TOSHIBA

TOSHIBA Original CMOS 16-Bit Microcontroller

TLCS-900/L1 Series

TMP91C016FG
JTMP91C016S

TOSHIBA CORPORATION

Semiconductor Company

Preface

Thank you very much for making use of Toshiba microcomputer LSIs.
Before use this LSI, refer the section, “Points of Note and Restrictions”.
Especially, take care below cautions.

CAUTION
How to release the HALT mode

Usually, interrupts can release all halts status. However, the interrupts = ($\overline{\text{NMI}}$, INT0 to INT3, INTRTC, INTALM0 to INTALM4, INTKEY, INTVLD0 to INTVLD2), which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 5 clocks of f_{FPH}) with IDLE1 or STOP mode (IDLE2 is not applicable to this case). (In this case, an interrupt request is kept on hold internally.)

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

CMOS 16-Bit Microcontrollers

TMP91C016FG/JTMP91C016S

1. Outline and Features

TMP91C016 is a high-speed 16-bit microcontroller designed for the control of various mid- to large-scale equipment.

TMP91C016FG comes in a 100-pin flat package. JTMP91C016S is a 100-pad chip product.

Listed below are the features.

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

- Instruction mnemonics are upward compatible with TLCS-90/900
- 16 Mbytes of linear address space
- General-purpose registers and register banks
- 16-bit multiplication and division instructions; bit transfer and arithmetic instructions
- Micro DMA: 4 channels (592ns/ 2bytes at 27MHz)

(2) Minimum instruction execution time: 148 ns (at 27 MHz)

RESTRICTIONS ON PRODUCT USE

20070701-EN

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

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

(4) External memory expansion
- Expandable up to 105 Mbytes (Shared program/data area)
- Can simultaneously support 8-/16-bit width external data bus: Dynamic data bus sizing
- Separate bus system

(5) 8-bit timers: 4 channels

(6) General-purpose serial interface: 2 channels

Channel 0
- UART mode
- IrDA Ver.1.0 (115.2 kbps) mode selectable

Channel 1
- UART mode
- Synchronous mode selectable

(7) LCD controller
- Adapt to both Shift register type and Built-in RAM type LCD driver

(8) Timer for real time clock (RTC)
- Based on TC8521A

(9) Key-on wakeup (Interrupt key input)

(10) Watchdog timer

(11) Melody/alarm generator
- Melody: Output of clock 4 to 5461 Hz
- 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) MMU
- Expandable up to 105 Mbytes (4 local area/8 bank method)

(14) Display data reciprocal conversion function between the vertical and horizontal (8 × 8)

(15) Interrupts: 40 interrupts
- 9 CPU interrupts: Software interrupt instruction and illegal instruction
- 25 internal interrupts: 7 priority levels are selectable
- 9 external interrupts: 7 priority levels are selectable
  (among 4 interrupts are selectable edge mode)

(16) Input/output ports: 31 pins (at External 16-bit data bus memory)

(17) Standby function
Three HALT modes: IDLE2 (Programmable), IDLE1 and STOP

(18) DRAM controller
- $\overline{2CAS}$ mode

(19) Voltage compare circuit: 3 channels

(20) Triple-clock controller

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

(21) Operating voltage

- VCC = 2.7 V to 3.6 V (fc max = 27 MHz)
- VCC = 1.8 V to 3.6 V (fc max = 10 MHz)

(22) Package

- 100-pin QFP: LQFP100-P-1414-0.50F
- Chip form supply also available. For details, contact your local Toshiba sales representative.

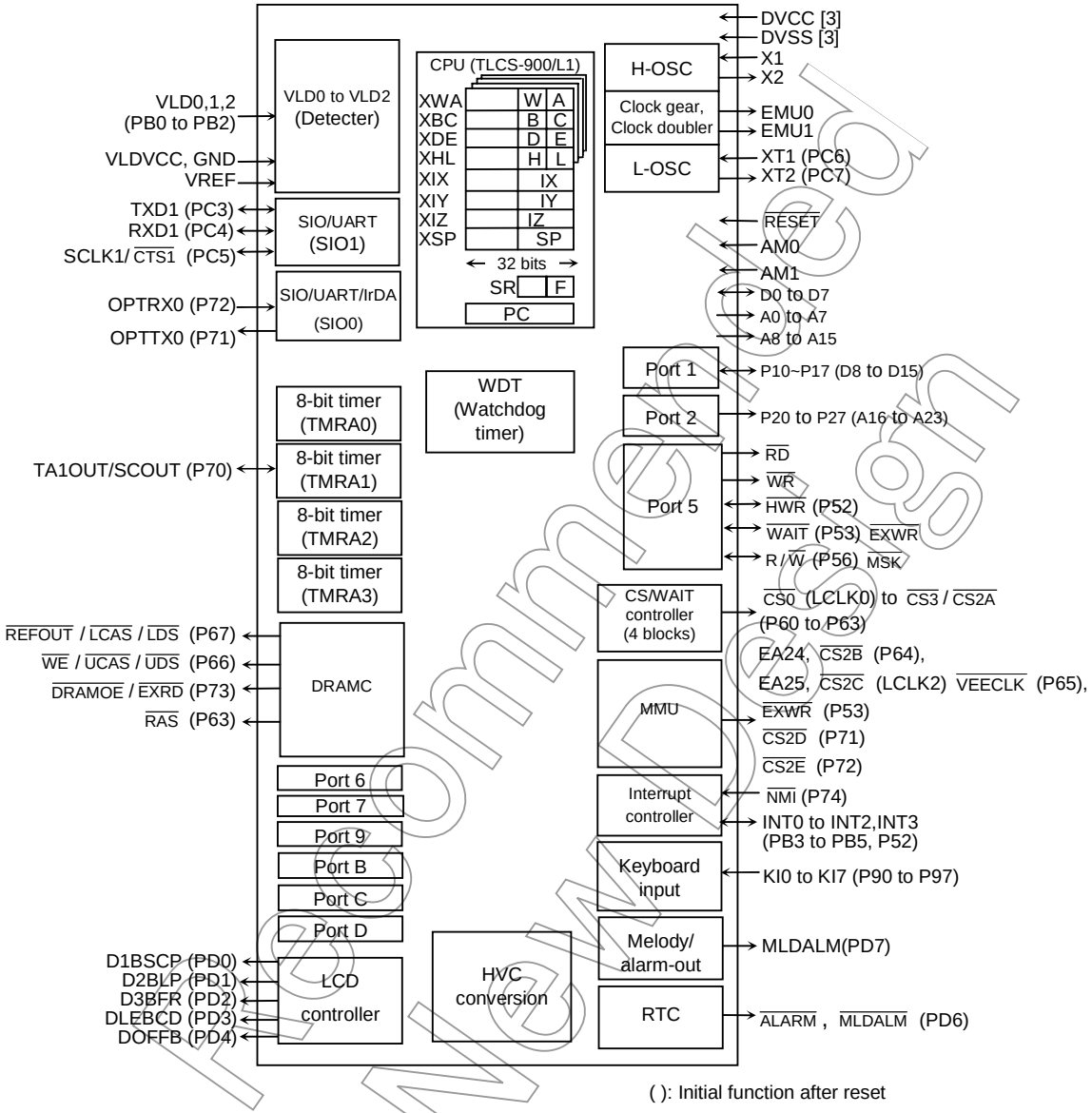


Figure 1.1 TMP91C016 Block Diagram

2. Pin Assignment and Pin Functions

The assignment of input/output pins for the TMP91C016, their names and functions are as follows:

2.1 Pin Assignment Diagram

Figure 2.1.1 shows the pin assignment of the TMP91C016FG.

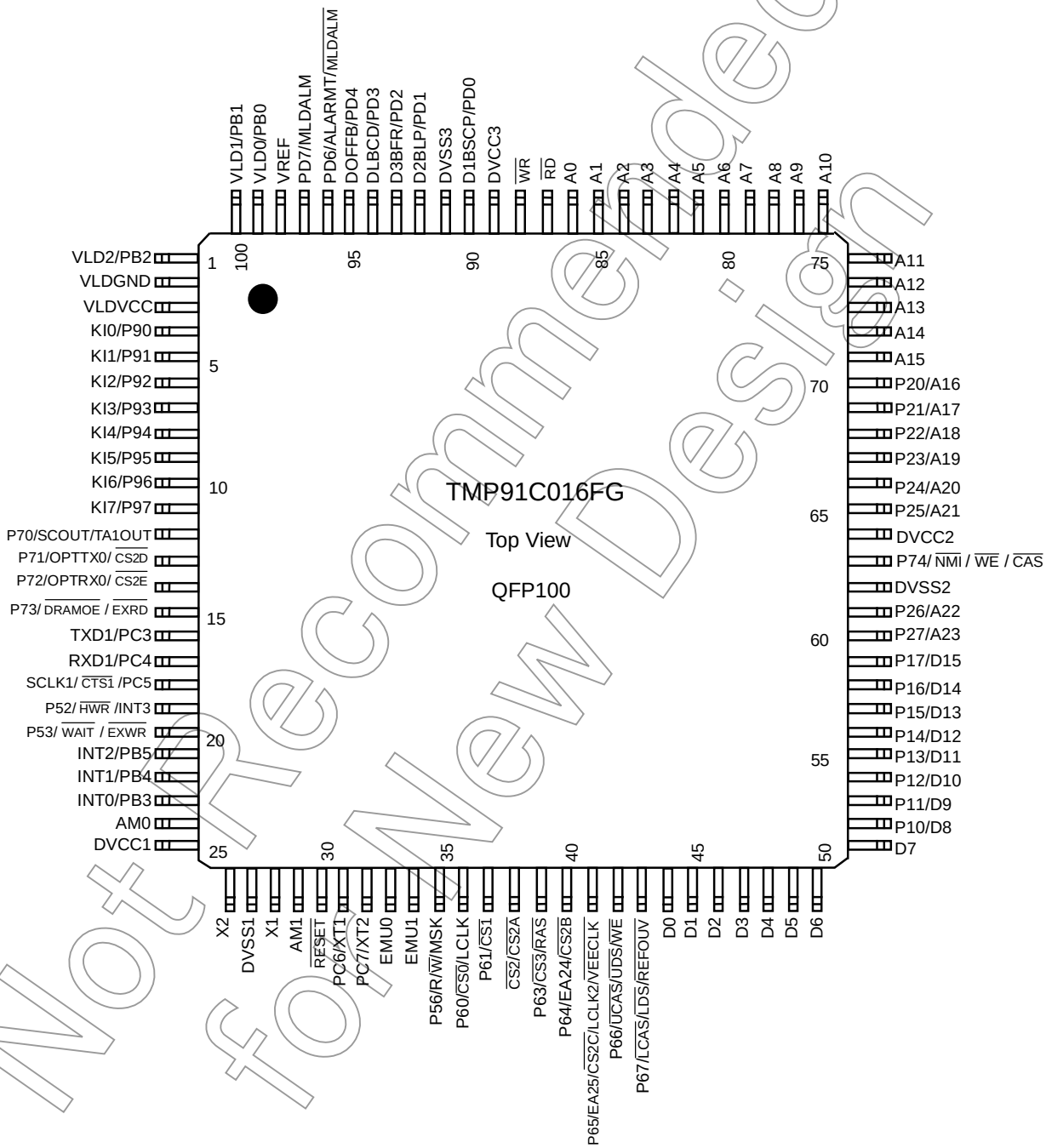


Figure 2.1.1 Pin Assignment Diagram (100-pin QFP)

2.1.1 Pad Layout

Table 2.1.1 PAD Layout

(Chip size 4.38 mm × 4.43 mm) Unit: μm

Pin No	Name	X Point	Y Point	Pin No	Name	X Point	Y Point	Pin No	Name	X Point	Y Point
1	PB2	−2057	1531	35	P56	−239	−2082	69	P21	2053	850
2	VLDGND	−2057	1417	36	P60	−125	−2082	70	P20	2053	964
3	VLDVCC	−2057	1303	37	P61	−11	−2082	71	A15	2053	1078
4	P90	−2057	990	38	P62	103	−2082	72	A14	2053	1192
5	P91	−2057	876	39	P63	217	−2082	73	A13	2053	1306
6	P92	−2057	762	40	P64	331	−2082	74	A12	2053	1420
7	P93	−2057	648	41	P65	479	−2082	75	A11	2053	1534
8	P94	−2057	534	42	P66	593	−2082	76	A10	1503	2082
9	P95	−2057	420	43	P67	707	−2082	77	A9	1389	2082
10	P96	−2057	306	44	D0	821	−2082	78	A8	1275	2082
11	P97	−2057	192	45	D1	935	−2082	79	A7	1160	2082
12	P70	−2057	55	46	D2	1049	−2082	80	A6	1046	2082
13	P71	−2057	−59	47	D3	1163	−2082	81	A5	932	2082
14	P72	−2057	−174	48	D4	1277	−2082	82	A4	818	2082
15	P73	−2057	−290	49	D5	1391	−2082	83	A3	704	2082
16	PC3	−2057	−404	50	D6	1505	−2082	84	A2	590	2082
17	PC4	−2057	−521	51	D7	2053	−1534	85	A1	476	2082
18	PC5	−2057	−638	52	P10	2053	−1420	86	A0	362	2082
19	P52	−2057	−755	53	P11	2053	−1306	87	$\overline{RD}$	248	2082
20	P53	−2057	−870	54	P12	2053	−1192	88	$\overline{WR}$	134	2082
21	PB5	−2057	−991	55	P13	2053	−1078	89	DVCC3	20	2082
22	PB4	−2057	−1105	56	P14	2053	−964	90	PD0	−180	2082
23	PB3	−2057	−1219	57	P15	2053	−850	91	DVSS3	−294	2082
24	AM0	−2057	−1333	58	P16	2053	−736	92	PD1	−408	2082
25	DVCC1	−2057	−1447	59	P17	2053	−606	93	PD2	−522	2082
26	X2	−1507	−2082	60	P27	2053	−450	94	PD3	−638	2082
27	DVSS1	−1342	−2082	61	P26	2053	−295	95	PD4	−752	2082
28	X1	−1176	−2082	62	DVSS2	2053	−140	96	PD6	−866	2082
29	AM1	−1060	−2082	63	P74	2053	17	97	PD7	−980	2082
30	$\overline{RESET}$	−946	−2082	64	DVCC2	2053	171	98	VREF	−1274	2082
31	PC6	−831	−2082	65	P25	2053	326	99	PB0	−1388	2082
32	PC7	−583	−2082	66	P24	2053	482	100	PB1	−1506	2082
33	EMU0	−467	−2082	67	P23	2053	622				
34	EMU1	−353	−2082	68	P22	2053	736				

2.2 Pin Names and Functions

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

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	8	I/O	Port 1: I/O port that allows I/O to be selected at the bit level (When used to the external 8-bit bus)
D8 to D15		I/O	Data (Upper): Bits 8 to15 of data bus
P20 to P27	8	Output	Port 2: Output port
A16 to A23		Output	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
$\overline{RD}$	1	Output	Read: Strobe signal for reading external memory. P5 <RDE>=0, output RD when reading internal area.
$\overline{WR}$	1	Output	Write: Strobe signal for writing data to pins D0 to D7
P52	1	I/O	Port 52: I/O port (with pull-up resistor)
$\overline{HWR}$		Output	High Write: Strobe signal for writing data to pins D8 to D15
$\overline{INT3}$		Input	Interrupt request pin 3: Interrupt request pin with programmable rising/falling edge
P53	1	I/O	Port 53: I/O port (with pull-up resistor)
$\overline{WAIT}$		Input	Wait: Pin used to request CPU bus wait ((1 + N) WAIT mode)
$\overline{EXWR}$		Output	Ex write: Strobe signal for writing data for RAM
P56	1	I/O	Port 56: I/O port (with pull-up resistor)
R / $\overline{W}$		Output	Read/write: 1 represents read or dummy cycle; 0 represents write cycle.
$\overline{MSK}$		Input	Request VEECLK clock for external LCD-driver.
P60	1	I/O	Port 60: I/O port (with pull-up resistor)
$\overline{CS0}$		Output	Chip select 0: Outputs 0 when address is within specified address area.
LCLK0		Output	Lcd CLK: Command controll C/S for S/R type lcdd.
P61	1	I/O	Port 61: I/O port (with pull-up resistor)
$\overline{CS1}$		Output	Chip select 1: Outputs 0 when address is within specified address area
CS2	1	Output	Chip select 2: Outputs 0 when address is within specified address area
$\overline{CS2A}$		Output	Expand chip select: 2A: Outputs 0 when address is within specified address area
P63	1	I/O	Port 63: I/O port (with pull-up resistor)
CS3		Output	Chip select 3: Outputs 0 when address is within specified address area
$\overline{RAS}$		Output	Row address strobe: RAS strobe row address area for DRAM
P64	1	I/O	Port 64: I/O port (with pull-up resistor)
EA24		Output	Chip select 24: Outputs 0 when address is within specified address area
$\overline{CS2B}$		Output	Expand chip select 2B: Outputs 0 when address is within specified address area
P65	1	I/O	Port 65: I/O port (with pull-up resistor)
EA25		Output	Chip select 25: Outputs 0 when address is within specified address area
$\overline{CS2C}$		Output	Expand chip select 2C: Outputs 0 when address is within specified address area
LCLK2		Output	Lcd CLK: Command controll C/S for S/R type lcdd.
$\overline{VEECLK}$		Output	Pomp-up CLK for external LCD driver
P66	1	I/O	Port 66: I/O port (with pull-up resistor)
$\overline{UCAS}$		Output	Upper column address strobe: Upper CAS strobe for 2CAS type DRAM.
$\overline{UDS}$		Output	Upper data enable strobe
$\overline{WE}$		Output	Write strobe for DRAM (only 8-bit access)
P67	1	I/O	Port 67: I/O port (with pull-up resistor)
$\overline{LCAS}$		Output	Lower column address strobe: Upper CAS strobe for 2CAS type DRAM.
$\overline{LDS}$		Output	Lower data enable strobe
$\overline{REFOUT}$		Output	Refresh cycle state singanl for DRAM (only 8-bit access)

Pin Name	Number of Pins	I/O	Functions
P70 SCOUT TA1OUT	1	I/O Output Output	Port 70: I/O port (with pull-up resistor) System clock output: Selectable f_{FPH} or fs 8-bit timer output: Timer 0 or timer 1 out
P71 OPTTX0 $\overline{CS2D}$	1	I/O Output Output	Port 71: I/O port (with pull-up resistor) SIO0 trance port Expond chip select 2D: Outputs 0 when address is within specified address area
P72 OPTRX0 $\overline{CS2E}$	1	I/O Input Output	Port 72: I/O port (Shummit input, with pull-up/pull-down resistor) SIO0 receive port Expond chip select 2E: Outputs 0 when address is within specified address area
P73 $\overline{DRAMOE}$ $\overline{EXRD}$	1	I/O Output Output	Port 73: I/O port (with pull-up resistor) DRAMOE: Strobe signal for reading external DRAM External read: Strobe signal for reading external memory
P74 $\overline{NMI}$ $\overline{WE}$ $\overline{CAS}$	1	I/O Input Output Output	Port 74: I/O port (with pull-up resistor) Non-maskable interrupt request pin: Interrupt request pin with programmable falling edge level or with both edge levels programmable Strobe signal for writing data for DRAM (only 2CAS) Coulmn address strobe: Outputs 0 when address is within specified DRAM column address area (only 8 bits access)
P90 to P97 KI0 to KI7	8	Input Input	Port: 90 to 97 port: Pin used to input ports Key input 0 to 7: Pin used of key on wake-up 0 to 7 (Schmitt input, with pull-up resistor)
PB0 VLD0	1	I/O Input	Port B0: I/O port (with pull-up resistor) Voltage level detector 0: For main battery, Interrupt request with edge, too
PB1 VLD1	1	I/O Input	Port B1: I/O port (with pull-up resistor) Voltage level detector 1: For back up battery, Interrupt request with edge, too
PB2 VLD2	1	I/O Input	Port B2: I/O port (with pull-up resistor) Voltage level detector 2: For micon battery, Interrupt request with edge, too
PB3 INT0	1	I/O Input	Port B3: I/O port (Schmitt input, with pull-up resistor) Interrupt request pin 0: Interrupt request pin with programmable level/rising/falling edge
PB4 to PB5 INT1 to INT2	2	I/O Input	Port B4 to B5: I/O port (Schmitt input, with pull-down resistor) Interrupt request pin 1 to 2: Interrupt request pin with programmable rising/falling edge
PC3 TXD1	1	I/O Output	Port C3: I/O port (with pull-up resistor) Serial 1 send data: Open-drain output pin by programmable
PC4 RXD1	1	I/O Input	Port C4: I/O port (Schmitt input, with pull-up/pull-down resistor) Serial 1 recive data
PC5 SCLK1 $\overline{CTS1}$	1	I/O I/O Input	Port C5: I/O port (Schmitt input, with pull-up/pull-down resistor) Serial clock I/O 1 Clear to send
PC6 XT1	1	I/O Input	Port C6: I/O port (Open-drain output) Low-frequency oscillator connection pins
PC7 XT2	1	I/O Output	Port C7: I/O port (Open-drain output) Low-frequency oscillator connection pins

Pin Name	Number of Pins	I/O	Functions
PD0 D1BSCP	1	I/O Output	Port D0: I/O port (with pull-up resistor) LCD driver output pin
PD1 D2BLP	1	I/O Output	Port D1: I/O port (with pull-up resistor) LCD driver output pin
PD2 D3BFR	1	I/O Output	Port D2: I/O port (with pull-up resistor) LCD driver output pin
PD3 DLEBCD	1	I/O Output	Port D3: I/O port (with pull-up resistor) LCD driver output pin
PD4 DOFFB	1	I/O Output	Port D4: I/Ot port (with pull-up resistor) LCD driver output pin
PD6 $\overline{\text{ALARM}}$ $\overline{\text{MLDALM}}$	1	I/O Output Output	Port D6: I/O port (with pull-up resistor) RTC alarm output pin Logical invert for Melody/alarm output pin
PD7 MLDALM	1	I/O Output	Port D7: I/O port (with pull-up resistor) Melody/alarm output pin
AM0 to AM1	2	Input	Operate 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
$\overline{\text{RESET}}$	1	Input	Reset: Initializes TMP91C016. (with pull-up resistor)
VREF	1	Input	Power supply pin for Low-frequency oscillator, RTC and VLD.
VLDVCC	1		For VLD power supply pin
VLDVSS	1		For VLD: GND pins (0 V) (All pins should be connected with GND (0 V).)
X1/X2	2		High-frequency oscillator connection pins
DVCC	3		Power supply pins (All Vcc pins should be connected with the power Supply pin).
DVSS	3		GND pins (0 V) (All pins should be connected with GND (0V).)

3. Operation

This following describes block by block the functions and operation of the TMP91C016.

Notes and restrictions for eatch book are outlined in 6. “Points of Note and Restrictions” at the end of this manual.

3.1 CPU

The TMP91C016 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 TMP91C016; these functions are not covered in the TLCS-900/L1 CPU section.

3.1.1 Reset

When resetting the TMP91C016 microcontroller, ensure that the power supply voltage is within the operating voltage range, and that the internal high-frequency oscillator has stabilized. Then set the $\overline{\text{RESET}}$ input to low level at least for 10 system clocks (12 μs at 27 MHz).

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

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

When 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<0:7> ← Value at FFFF00H address

  PC<15:8> ← Value at FFFF01H address

  PC<23:16> ← Value at FFFF02H address
- Sets the stack pointer (XSP) to 100H.
- Sets bits <IFF2:0> of the status register (SR) to 111 (Sets the interrupt level mask register to level 7).
- Sets the <MAX> bit of the status register (SR) to 1 (Max mode).
  (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.

Note: 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 TMP91C016.

