SBI0CR2/SBI0SR
4H	SBI0BR0
5H	SBI0BR1
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[10] 10bit ADC

Address	Name
02A0H	ADREG04L
1H	ADREG04H
2H	ADREG15L
3H	ADREG15H
4H	ADREG26L
5H	ADREG26H
6H	ADREG37L
7H	ADREG37H
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Name
02B0H	ADMOD0
1H	ADMOD1
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Note: Do not access to the unnamed addresses, i.e. addresses to which no register has been allocated.

Table 5.3 Address map SFRs

[11] WDT

Address	Name
0300H	WDMOD
1H	WDCR
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[12] RTC

Address	Name
0320H	SECR
1H	MINR
2H	HOURR
3H	DAYR
4H	DATER
5H	MONTHR
6H	YEWRR
7H	PAGER
8H	RESTR
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[13] MLD

Address	Name
0330H	ALM
1H	MELALMC
2H	MELFL
3H	MELFH
4H	ALMINT
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[13] MMU

Address	Name
0350H	LOCAL0
1H	LOCAL1
2H	LOCAL2
3H	LOCAL3
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Note: Do not access to the unnamed addresses, i.e. addresses to which no register has been allocated.

(1) I/O Ports

Symbol	Name	Address	7	6	5	4	3	2	1	0
P1	PORT1	01H	P17	P16	P15	P14	P13	P12	P11	P10
			R/W							
			0	0	0	0	0	0	0	0
			Input Mode							
P2	PORT2	06H	P27	P26	P25	P24	P23	P22	P21	P20
			R/W							
			1	1	1	1	1	1	1	1
			Input Mode							
							Input Mode			
P5	PORT5	0DH		P56	P55	P54				
				R/W						
				1						
				Input Mode (Pull Up)						
P6	PORT6	12H	P67	P66	P65	P64	P63	P62	P61	P60
			R/W							
			1	1	1	1	1	0	1	1
P7	PORT7	13H						P72	P71	P70
			R/W							
								1	1	1
			Input Mode							
P8	PORT8	18H		P86	P85	P84	P83	P82	P81	P80
			R							
			Input Mode							
PB	PORTB	22H		PB6	PB5	PB4	PB3	PB2	PB1	PB0
			R/W							
				1	1	1	1	1	1	1
			Input Mode							
PC	PORTC	23H			PC5	PC4	PC3	PC2	PC1	PC0
					R/W					
					1	1	1	1	1	1
					Input Mode					
PD	PORTD	29H	PD7	PD6	PD5					
			R/W							
			1	1	1					
PZ	PORTZ	7DH					PZ3	PZ2		RDE
							R/W	R/W		
							1	1		1

(2) I/O Port Control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P1CR	PORT1 Control	04H (Prohibit RWM)	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
			W							
			0	0	0	0	0	0	0	0
			0: IN 1: OUT							
P2FC	PORT2 Function	09H (Prohibit RWM)	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
			W							
			1	1	1	1	1	1	1	1
			0: Port, 1:Address bus (A23 to A16)							
P5CR	PORT5 Control	0AH (Prohibit RWM)		P56C	P55C	P54C				
			W							
				0	0	0				
			0 : IN 1 : OUT							
P5FC	PORT5 Function	0BH (Prohibit RWM)			P55F	P54F				
			W							
					0	0				
					0: PORT 1: $\overline{\text{BUSAK}}$	0: PORT 1: $\overline{\text{BUSRQ}}$				
P6FC	PORT6 Function	15H (Prohibit RWM)			P65F	P64F	P63F	P62F	P61F	P60F
					W					
					0	0	0	0	0	0
			Always write 0		0: PORT 1: EA25	0: PORT 1: EA24	0: PORT 1: /CS3	0: PORT 1: /CS2	0: PORT 1: /CS1	0: PORT 1: /CS0
P6FC2	PORT6 Function2	1BH (Prohibit RWM)	P67F2	P66F2	P65F2	P64F2		P62F2		
			W					W		
			0					0		
			0:<P67F> 1:/CS2E	0: <P66F> 1: /CS2D	0: <P65F> 1: /CS2C	0: <P64F> 1: /CS2B	Always write 0	0: <P62F> 1: /CS2A	Always write 0	
P7CR	PORT7 Control	16H (Prohibit RWM)						P72C	P71C	P70C
			W							
								0	0	0
			0 : IN 1 : OUT							
P7FC	PORT7 Function	17H (Prohibit RWM)						P72F	P71F	P70F
			W							
								0	0	0
								0: PORT 1: SCL	0: PORT 1: SDA/SO	0: PORT 1: SCK
P7FC2	PORT7 Function2	1CH (Prohibit RWM)						P72F2	P71F2	P70F2
			W							
								0	0	0
									0: <P71F> 1: OPTTX0	PIN SELECT 0: RXD0(PC1) 1: PTRX0(P70)