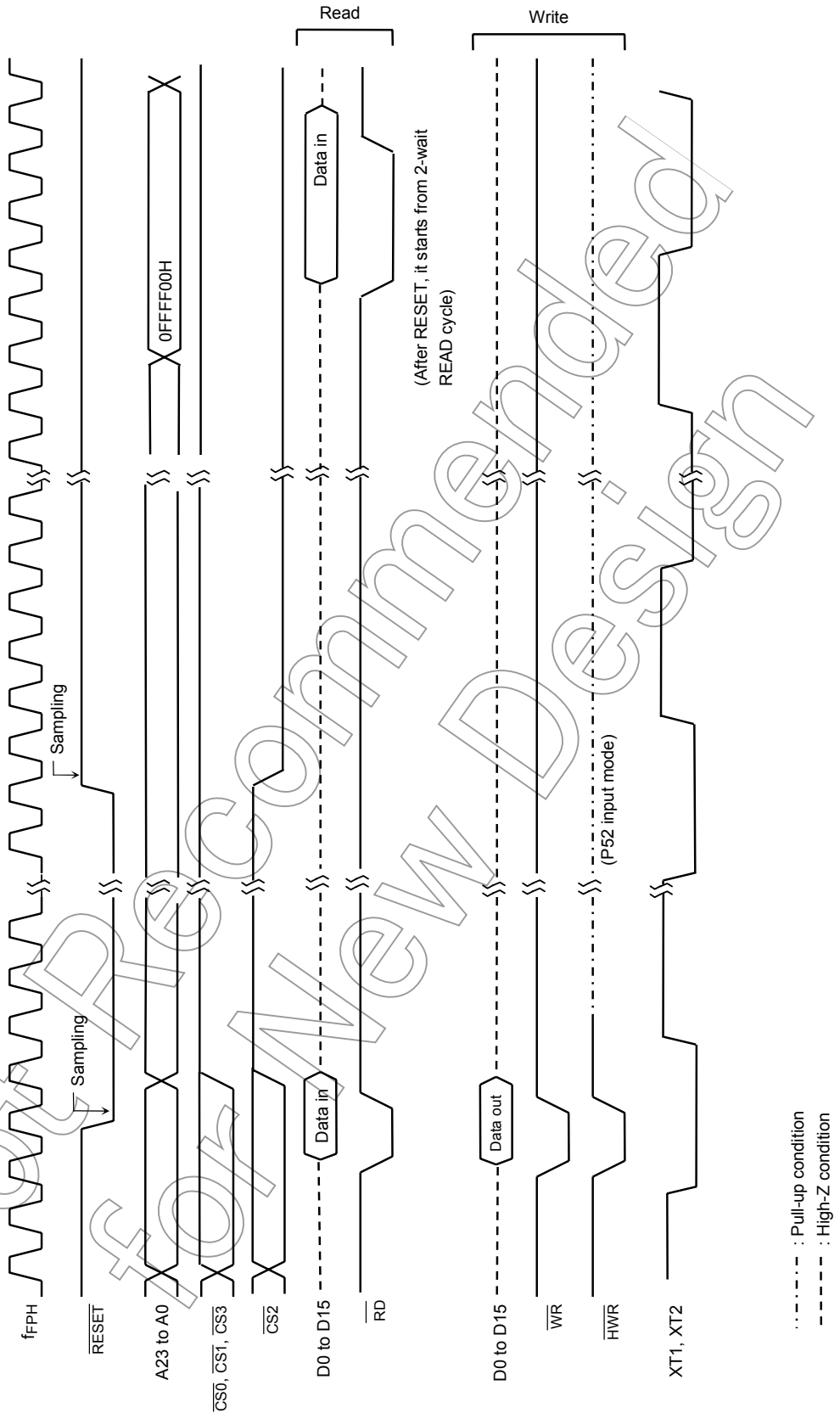


Figure 3.1.1 TMP91C016 Reset Timing Chart

3.2 Memory Map

Figure 3.2.1 is a memory map of the TMP91C016.

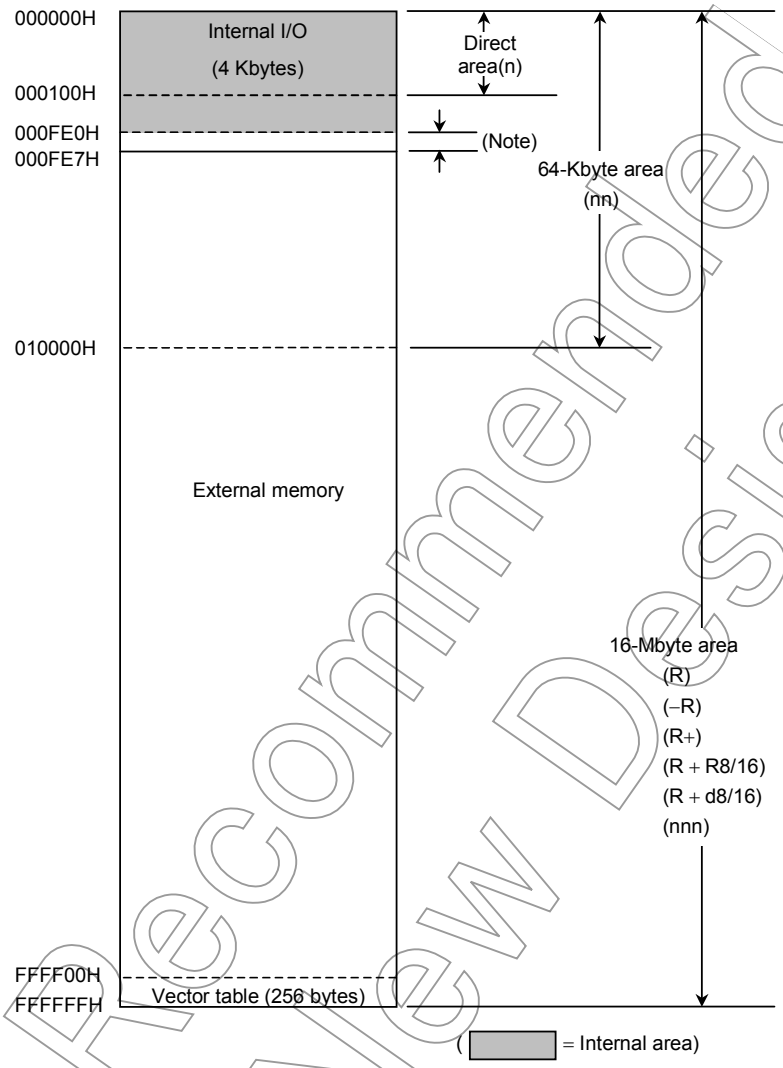


Figure 3.2.1 Memory Map

Note: Address 000FE0H to 00FE7H are assigned for the external memory area of built-in RAM type LCD driver.

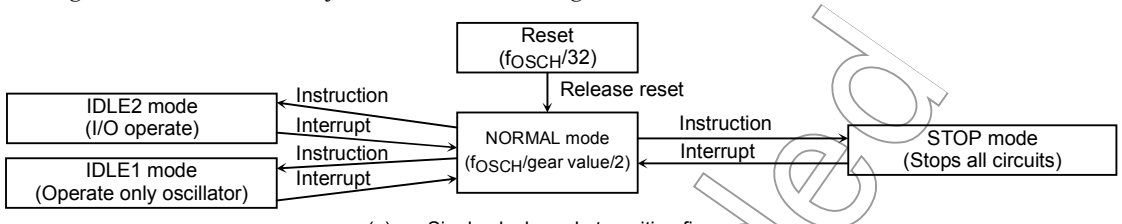
3.3 Triple Clock Function and Standby Function

TMP91C016 contains (1) clock gear, (2) clock doubler (DFM), (3) standby controller and (4) noise-reduction circuit. It is used for low-power, low-noise systems.

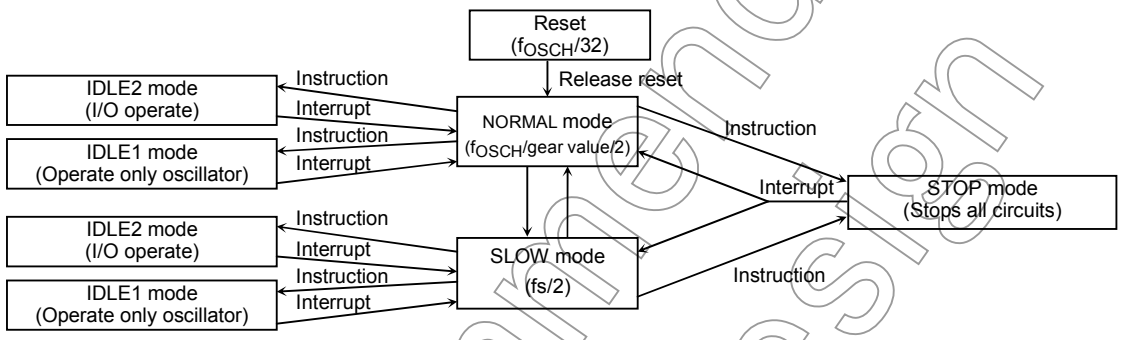
This chapter is organized as follows:

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).

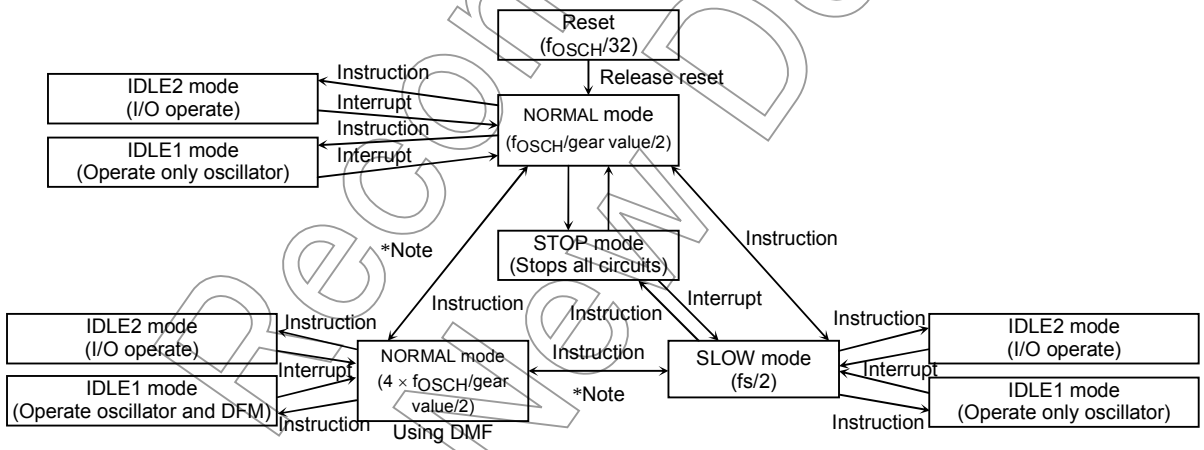
Figure 3.3.1 shows the system clock block diagrams.



(a) Single clock mode transition figure



(b) Dual clock mode transition figure



(c) Triple clock mode trasition figure

Note 1: It's prohiibited 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> register)

Note 2: If you shift from NORMAL mode with use of DFM to NORMAL mode, the above two instructions should be separated into two procedures as below. Change CPU clock → Stop DFM circuit.

Note 3: 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 regret to as one state.

3.3.1 Block Diagram of System Clock

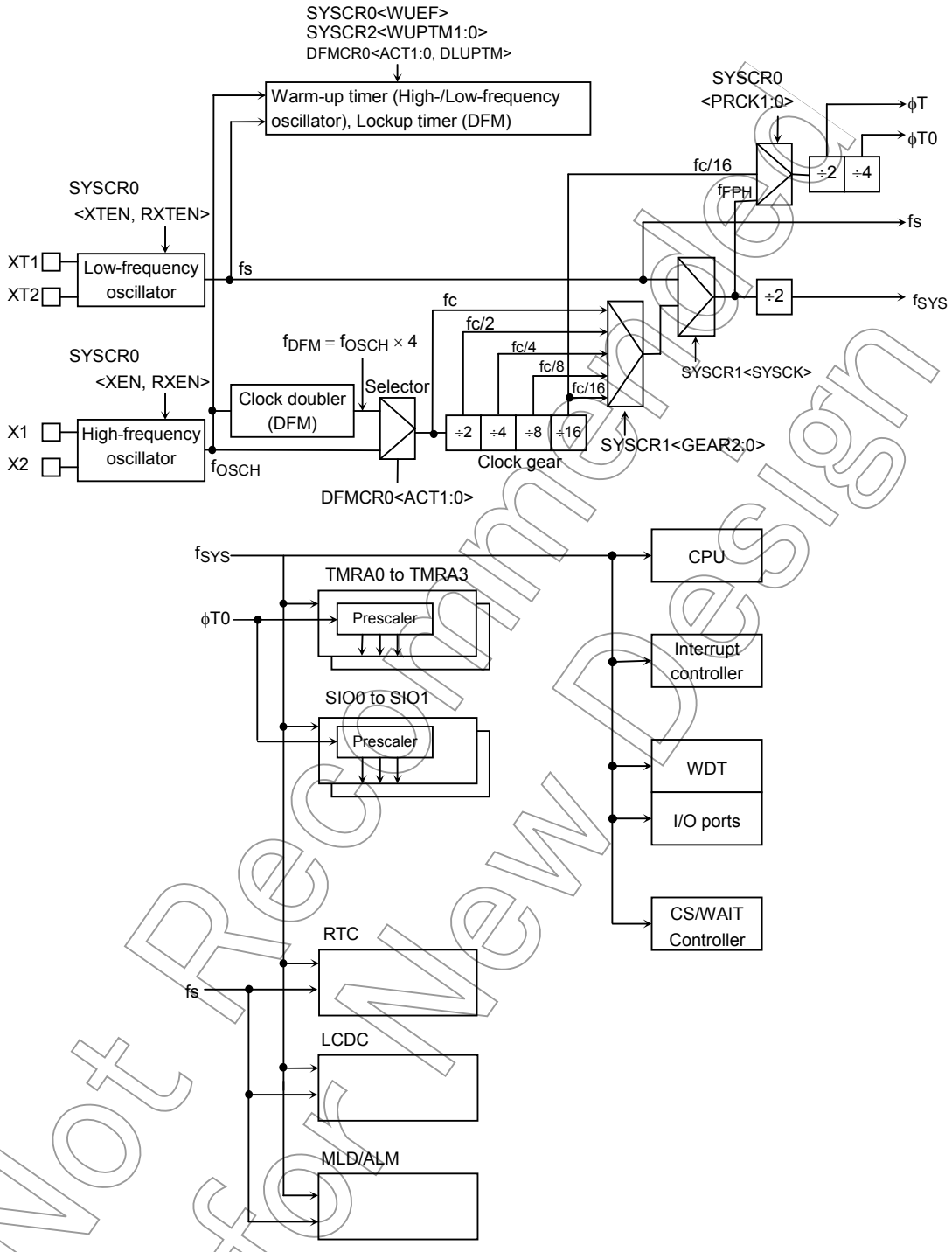


Figure 3.3.2 Block Diagram of System Clock

3.3.2 SFRs

		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	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} 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_{FPH}	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
VLDCTL (0449H)	Bit symbol					XT1VSEL	VLD2VSE	VLD1VSE	VLD0VSE
	Read/Write					W	R/W	R/W	R/W
	After reset					0	0	0	0
	Function					0: Vcc operation 1: Vref operation	0: VLD don't use 1: VLD use	0: VLD don't use 1: VLD use	0: VLD don't use 1: VLD use

Note1: SYSCR1<bit7:4>,SYSCR2<bit7> are read as undefined value.

Note2: By reset, low-frequency oscillator become to enable condition.

Figure 3.3.3 SFR for System Clock

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}		Lockup status flag	Lockup time				
			00 STOP STOP f_{OSCH}		0: End	0: $2^{12}/f_{OSCH}$				
			01 RUN RUN f_{OSCH}		1: Not end	1: $2^{10}/f_{OSCH}$				
			10 RUN STOP f_{DFM}							
			11 RUN STOP f_{OSCH}							
DFMCR1	DFM Control Register 1	E9H	D7	D6	D5	D4	D3	D2	D1	D0
			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 to 6.75 MHz (at 2.7 V to 3.6 V): Write 0BH Input frequency 2 to 2.5 MHz (at 2.0 ± 10%): Write 1BH							

Figure 3.3.4 SFR for DFM

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 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	–	–	–	–	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	Always Write "0"	Always Write "1"	Always Write "0"	Always Write "0"	1: External clock	fc oscillator drivability 1: Normal 0: Weak	fs oscillator drivability 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 When reading 0: Not written 1: Written When writing 0: Clear flag	CS2B-2G write operation flag	CS2A write operation flag
EMCCR4 (00E7H)	Bit symbol							TA3MLDE	TA3LCDE
	Read/Write							R/W	R/W
	After reset							0	0
	Function							MLD CLK: 0: 32 kHz 1: TA3	LCD CLK: 0: 32 kHz 1: TA3

Note1: In case restarting the oscillator in the stop oscillation state (e.g. Restart the oscillator in STOP mode), set EMCCR0<DRVOSCH>, <DRVOSCL>="1".

Note2: When VCC=2V±10%, set EMCCR0<DRVOSCH> to "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<GEAR0:2> 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 <GEAR0:2> = 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 0.84 MHz when the 27-MHz oscillator is connected to the X1 and X2 pins. And TMP91C016 has another power terminal: VREF except DVCC, this VREF power terminal supply to low-frequency oscillator operation and reference voltage for VLD operation. That can controll low-frequency oscillator's power DVCC or VREF by VLDCTL<XTVSEL>.

(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<WUPTM0:1>.

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 Warm-up Times

Warm-up Time SYSCR2 <WUPTM1:0>	Change to NORMAL Mode	Change to SLOW Mode
01 (2^8/frequency)	9.0 μs	7.8 ms
10 (2^{14}/frequency)	0.607 ms	500 ms
11 (2^{16}/frequency)	2.427 ms	2000 ms

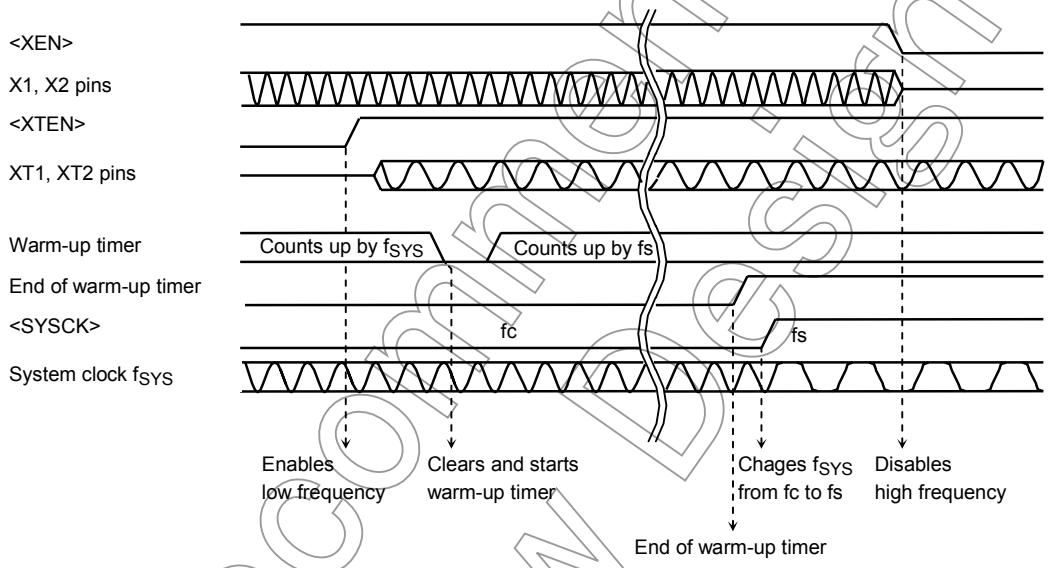
at f_{OSCH} = 27 MHz, fs = 32.768 kHz

Example 1: Setting the clock

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

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

X: Don't care, –: No change

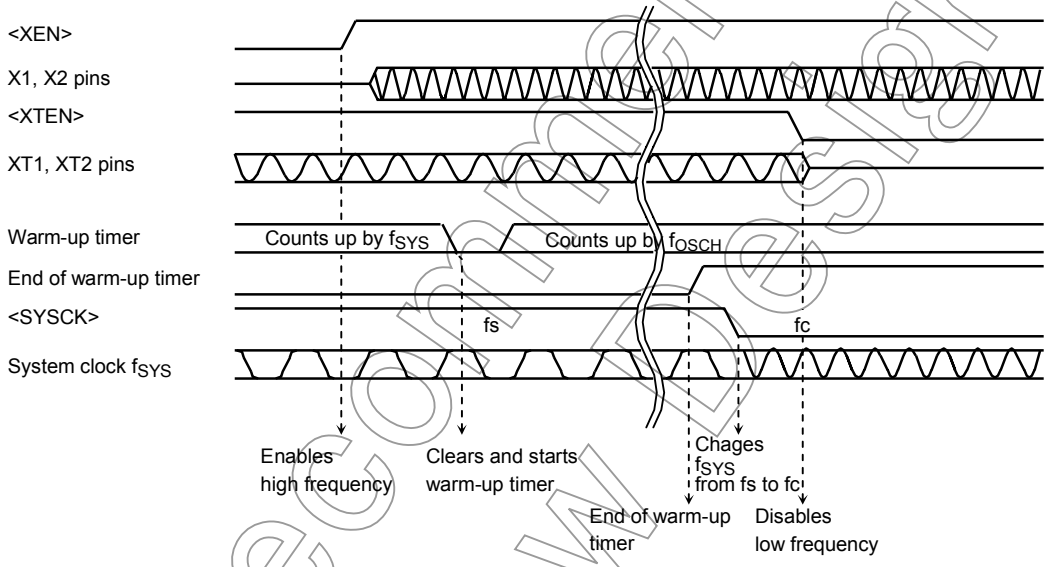


Example 2: Setting the clock

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)            ;  } Detects stopping of warm-up timer.
         JR    NZ, WUP                ;  }
         RES   3, (SYSCR1)            ;  Changes f_SYS from fs to fc.
         RES   6, (SYSCR0)            ;  Disables low-frequency oscillation.
```

X: Don't care, –: No change



(2) Clock gear controller

When the high-frequency clock fc is selected by setting SYSCR1<SYSCK> = 0, f_{FPH} is set according to the contents of the clock gear select register SYSCR1<GEAR0:2> 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 warm-up time until changing after writing the register value.

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

Example:

```
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_{FPH} or fs) from P70 (TA1OUT, SCOUT).

P70 pin function is set to SCOUT output by the following bit setting.

: P7CR<P70F> = 1, P7FC<P70F> = 0, P7FC2<P70F2> = 1

Output clock select

: Refer to SYSCR2<SCOSEL> bit setting

Table 3.3.2 SCOUT Output Condition

HALT Mode / SCOUT Select	NORMAL Mode SLOW Mode	HALT Mode		
		IDLE2 Mode	IDEL1 Mode	STOP Mode
<SCOSEL> = 0	fs clock out			
<SCOSEL> = 1	f_{FPH} clock out		0 or 1 fix out	

3.3.4 Prescaler Clock Controller

For the internal I/O (TMRA01 to TMRA23, SIO0 to SIO1, SBI) there is a prescaler which can divide the clock.

The φT clock input to the prescaler is either the clock f_{FPH} divided by 2 or the clock fc/16 divided by 2. The setting of the SYSCR0<PRCK0:1> register determines which clock signal is input.

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<PRCK0:1> register determines which clock signal is input.

3.3.5 Clock Doubler (DFM)

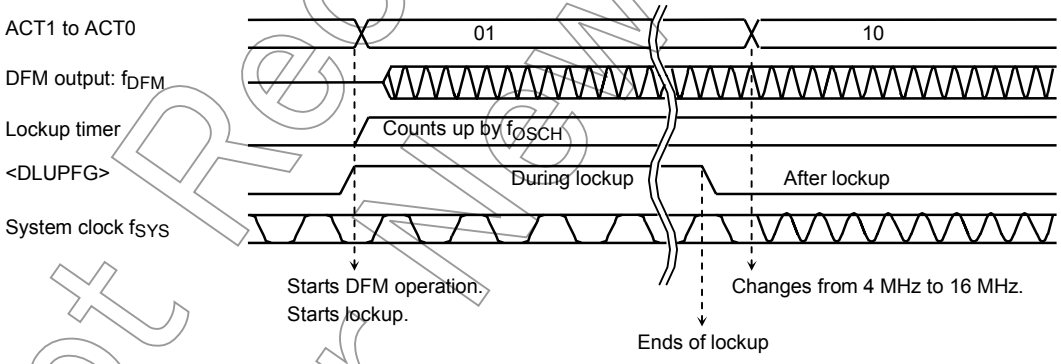
DFM outputs the f_{DFM} clock signal, which is four times as fast as f_{OSCH}. It can use the low-frequency oscillator, even though the internal clock is high-frequency.

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 lockup time to 2^12/4 MHz
                                         Enables DFM operation and starts lockup.
LUP:      BIT   5, (DFMCR0)           ;
          JR    NZ, LUP               ;  Detects end of lockup
          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 to 6.75 MHz (Vcc = 2.7 to 3.6 V): Write 0BH to DFMCR1

f_{OSCH} = 2 to 2.5 MHz (Vcc = 2.0 V ± 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 fDFM → 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 setting are below.