I/O Port Control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P7ODE	PORT7 Open Drain	1FH (Prohibit RWM)						ODEP72	ODEP71	
								W		
								0	0	
								0: 3STATE 1: Open Drain		
PBCR	PORTB Control	24H (Prohibit RWM)		PB6C	PB5C	PB4C	PB3C	PB2C	PB1C	PB0C
			W							
				0	0	0	0	0	0	0
			0: IN 1: OUT							
PBFC	PORTB Function	25H (Prohibit RWM)		PB6F	PB5F	PB4F	PB3F	PB2F	PB1F	
				W	W	W	W	W	W	
				0	0	0	0	0	0	
				0: PORT 1: INT3	0: PORT 1: INT2	0: PORT 1: INT1	0: PORT 1: INT0	0: PORT 1:TA3OUT	0: PORT 1: TA1OUT	
PCCR	PORTC Control	26H (Prohibit RWM)			PC5C	PC4C	PC3C	PC2C	PC1C	PC0C
			W							
					0	0	0	0	0	0
			0: IN 1: OUT							
PCFC	PORTC Function	27H (Prohibit RWM)			PC5F		PC3F	PC2F		PC0F
					W		W	W		W
					0		0	0		0
					0: PORT 1 : SCLK1		0: PORT 1: TXD1	0: PORT 1 : SCLK0		0: PORT 1: TXD0
PCODE	PORTC Open Drain	28H (Prohibit RWM)					ODEPC3			ODEPC0
							W			W
							0			0
							0: CMOS 1:Open Drain			0: CMOS 1:Open Drain
PDFC	PORTD Function	2AH (Prohibit RWM)	PD7F	PD6F	PD5F					
			W	W	W					
			0	0	0					
			0: PORT 1:MLDALM	0: PORT 1: /ALARM	0: PORT 1: SCOUT					
PZCR	PORTZ Control	7DH (Prohibit RMW)					PZ3C	PZ2C		
							W			
							0	0		
							0:IN 1:OUT			
PZFC	PORTZ Function	7FH (Prohibit RMW)					PZ3F	PZ2F		
							W			
							0	0		
							0:PORT 1:R/W	0:PORT 1:/HWR		

(3) Interrupt Control (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE-0AD	Interrupt Enable 0 & A/D	90H	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTAD	Interrupt level			1: INT0	Interrupt level		
INTE12	Interrupt Enable 2/1	91H	INT2				INT1			
			I2C	I2M2	I2M1	I2M0	I1C	I1M2	I1M1	I1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INT2	Interrupt level			1: INT1	Interrupt level		
INTE3-ALM4	Interrupt Enable 3 & ALM4	92H	INTALM4				INT3			
			IA4C	IA4M2	IA4M1	IA4M0	I3C	I3M2	I3M1	I3M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INTALM4	Interrupt level			1: INT3	Interrupt level		
INTE-ALM01	Interrupt Enable ALM0/1	93H	INTALM1				INTALM0			
			IA1C	IA1M2	IA1M1	IA1M0	IA0C	IA0M2	IA0M1	IA0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INTALM1	Interrupt level			1:INTALM0	Interrupt level		
INTE-ALM23	Interrupt Enable ALM2/3	94H	INTALM3				INTALM2			
			IA3C	IA3M2	IA3M1	IA3M0	IA2C	IA2M2	IA2M1	IA2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INTALM3	Interrupt level			1:INTALM2	Interrupt level		
INTE-TA01	Interrupt Enable Timer A 1/0	95H	INTTA1(TMRA1)				INTTA0 (TMRA0)			
			ITA1C	ITA1M2	ITA1M1	ITA1M0	ITA0C	ITA0M2	ITA0M1	ITA0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTA1	Interrupt level			1: INTTA0	Interrupt level		
INTE-TA23	Interrupt Enable Timer A 3/2	96H	INTTA3 (TMRA5)				INTTA2 (TMRA4)			
			ITA3C	ITA3M2	ITA3M1	ITA3M0	ITA2C	ITA2M2	ITA2M1	ITA2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTA3	Interrupt level			1: INTTA2	Interrupt level		
INTE-RTC	Interrupt Enable RTC & KEY	97H					INTRTC			
							IRC	IRM2	IRM1	IRM0
							R	R/W		
							0	0	0	0
							1: INTRTC	Interrupt level		

(3) Interrupt Control (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTES0	Interrupt Enable Serial 0	98H	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTX0	Interrupt level			1: INTRX0	Interrupt level		
INTES1	Interrupt Enable Serial 1	99H	INTTX1				INTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1:INTTX1	Interrupt level			1:INTRX1	Interrupt level		
INTES2	Interrupt Enable Serial 2 /LCD	9AH					INTS2			
							IS2C	IS2M2	IS2M1	IS2M0
							R	R/W		
							0	0	0	0
							1:INTS2	Interrupt level		
INTETG01	Interrupt Enable TC0/1	9BH	INTTC1				INTTC0			
			ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETG23	Interrupt Enable TC2/3	9CH	INTTC3				ITC2M0			
			ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTEP01	Interrupt Enable PC0/1	9DH	INTP1				IP0M0			
			IP1C	IP1M2	IP1M1	IP1M0	IP0C	IP0M2	IP0M1	IP0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0

(3) Interrupt Control (3/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMA0V	DMA 0 Request Vector	80H			DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
					R/W					
					0	0	0	0	0	0
					DMA0 Start vector					
DMA1V	DMA 1 Request Vector	81H			DMA1V5	DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
					R/W					
					0	0	0	0	0	0
					DMA1 Start vector					
DMA2V	DMA 2 Request Vector	82H			DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
					R/W					
					0	0	0	0	0	0
					DMA2 Start vector					
DMA3V	DMA 3 Request Vector	83H			DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
					R/W					
					0	0	0	0	0	0
					DMA3 Start vector					
INTCLR	Interrupt Clear Control	88H (Prohibit RMW)			CLRV5	CLRV4	CLRV3	CLRV2	CLRV1	CLRV0
					W					
					–	–	–	–	–	–
					Clears interrupt request flag by writing to DMA start vector					
DMAR	DMA Software Request Register	89H					DMAR3	DMAR2	DMAR1	DMAR0
							R/W	R/W	R/W	R/W
							0	0	0	0
							1: DMA request in software			
DMAB	DMA Burst Request Register	8AH					DMAB3	DMAB2	DMAB1	DMAB0
							R/W	R/W	R/W	R/W
							0	0	0	0
							1 : DMA request on Burst Mode			
IIMC	Interrupt Input Mode Control	8CH (Prohibit RMW)			I3EDGE	I2EDGE	I1EDGE	I0EDGE	I0LE	NMIREE
			W		W	w	W	W	W	W
			0		0	0	0	0	0	0
			Always write 0	Always write 0	INT3 edge 0: Rising 1: Falling	INT2 edge 0: Rising 1: Falling	INT1 edge 0: Rising 1: Falling	INT0 edge 0: Rising 1: Falling	INT0 0: edge 1:level	1: operation even on $\overline{\text{NMI}}$ rising edge

(4) Chip Select/Wait Control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
B0CS	Block 0 CS/WAIT control Register	C0H (Prohibit RMW)	B0E		B00M1	B00M0	B0BUS	B0W2	B0W1	B0W0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0

			0: DIS
			1: EN
					00: ROM/SRAM
					01: }
					10: } Reserved
					11: }
							Data bus width
							0: 16 bit
							1: 8 bit
								000: 2WAIT 100: Reserved
								001: 1WAIT 101: 3WAIT
								010: 1 + NWAIT 110: 4WAIT
								011: 0WAIT 111: 8WAIT		

B1CS	Block 1 CS/WAIT control Register	C1H (Prohibit RMW)	B1E		B10M1	B10M0	B1BUS	B1W2	B1W1	B1W0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0

			0: DIS
			1: EN
					00: ROM/SRAM
					01: }
					10: } Reserved
					11: }
							Data bus width
							0: 16 bit
							1: 8 bit
								000: 2WAIT 100: Reserved
								001: 1WAIT 101: 3WAIT
								010: 1 + NWAIT 110: 4WAIT
								011: 0WAIT 111: 8WAIT		

B2CS	Block 2 CS/WAIT control Register	C2H (Prohibit RMW)	B2E	B2M	B20M1	B20M0	B2BUS	B2W2	B2W1	B2W0
			W	W	W	W	W	W	W	W
			1	0	0	0	0	0	0	0

			0: DIS
			1: EN
				0: 16 M Area
				1: Area set
					00: ROM/SRAM
					01: }
					10: } Reserved
					11: }
							Data bus width
							0: 16 bit
							1: 8 bit
								000: 2WAIT 100: Reserved
								001: 1WAIT 101: 3WAIT
								010: 1 + NWAIT 110: 4WAIT
								011: 0WAIT 111: 8WAIT		

B3CS	Block 3 CS/WAIT control Register	C3H (Prohibit RMW)	B3E		B30M1	B30M0	B3BUS	B3W2	B3W1	B3W0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0

			0: DIS
			1: EN
					00: ROM/SRAM
					01: }
					10: } Reserved
					11: }
							Data bus width
							0: 16 bit
							1: 8 bit
								000: 2WAIT 100: Reserved
								001: 1WAIT 101: 3WAIT
								010: 1 + NWAIT 110: 4WAIT
								011: 0WAIT 111: 8WAIT		

BEXCS	External CS/WAIT control Register	C7H (Prohibit RMW)					BEXBUS	BEXW2	BEXW1	BEXW0
							W	W	W	W
							0	0	0	0
							Data bus width 0: 16 bit 1: 8 bit	000: 2WAIT 100: Reserved 001: 1WAIT 101: 3WAIT 010: 1 + NWAIT 110: 4WAIT 011: 0WAIT 111: 8WAIT		
MSAR0	Memory Start Address Reg0	C8H	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			Start address A23 to A16							
MAMR0	Memory Address Mask Reg0	C9H	V20	V19	V18	V17	V16	V15	V14~9	V8
			R/W							
			1	1	1	1	1	1	1	1
			CS0 area size 0: enable to address comparison							
MSAR1	Memory Start Address Reg1	CAH	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			Start address A23 to A16							
MAMR1	Memory Address Mask Reg1	CBH	V21	V20	V19	V18	V17	V16	V15~9	V8
			R/W							
			1	1	1	1	1	1	1	
			CS0 area size 0: enable to address comparison							

Chip Select/Wait Control (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
MSAR2	Memory Start Address Reg2	CCH	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			Start address A23 to A16							
MAMR2	Memory Address Mask Reg2	CDH	V22	V21	V20	V19	V18	V17	V16	V15
			R/W							
			1	1	1	1	1	1	1	1
			CS0 area size 0: enable to address comparison							
MSAR3	Memory Start Address Reg3	CEH	S23	S22	S21	S20	S19	S18	S17	S16
			R/W							
			1	1	1	1	1	1	1	1
			Start address A23 to A16							
MAMR3	Memory Address Mask Reg3	CFH	V22	V21	V20	V19	V18	V17	V16	V15
			R/W							
			1	1	1	1	1	1	1	1
			CS0 area size 0: enable to address comparison							