(1) Start-up/change control

(OK) Low-frequency oscillator operation mode (fs) (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/warm-up start
WUP:  BIT   2, (SYSCR0)           ; } Check for the flag of warm-up end
      JR    NZ, WUP               ; }
      LD    (SYSCR1),   ----0---B ; Change the system clock fs to fOSCH
      LD    (DFMCR0),   01-0----B ; DFM start-up/lockup start
LUP:  BIT   5, (DFMCR0)           ; } Check for the flag of lock up end
      JR    NZ, LUP               ; }
      LD    (DFMCR0),   10-0----B ; Change the system clock fOSCH to fDFM
```

(OK) Low-frequency oscillator operation mode (fs) (High-frequency oscillator Operator) → 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/lockup start
LUP:  BIT   5, (DFMCR0)           ; } Check for the flag of lockup end
      JR    NZ, LUP               ; }
      LD    (DFMCR0),   10-0----B ; Change the system clock fOSCH to fDFM
```

(OK) 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/warm-up start
WUP:  BIT   2, (SYSCR0)           ; } Check for the flag of warm-up end
      JR    NZ, WUP               ; }
      LD    (DFMCR0),   01-0----B ; DFM start-up/lockup start
LUP:  BIT   5, (DFMCR0)           ; } Check for the flag of lockup end
      JR    NZ, LUP               ; }
      LD    (DFMCR0),   10-0----B ; Change the system 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 (fs) → 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      (SYSCR 0),  0-------B ; High-frequency oscillator stop
```

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

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

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

```
LD      (SYSCR2),   ----01--B ; Set 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
```

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

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

3.3.6 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

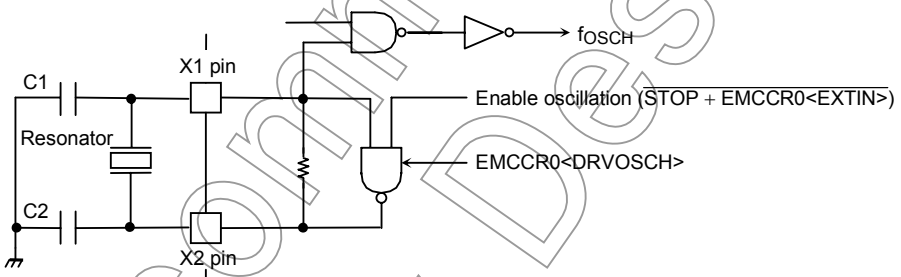
The above functions are performed by making the appropriate settings in the EMCCR0 to EMCCR3 registers.

(1) Reduced drivability for high-frequency oscillator

(Purpose)

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

(Block diagram)



(Setting method)

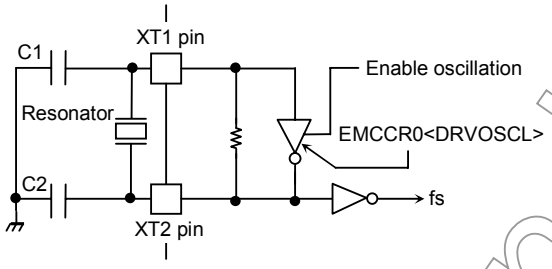
The drivability of the oscillator is reduced by writing 0 to EMCCR0<DRVOSCH> register. By reset, <DRVOSCH> is initialized to 1 and the oscillator starts oscillation by normal-drivability when the power-supply is on. When VCC=2V±10%, don't set EMCCR0<DRVOSCH> to "0".

(2) Reduced drivability for low-frequency oscillator

(Purpose)

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

(Block diagram)



(Setting method)

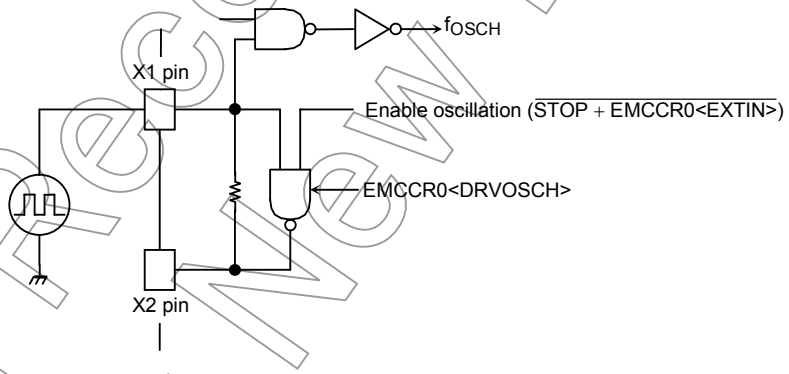
The drivability of the oscillator is reduced by writing 0 to the EMCCR0<DRVOSCL> register. By reset, <DRVOSCL> is initialized to 1.

(3) Single drive for high-frequency oscillator

(Purpose)

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

(Block diagram)



(Setting method)

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

By reset, <EXTIN> is initialized to 0.

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

(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
5. PORT
   P2FC, P5CR, P5FC, P5FC2, P6CR, P6FC, P6FC2
   P7CR, P7FC, P7FC2, PDCR, PDFC
6. DRAMC
   DREFCR, DMEMCR

(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 to 7FFFFFH
2. Data ROM: Address 800000H to BFFFFFH
3. Program ROM: Address C00000H to 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.

(6) <EMCCR4> register explanation

It is assigned <TA3LCDE> at bit0 and <TA3MLDE> at bit1, of EMCCR4 register (00E7hex). These bits are used when you want to operate LCDD and MELODY circuit without low-frequency clock (XTIN, XTOUT). After reset these two bits set to 0 and low clock is supplied each LCDD and MELODY circuit. If you write these bits to 1, TA3 (Generate by timer 3) is supplied each LCDD and MELODY circuit. In this case, you should set 32 kHz timer 3 frequency. For detail, look AC specification characteristics.

3.3.7 Standby Controller

(1) HALT modes

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

The subsequent actions performed in each mode are as follows:

a. IDLE2: Only the CPU HALTs.
The internal I/O is available to select operation during IDLE2 mode. By setting the following register.
Table 3.3.3 shows the registers of setting operation during IDLE2 mode.

Table 3.3.3 SFR Seting Operation During IDLE2 Mode

Internal I/O	SFR
TMRA01	TA01RUN<I2TA01>
TMRA23	TA23RUN<I2TA23>
SIO0	SC0MOD1<I2S0>
SIO1	SC1MOD1<I2S1>
WDT	WDMOD<I2WDT>

b. IDLE1: Only the oscillator and the RTC (Real time clock) and MLD continue to operate.

c. STOP: All internal circuits stop operating.

The operation of each of the different HALT modes is described in Table 3.3.4.

Table 3.3.4 I/O Operation During HALT Modes

<table>
<tr><td colspan="2">HALT Mode</td><td>IDLE2</td><td>IDLE1</td><td>STOP</td></tr>
<tr><td colspan="2">SYSCR2<HALTM1:0></td><td>11</td><td>10</td><td>01</td></tr>
<tr><td rowspan="8">Block</td><td>CPU</td><td colspan="3">Stop</td></tr>
<tr><td>I/O ports</td><td colspan="3">Keep the state when the HALT instruction is executed. See Table 3.3.7, Table 3.3.8 are</td></tr>
<tr><td>TMRA</td><td rowspan="5">Available to select operation block</td><td></td><td rowspan="5"></td></tr>
<tr><td>RTC, MLD</td><td>Operational available</td></tr>
<tr><td>SIO</td><td>Stop</td></tr>
<tr><td>DRAMC</td><td>Note: Operational available</td></tr>
<tr><td>WDT</td><td rowspan="2"></td></tr>
<tr><td>LCDC, Interrupt controller</td><td>Operate</td></tr>
</table>

Note: It is only self refresh mode of DRAM. It can't move normal operation and interval refresh mode of DRAM.

(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 3.3.5.

- 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 to INT3, INTRTC, INTALM0 to INTALM4, INTKEY 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 ($\overline{NMI}$, INT0 to INT3, INTKEY, INTRTC, INTALM0 to INTALM4, INTVLD0 to INTVLD2) which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 5 clocks of f_{FPH}) with IDLE1 or STOP mode (IDLE2 is not applicable to this case). (In this case, an interrupt request is kept on hold internally.)
If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficulty. The priority of this interrupt is compared with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.

- Releasing by resetting

Releasing all halt status is executed by resetting.

When the STOP mode is released by reset, it is necessry enough resetting time (See Table 3.3.6) to set the operation of the oscillator to be stable.

Table 3.3.5 Source of Halt State Clearance and Halt Clearance Operation

Status of Received Interrupt			Interrupt Enabled (Interrupt level) ≥ (Interrupt mask)			Interrupt Disabled (Interrupt level) < (Interrupt mask)		
HALT mode			IDLE2	IDLE1	STOP	IDLE2	IDLE1	STOP
Source of halt state clearance	Interrupt	NMI	♦	♦	♦*1	–	–	–
		INTWD	♦	×	×	–	–	–
		INT0 to INT3 (Note 1)	♦	♦	♦*1	○	○	○*1
		INTALM0 to INTALM4	♦	♦	×	○	○	×
		INTTA0 to INTTA3	♦	×	×	×	×	×
		INTRX0 to INTRX1, TX0 to TX1	♦	×	×	×	×	×
		INTKEY	♦	♦	♦*1	○	○	○*1
		INTRTC	♦	♦	×	○	○	×
		INTLCD	♦	×	×	×	×	×
		INTVLD0 to INTVLD2*2	♦	♦	♦*1	–	–	–
		RESET	Reset initializes the LSI					

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

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

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

–: The priority level (Interrupt request level) of non-maskable interrupts is fixed to 7, the highest priority level. There is not this combination type.

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

*2: INTVLD0 to INTVLD2 are NMI (Non maskable interrupt) class in point of view from interrupt circuit, but these signals are actually maskable signals. If you want to mask these signals, you can controll by VLD circuit.

Note: 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
8200H   LD      (PBFC), 08H     ; Sets PB3 to INT0
8203H   LD      (IIMC), 00H     ; Selects INT0 interrupt rising edge.
8206H   LD      (INTE0), 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

a. IDLE2 mode

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

Figure 3.3.6 illustrates an example of the timing for clearance of the IDLE2 mode halt state by an interrupt.

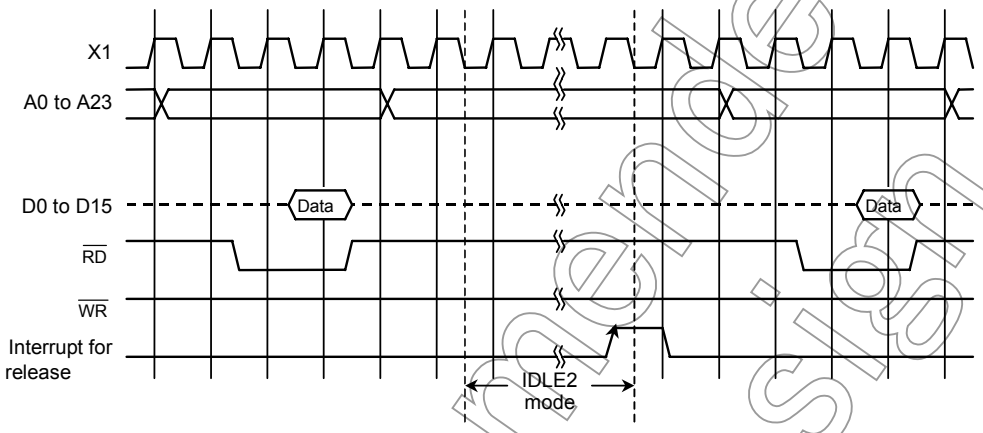


Figure 3.3.6 Timing Chart for IDLE2 Mode Halt State Cleared by Interrupt

b. 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 to DRVE>. Table 3.3.7, Table 3.3.8 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 (e.g., 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.

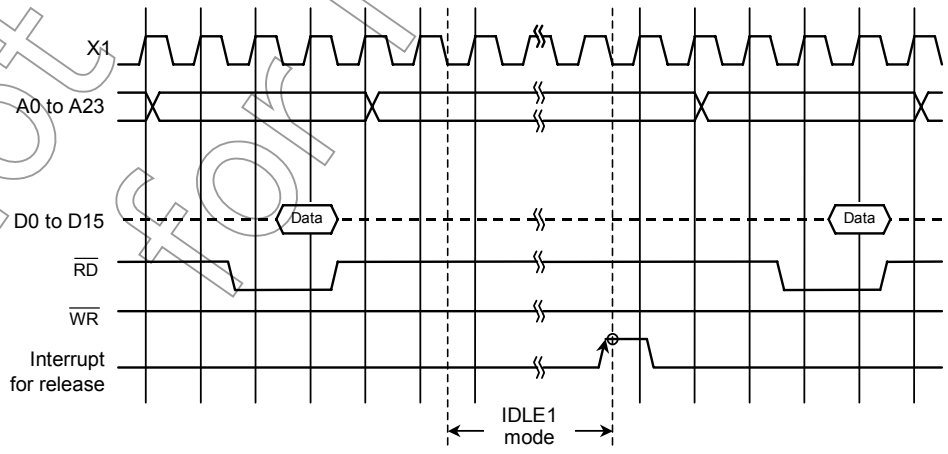


Figure 3.3.7 Timing Chart for IDLE1 Mode Halt State Cleared by Interrupt

c. STOP mode

When STOP mode is selected, all internal circuits stop, including the internal oscillator pin status in STOP mode depends on the settings in the SYSCR2<DRVE> register. Table 3.3.7, Table 3.3.8 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 3.3.6.

Figure 3.3.8 illustrates the timing for clearance of the STOP mode halt state by an interrupt.

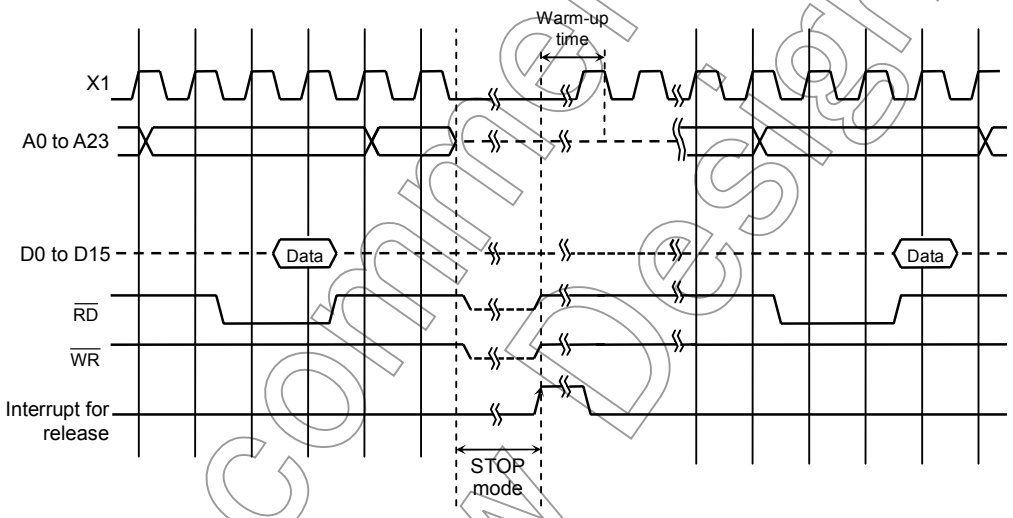


Figure 3.3.8 Timing Chart for STOP Mode Halt State Cleared by Interrupt

Table 3.3.6 Sample Warm-up Times after Clearance of STOP Mode

at f_{OSCH} = 27 MHz, fs = 32.768 kHz

SYSCR0 <RSYSCK>	SYSCR2<WUPTM1:0>		
	01 (2^8)	10 (2^{14})	11 (2^{16})
0 (fc)	9.0 μs	0.607 ms	2.427 ms
1 (fs)	7.8 ms	500 ms	2000 ms

Example:

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