(5) Clock Gear (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SYSCR0	System Clock Control Registe 0	E0H	XEN	XTEN	RXEN	RXTEN	RSYSCK	WUEF	PRCK1	PRCK0
			R/W							
			1	0	1	0	0	0	0	0
			High-frequency oscillator (fc) 0: stopped 1: oscillation	Low-frequency oscillator (fs) 0: stopped 1: oscillation	High-frequency oscillator (fc after release of STOP Mode 0: stopped 1: oscillation	Low-frequency oscillator (fs after release of STOP Mode 0: stopped 1: oscillation	Select clock after release of STOP Mode 0: fc 1: fs	Warm-up timer 0 write: Don’t care 1 write: start timer 0 read: end warm-up 1 read: not end warm-up	Select prescaler clock 00: f_{FPH} 01: reserved 10: fc/16 11: reserved	
SYSCR1	System Clock Control Register 1	E1H					SYSCK	GEAR2	GEAR1	GEAR0
							R/W			
							0	1	0	0
							System clock selection 0: fc 1: fs (Note 2)	High-frequency gear value selection (fc) 000: fc 001: fc/2 010: fc/4 011: fc/8 100: fc/16 101: (reserved) 110: (reserved) 111: (reserved)		
SYSCR2	System Clock Control Register 2	E2H		SCOSEL	WUPTM1	WUPTM0	HALTM1	HALTM0		DRVE
				R/W	R/W	R/W	R/W	R/W		R/W
				0	1	0	1	1		0
				0: fs 1: f_{FPH}	Warming-up time 00: reserved 01: 2^8/inputt frequency 10: 2^{14} 11: 2^{16}		00: reserved 01: STOP Mode 10: IDLE1 Mode 11: IDLE2 Mode		<Drive> Mode Select 0:STOP 1:IDLE	1: Drive the pin in STOP / IDLE1 Mode

(5) Clock Gear (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
EMCCR0	EMC Control Register 0	E3H	PROTECT	TA3LCDE				EXTIN	DRVOSCH	DRVOSCL
			R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	1	0	0	0	1	1
			Protection flag 0: OFF 1: ON	LCDC Source clk 0: 32KHz 1: TA3OUT	Always write 1	Always write 0	e	1: fc is external clock.	fc oscillator drivability 1: Normal 0: Weak	fs oscillator drivability 1: Normal 0: Weak
EMCCR1	EMC Control Register 1	E4H	Follows: 1^{ST}-KEY and 2^{ND}-KEY needs to lock Continuation writes in 1^{ST}-KEY:EMCCR1=5AH,EMCCR2=A5H. Continuation writes in 2^{ND}-KEY:EMCCR1=A5H, EMCCR2=5AH.							
EMCCR2	EMC Control Register 2	E5H								
EMCCR3	EMC Control Register 3	E6H		ENFROM	ENDROM	ENPROM		FFLAG	DFLAG	PFLAG
				R/W	R/W	R/W		R/W	R/W	R/W
				0	0	0		0	0	0
				CS1A area det ect enable 0:disable 1:enable	CS2B-2G area detect Enable 0:disable 1:enable	CS2A area detect enable 0:disable 1:enable		CS1A Write operation flag	CS2B-2G Write operation flag	CS2A Write operation flag
								When reading "0": not written "1": written	When writing "0": clear flag	

(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/W	R	R/W				
			0	0	0	0				
			DFM LUP f_{FPH} 00 STOP STOP f_{OSCH} 01 RUN RUN f_{OSCH} 10 RUN STOP f_{DFM} 11 RUN STOP f_{OSCH}		Lock-up falg 0: End LUP 1: Do not end LUP	Lock-up time 0: $2^{12}/f_{OSCH}$ 1: $2^{10}/f_{OSCH}$				
DFMCR1	DFM Control Register 1	E9H								
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	1	0	0	1	1
			DFM correction Input frequency 4 ~ 6.75MHz (@2.7-3.6V) : write 0BH Input frequency 2 ~ 2.5MHz (@2.0V ±10%) : write 1BH							

(7) 8-Bit Timer

(7–1) TMRA01

Symbol	Name	Address	7	6	5	4	3	2	1	0
TA01–RUN	Timer RUN	100H	TA0RDE				I2TA01	TA01PRUN	TA1RUN	TA0RUN
			R/W				R/W	R/W	R/W	R/W
			0				0	0	0	0
			Double Buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	8-Bit Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		
TA0REG	8-Bit Timer Register 0	102H (Prohibit RMW)	–							
			W							
			Undefined							
TA1REG	8-Bit Timer Register 1	103H (Prohibit RMW)	–							
			W							
			Undefined							
TA01–MOD	8-Bit Timer Source CLK & MODE	104H	TA01M1	TA01M0	PWM01	PWM00	TA1CLK1	TA1CLK0	TA0CLK1	TA0CLK0
			R/W							
			0	0	0	0	0	0	0	0
			00: 8-Bit Timer 01: 16-Bit Timer 10: 8-Bit PPG 11: 8-Bit PWM		00: Reserved 01: $2^6 - 1$ PWM cycle 10: $2^7 - 1$ 11: $2^8 - 1$		00: TA0TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: TA0IN pin 01: ϕT1 10: ϕT4 11: ϕT16	
TA1FFCR	8-Bit Timer Flip-Flop Control	105H (Prohibit RMW)					TAFF1C1	TAFF1C0	TAFF1IE	TAFF1IS
							R/W		R/W	
							1	1	0	0
							00: Invert TA1FF 01: Set TA1FF 10: Clear TA1FF 11: Don't care		1: TA1FF Invert Enable	0: TMRA0 1: TMRA1 inversion

(7–2) TMRA23

Symbol	Name	Address	7	6	5	4	3	2	1	0
TA23-RUN	Timer RUN	108H	TA2RDE				I2TA23	TA23PRUN	TA3RUN	TA2RUN
			R/W				R/W	R/W	R/W	R/W
			0				0	0	0	0
			Double Buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	8-Bit Timer Run/Stop control 0: Stop & Clear 1: Run (count up)		
TA2REG	8-Bit Timer Register 0	10AH (Prohibit RMW)	–							
			W							
			Undefined							
TA3REG	8-Bit Timer Register 1	10BH (Prohibit RMW)	–							
			W							
			Undefined							
TA23-MOD	8-Bit Timer Source CLK & MODE	10CH	TA23M1	TA23M0	PWM21	PWM20	TA3CLK1	TA3CLK0	TA2CLK1	TA2CLK0
			R/W							
			0	0	0	0	0	0	0	0
			00: 8-Bit Timer 01: 16-Bit Timer 10: 8-Bit PPG 11: 8-Bit PWM		00: Reserved 01: $2^6 -1$ PWM cycle 10: $2^7 - 1$ 11: $2^8 - 1$		00: TA2TRG 01: ϕT1 10: ϕT16 11: ϕT256		00: Reserved 01: ϕT1 10: ϕT4 11: ϕT16	
TA3FFCR	8-Bit Timer Flip-Flop Control	10DH (Prohibit RMW)					TAFF3C1	TAFF3C0	TAFF3IE	TAFF3IS
							R/W		R/W	
							1	1	0	0
							00: Invert TA3FF 01: Set TA3FF 10: Clear TA1FF 11: Don't care		1: TA3FF Invert Enable	0: TMRA2 1: TMRA3 inversion

(8) UART/Serial Channel (1/2)

(8-1) UART/SIO Channel 0

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC0BUF	Serial Channel 0 Buffer	200H	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
			R (Receiving)/W (Transmission)							
			Undefined							
SC0CR	Serial Channel 0 Control	201H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Cleared to 0 by reading)			R/W	
			0	0	0	0	0	0	0	0
			Receiving data bit 8	Parity 0: Odd 1: Even	1: Parity Enable	1: Error Over Run	Parity	Framing	0:SCLK0↑ 1:SCLK0↓	1: Input SCLK0 pin
SC0-MOD0	Serial Channel 0 Mode0	202H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Transmission data bit 8	1: CTS Enable	1: Receive Enable	1: Wake-up Enable	00: I/O Interface 01: UART 7-Bit 10: UART 8-Bit 11: UART 9-Bit		00: TA0TRG 01: baud rate generator 10: internal clock f_{SYS} 11: external clock SCLK0	
BR0CR	Baud Rate Control	203H	—	BR0ADD	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
			R/W							
			0	0	0		0	0	0	0
			Always write 0.	1: (16-K) /16 divided Enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Set the dividiing value. 0 to F			
BR0-ADD	Serial Channel 0 K setting Reg	204H					BR0K3	BR0K2	BR0K1	BR0K0
							R/W			
							0	0	0	0
							Baud Rate0 K 1 to F			
SC0-MOD1	Serial Channel 0 Mode1	205H	I2S0	FDPX0						
			R/W	R/W						
			0	0						
			IDLE2 0: Stop 1: Operate	I/O interface 1: Full Duplex 0: Half Duplex						

(8-2) IrDA

Symbol	Name	Address	7	6	5	4	3	2	1	0
SIRCR	IrDA Control Register	207H	PLSEL	RXSEL	TXEN	RXEN	SIRWD3	SIRWD2	SIRWD1	SIRWD0
			R/W	R/W	R/W	R/W	R/W			
			0	0	0	0	0	0	0	0
			Transmission pulse width 0: 3/16 1: 1/16	Receiving Data 0: "H" pulse 1: "L" pulse	Transmission 0: Disable 1: Enable	Receiving 0: Disable 1: Enable	Set the effective SIRRxD pulse width Pulse width more than "2x × (set value + 1)" Possible : 1 to 14 Not possible : 0, 15			

UART/Serial Channel (2/2)

(8-3) UART/SIO Channel1

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC1BUF	Serial Channel 1 Buffer	208H	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 by reading)			R/W	
			0	0	0	0	0	0	0	0
			Receiving data bit 8	Parity 0: Odd 1: Even	1: Parity Enable	1: Error			0: SCLK1↑ 1: SCLK1↓	1: Input SCLK1 pin
						Over Run	Parity	Framing		
SC1-MOD0	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 bit 8	1: CTS Enable	1: Receive Enable	1: Wake up Enable	00: I/O Interface 01: UART 7 bit 10: UART 8 bit 11: UART 9 bit		00: TA0TRG 01: baud rate generater 10: internal clock f_{SYS} 11: external clock SCLK1	
BR1CR	Baud Rate Control	20BH	—	BR1ADD	BR1CK1	BR1CK	BR1S3	BR1S2	BR1S1	BR1S0
			R/W							
			0	0	0	0	0	0	0	0
			Always write 0.	1: (16-K)/16 divided Enable	00: φT0 01: φT2 10: φT8 11: φT32		Dividing value 0 to F			
BR1-ADD	Serial Channel 1 K setting Reg	20CH					BR1K3	BR1K2	BR1K1	BR1K0
							R/W			
							0	0	0	0
							Set the frequency divisor K 1 to F			
SC1-MOD1	Serial Channel 1 Mode1	20DH	I2S1	FDPX1						
			R/W	R/W						
			0	0						
			IDLE2 0: Stop 1: Operate	I/O interface mode 1: full Duplex 0: Half Duplex						