```
Address
         SYSCR0   EQU   00E0H
         SYSCR1   EQU   00E1H
         SYSCR2   EQU   00E2H
8FFDH            LD    (SYSCR1), 08H          ; fSYS = fs/2
9000H            LD    (SYSCR2), X –1001X1B   ; Sets warm-up time to 2^14/fOSCH
9002H            LD    (SYSCR0), 011000 – –B  ; Operates high frequency after released.
                                                –: No change
9005H            HALT
         NMI
                                                Clears and starts hit
                                                warm-up timer
                                                (High frequency)
                                                End
                                                NMI interrupt routine
9006H            LD    XX, XX                                    RETI
```

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

Table 3.3.7 Input Buffer State Table

Port Name	Input Function Name		Input Buffer State								
			During Reset	When the CPU is operating		In HALT mode (IDLE2)		In HALT mode(IDLE1/STOP)			
								Condition A (Note)		Condition B (Note)	
				When Used as function Pin	When Used as Input Port	When Used as function Pin	When Used as Input Port	When Used as function Pin	When Used as Input Port	When Used as function Pin	When Used as Input Port
D0-7	–		OFF	ON upon external read	–	OFF	–	OFF	–	OFF	–
P10-17	D8-15				ON		OFF		OFF		OFF
P52(*1)	INT3			ON		ON		ON		ON	
P53(*1)	$\overline{WAIT}$		ON				ON	OFF			ON
P56(*1)	$\overline{MSK}$								OFF		
P60-67(*1)	–		OFF	–		–	OFF	–		–	OFF
P70-71(*1)	–										
P72(*1)	OPTRX0		ON	ON		ON	ON	OFF		ON	ON
P73(*1)	–		OFF	–		–	OFF	–		–	OFF
P74(*1)	$\overline{NMI}$			ON		ON		ON		ON	
P90-97(*1)	KI0-7		ON				ON		ON		ON
PB0-B2 (*1)(*2)	–		OFF	–		–	OFF	–		–	OFF
PB3-B5(*1)	INT0-2			ON		ON		ON		ON	
PC3(*1)	–			–		–		–		–	
PC4(*1)	RXD1		ON	ON		ON	ON	OFF		ON	ON
PC5(*1)	SCLK1 $\overline{CTS1}$										
PC6	XT1	For oscillator			OFF		OFF			OFF	OFF
		For port		OFF		OFF					ON
PC7	–		OFF	–	ON	–	ON	–	OFF	–	
PD0-D4, PD6-D7(*1)	–			–		–	OFF	–		–	OFF
$\overline{MSK}$	–		ON	ON	–	ON	–	ON	–	ON	–
AM0,AM1	–				–		–		–		–
X1	–		ON	ON	–	ON	–	IDLE1 : ON , STOP : OFF			

ON:The buffer is always turned on.A current flows the input buffer if the input pin is not driven.

OFF:The buffer is always turned off.

—: No applicable

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

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

Note: Condition A/B are as follows.

(SYSCR2) register setting		HALT mode	
<DRVE>	<SELDRV>	IDLE1	STOP
0	0	Condition B	Condition A
0	1	Condition A	
1	0	Condition B	Condition B
1	1		

Table 3.3.8 Output buffer State Table

Port Name	Output Function Name		Output Buffer State							
			During Reset	When the CPU is Operating		In HALT mode (IDLE2)		In HALT mode(IDLE1/STOP)		
								Condition A (Note)		Condition B (Note)	
				When Used as function Pin	When Used as Output Port	When Used as function Pin	When Used as Output Port	When Used as function Pin	When Used as Output Port	When Used as function Pin	When Used as Output Port
D0-D7	–		OFF	ON upon external write	–	OFF	–	OFF	–	OFF	–
P10-17	D8-15				ON		ON		OFF		ON
P20-27	A16-23		ON	ON	–	ON	–	OFF	–	ON	–
A0-15	–										
$\overline{RD}$	–										
$\overline{WR}$	–										
P52(*1)	$\overline{HWR}$		OFF		ON		ON		OFF		ON
P53(*1)	$\overline{EXWR}$										
P56(*1)	R/W										
P60(*1)	$\overline{CS0}$,LCLK0										
P61(*1)	$\overline{CS1}$										
$\overline{CS2}$, $\overline{CS2A}$	–		ON		–		–		–		–
P63(*1)	$\overline{CS3}$, $\overline{RAS}$		OFF		ON		ON		OFF		ON
P64(*1)	EA24 , $\overline{CS2B}$										
P65(*1)	EA25 , $\overline{CS2C}$, LCLK , $\overline{VEECLK}$										
P66(*1)	$\overline{UCAS}$, $\overline{UDS}$, $\overline{WE}$										
P67(*1)	$\overline{LCAS}$, $\overline{LDS}$, REFOUT				ON		ON		OFF		ON
P70(*1)	SCOUT,TA1OUT										
P71(*1)	OPTTX0, $\overline{CS2D}$										
P72(*1)	$\overline{CS2E}$										
P73(*1)	$\overline{DRAMOE}$, $\overline{EXRD}$										
P74(*1)	$\overline{WE}$, $\overline{CAS}$										
PB0-B2(*1) (*2)	–		OFF	–	ON	–	ON	–	OFF	–	ON
PB3-B5 (*1)	–			–		–		–		–	
PC3(*1)	TXD1			ON		ON		OFF		ON	
PC4(*1)	–			–		–		–		–	
PC5(*1)	SCLK1			ON		ON		OFF		ON	
PC6	–			–		–		–		–	
PC7	XT2	For oscillator	ON	ON	OFF	ON	OFF	OFF	OFF	OFF	OFF
		For port	OFF	OFF	ON	OFF	ON				
PD0(*1)	D1BSCP		OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
PD1(*1)	D2BLP										
PD2(*1)	D3BFR										
PD3(*1)	DLEBCD										
PD4(*1)	DOFFB										
PD6(*1)	MLDALM,ALARM										
PD7(*1)	MLDALM										
X2	–		ON	ON	–	ON	–	IDLE1 : ON , STOP : output "H" level			

ON: The buffer is always turned on.When the bus is released, however, output buffers for some pins are turned off.

OFF:The buffer is always turned off.

－: No applicable

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

*2: If one of VLD0-2 pin is used as VLD function, others cannot be used as output port even if set port function.

Note: Condition A/B are as follows.

SYSCR2 register setting		HALT mode	
<DRVE>	<SELDRV>	IDLE1	STOP
0	0	Condition B	Condition A
0	1	Condition A	
1	0	Condition B	Condition B

3.4 Interrupts

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

The TMP91C016 has a total of 40 interrupts divided into the following three types:

- Interrupts generated by CPU: 9 sources
  (Software interrupts, illegal instruction interrupt)
- Internal interrupts: 25 sources
- Interrupts on external pins ($\overline{\text{NMI}}$ and INT0 to INT3, INTKEY): 6 sources

A Fixed individual interrupt vector number is assigned to each interrupt.

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

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

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

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

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

For example, specifying "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 (<IFF2:0> = 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 1 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, TMP91C016 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.

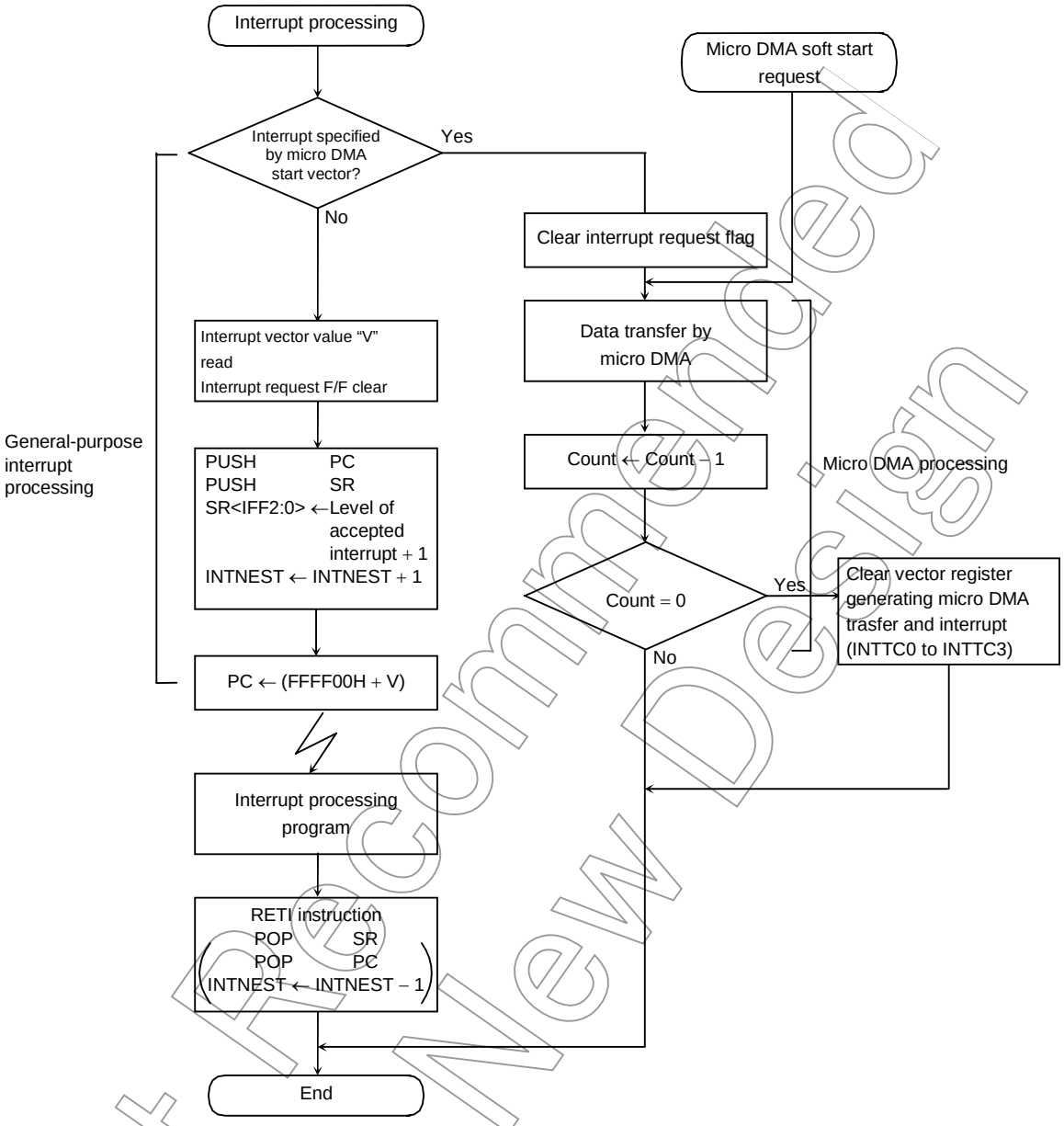


Figure 3.4.1 Overall Interrupt Processing Flow

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 <IFF2:0>. However, if the priority level of the accepted interrupt is 7, the register's value is set to 7.

(4) The CPU increases the interrupt nesting counter INTNEST by 1 (+1).

(5) The CPU jumps to the address indicated by the data at address "FFFF00H + interrupt vector" and starts the interrupt processing routine.

The above processing time is 18 states (1.33 μs at 27 MHz) as the best case (16-bit data-bus width and 0 waits).

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 <IFF2:0> comes out, the CPU accepts its interrupt. Then, the CPU interrupt mask register <IFF2:0> 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 <IFF2:0> to 111, disabling all maskable interrupts.

Table 3.4.1 shows the TMP91C016 interrupt vectors and micro DMA start vectors. The address FFFF00H to FFFFFFH (256 bytes) is assigned for the interrupt vector area.

(6) INTVLD0 to INTVLD2 are treated non-maskable interrupt in this interrupt circuit, but these interruption actually are maskable at VLD circuit source level.

Table 3.4.1 TMP91C016 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	–
11	(Note) Non-maskable	INTVLD0 pin	0098H	FFFF98H	–
12		INTVLD1 pin	009CH	FFFF9CH	–
13		INTVLD2 pin	00A0H	FFFFA0H	–
–	maskable	Micro DMA (MDMA)	–	–	–
14		INT0 pin	0028H	FFFF28H	0AH
15		INT1 pin	002CH	FFFF2CH	0BH
16		INT2 pin	0030H	FFFF30H	0CH
17		INT3 pin	0034H	FFFF34H	0DH
18		INTALM0: ALM0 (8k Hz)	0038H	FFFF38H	0EH
19		INTALM1: ALM1 (512Hz)	003CH	FFFF3CH	0FH
20		INTALM2: ALM2 (64 Hz)	0040H	FFFF40H	10H
21		INTALM3: ALM3 (2 Hz)	0044H	FFFF44H	11H
22		INTALM4: ALM4 (1 Hz)	0048H	FFFF48H	12H
23		INTTA0: 8-bit timer 0	004CH	FFFF4CH	13H
24		INTTA1: 8-bit timer 1	0050H	FFFF50H	14H
25		INTTA2: 8-bit timer 2	0054H	FFFF54H	15H
26		INTTA3: 8-bit timer 3	0058H	FFFF58H	16H
27		INTRX0: Serial reception (Channel 0)	005CH	FFFF5CH	17H
28		INTTX0: Serial transmission (Channel 0)	0060H	FFFF60H	18H
29		INTRX1: Serial reception (Channel 1)	0064H	FFFF64H	19H
30		INTTX1: Serial transmission (Channel 1)	0068H	FFFF68H	1AH
31		INTKEY: Key wake up	0070H	FFFF70H	1CH
32		INTRTC: RTC (Alarm interrupt)	0074H	FFFF74H	1DH
33		INTLCD: LCDC/LP pin	007CH	FFFF7CH	1FH
34		INTP0: Protect 0 (WR to Special SFR)	0080H	FFFF80H	20H
35		INTP1: Protect 1 (WR to ROM)	0084H	FFFF84H	21H
36		INTTC0: Micro DMA End (Channel 0)	0088H	FFFF88H	–
37		INTTC1: Micro DMA End (Channel 1)	008CH	FFFF8CH	–
38		INTTC2: Micro DMA End (Channel 2)	0090H	FFFF90H	–
39		INTTC3: Micro DMA End (Channel 3)	0094H	FFFF94H	–
		(Reserved) to (Reserved)	0098H to 00FCH	FFFF98H to FFFFFCH	– to –

Note: INTVLD0 to INTVLD2 are controlled by VLDCRx register. (Maskable: Source level)

3.4.2 Micro DMA Processing

In addition to general-purpose interrupt processing, the TMP91C016 supports 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 standby mode (STOP, IDLE1 and IDLE2) by HALT instruction, the requirement of micro DMA will be ignored (Pending) and DMA transfer is started after release HALT.

(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 <IFF2:0> = "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. (Note) 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 16 Mbytes (The upper eight bits of the 32 bits are not valid).

Note: If the priority level of micro DMA is set higher than that of other interrupts, CPU operates as follows.
In case INTxxx interrupt is generated first and then INTyyy interrupt is generated between checking "Interrupt specified by micro DMA start vector" (in the Table 3.4.1) and reading interrupt vector with setting below. The vector shifts to that of INTyyy at the time.
This is because the priority level of INTyyy is higher than that of INTxxx.
In the interrupt routine, CPU reads the vector of INTyyy because cheking of micro DMA has finished.
And INTyyy is generated regardless of transfer counter of micro DMA.
INTxxx: level 1 without micro DMA
INTyyy: level 6 with micro DMA

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) "Detailed description of the 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 34 interrupts shown in the micro DMA start vectors of Table 3.4.1 and by the micro DMA soft start, making a total of 35 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, transfer source/transfer destination addresses both even-numberd values).

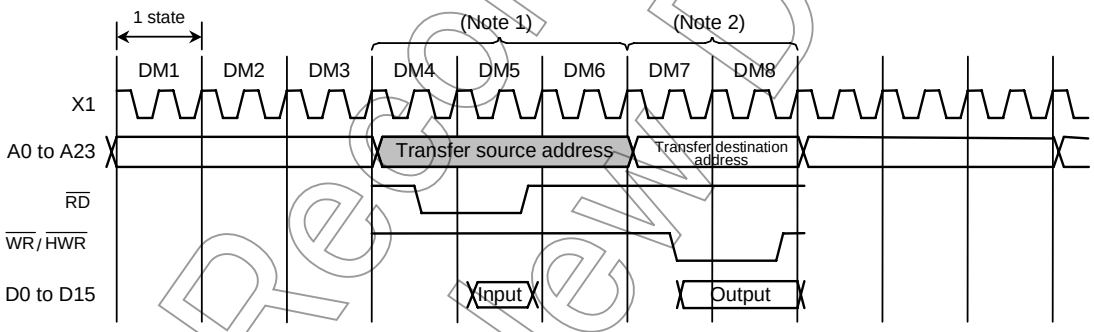


Figure 3.4.2 Timing for Micro DMA Cycle

States 1 to 3: Instruction fetch cycle (Gets next address code).
If 3 bytes and more instruction codes are inserted in the instruction queue buffer, this cycle becomes a dummy cycle.

States 4 to 5: Micro DMA read cycle

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

States 7 to 8: Micro DMA write cycle

Note 1: If the source address area is an 8-bit bus, it is increased by two states.
If the source address area is a 16-bit bus and the address starts from an odd number, it is increased by two states.

Note 2: If the destination address area is an 8-bit bus, it is increased by two states.
If the destination address area is a 16-bit bus and the address starts from an odd number, it is increased by two states.

(2) Soft start function

In addition to starting the micro DMA function by interrupts, TMP91C016 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 (If write "0" to each bit, micro DMA doesn't operate). At the end of transfer, the corresponding bit of the DMAR register is automatically cleared to "0".

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

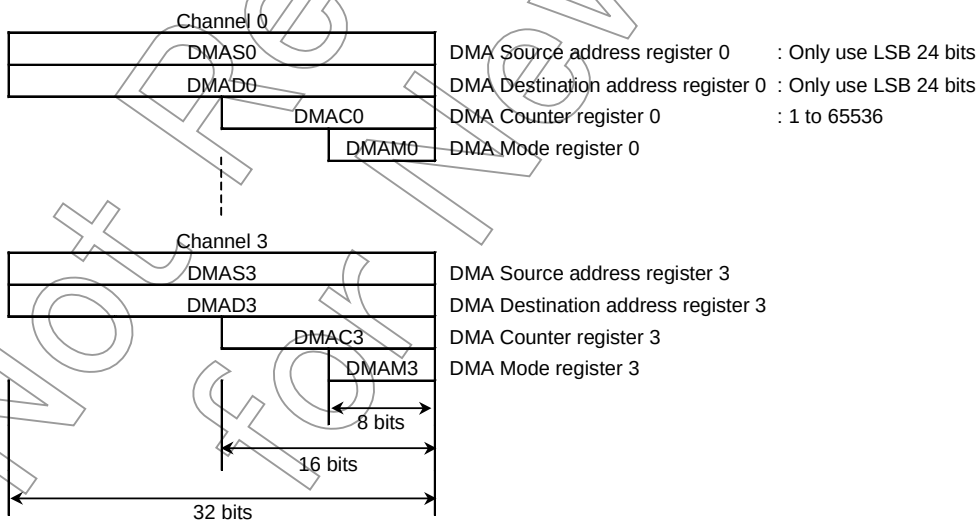
When writing again "1" to the DMAR register, check whether the bit is 0 before writing "1". If read "1", micro DMA transfer isn't started yet.

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. If execute soft start during micro DMA transfer by interrupt source, micro DMA transfer counter doesn't change. Don't use Read-modify-write instruction to avoid writing to other bits by mistake.

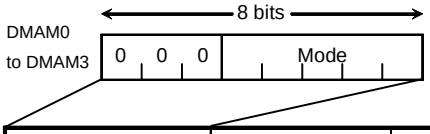
Symbol	Name	Address	7	6	5	4	3	2	1	0
DMAR	DMA request register	89H (Prohibit RMW)					DMA request			
							DMAR3	DMAR2	DMAR1	DMAR0
							R/W			
							0	0	0	0

(3) Transfer control registers

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



(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 at fc = 27 MHz
000 (Fixed)	000	00	Byte transfer	Transfer destination address INC mode .....I/O to memory	8 states	593 ns
		01	Word transfer	(DMADn+) ← (DMASn) DMACn ← DMACn − 1		
		10	4-byte transfer	If DMACn = 0, then INTTCn is generated.	12 states	889 ns
	001	00	Byte transfer	Transfer destination address DEC mode .....I/O to memory	8 states	593 ns
		01	Word transfer	(DMADn−) ← (DMASn) DMACn ← DMACn − 1		
		10	4-byte transfer	If DMACn = 0, then INTTCn is generated.	12 states	889 ns
	010	00	Byte transfer	Transfer source address INC mode ..... Memory to I/O	8 states	593 ns
		01	Word transfer	(DMADn) ← (DMASn+) DMACn ← DMACn − 1		
		10	4-byte transfer	If DMACn = 0, then INTTCn is generated.	12 states	889 ns
	011	00	Byte transfer	Transfer source address DEC mode ..... Memory to I/O	8 states	593 ns
		01	Word transfer	(DMADn) ← (DMASn−) DMACn ← DMACn − 1		
		10	4-byte transfer	If DMACn = 0, then INTTCn is generated.	12 states	889 ns
	100	00	Byte transfer	Fixed address mode .....I/O to I/O	8 states	593 ns
		01	Word transfer	(DMADn) ← (DMASn−) DMACn ← DMACn − 1		
		10	4-byte transfer	If DMACn = 0, then INTTCn is generated.	12 states	889 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 states	370 ns

Note 1: “n” is the corresponding micro DMA channels 0 to 3
DMADn+/DMASn+: Post-increment (Increment register value after transfer)
DMADn−/DMASn−: Post-decrement (Decrement register value after transfer)
The I/Os in the table mean fixed address and the memory means increment (INC) or decrement (DEC) addresses.

Note 2: Execution time is under the condition of:
16-bit bus width (Both translation and destination address area)/0 waits/
fc = 27 MHz/selected high frequency mode (fc × 1)

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

3.4.3 Interrupt Controller Operation

The block diagram inFigure 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 "0" 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 watchdog timer interrupts) is fixed at 7. If interrupt request with the same level are generated at the same time, the default priority (The interrupt with the lowest priority or, in other words, the interrupt with the lowest vector value) is used to determine which interrupt request is accepted first.

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

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

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

The interrupt controller also has registers (4 channels) used to store the micro DMA start vector. Writing the start vector of the interrupt source for the micro DMA processing (See Table 3.4.1), 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.

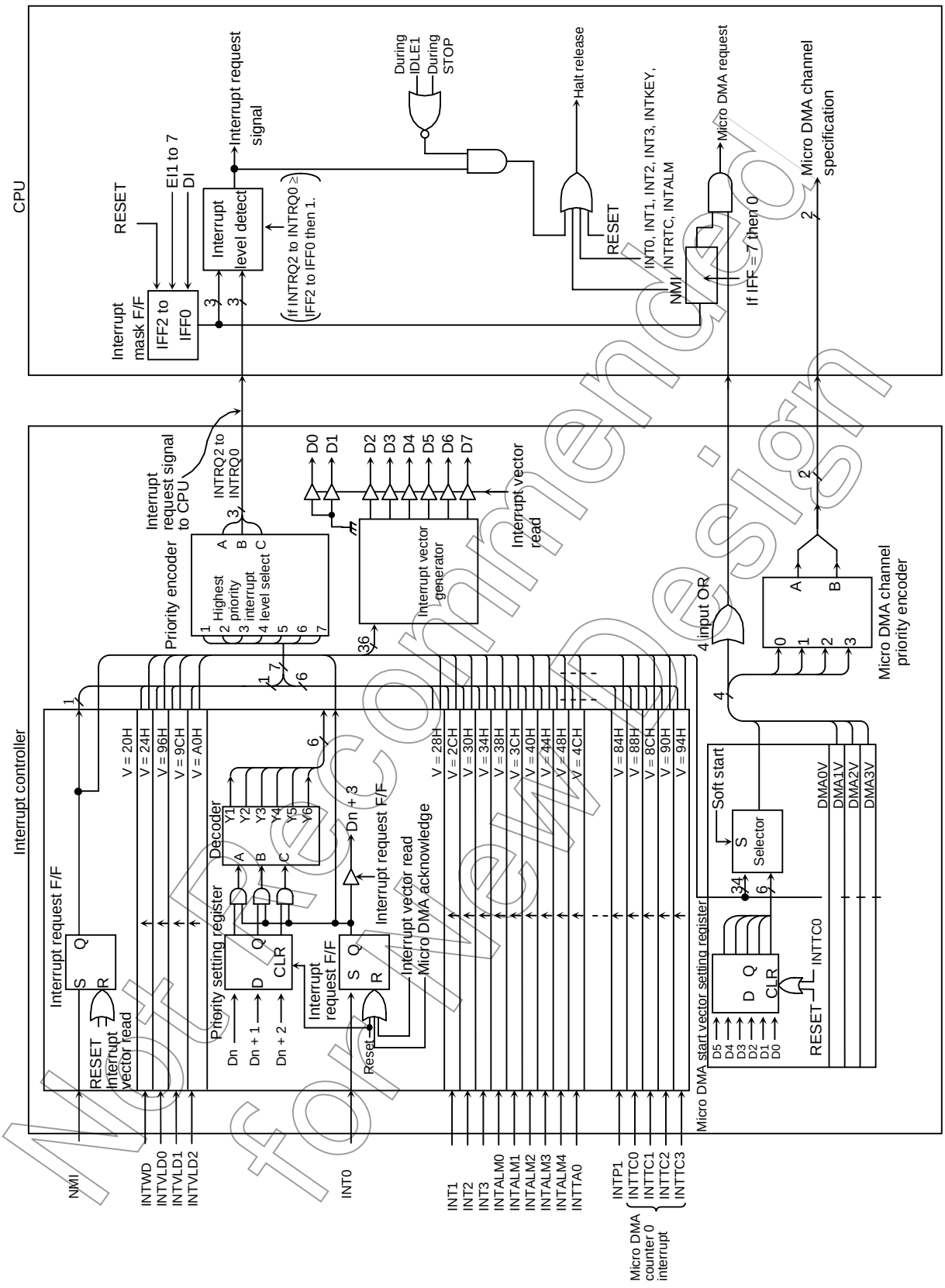


Figure 3.4.3 Block Diagram of Interrupt Controller

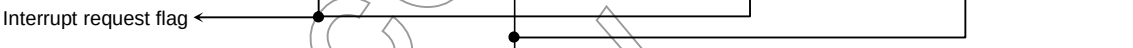
(1) Interrupt priority setting registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE0	INT0 enable	90H					INT0			
							I0C	I0M2	I0M1	I0M0
							R	R/W		
							0	0	0	0
INTE12	INT1 and 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
INTE3ALM4	INT3 and INTALM4 enable	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
INTEALM01	INTALM0 and INTALM1 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
INTEALM23	INTALM2 and INTALM3 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 and 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 and 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
INTERTCKEY	INTRTC and INTKEY enable	97H	INTKEY				INTRTC			
			IKC	IKM2	IKM1	IKM0	IRC	IRM2	IRM1	IRM0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0

Interrupt request flag

IxxM2	IxxM1	IxxM0	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 and 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
INTELCD	INTLCD enable	9AH	INTLCD							
			ILCD1C	ILCDM2	ILCDM1	ILCDM0				
			R	R/W						
			0	0	0	0				
INTETC01	INTTC0 and 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 and 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 and 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



IxxM2	IxxM1	IxxM0	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 (Prohibit 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 $\overline{NMI}$

INT0 level enable

0	Edge detect INT
1	H level INT

$\overline{NMI}$ rising edge enable

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

(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 (Prohibit 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	DMA1V2	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 "0" after micro DMA start. Setting a bit which corresponds to the micro DMA channel of the DMAB registers mentioned below to "1" specifies a burst.

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMAR	DMA software request register	89H (Prohibit RMW)					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
							1: DMA burst request			

(6) Attention point

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

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

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

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

INT0 level mode	In Level mode INT0 is not an edge-triggered interrupt. Hence, in Level mode the interrupt request flip-flop for INT0 does not function. The peripheral interrupt request passes through the S input of the flip-flop and becomes the Q output. If the interrupt input mode is changed from edge mode to level mode, the interrupt request flag is cleared automatically.

	If the CPU enters the interrupt response sequence as a result of INT0 going from 0 to 1, INT0 must then be held at 1 until the interrupt response sequence has been completed. If INT0 is set to level mode so as to release a halt state, INT0 must be held at 1 from the time INT0 changes from 0 to 1 until the halt state is released. (Hence, it is necessary to ensure that input noise is not interpreted as a 0, causing INT0 to revert to 0 before the halt state has been released.)
	When the mode changes from level mode to edge mode, interrupt request flags which were set in level mode will not be cleared. Interrupt request flags must be cleared using the following sequence.
	DI
	LD (IIMC), 00H ; Switches interrupt input mode from level mode to edge mode.
	LD (INTCLR), 0AH ; Clears interrupt request flag.
	NOP ; Wait EI instruction
	EI

INTRX	The interrupt request flip-flop can only be cleared by a reset or by reading the serial channel receive buffer. It cannot be cleared by writing INTCLR register.

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

INT0: Instructions which switch to level mode after an interrupt request has been generated in edge mode.
The pin input change from high to low after interrupt request has been generated in level mode. (H → L)

INTRX: Instruction which read the receive buffer

3.5 Port Functions

The TMP91C016 features 57-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/D = with programmable pull-up/pull-down resistor
PU = with programmable pull-up resistor
PD = with programmable 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	P52	1	I/O	PU/D	Bit	$\overline{HWR}$, INT3
	P53	1	I/O	PU	Bit	$\overline{WAIT}$, $\overline{EXWR}$
	P56	1	I/O	PU	Bit	R/W, MSK
Port 6	P60	1	I/O	PU	Bit	$\overline{CS0}$, LCLK0
	P61	1	I/O	PU	Bit	$\overline{CS1}$
	P63	1	I/O	PU	Bit	$\overline{CS3}$, $\overline{RAS}$
	P64	1	I/O	PU	Bit	EA24, $\overline{CS2B}$
	P65	1	I/O	PU	Bit	EA25, $\overline{CS2C}$, LCLK, VEECLK
	P66	1	I/O	PU	Bit	$\overline{UCAS}$, $\overline{UDS}$, $\overline{WE}$
	P67	1	I/O	PU	Bit	$\overline{LCAS}$, $\overline{LDS}$, $\overline{REFOUT}$
Port 7	P70	1	I/O	PU	Bit	SCOUT, TA1OUT
	P71	1	I/O	PU	Bit	OPTTX0, $\overline{CS2D}$
	P72	1	I/O	PU/D	Bit	OPTRX0, $\overline{CS2E}$
	P73	1	I/O	PU	Bit	$\overline{DRAMOE}$, $\overline{EXRD}$
	P74	1	I/O	PU	Bit	$\overline{WE}$, $\overline{NMI}$, $\overline{CAS}$
Port 9	P90 to P97	8	Input	PU	(Fixed)	KI0 to KI7
Port B	PB0	1	I/O	PU	Bit	VLD0
	PB1	1	I/O	PU	Bit	VLD1
	PB2	1	I/O	PU	Bit	VLD2
	PB3	1	I/O	PU	Bit	INT0
	PB4	1	I/O	PU/D	Bit	INT1
	PB5	1	I/O	PU/D	Bit	INT2
Port C	PC3	1	I/O	PU	Bit	TXD1
	PC4	1	I/O	PU/D	Bit	RXD1
	PC5	1	I/O	PU/D	Bit	SCLK1, $\overline{CTS1}$
	PC6	1	I/O	–	Bit	XT1
	PC7	1	I/O	–	Bit	XT2
Port D	PD0	1	I/O	PU	Bit	D1BSCP
	PD1	1	I/O	PU	Bit	D2BLP
	PD2	1	I/O	PU	Bit	D3BFR
	PD3	1	I/O	PU	Bit	DLEBCD
	PD4	1	I/O	PU	Bit	DOFFB
	PD6	1	I/O	PU	Bit	ALARM, $\overline{MLDALM}$
	PD7	1	I/O	PU	Bit	MLDALM

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

Port	Pin Name	Specification	I/O Register Setting Data				
			Pn	PnCR	PnFC	PnFC2	PnFC3
Port 1 (Note 1)	P10 to P17	Input port	X	0	–	–	–
		Output port	X	1	–	–	–
		D8 to D15 bus	X	X	–	–	–
Port 2	P20 to P27	Output port	X	–	0	–	–
		A16 to A23 output	X	–	1	–	–
Port 5	P52, P53, P56	Input port	X	0	0	X	–
		Output port	X	1	0	X	–
	P52	$\overline{HWR}$ output	X	1	1	0	–
		INT3 input	X	0	X	1	–
	P53	$\overline{WAIT}$ input (Note 2)	–	–	–	–	–
		$\overline{EXWR}$ output	X	1	1	–	–
	P56	R/$\overline{W}$ output	X	1	1	–	–
		MSK input (Note 3)	X	X	X	Logical selection	–
Port 6	P60, P61, P63 to P67	Input port	X	0	0	0	0
		Output port	X	1	0	0	0
	P60	$\overline{CS0}$ output (Note 10)	X	1	1	–	0
		LCLK output (Note 10)	X	1	1	–	1
	P61	$\overline{CS1}$ output	X	1	1	0	0
		$\overline{CS2}$ output	X	1	0	–	–
		$\overline{CS2A}$ output	X	1	1	–	–
	P63	$\overline{CS3}$ output (Note 14)	X	1	1	0	–
		$\overline{RAS}$ output (Note 14)	X	1	1	0	–
	P64	EA24 output	X	1	1	0	–