(9) I^{2}CBUS/Serial Interface

Symbol	Name	Address	7	6	5	4	3	2	1	0
SBI0CR1	Serial Bus Interface Control Register 1	240H (I2C Bus Mode) (Prohibit RMW)	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0 /SWRMON
			W			R/W		W	W	R/W
			0	0	0	0		0	0	0
			Number of transfer bits 000: 8, 001: 1, 010: 2 011: 3, 100: 4, 101: 5 110: 6, 111: 7			Acknowledge Mode 0: Disable 1: Enable		Setting for the devisor value n 000: 4, 001: 5, 010: 6 011: 7, 100: 8, 101: 9 110: 10, 111: (reserved)		
		240H (SIO Mode) (Prohibit RMW)	SIOS	SIOINH	SIOM1	SIOM0		SCK2	SCK1	SCK0 /SWRMON
			W	W	W	W		W	W	R/W
			0	0	0	0		0	0	0
			Transfer 0: Stop 1: Start	Transfer 0: Continue 1: Abort	Transfer mode 00: 8-Bit Transmit Mode 10: 8-Bit Transmit/Receive Mode 11: 8-Bit received Mode			Setting for the divisor value n 000: 3, 001: 4, 010 : 5 011: 6, 100: 7, 101 : 8 110: 9, 111: SCK pin		
SBI0-DBR	SBI Buffer Register	241H (Prohibit RMW)	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	PB0/TB0
			R (receiving)/W (transmission)							
			Undefined							
I2C0AR	I2CBUS Address Register	242H (Prohibit RMW)	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
			W	W	W	W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			Setting slave address							Address recognition 0: Enable 1: Disable
SBI0-CR2 (SBI0SR)	Serial Bus Interface Control Register 2	243H (I^{2}C bus Mode) (Prohibit RMW)	MST	TRX	BB	PIN	AL/SBIM1	AAS/SBIM0	AD0/ SWRST	LRB/ SWRST 0
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	1	0	0	0	0
			0: Slave 1: Master	0: receiver 1: transmit	Bus status monitor 0: Free 1: Busy	INTS2 request monitor 0: Request 1: Cancel	Arbitration lost detection monitor 1: Detect	Slave address match detection monitor 1: Detect	GENERAL CALL detection monitor 1: Detect	Lost receive bit monitor 0: 0 1: 1
		243H (SIO Mode) (Prohibit RMW)	—	—	—	—	SIOF	SEF	—	—
							R	R		
							0	0		
							Transfer status monitor 0: stopped 1: terminated in process	Shift operation status monitor 0: stopped 1: terminated in process		
SBI0-BR0	Serial Bus Interface Baud Rate Register 0	244H		I2SBI0						
				R/W						
				0						
			Always write 0.	IDLE2 0: Abort 1: Operate						
SBI0-BR1	Serial Bus Interface Baud Rate Register 1	245H	P4EN							
			R/W							
			0							
			Internal Clock 0: Abort 1: Operate							

(10) A/D Converter

Symbol	Name	Address	7	6	5	4	3	2	1	0
ADMOD0	A/D MODE Reg0	2B0H	EOCF	ADBF	—	ITM1	ITM0	REPEAT	SCAN	ADS
			R		R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			1: End	1: busy		Interrupt in Repeat Mode		1: Repeat	1: Scan	1: Start
ADMOD1	A/D MODE Reg1	2B1H	VREFON	I2AD			ADTRGE	ADCH2	ADCH1	ADCH0
			R/W	R/W			R/W	R/W		
			0	0			0	0	0	0
			1: VREF On	IDLE2 0: Abort 1: Operate			1: Enable for external start	Input channel 000: AN0 AN0 001: AN1 AN0 →AN1 010: AN2 AN0 → AN1 → AN2 011: AN3 AN0 → AN1 → AN2 → AN3 100: AN4 AN4 101: AN5 AN4 → AN5 110: AN6 AN4 → AN5 → AN6 111: AN7 AN4 → AN5 → AN6 → AN7		
ADREG04L	AD Result Reg 0/4 low	2A0H	ADR01	ADR00						ADR0RF
			R							R
			Undefined							0
ADREG04H	AD Result Reg 0/4 high	2A1H	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
			R							
			Undefined							
ADREG15L	AD Result Reg 1/5 low	2A2H	ADR11	ADR10						ADR1RF
			R							R
			Undefined							0
ADREG15H	AD Result Reg 1/5 high	2A3H	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
			R							
			Undefined							
ADREG26L	AD Result Reg 2/6 low	2A4H	ADR21	ADR20						ADR2RF
			R							R
			Undefined							0
ADREG26H	AD Result Reg 2/6 high	2A5H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
			R							
			Undefined							
ADREG37L	AD Result Reg 3/7 low	2A6H	ADR31	ADR30	—	—	—	—	—	ADR3RF
			R							R
			Undefined							0
ADREG37H	AD Result Reg 3/7 high	2A7H	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
			R							
			Undefined							

(11) Watchdog Timer

Symbol	Name	Address	7	6	5	4	3	2	1	0
WDMOD	WDT MODE Reg	300H	WDTE	WDTP1	WDTP0			I2WDT	RESCR	—
			R/W	R/W	R/W			R/W	R/W	R/W
			1	0	0			0	0	0
			1: WDT Enable	00: $2^{15}/f_{sys}$ 01: $2^{17}/f_{sys}$ 10: $2^{19}/f_{sys}$ 11: $2^{21}/f_{sys}$				IDLE2 0: Abort 1: Operate	1: RESET connect internally WDT out to Reset pin	Always write 0.
WDCR	WD Control	301H	—							
			W							
			—							
			B1H: WDT Disable 4EH: WDT Clear							