		$\overline{CS2B}$ output	X	1	X	1	–
	P65	EA25 output	X	1	1	0	0
		$\overline{CS2C}$ output (Note 5, 11)	X	1	X	1	0
		VEECLK output (Note 5)	X	1	X	X	0
		LCLK output (Note 11)	X	1	X	1	1
	P66	$\overline{UCAS}$ output (Note 6)	X	1	1	0	–
		$\overline{UDS}$ output	X	1	X	1	–
		$\overline{WE}$ output (Note 6)	X	1	1	0	–
	P67	$\overline{LCAS}$ output (Note 7)	X	1	1	0	–
		$\overline{LDS}$ output	X	1	X	1	–
		$\overline{REFOUT}$ output (Note 7)	X	1	1	0	–
Port 7	P70 to P74	Input port	X	0	0	0	–
		Output port	X	1	0	0	–
	P70	SCOUT output	X	1	X	1	–
		TA1OUT output	X	1	1	0	–
	P71	OPTTX0 output (Note 4)	X	1	X	1	–
		$\overline{CS2D}$ output	X	1	1	0	–
	P72	OPTRX0 input (Note 4)	X	0	0	–	–
		$\overline{CS2E}$ output	X	1	1	–	–
	P73	$\overline{DRAMOE}$ output	X	1	X	1	–
		$\overline{EXRD}$ output	X	1	1	0	–
	P74	$\overline{WE}$ output (Note 8)	X	1	X	1	–
		$\overline{NMI}$ input	X	0	1	X	–
		$\overline{CAS}$ output (Note 8)	X	1	X	1	–

X: Don't care

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

Port	Pin Name	Specification	I/O Register Setting Data				
			Pn	PnCR	PnFC	PnFC2	PnFC3
Port 9	P90 to P97	Input port	X	–	0	–	–
		KI0 to KI7 input	X	–	1	–	–
Port B	PB0 to PB5	Input port	X	0	0	–	–
		Output port	X	1	0	–	–
	PB0	VLD0 input (Note 12)	X	0	–	–	–
	PB1	VLD1 input (Note 12)	X	0	–	–	–
	PB2	VLD2 input (Note 12)	X	0	–	–	–
	PB3	INT0 input	X	0	1	–	–
	PB4	INT1 input	X	0	1	–	–
	PB5	INT2 input	X	0	1	–	–
Port C	PC3 to PC5 PC6, PC7	Input port	X	0	0	–	–
		Output port	X	1	0	–	–
	PC3	TXD1 output (Note 4)	X	1	1	–	–
	PC4	RXD1 input (Note 4)	X	0	–	–	–
	PC5	SCLK1 input (Note 4, 13)	X	0	0	–	–
		SCLK1 output (Note 4, 13)	X	1	1	–	–
		$\overline{CTS1}$ input (Note 4, 13)	X	0	0	–	–
	PC6	XT1 input (Note 9)	X	X	X	–	–
	PC7	XT2 output (Note 9)	X	X	X	–	–
Port D	PD0 to PD7	Input port	X	–	0	–	–
		Output port	X	0	0	–	–
	PD0	D1BSCP output	X	1	1	–	–
	PD1	D2BLP output	X	1	1	–	–
	PD2	D3BFR output	X	1	1	–	–
	PD3	DLEBCD output	X	1	1	–	–
	PD4	DOFFB output	X	1	1	–	–
	PD6	$\overline{MLDALM}$ output	1	1	1	–	–
		$\overline{ALARM}$ output	0	1	1	–	–
	PD7	MLDALM output	X	1	1	–	–

X: Don't care

Note 1: Port1 is able to set port function or data bus by AM1, AM0 setting.

Note 2: If you want to use WAIT input, it needs BxCS register (1 + N) wait setting.

Note 3: In case of P76/MSK set MSK input, it can set logical selection by P7FC<P76F>.

Note 4: OPTRX0, OPTTX0, TXD1, RXD1, SCLK1, $\overline{CTS1}$:
These pins can set input/output data's logical selection by each Pn register.

Note 5: In case of P65F2D and P65F2, both write 1, it set P65F2D (VEECLK).

Note 6: Selection of $\overline{UCAS}$ and $\overline{WE}$ depend on CS/WAIT bus width control (8 bits or 16 bits).

Note 7: Selection of $\overline{LCAS}$ and $\overline{REFOUT}$ depend on CS/WAIT bus width control (8 bits or 16 bits).

Note 8: Selection of $\overline{WE}$ and $\overline{CAS}$ depend on CS/WAIT bus width control (8 bits or 16 bits).

Note 9: Oscillator setting of XT1 and XT2 is controlled by SYSCR0<XTEN> and this control have priority over other setting.

Note 10: Selection of $\overline{CS0}$ and LCLK is set by P6FC3<P60F3>.

Note 11: Selection of $\overline{CS2C}$ and LCLK is set by P6FC3<P65F3>.

Note 12: If One of PB0 to PB2 is set VLD function, other PBx pin can't output function even port function setting. And these pin can only VLD input or port output. VLD function is set by VLDCTL<VLD*USE>.

Note 13: Selection of SCLK and $\overline{CTS}$ is set by SC1MOD0<CTSE>.

Note 14: Selection of $\overline{CS3}$ and $\overline{RAS}$ is set by B3CS<B3OM1:0>.

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).

When AM1 = 0 and AM0 = 1, port 10 to 17 always operate data bus function, even if it changes P1CR setting.

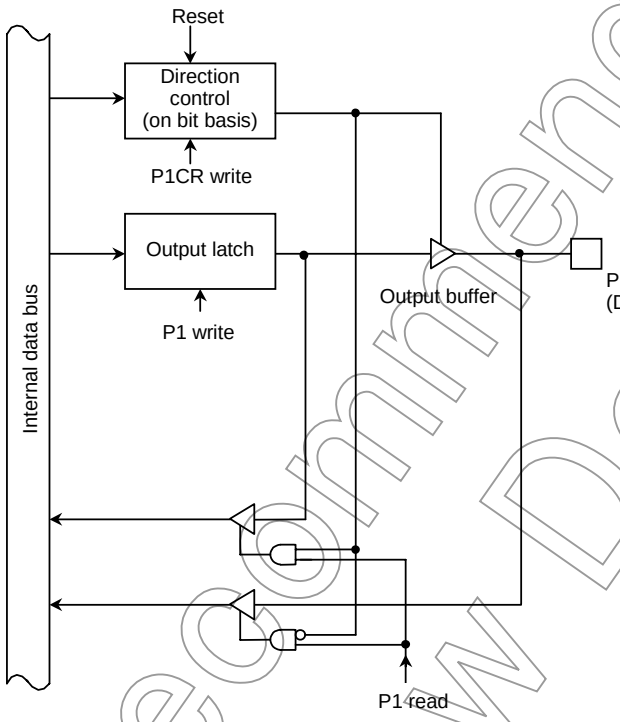


Figure 3.5.1 Port 1

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 bits 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.

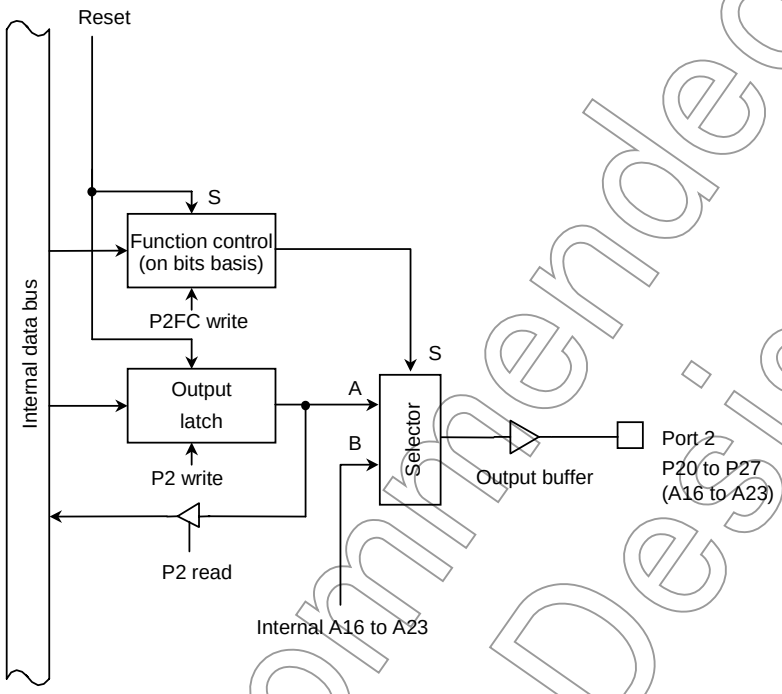


Figure 3.5.2 Port 2

Port 1 Register

P1 (0001H)

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

Port 1 Control Register

P1CR (0004H)

	7	6	5	4	3	2	1	0
Bit symbol	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
Read/Write	W							
After reset (Note)	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
Function	0: Input 1: Output							

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

Port 2 Register

P2 (0006H)

	7	6	5	4	3	2	1	0
Bit symbol	P27	P26	P25	P24	P23	P22	P21	P20
Read/Write	R/W							
After reset	1							

Port 2 Function Register

P2FC (0009H)

	7	6	5	4	3	2	1	0
Bit symbol	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
Read/Write	W							
After reset	1	1	1	1	1	1	1	1
Function	0: Port 1: Address bus (A23 to A16)							

Note1: Read-modify-write is prohibited for P1CR and P2FC.

Note2: It is set to “Port” or “Data bus” by AM pins state.

Figure 3.5.3 Registers for Ports 1 and 2

3.5.3 Port 5 (P52, P53, P56)

Port 5 is an 3-bit general-purpose I/O port. This I/O port is set using control register P5CR, P5FC, P5FC2 and P5UDE. And P52 port have $\overline{HWR}$ output, INT2 input, P53 port have $\overline{WAIT}$ input, $\overline{EXWR}$ output, P56 port have R/W output, MSK input, except port function.

Resetting resets all bits of P5 and bit 3, 5 of P5UDE to 1, all bits of P5CR, P5FC and P5FC2 to 0. And sets P52, P53, P56 to input mode with pull-up resistor.

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.

When the P5<RDE> register clearing to 0, outputs the $\overline{RD}$ strobe (used for the peused static RAM) of the $\overline{RD}$ pin even when the internal addressed.

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

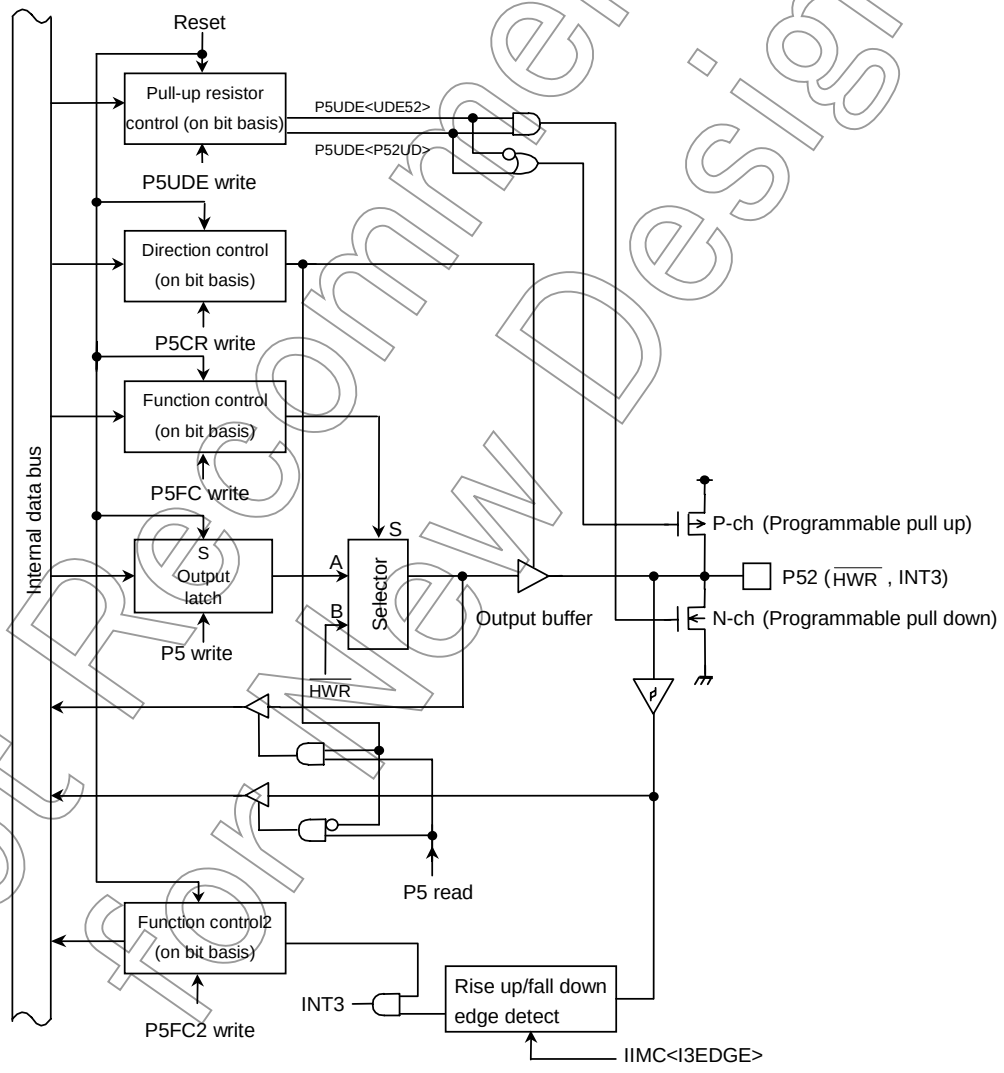


Figure 3.5.4 Port 5 (P52)

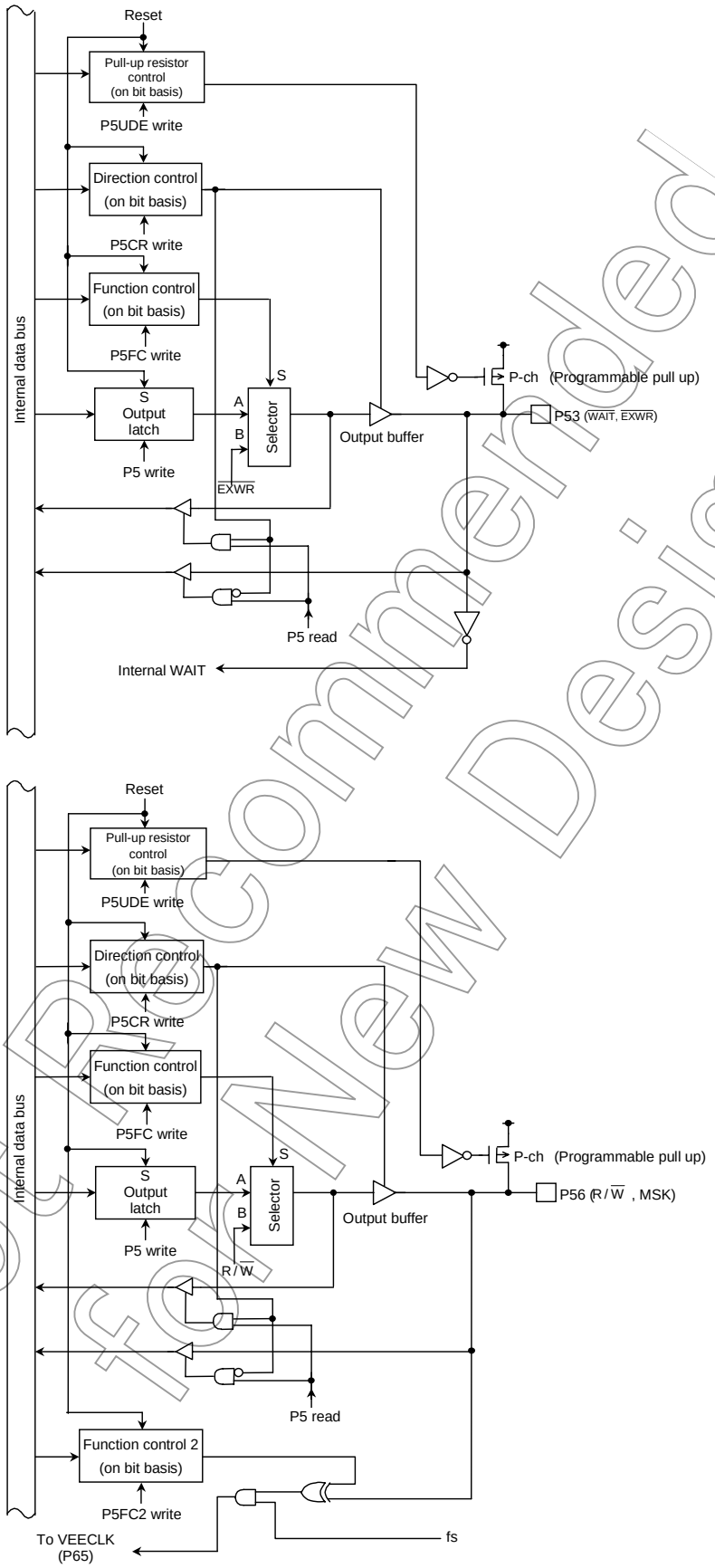


Figure 3.5.5 Port 5 (P53, P56)

Port 5 Register

P5 (000DH)

	7	6	5	4	3	2	1	0
Bit symbol		P56			P53	P52		RDE
Read/Write	R/W							
After reset	Data from external port (Output latch register is set to 1)							

Port 5 Control Register

P5CR (000AH)

	7	6	5	4	3	2	1	0
Bit symbol		P56C			P53C	P52C		
Read/Write	W							
After reset		0			0	0		
Function	0: Input 1: Output							

I/O setting

0	Input
1	Output

Port 5 Function Register

P5FC (000BH)

	7	6	5	4	3	2	1	0
Bit symbol		P56F			P53F	P52F		
Read/Write	W							
After reset		0			0	0		
Function		0: Port 1: R / $\overline{W}$			0: Port 1: $\overline{EXWR}$	0: Port 1: $\overline{HWR}$		

Port 5 Function Register 2

P5FC2 (000CH)

	7	6	5	4	3	2	1	0
Bit symbol		P56F2				P52F2		
Read/Write	W							
After reset		0				0		
Function		Logic select 0: 1 CLK 1: 0 CLK				0: <P52F> 1: INT3		

Port 5 UDE Register

P5UDE (000EH)

	7	6	5	4	3	2	1	0
Bit symbol		P56U			P53U	UDEP52	P52UD	
Read/Write	W							
After reset		1			1	1	0	
Function		Pull-up resistance 0: Disable 1: Enable			Pull-up resistance 0: Disable 1: Enable	Pull-up resistance 0: Disable 1: Enable	0: Pull up 1: Pull down	

Note 1: Read-modify-write is prohibited for register P5CR, P5FC, P5FC2 and P5UDE.

Note 2: When P53 pin is used as a $\overline{WAIT}$ pin, set P5CR<P53C> to 0 and chip select/wait control register <BnW2:0> to 010.

Figure 3.5.6 Registers for Port 5

3.5.4 Port 6 (P60, P61, P63 to P67)

Port 6 is 7-bit I/O port. This I/O port have standard chip select signal output function ($\overline{CS0}$, $\overline{CS1}$, $\overline{CS3}$), expand address signal output function (EA24, EA25), expand chip select signal output function ($\overline{CS2B}$, $\overline{CS2C}$), clock output for LCDD (VEECLK), chip select for special command for Sift Register type (LCLK), and special signals for dynamic RAM access function ($\overline{RAS}$, $\overline{CAS}$, $\overline{WE}$, $\overline{LCAS}$, $\overline{UCAS}$, $\overline{LDS}$, $\overline{UDS}$, $\overline{REFOUV}$). These function is set by P6FC and P6FC2 register. Resetting resets all bits of P6CR, P6FC, P6FC2, and 3, 6, 7 bits of P6UE to 0 and 0, 1, 4, 5 bits of P6UE to 1. And P63, P66, P67 set to cut off resistance, P60, P61, P64, P65 set to pull-up resistance input mode.

Selection of $\overline{CS2}$ and $\overline{CS2A}$ is set by P6FC<P62F>. (This terminal don't have pull-up resistance and port function)

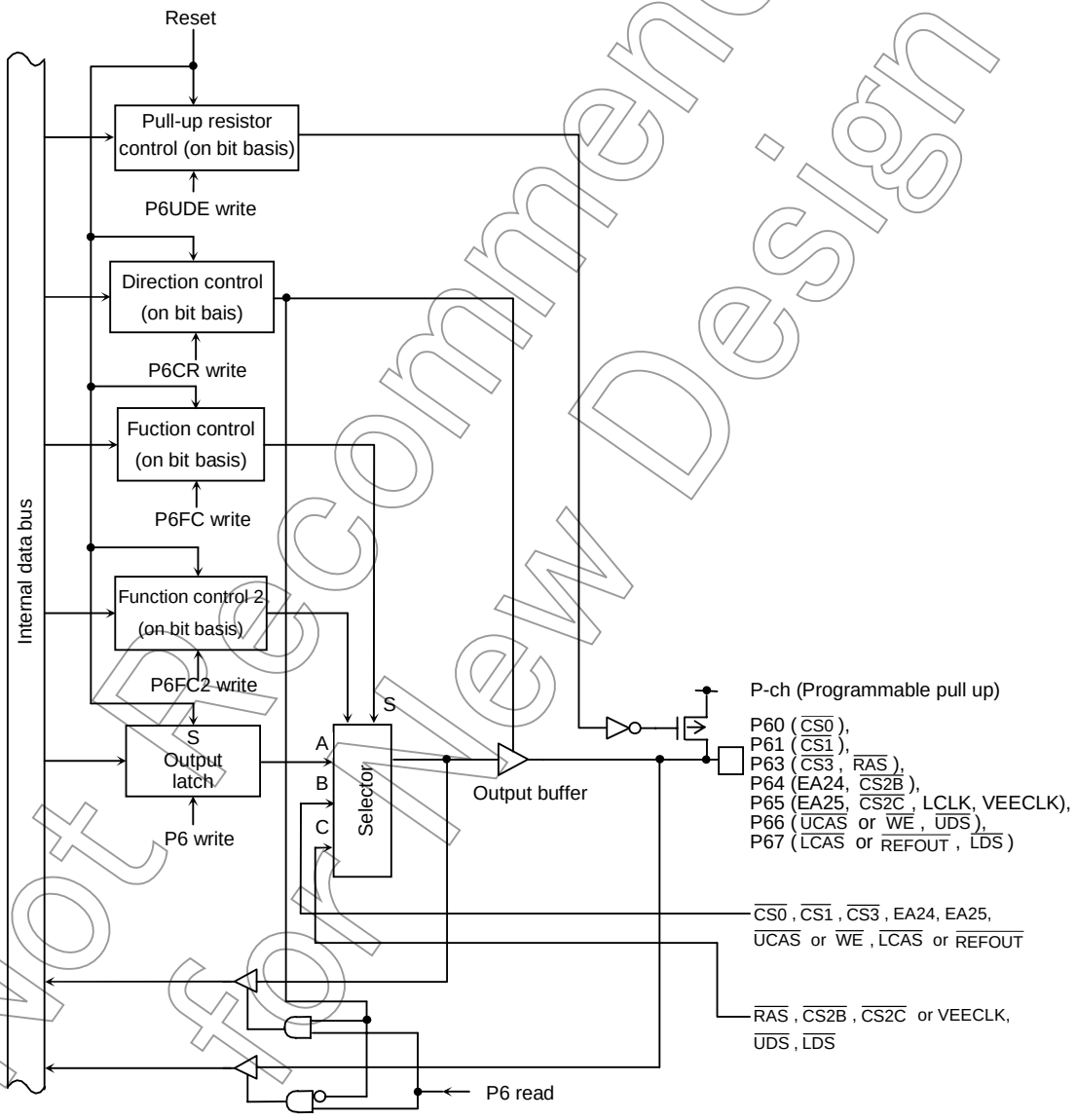


Figure 3.5.7 Port 6

Port 6 Register

P6 (0012H)

	7	6	5	4	3	2	1	0
Bit symbol	P67	P66	P65	P64	P63		P61	P60
Read/Write	R/W							
After reset	Data from external port (Output latch register is set to 1)							

Port 6 Control Register

P6CR (0014H)

	7	6	5	4	3	2	1	0
Bit symbol	P67C	P66C	P65C	P64C	P63C		P61C	P60C
Read/Write	W							
After reset	0						0	
Function	0: Input 1: Output						0: Input 1: Output	

Port 6 Function Register

P6FC (0015H)

	7	6	5	4	3	2	1	0
Bit symbol	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
Read/Write	W							
After reset	0							
Function	0: Port 1: $\overline{LCAS}$ or $\overline{REFOUV}$	0: Port 1: $\overline{UCAS}$ or $\overline{WE}$	0: Port 1: EA25	0: Port 1: EA24	0: Port 1: $\overline{CS3}$	0: $\overline{CS2}$ 1: $\overline{CS2A}$	0: Port 1: $\overline{CS1}$	0: Port 1: $\overline{CS0}$

Port 6 Function Register2

P6FC2 (001BH)

	7	6	5	4	3	2	1	0
Bit symbol	P67F2	P66F2	P65F2	P64F2	–	P65F2D		
Read/Write	W							
After reset	0							
Function	0: <P67F> 1: $\overline{LDS}$	0: <P66F> 1: $\overline{UDS}$	0: <P65F> 1: $\overline{CS2C}$	0: <P64F> 1: $\overline{CS2B}$	Always write “0”	0: <P65F2> 1: VEECLK		

Port 6 UE Register

P6UE (0018H)

	7	6	5	4	3	2	1	0
Bit symbol	P67U	P66U	P65U	P64U	P63U		P61U	P60U
Read/Write	W						W	
After reset	0		1		0		1	
Function	Pull-up resistor 0: Disable 1: Enable	Pull-up resistor 0: Disable 1: Enable	Pull-up resistor 0: Disable 1: Enable	Pull-up resistor 0: Disable 1: Enable	Pull-up resistor 0: Disable 1: Enable		Pull-up resistor 0: Disable 1: Enable	Pull-up resistor 0: Disable 1: Enable

Port 6 Function Register3

P6FC3 (0010H)

	7	6	5	4	3	2	1	0
Bit symbol			P65F3				–	P60F3
Read/Write			W				W	W
After reset			0				0	0
Function			LCLK2 selection 0: Normal 1: LCLK2				Always write “0”	LCLK0 selection 0: Normal 1: LCLK0

Note 1: Read-modify-write is prohibited for registers P6CR, P6FC, P6FC2, P6FC3 and P6UE.

Note 2: When P63 pin is used as a $\overline{CS3}$ pin and $\overline{RAS}$, set chip select/wait control register B3CS<B3OM1:0> to 10.

Figure 3.5.8 Port 6 Register

3.5.5 Port 7 (P70 to P74)

Port 7 is 5-bit general-purpose I/O port. This port can be set I/O on bit basis. Resetting resets all bits of P7CR, P7FC and P7FC2 to P7FC0, and become to input port, and all bits of P7 to P1.

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

1. Output function for 8-bit timer (TA1OUT)
2. Output function for internal clock (SCOUT)
3. Input/output function for IrDA (OPTRX0, OPTTX0)
4. Extend chip-select output ($\overline{CS2E}$, $\overline{CS2D}$)
5. DRAM control output ($\overline{WE}$, $\overline{CAS}$, $\overline{DRAMOE}$)
6. Extend read signal output ($\overline{EXRD}$)
7. Non maskable interrupt request input ($\overline{NMI}$)

Writing 1 in the corresponding bit of P7FC, P7FC2 enables the respective functions.
Resetting resets the P7FC, P7FC2 to P7FC0, and sets all bits to input ports.

(1) Port 70 (SCK, OPTRX0)

Port 70 is a general-purpose I/O port. It is also used as TA1OUT (8-bit timer output function) and SCOUT (Internal clock output function). In case of used as TA1OUT, it set to P7FC<P70F> = 1 and using SCOUT, set to P7FC2<P70F2> = 1.

Setting to P7UDE<P70U> = 1, set to pull-up resistor.

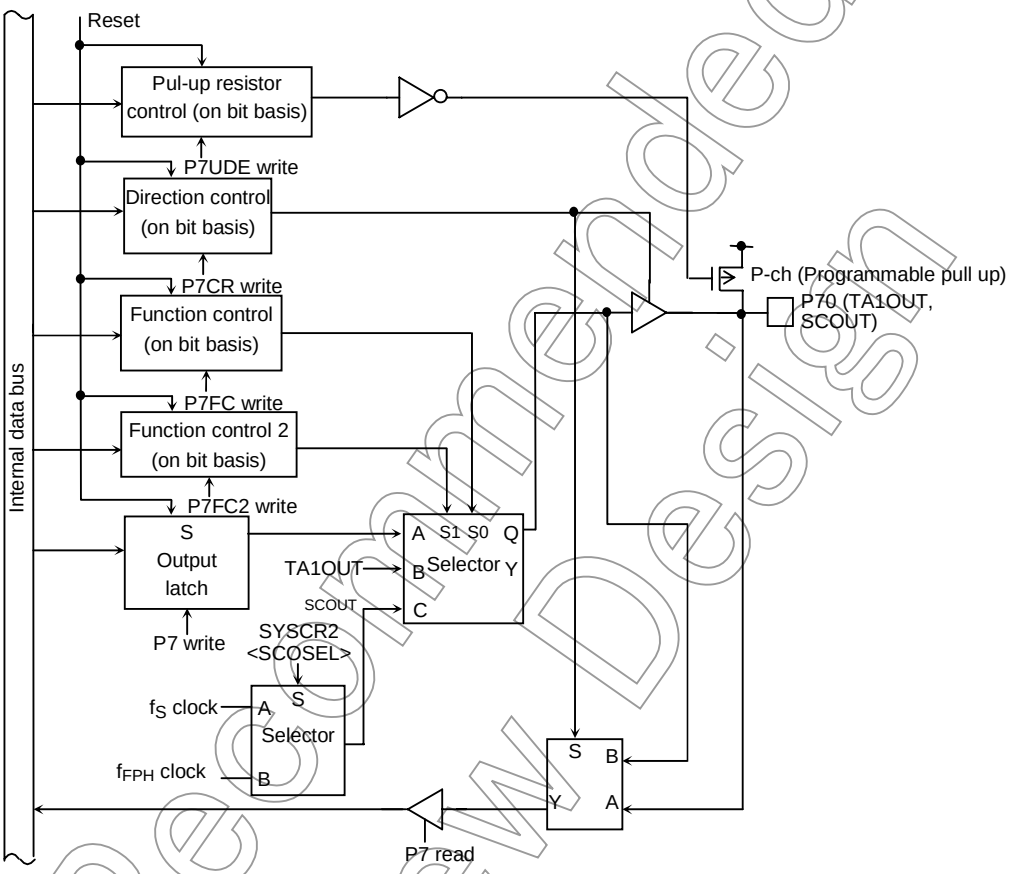


Figure 3.5.9 Port 70

(2) Port 71 ($\overline{CS2D}$, OPTTX0)

Port 71 also function as extend chip-select output ($\overline{CS2D}$) and transmitting output for IrDA mode of SIO0 (OPTTX0). When P71 is used to OPTTX0 function, it possible to control logical reverse by P7<P71>.

Setting to P7UDE<P71U> = 1, set to pull-up resistor.

Resetting it becomes to cut off pull-up resistor and become to input mode.

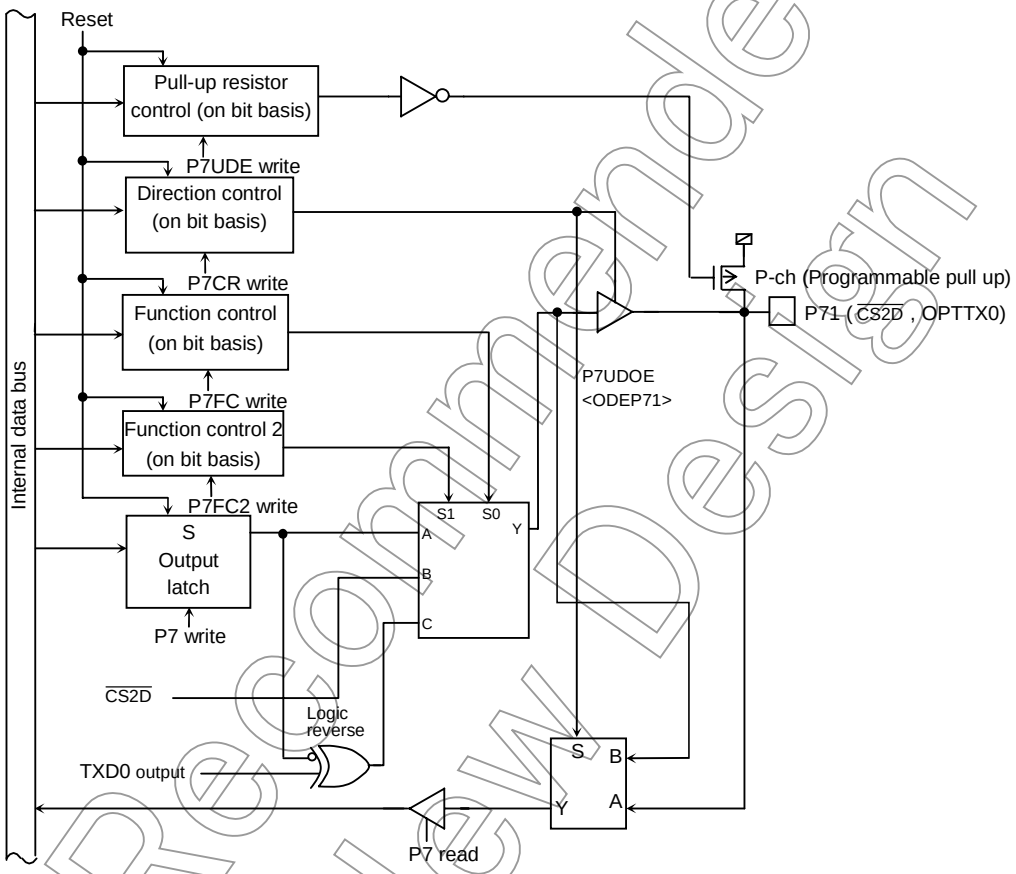


Figure 3.5.10 Port 71 ($\overline{CS2D}$, OPTTX0)

(3) Port 72 ($\overline{CS2E}$, OPTRX0)

Port 72 have also function as extend chip-select output ($\overline{CS2E}$) and receiving input for IrDA mode of SIO0 (OPTRX0). When P72 is used to OPTRX0 function, it possible to control logical reverse by P7<P72>.

Selection of pull-up or pull-down is decided with P7UDE<P72UD> and selection of enable or disable of that resistor's situation by P7UDE<P72U>. It become to input mode without pull-up/pull-down resistor by reset operation.

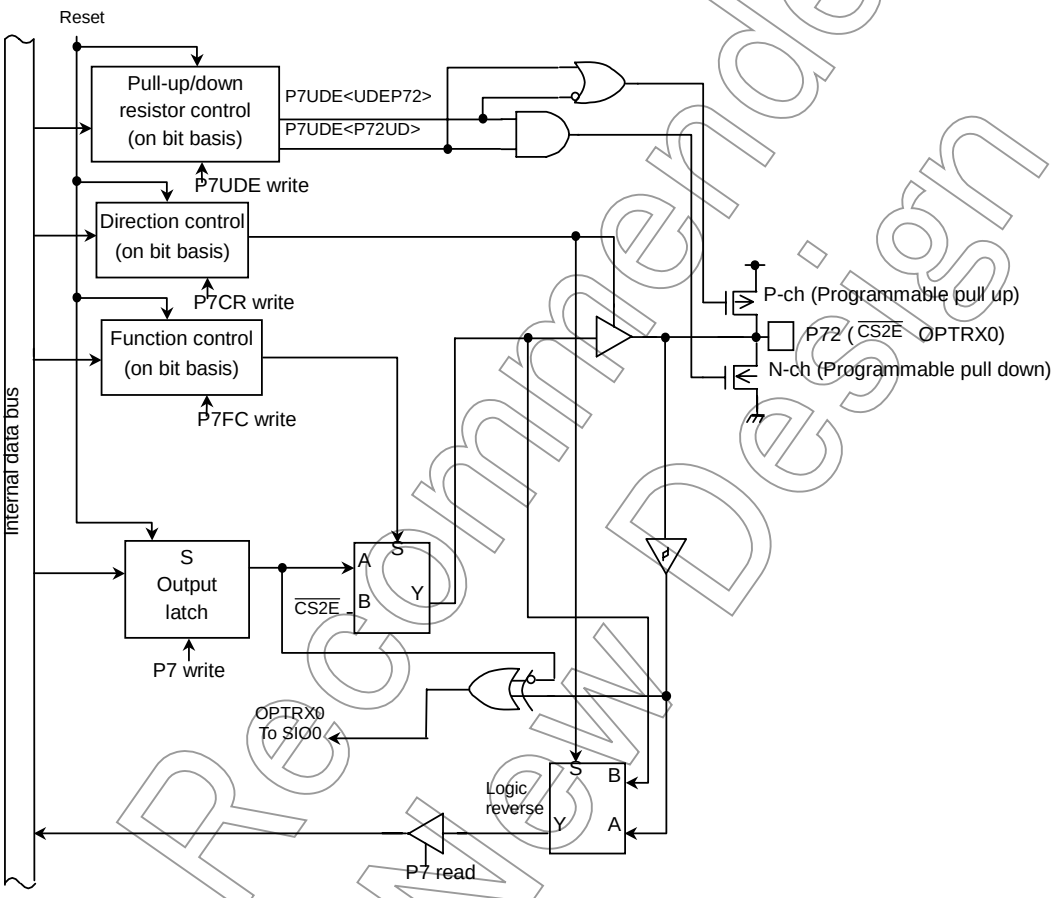


Figure 3.5.11 Port 72

(4) Port 73 ($\overline{\text{EXRD}}$, $\overline{\text{DRAMOE}}$)

Port 73 have also function as DRAM control output ($\overline{\text{DRAMOE}}$) and extend read output ($\overline{\text{EXRD}}$). $\overline{\text{EXRD}}$ output same timing as $\overline{\text{RD}}$ signal.

Setting to P7UDE<P73U> = 1, set to pull-up resistor. It become to pull-up situation by reset operation.

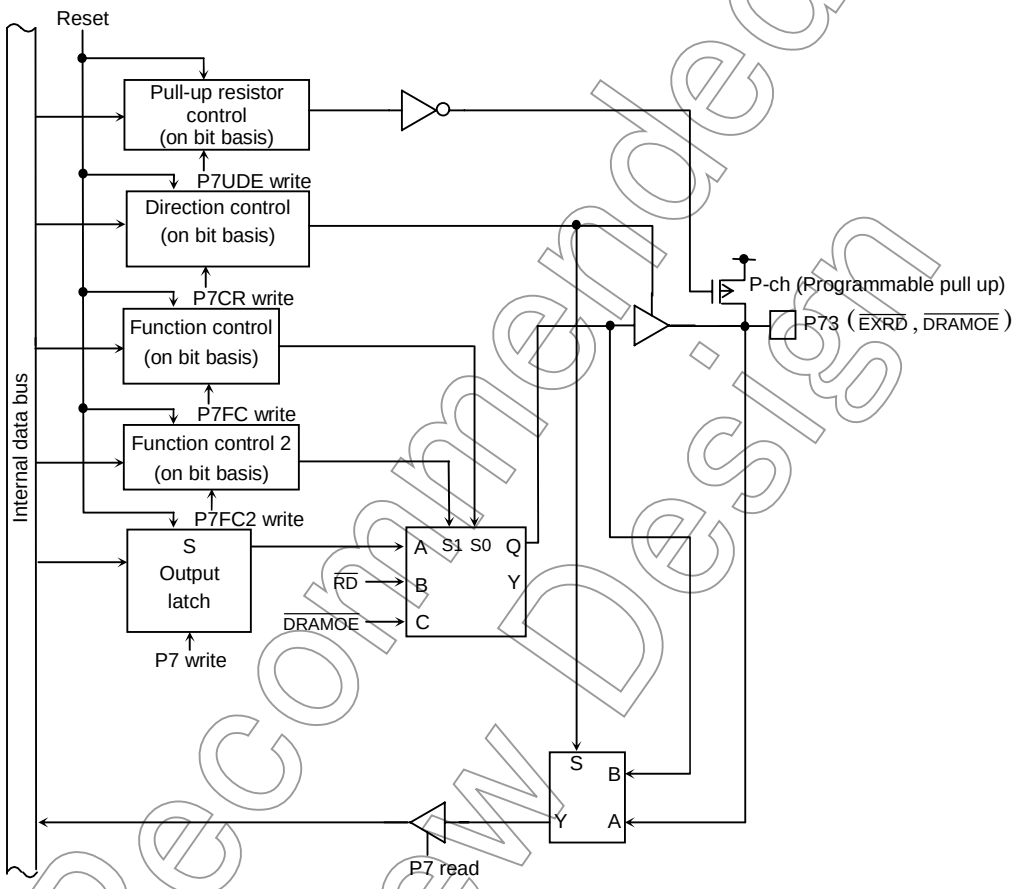


Figure 3.5.12 Port 73

(5) Port 74 ($\overline{NMI}$, $\overline{WE}$, $\overline{CAS}$)

Port 74 have also function $\overline{NMI}$ input and DRAM control output ($\overline{WE}$, $\overline{CAS}$).

And setting P7UDE<P74U> = 1, set to pull-up resistor. It become to pull-up situation by reset operation.

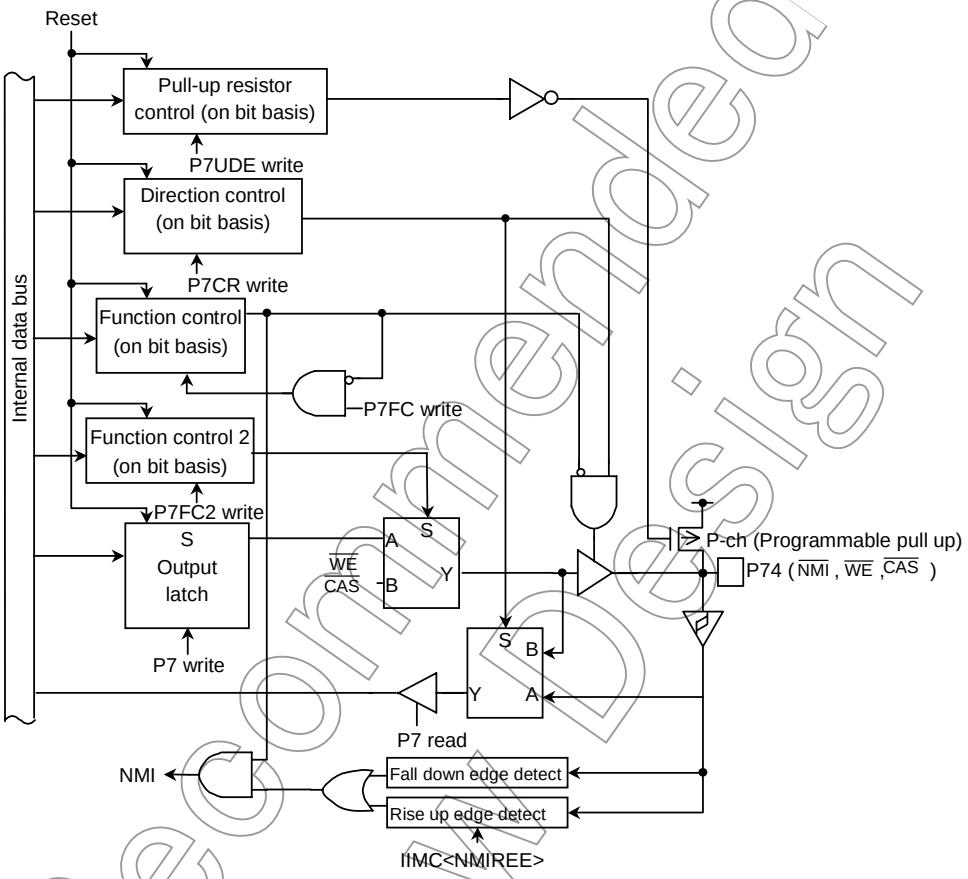


Figure 3.5.13 Port 74

Port 7 Register

P7 (0013H)

	7	6	5	4	3	2	1	0
Bit symbol				P74	P73	P72	P71	P70
Read/Write				R/W				
After reset				Data from external port (Output latch register is set to 1)				

Port 7 Control Register

P7CR (0016H)

	7	6	5	4	3	2	1	0
Bit symbol				P74C	P73C	P72C	P71C	P70C
Read/Write				W				
After reset				0	0	0	0	0
Function				0: Input 1: Output				

Port 7 Function Register

P7FC (0017H)

	7	6	5	4	3	2	1	0
Bit symbol				P74F	P73F	P72F	P71F	P70F
Read/Write				W				
After reset				0				
Function				0: Port 1: $\overline{\text{NMI}}$ input	0: Port 1: $\overline{\text{EXRD}}$ output	0: Port 1: $\overline{\text{CS2E}}$ output	0: Port 1: $\overline{\text{CS2D}}$ output	0: Port 1: TA1OUT output

Port 7 Function Register 2

P7FC2 (001CH)

	7	6	5	4	3	2	1	0
Bit symbol				P74F2	P73F2		P71F2	P70F2
Read/Write				W	W		W	W
After reset				0	0		0	0
Function				0: <P74F> 1: $\overline{\text{WE}}$, $\overline{\text{CAS}}$ output	0: <P73F> 1: $\overline{\text{DRAMOE}}$		0: <P71F> 1: OPTTX0 output	0: <P70F> 1: SCOUT output

Port 7 Pull-up/Pull-down Resistor, Open-drain Enable Register

P7UDE (001FH)

	7	6	5	4	3	2	1	0
Bit symbol			P72UD	P74U	P73U	UDEP72	P71U	P70U
Read/Write			W					
After reset			0	1	1	0	0	0
Function			0: Pull up 1: Pull down	Pull-up resistor 0: Disable 1: Enable		Resistor control 0: Disable 1: Enable	Pull-up resistor 0: Disable 1: Enable	

Note: Read-modify-write is prohibited for P7CR, P7FC, P7FC2 and P7UDE.

Figure 3.5.14 Port 7 Register

3.5.6 Port 9 (P90 to P97)

Port 90 to 97 are 8-bit input ports with pull-up resistors. In addition to functioning as general-purpose I/O port, port 90 to 97 can also key-on wakeup function as keyboard interface. The various functions can each be enabled by writing 1 to the corresponding bit of the Port 9 function register (P9FC).

Resetting resets all bits of the register P9FC to 0 and sets all pins to be input port.

And resetting resets all bits of the register P9UE to 1 and sets all pins to be pull-up port.

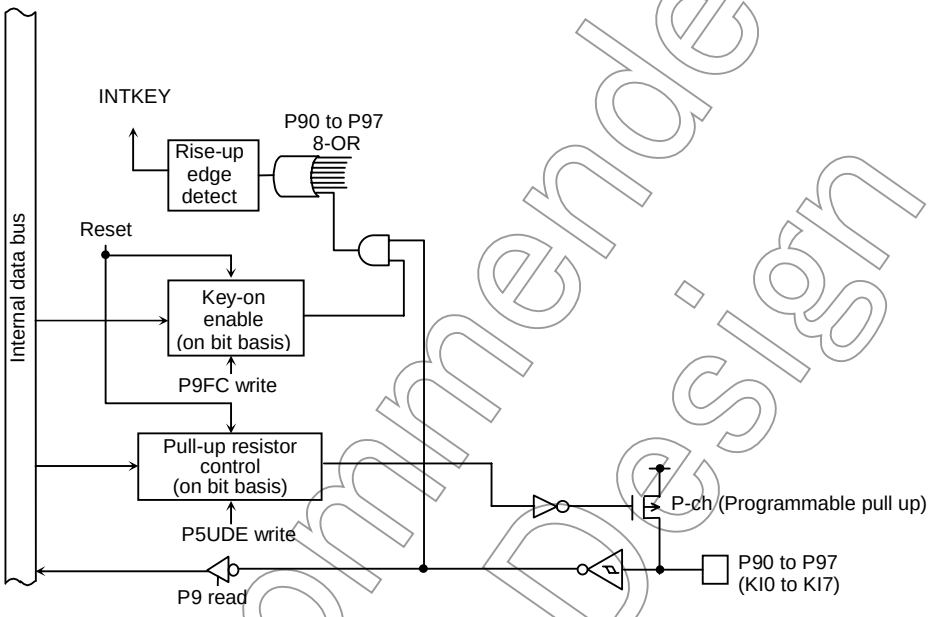


Figure 3.5.15 Port 9

When P9FC = 1, if either of input of KI0 to KI7 pins falls down, INTKEY interrupt is generated. INTKEY interrupt can be used release all HALT mode.

Port 9 Register

P9 (0019H)

	7	6	5	4	3	2	1	0
Bit symbol	P97	P96	P95	P94	P93	P92	P91	P90
Read/Write	R							
After reset	Data from external port							

Port 9 Function Register

P9FC (001DH)

	7	6	5	4	3	2	1	0
Bit symbol	P97F	P96F	P95F	P94F	P93F	P92F	P91F	P90F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0: Key-in disable 1: Key-in enable							

Key-in of Port 9

Disable	0
Enable	1

Port 9 UE Register 2

P9UE (001EH)

	7	6	5	4	3	2	1	0
Bit symbol	P97U	P96U	P95U	P94U	P93U	P92U	P91U	P90U
Read/Write	W							
After reset	1							
Function	Pull-up resistor 0: Disable 1: Enable	Pull-up resistor 0: Disable 1: Enable	Pull-up resistor 0: Disable 1: Enable	Pull-up resistor 0: Disable 1: Enable	Pull-up resistor 0: Disable 1: Enable	Pull-up resistor 0: Disable 1: Enable	Pull-up resistor 0: Disable 1: Enable	Pull-up resistor 0: Disable 1: Enable

Note: Read-modify-write is prohibited for the registers P9FC.

Figure 3.5.16 Port 9 Registers

3.5.7 Port B (PB0 to PB5)

Port B is 6-bit general-purpose I/O port. This I/O port have voltage level detector function (VLD0 to VLD2), external interrupt input function (INT0 to INT2). It can be controlled by IIMC register's setting to select of rise up/fall down for interruption.

External interrupt function is set by writing to 1 correspond bit of PBFC register. And it can set pull-up resistor to port B0 to B3, pull-up/pull-down register to port B4, B5. Selection of pull-up or pull-down, is set by writing 1 corresponding bit of PBUDE register.

Resetting resets to PBCR, PBFC, PBUDE register, port B0 to B2, B4, B5 input without resistor. Only port B3 become to input with pull-up resistor by reset operation.

(1) PB0 to PB2 (VLD0 to VLD2)

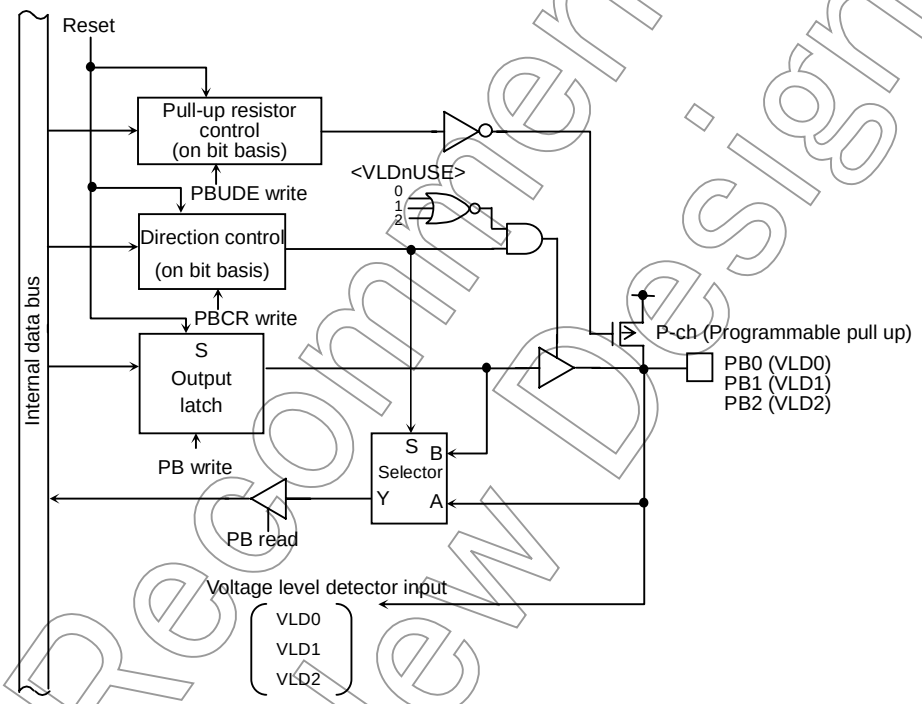


Figure 3.5.17 Port B0 to B2

(2) PB3 (INT0)

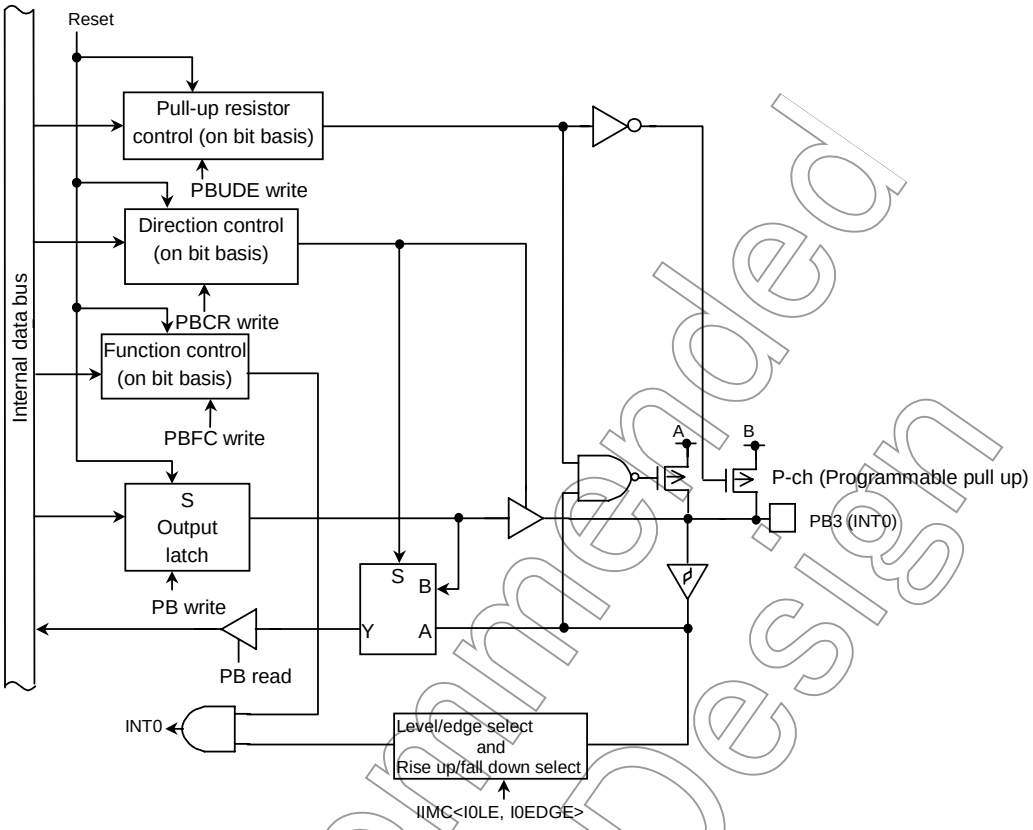


Figure 3.5.18 Port B3

(3) PB4 (INT1), PB5 (INT2)

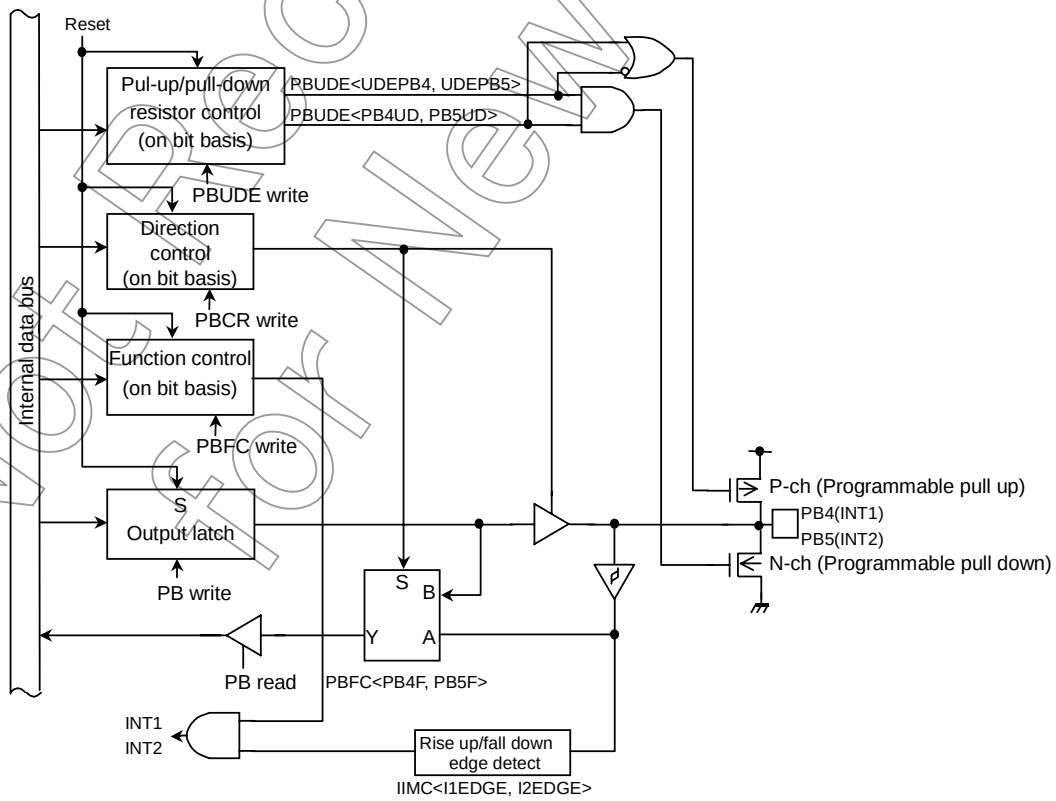


Figure 3.5.19 Port B4 to B5

Port B Register

PB (0022H)

	7	6	5	4	3	2	1	0
Bit symbol			PB5	PB4	PB3	PB2	PB1	PB0
Read/Write			R/W					
After reset			Data from external port (Output latch register is set to 1)					

Port B Control Register

PBCR (0024H)

	7	6	5	4	3	2	1	0
Bit symbol			PB5C	PB4C	PB3C	PB2C	PB1C	PB0C
Read/Write			W					
After reset			0					
Function			0: Input 1: Output					

Port B Function Register

PBFC (0025H)

	7	6	5	4	3	2	1	0
Bit symbol			PB5F	PB4F	PB3F			
Read/Write			W					
After reset			0					
Function			0: Port 1: INT2	0: Port 1: INT1	0: Port 1: INT0			

Port B Pull-up/Pull-down Resistor Control Register

PBUDE (0020H)

	7	6	5	4	3	2	1	0
Bit symbol	PB5UD	PB4UD	UDEPB5	UDEPB4	PB3U	PB2U	PB1U	PB0U
Read/Write	W							
After reset	0				1	0		
Function	Pull-up/Pull-down control 0: Pull-up resistor 1: Pull-down resistor		Resistor control 0: Disable 1: Enable		Pull-up resistor 0: Disable 1: Enable			

Note 1: Read-modify-write is prohibited PBCR, PBFC and PBUDE.

Note 2: Because PB0/VLD0, PB1/VLD1 and PB2/VLD2 can't be controlled those terminal's function by register, VLD circuit also receive signals operating input port function.

Figure 3.5.20 Port B Register

3.5.8 Port C (PC3 to PC5, PC6, PC7)

Port C is 5-bit general-purpose I/O port. By reset, these ports become to input port and set to 1 of all output latch.

Except I/O port function, this port have serial channel I/O function (SIO0, SIO1). This function is set by writing 1 data to correspond bit of PCFC register. All the data of PCCR, PCFC register, all port become to input port.

(1) Port C3 (TXD1)

Port C3 have also function as serial channel output (TXD1). When it is used to TXD1function, it possible logical reverse output by PC<PC3> register setting.

And this port's output buffer have also open-drain type except push-pull type and this selection is set by PCUDOE<ODEPC0> register.

Port C3 can set pull-up resistor by writing 1 data to PCUDOE<PC3U> register. This port become to input port without pull up, by reset operation.

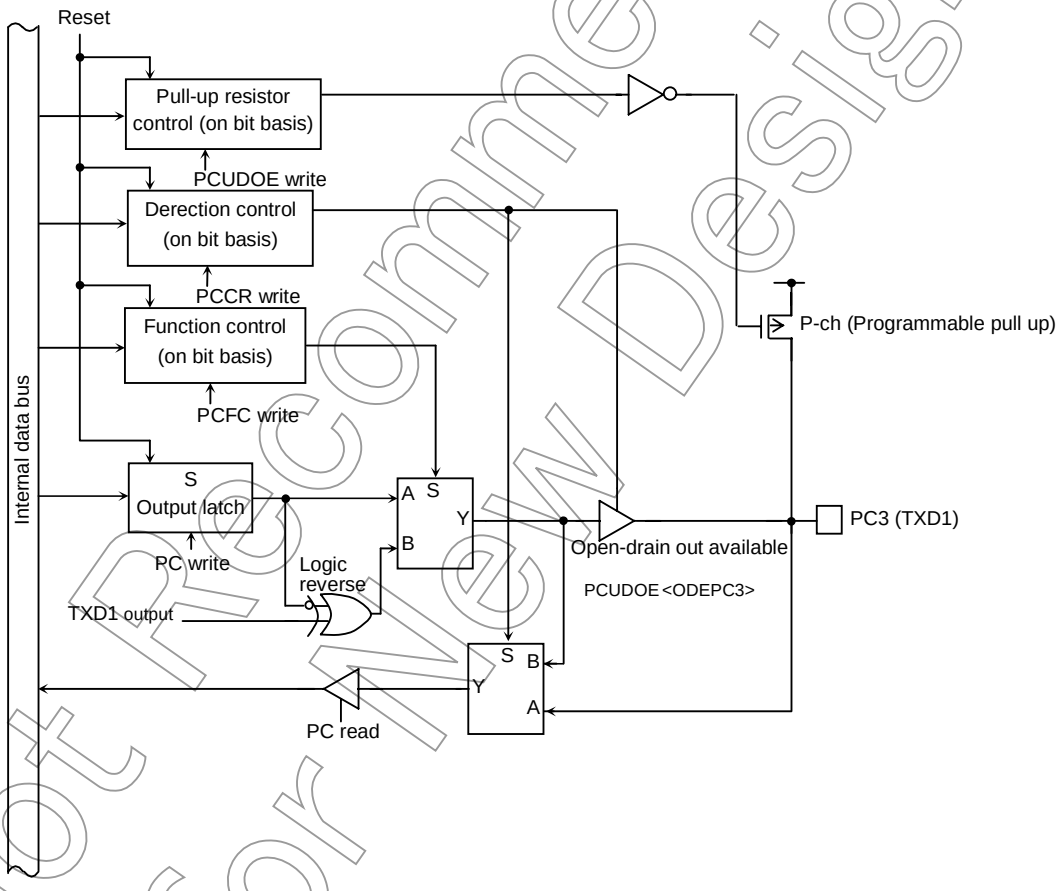


Figure 3.5.21 Port C3

(2) Port C4 (RXD1)

Port C4 have also function as serial channel input (RXD1). When it used to RXD1 function, it possible to out logical reverse by PC<PC4> register setting.

Port C4 can set pull-up or pull-down resistor by writing 1 data to PCUDOE<UDEPC4>. Selection of pull-up or pull-down, is set by PCUDOE<PC4UD>. This port become to input port without pull-up/down resistor by reset operation.

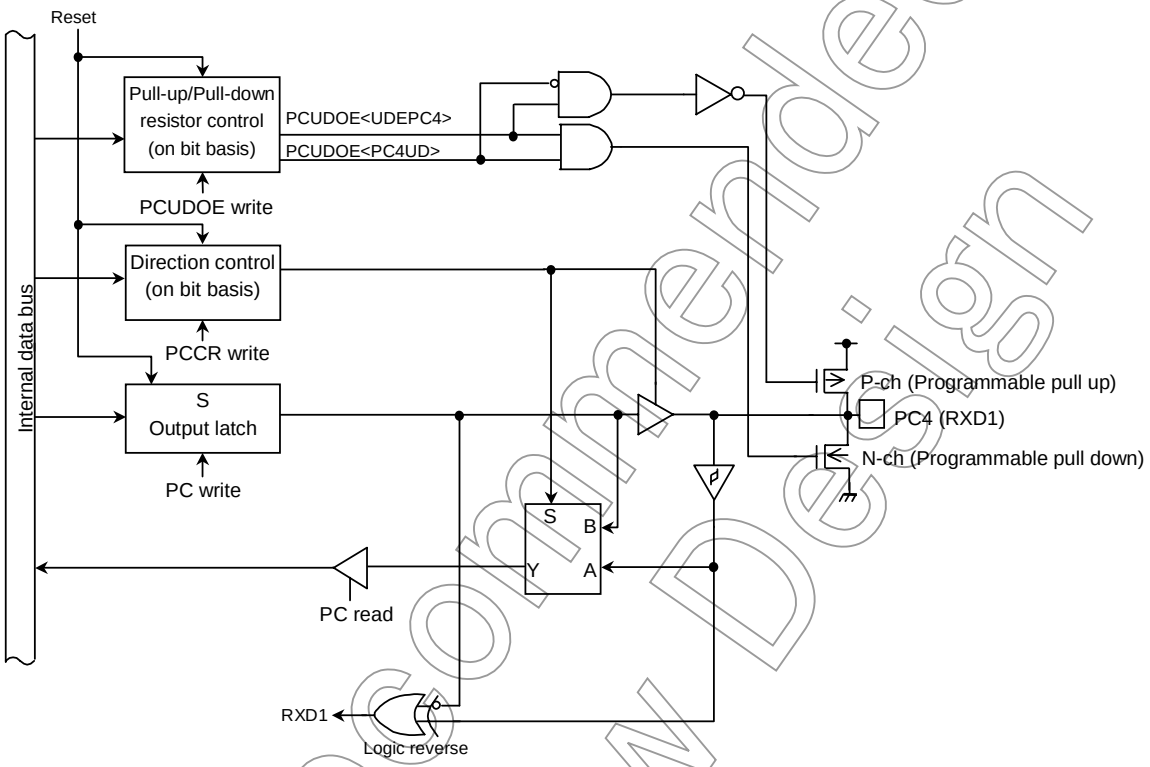


Figure 3.5.22 Port C4

(3) Port C5 ($\overline{CTS1}$, SCLK1)

Port C5 have also function as serial channel I/O ($\overline{CTS1}$) and clock I/O for SIO (SCLK1). When it used to serial channel port, it possible to set logical reverse I/O by PC<PC5>.

Port C5 can set pull-up or pull-down resistor by writing 1 data of PCUDOE<UDEPC5>.

Selection of pull-up or pull-down resistor is set by PCUDOE<PC5UD>. This port is to input port without pull-up/pull-down resistor by reset operation.

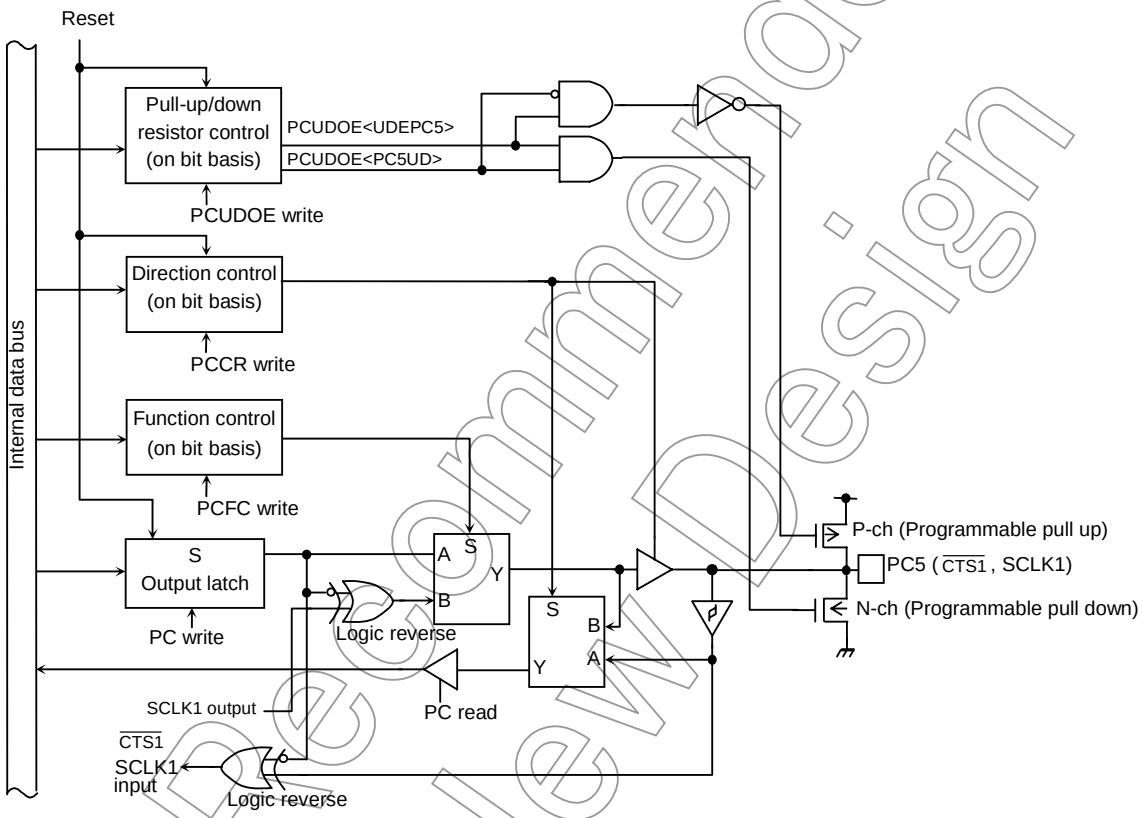


Figure 3.5.23 Port C5

(4) Port C6 (XT1), C7 (XT2)

Port C6, C7 have low-frequency oscillator function, except I/O port function.

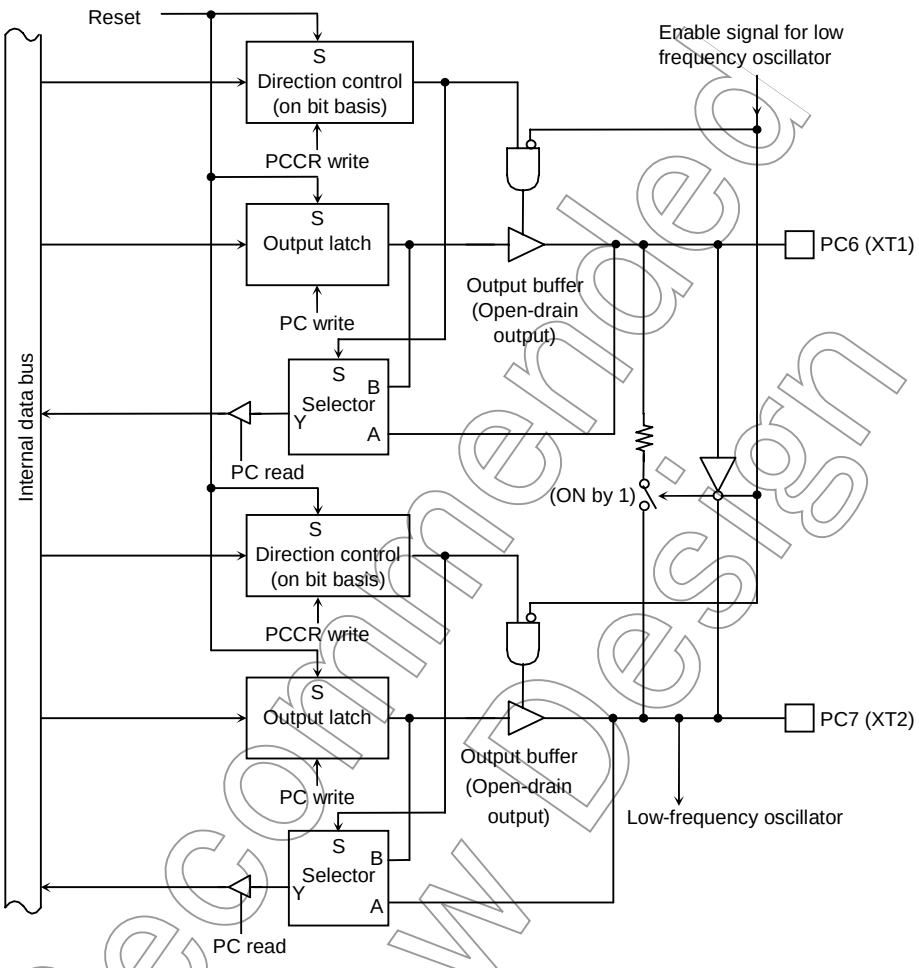


Figure 3.5.24 Port C6, C7

Port C Register

PC (0023H)

	7	6	5	4	3	2	1	0
Bit symbol	PC7	PC6	PC5	PC4	PC3			
Read/Write	R/W		R/W					
Function	Data from external port (Output latch register is set to 1)							

Port C Control Register

PCCR (0026H)

	7	6	5	4	3	2	1	0
Bit symbol	PC7C	PC6C	PC5C	PC4C	PC3C			
Read/Write	W	W	W					
After reset	1	1	0	0	0			
Function	0: Input 1: Output		0: Input 1: Output					

Port C Function Register

PCFC (0027H)

	7	6	5	4	3	2	1	0
Bit symbol			PC5F		PC3F			
Read/Write			W		W			
After reset			0		0			
Function			0: Port 1: SCLK1 output		0: Port 1: TXD1			

Port C Pull-up/Pull-down Resistor, Open-drain Register

PCUDOE (0028H)

	7	6	5	4	3	2	1	0
Bit symbol			ODEPC3	PC5UD	PC4UD	UDEPC5	UDEPC4	PC3U
Read/Write			W					
After reset			0					
Function			0: 3 states 1: Open drain	Pull-up/Pull-down control 0: Pull up 1: Pull down		Resistor control 0: Disable 1: Enable		Pull-up resistor 0: Disable 1: Enable

Note 1: Read-modify-write is prohibited for PCCR, PCFC and PCUDOE.

Note 2: Because PC4/RXD1 can't be controlled those terminal's function by register, SIO circuit also receive signals operating input port function.

Figure 3.5.25 Port C registers