(12) RTC (Real-Time Clock)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SECR	Second Reg	320H		SE6	SE5	SE4	SE3	SE2	SE1	SE0
				R/W						
				Undefined						
			“0”	40 sec.	20 sec.	10 sec.	8 sec.	4 sec.	2 sec.	1 sec.
MINR	Minute Reg	321H		MI6	MI5	MI4	MI3	MI2	MI1	MI0
				R/W						
				Undefined						
			“0”	40 min.	20 min.	10 min.	8 min.	4 min.	2 min.	1min.
HOURR	Hour Reg	322H			HO5	HO4	HO3	HO2	HO1	HO0
					R/W					
					Undefined					
			“0”	“0”	20 hour (PM/AM)	10 hour	8 hour	4 hour	2 hour	1 hour
DAYR	Day Reg	323H						WE2	WE1	WE0
								R/W		
								Undefined		
			“0”	“0”	“0”	“0”	“0”	WE2	WE1	WE0
DATER	Date Reg	324H			DA5	DA4	DA3	DA2	DA1	DA0
					R/W					
					Undefined					
			“0”	“0”	20 day	10 day	8 day	4 day	2 day	1 day
MONTHR	Month Reg	325H				MO4	MO3	MO2	MO1	MO0
						R/W				
						Undefined				
		PAGE0	“0”	“0”	“0”	10 month	8 month	4 month	2 month	1 month
		PAGE1				--	--	--	--	0: Indicator for 12 hours 1: Indicator for 24 hours
YEARR	Year Reg	326H	YE7	YE6	YE5	YE4	YE3	YE2	YE1	YE0
			R/W							
			Undefined							
		PAGE0	80 year	40 year	20 year	10 year	8 year	4 year	2 year	1 year
		PAGE1	--	--	--	--	--	--	Leap year setting	
PAGER	Page Reg(Prohibit RMW)	327H	INTRTC			ADJUST	ENATMR	ENAALM		PAGE
			R/W			W	R/W			R/W
			0			--	Undefined			Undefined
			INT ENABLE			ADJUST	TIMER ENABLE	ALARM ENABLE		PAGE setting
RESTR	Reset Reg(Prohibit RMW)	328H	DIS1HZ	DIS16HZ	RSTTMR	RSTALM	RE3	RE2	RE1	RE0
			W							
			Undefined							
			0: 1HZ	0: 16HZ	1: RESET TIMER	1:RESET ALARM	Always write 0.			

(13) Melody/Alarm Generator

Symbol	Name	Address	7	6	5	4	3	2	1	0
ALM	Alarm – Pattern Reg	330H	AL8	AL7	AL6	AL5	AL4	AL3	AL2	AL1
			R/W							
			0							
			Alarm –Pattern set							
MEL-ALMC	Melody/ Alarm Control Reg	331H	FC1	FC0	ALMINV					MELALM
			R/W		R/W					R/W
			0		0					0
			Free-run counter Control 00: Hold 01: Restart 10: Clear 11: Clear & Start		Alarm Frequency Invert 1: Invert	Always write 0				Output Frequency 0: Alarm 1: Melody
MELFL	Melody Frequency L-Reg	332H	ML7	ML6	ML5	ML4	ML3	ML2	ML1	ML0
			R/W							
			0							
			Melody Frequency set (low 8bit)							
MELFH	Melody Frequency H-Reg	333H	MELON				ML11	ML10	ML9	ML8
			R/W				R/W			
			0				0			
			Melody counter Control 0: Stop and Clear 1: Start				Melody Frequency set (high 4bit)			
ALMINT	Alarm Interrupt Enable Reg	334H				IALM4E	IALM3E	IALM2E	IALM1E	IALM0E
						R/W				
						0				
					Always write 0	INTALM4 to INTALM0 Alarm Interrupt Enable				

(14) MMU

Symbol	Name	Address	7	6	5	4	3	2	1	0
LOCAL 0	LOCAL0 Control Reg	350H	L0E					L0EA22	L0EA21	L0EA20
			R/W					R/W		
			0					0		
			0: Disable 1: Enable					LOCAL0 area BANK set		
LOCAL 1	LOCAL1 Control Reg	351H	L1E					L1EA23	L1EA22	L1EA21
			R/W					R/W		
			0					0		
			0: Disable 1: Enable					LOCAL1 area ANK set		
LOCAL 2	LOCAL2 Control Reg	352H	L2E					L2EA23	L2EA22	L2EA21
			R/W					R/W		
			0					0		
			0: Disable 1: Enable					LOCAL2 area BANK set		
LOCAL 3	LOCAL3 Control Reg	353H	L3E			L3EA26	L3EA25	L3EA24	L3EA23	L3EA22
			R/W			R/W				
			0			0				
			0: Disable 1: Enable			LOCAL3 area BANK set				

6. Points to Note and Restrictions

(1) Notation

1. The notation for built-in / I/O registers is as follows register symbol <bit symbol>

   e.g.) 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

```
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)           RR     (mem)
SLA    (mem)           SRA    (mem)
SLL    (mem)           SRL    (mem)
RLD    (mem)           RRD    (mem)
```

3. fc, fs, f_{FPH}, f_{SYS} and one state

   The clock frequency input on ins X1 and 2 is called f_{OSCH}. The clock selected by DFMCR0 <ACT1~ACT0> is called fc.

   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 f_{SYS} is referred to as one state.

(2) Points to note

a) AM0 and AM1 pins

This pin is connected to the VCC or the VSS pin. Do not alter the level when the pin is active.

b) EMU0 and EMU1

Open pins.

c) Reserved address areas

The TMP91C815 does not have any reserved areas.

d) 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.

e) Programmable pull-up resistance

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. Px) are used to turn the pull-up/-down resistors ON/OFF. Consequently read-Modify-write instructions are prohibited.

f) Bus release function

It is described note point in "3.5 Port Function" that pin's condition at bus release condition.

Please refer that.

g) Watchdog timer

The watchdog timer starts operation immediately after a Reset is released. When the watchdog timer is not to be used, disable it.

h) AD converter

The string resistor between the VREFH and VREFL pins can be cut by a program so as to reduce power consumption. When STOP Mode is used, disable the resistor using the program before the HALT instruction is executed.

i) 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)).

j) Undefined SFR

The value of an undefined bit in an SFR is undefined when read.

k) POP SR instruction

Please execute the POP SR instruction during DI condition.

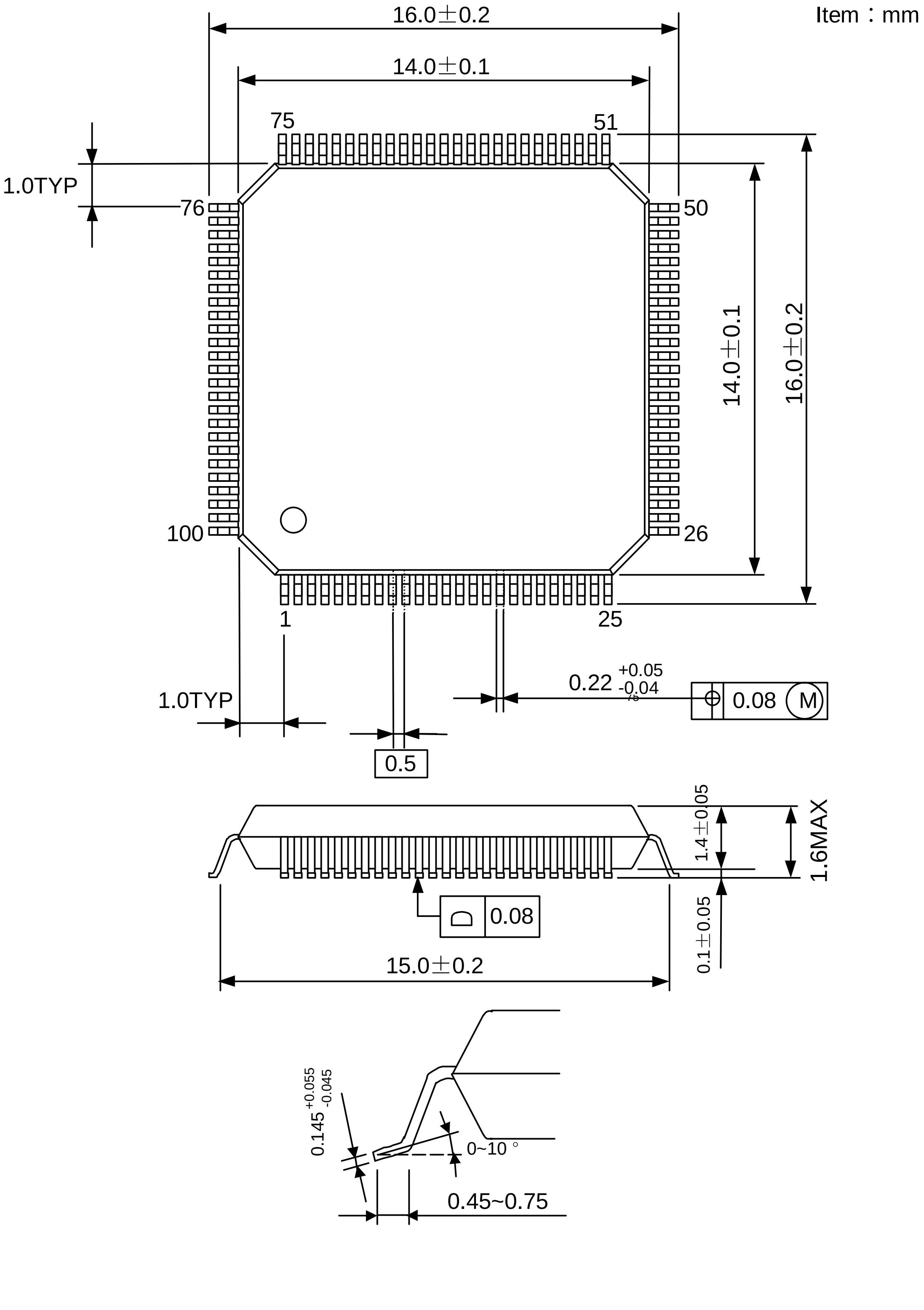
l) Releasing the HALT mode by requesting an interruption

Usually, interrupts can release all halts status. However, the interrupts (/NMI,INT0-3,INTKEY,INTRTC,INTALM0 to 4) 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).(In this case,an interrupt request is kept on hold internally)

If another interrupt is generated after it has shifted to HALT mode completely, release 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. 100 pin QFP (Flat Package)

PACKAGE NAME: P-LQFP100-P-1414-0.5D


Item：mm
16.0±0.2
14.0±0.1
75
51
1.0TYP
76
50
14.0±0.1
16.0±0.2
100
26
1
25
0.22 +0.05 -0.04
⊕ 0.08 Ⓜ
1.0TYP
0.5
1.4±0.05
1.6MAX
0.08
0.1±0.05
15.0±0.2
0.145 +0.055 -0.045
0~10 °
0.45~0.75