3.5.9 Port D (PD0 to PD4, PD6, PD7)

Port D is 7-bit general-purpose I/O port. And port D0 to D4, D6, D7 can be set pull-up resistor by setting 1 data correspond bit of PDUE register. Port D0 to D4 become to input with pull-up resistor and port D6, D7 become to input without pull-up resistor by reset operation.

Resetting set to 1 data for output latch of this port.

Except I/O port function, this port have also function LCD controller output (DIBSCP, D2BLP, D3BFR, DLEBCD, DOFFB), RTC alarm output ($\overline{\text{ALARM}}$), MLD output (MLDALM, $\overline{\text{MLDALM}}$).

Above setting is set by writing data to PDFC register. Only port D6 have two functions ($\overline{\text{ALARM}}$, $\overline{\text{MLDALM}}$) and this setting by PD<PD6> register's data.

(1) PD0 (D1BSCP), PD1 (D2BLP), PD2 (D3BFR), PD3 (DLEBCD), PD7 (MLDALM)

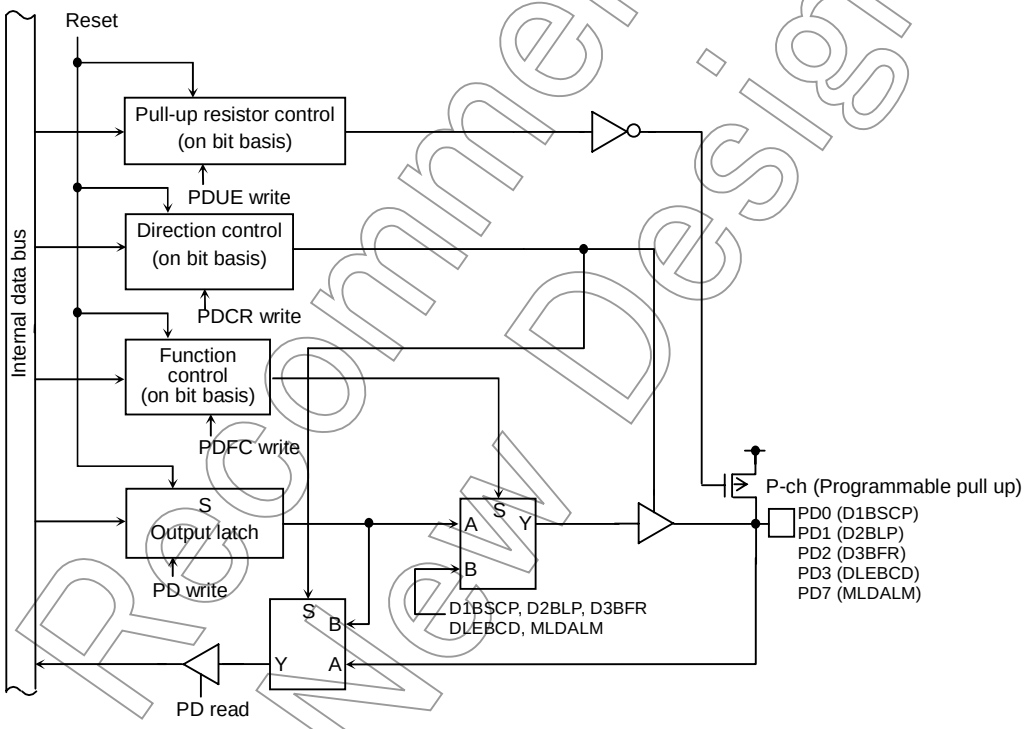


Figure 3.5.26 Port D0 to D3, D7

(2) PD4 (DOFFB)

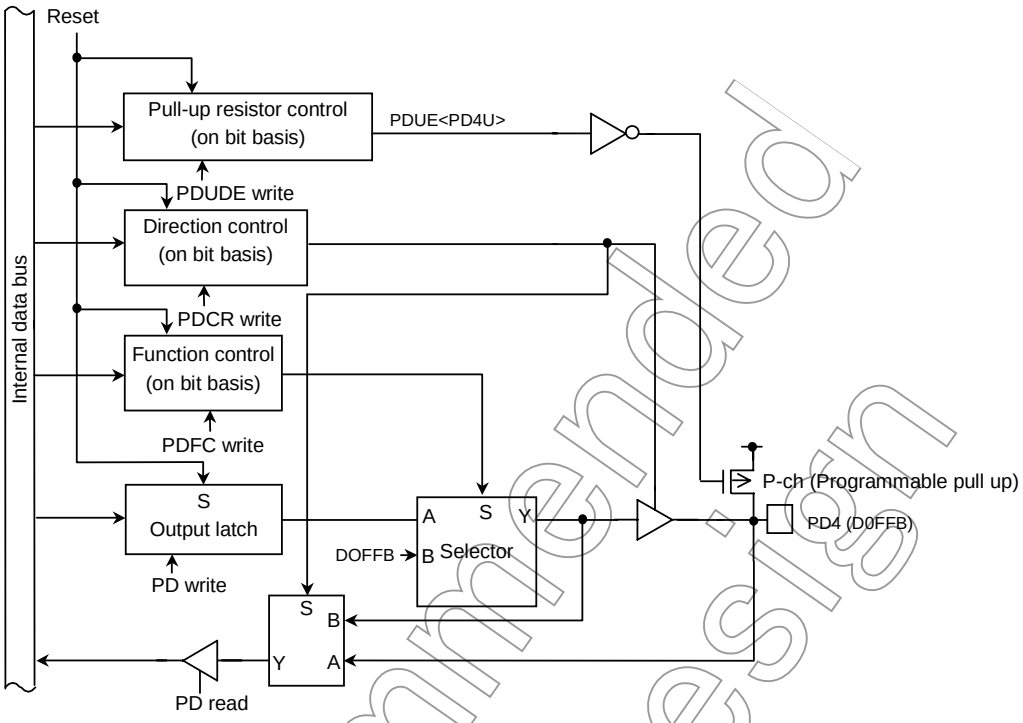


Figure 3.5.27 Port D4

(3) PD6 ($\overline{\text{MLDALM}}$, $\overline{\text{ALARM}}$)

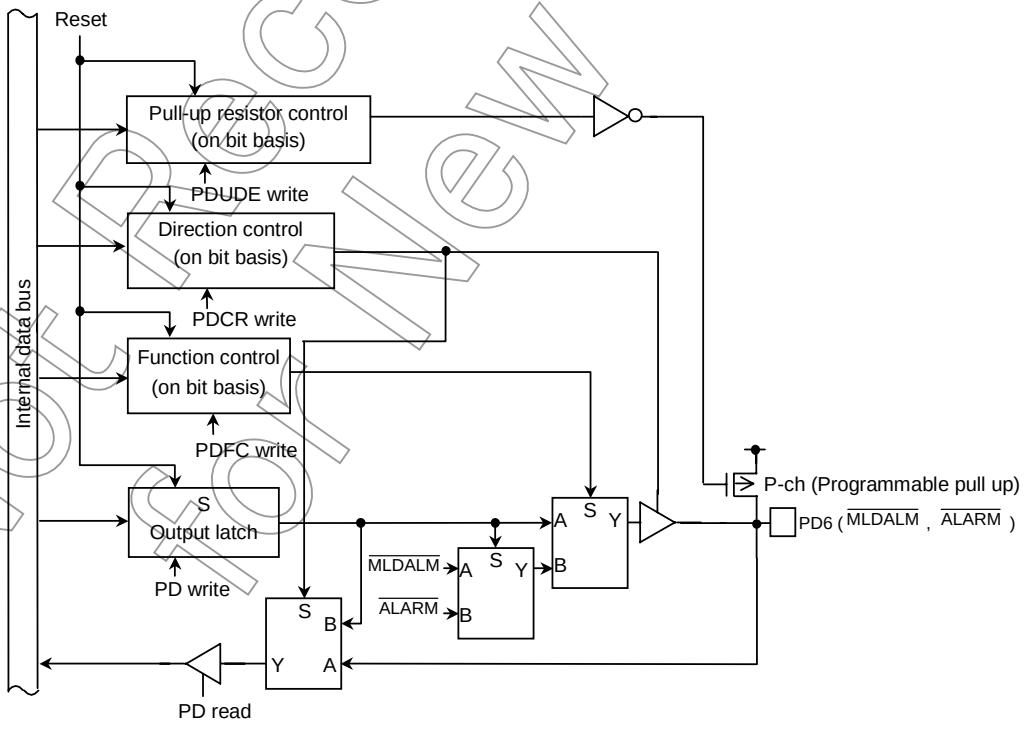


Figure 3.5.28 Port D6

Port D Register

PD (0029H)

	7	6	5	4	3	2	1	0
Bit symbol	PD7	PD6		PD4	PD3	PD2	PD1	PD0
Read/Write	R/W			R/W				
After reset	Data from external port (Output latch register is set to 1)							

Port D Control Register

PDCR (002BH)

	7	6	5	4	3	2	1	0
Bit symbol	PD7C	PD6C		PD4C	PD3C	PD2C	PD1C	PD0C
Read/Write	W			W				
After reset	0			0				
Function	0: Input 1: Output			0: Input 1: Output				

Port D Function Register

PDFC (002AH)

	7	6	5	4	3	2	1	0
Bit symbol	PD7F	PD6F		PD4F	PD3F	PD2F	PD1F	PD0F
Read/Write	W			W				
After reset	0			0				
Function	0: Port 1: MLDALM	0: Port 1: $\overline{\text{ALARM}}$ at <PD6> = 1 1: $\overline{\text{MLDALM}}$ at <PD6> = 0		0: Port 1: DOFFB	0: Port 1: DLEBCD	0: Port 1: D3BFR	0: Port 1: D2BLP	0: Port 1: D1BSCP

Port D Pull-up Resistor Control Register

PDUE (002CH)

	7	6	5	4	3	2	1	0
Bit symbol	PD7U	PD6U	–	PD4U	PD3U	PD2U	PD1U	PD0U
Read/Write	W		W	W				
After reset	0		0	1				
Function	Pull-up resistor 0: Disable 1: Enable		Always write "0"	Pull-up resistor 0: Disable 1: Enable				

Note: Read-modify-write is prohibited for PDCR, PDFC and PDUE.

Figure 3.5.29 Port D Register

3.6 Chip Select/Wait Controller

On the TMP91C016, four user-specifiable address areas (CS0 to CS3) can be set. The data bus width and the number of waits can be set independently for each address area (CS0 to CS3 and others).

The pins $\overline{CS0}$ to $\overline{CS3}$ (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 $\overline{CS0}$ to $\overline{CS3}$ 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.

$\overline{CS2A}$ to $\overline{CS2G}$ and $\overline{CSEXA}$ (CS pin except $\overline{CS0}$ to $\overline{CS3}$) are made by MMU.

These pins is $\overline{CS}$ 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 ($\overline{WAIT}$).

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 $\overline{CS0}$ to $\overline{CS3}$ 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 <S23:16>. 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)

		7	6	5	4	3	2	1	0
MSAR0 (00C8H) / MSAR1 (00CAH)	Bit symbol	S23	S22	S21	S20	S19	S18	S17	S16
	Read/Write	R/W							
MSAR2 (00CCH) / MSAR3 (00CEH)	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

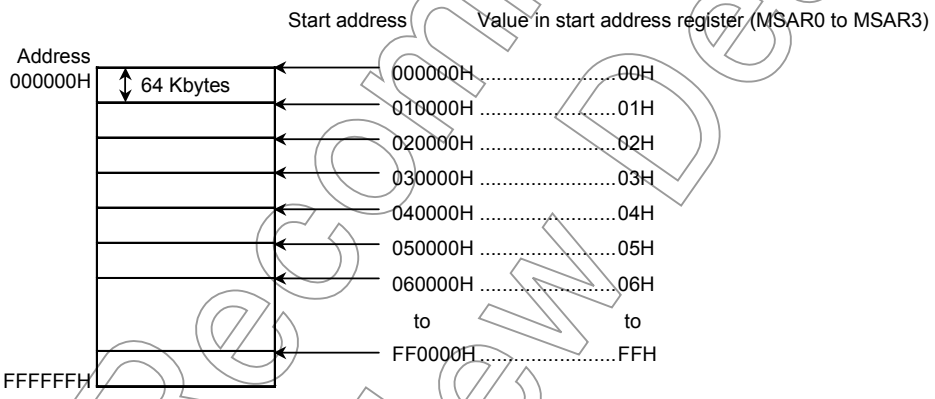


Figure 3.6.2 Relationship between Start Address and Start Address Register Value

(2) Memory address mask registers

Figure 3.6.3 shows the memory address mask registers. The 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)

		7	6	5	4	3	2	1	0
MAMR0 (00C9H)	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)

		7	6	5	4	3	2	1	0
MAMR1 (00CBH)	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 4 Mbytes.

Memory Address Mask Register (CS2, CS3)

		7	6	5	4	3	2	1	0
MAMR2 (00CDH) / MAMR3 (00CFH)	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 64-Kbyte address area starting from 010000H using the CS0 areas.

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

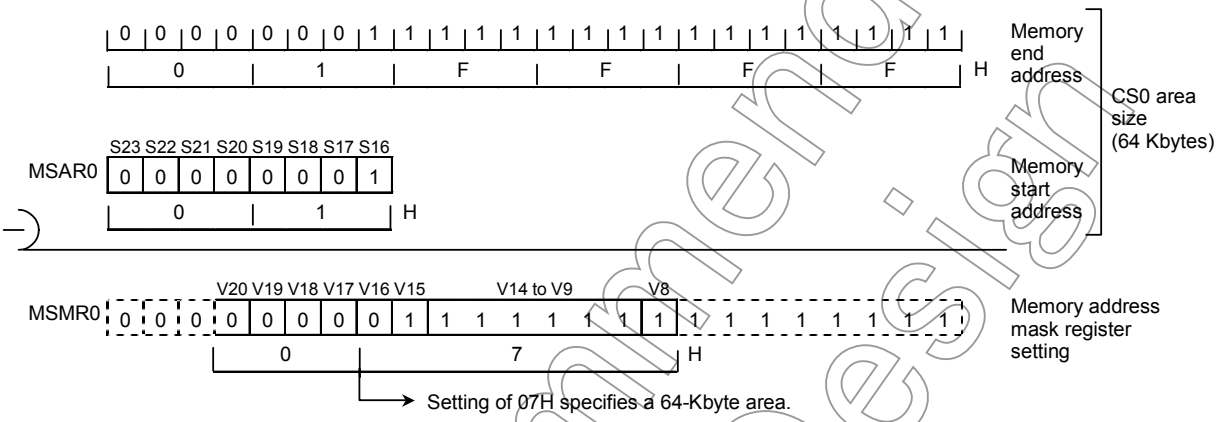


Figure 3.6.4 Example Showing how to Set the CS0 Area

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 to 000FFFH to 001000H to FFFFFFH in TMP91C016. 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”.)

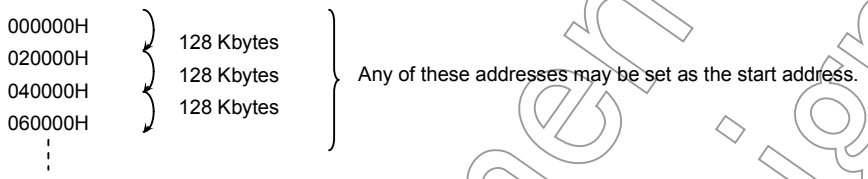
(4) Address area size specification

Table 3.6.1 shows the relationship between CS area and area size. The triangle (Δ) indicates in the table below that areas cannot be set by memory start address register and address mask register combinations. When setting an area size using a combination indicated by this symbol (Δ), set the start address mask register in the desired steps starting from 000000H.

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

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

a. Valid start addresses



b. Invalid start addresses

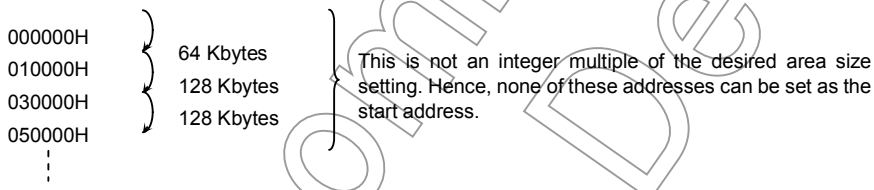


Table 3.6.1 Valid Area Sizes for Each CS Area

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	○	○	○	○	Δ	Δ	Δ	Δ	Δ		
CS1	○	○		○	Δ	Δ	Δ	Δ	Δ	Δ	
CS2			○	○	Δ	Δ	Δ	Δ	Δ	Δ	Δ
CS3			○	○	Δ	Δ	Δ	Δ	Δ	Δ	Δ

Δ: These areas 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 + N) waits 011: 0 waits	100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits	
B1CS (00C1H)	Bit Symbol	B1E		B1OM1	B1OM0	B1BUS	B1W2	B1W1	B1W0
	Read/Write	W		W					
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 + N) waits 011: 0 waits	100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits	
B2CS (00C2H)	Bit Symbol	B2E	B2M	B2OM1	B2OM0	B2BUS	B2W2	B2W1	B2W0
	Read/Write	W							
Read modify-write instructions are prohibited.	After reset	1	0	0	0	0	0	0	0
	Function	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 + N) waits 011: 0 waits	100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits	
B3CS (00C3H)	Bit Symbol	B3E		B3OM1	B3OM0	B3BUS	B3W2	B3W1	B3W0
	Read/Write	W		W					
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: Don't care 10: For DRAMC 11: Don't care		Data bus width 0: 16 bits 1: 8 bits	Number of waits 000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits	100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits	
BEXCS (00C7H)	Bit Symbol					BEXBUS	BEXW2	BEXW1	BEXW0
	Read/Write					0			
Read-modify-write instructions are prohibited.	After reset					0	0	0	0
	Function					Data bus width 0: 16 bits 1: 8 bits	Number of waits 000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits	100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits	

Master enable bit

0	Disable
1	Enable

CS2 area selection

0	16-Mbyte area
1	Specified address area

Chip select output waveform selection

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

For DRAM only CS3 setting

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

Bit7 (<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 is only CS2.

(2) Data bus width selection

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

This process of changing the data bus width according to the address being accessed is known as "Dynamic bus sizing". For details of this bus operation see Table 3.6.2.

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 to b0
		16 bits	2n + 0	xxxxx	b7 to b0
	2n + 1 (Odd number)	8 bits	2n + 1	xxxxx	b7 to b0
		16 bits	2n + 1	b7 to b0	xxxxx
16 bits	2n + 0 (Even number)	8 bits	2n + 0	xxxxx	b7 to b0
			2n + 1	xxxxx	b15 to b8
		16 bits	2n + 0	b15 to b8	b7 to b0
	2n + 1 (Odd number)	8 bits	2n + 1	xxxxx	b7 to b0
			2n + 2	xxxxx	b15 to b8
		16 bits	2n + 1	b7 to b0	xxxxx
			2n + 2	xxxxx	b15 to b8
32 bits	2n + 0 (Even number)	8 bits	2n + 0	xxxxx	b7 to b0
			2n + 1	xxxxx	b15 to b8
			2n + 2	xxxxx	b23 to b16
			2n + 3	xxxxx	b31 to b24
		16 bits	2n + 0	b15 to b8	b7 to b0
			2n + 2	b31 to b24	b23 to b16
	2n + 1 (Odd number)	8 bits	2n + 1	xxxxx	b7 to b0
			2n + 2	xxxxx	b15 to b8
			2n + 3	xxxxx	b23 to b16
			2n + 4	xxxxx	b31 to b24
		16 bits	2n + 1	b7 to b0	xxxxx
			2n + 2	b23 to b16	b15 to b8
			2n + 4	xxxxx	b31 to b24

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 to high-impedance; also, that the write strobe signal for the bus remains inactive.

(3) Wait control

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

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

Table 3.6.3 Wait Operation Settings

<BxW2:0>	No. of Waits	Wait Operation
000	2 waits	Inserts a wait of 2 states, irrespective of the $\overline{WAIT}$ pin state.
001	1 wait	Inserts a wait of 1 state, irrespective of the $\overline{WAIT}$ pin state.
010	(1 + N) waits	Samples the state of the $\overline{WAIT}$ pin after inserting a wait of one state. If the $\overline{WAIT}$ pin is low, the waits continue and the bus cycle is extended until the pin goes high.
011	0 waits	Ends the bus cycle without a wait, regardless of the $\overline{WAIT}$ pin state.
100	Reserved	Invalid setting
101	3 waits	Inserts a wait of 3 state, irrespective of the $\overline{WAIT}$ pin state.
110	4 waits	Inserts a wait of 4 state, irrespective of the $\overline{WAIT}$ pin state.
111	8 waits	Inserts a wait of 8 state, irrespective of the $\overline{WAIT}$ 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> (Bit6 of the chip select/wait control register for CS2) to 0 designates the 16-Mbyte area 000FE0H to 000FFFH, 003000H to 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 (e.g., if B2CS<B2M> = 1, CS2 is specified in the same manner as CS0, CS1 and CS3 are).

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

(6) Procedure for setting chip select/wait control

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

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 $\overline{CS0}$ to $\overline{CS3}$.

The CS0 to CS3 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 CS3 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 $\overline{CS0}$ to $\overline{CS3}$ pins.

Example:

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

MSAR0 = 01H ..... Start address: 010000H

MAMR0 = 07H ..... Address area: 64 Kbytes

B0CS = 83H..... ROM/SRAM, 16-bit data bus, 0 waits, CS0 area settings

3.6.3 Connecting External Memory

Figure 3.6.6 shows an example of how to connect external memory to the TMP91C016.

In this example the ROM is connected using a 16-bit bus. The RAM and I/O are connected using an 8-bit bus.

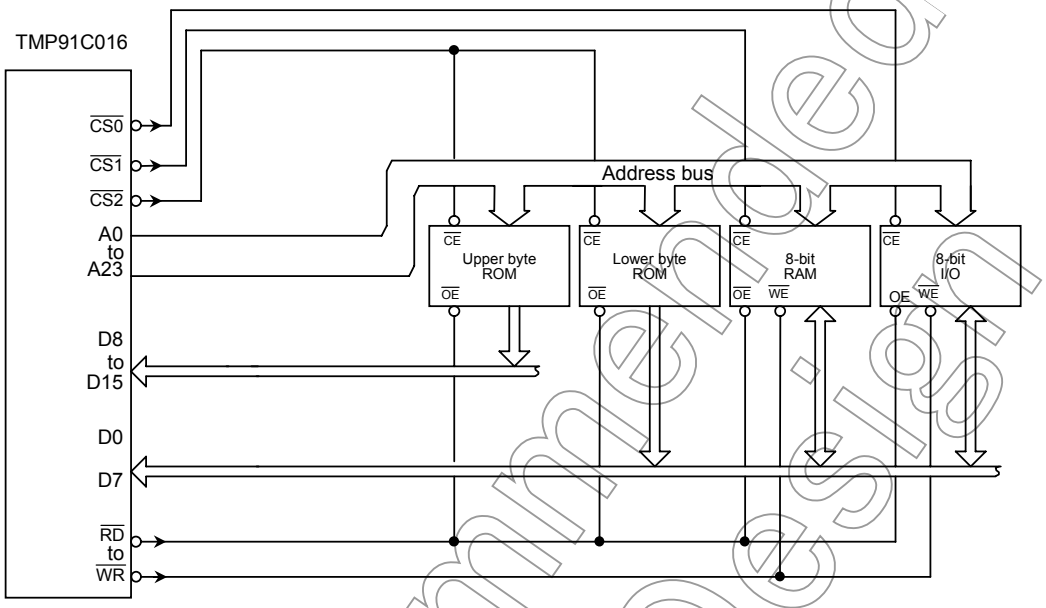


Figure 3.6.6 Example of External Memory Connection
(ROM uses 16-bit bus; RAM and I/O use 8-bit bus.)

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.

3.7 8-Bit Timers (TMRA)

The TMP91C016 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-flops are controlled by 5 bytes 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	None	None
	Output pin for timer flip-flop	TA1OUT (Shared with P70)	TA3OUT No external terminal (LCDC, MLD source clk use)
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

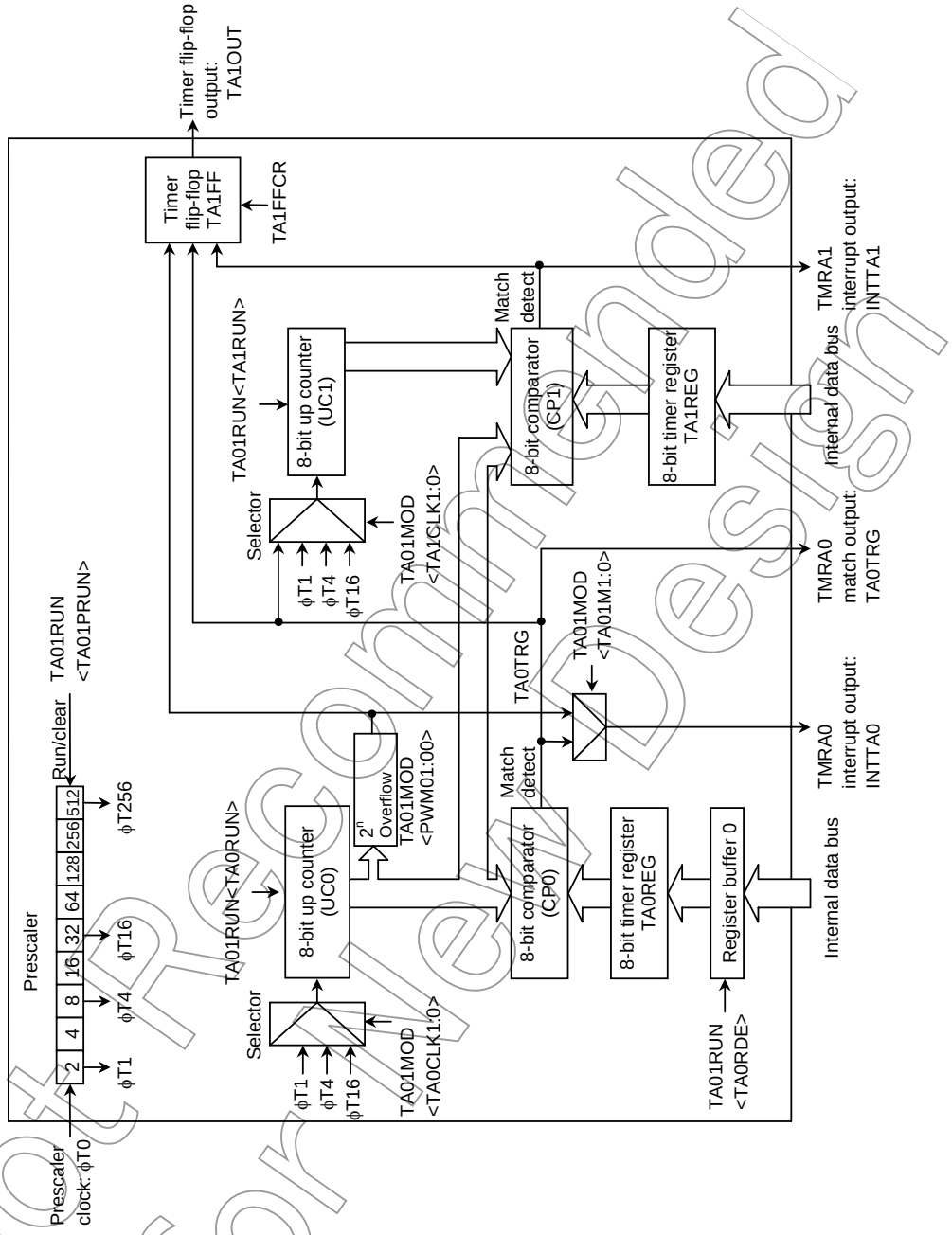


Figure 3.7.1 TMRA01 Block Diagram

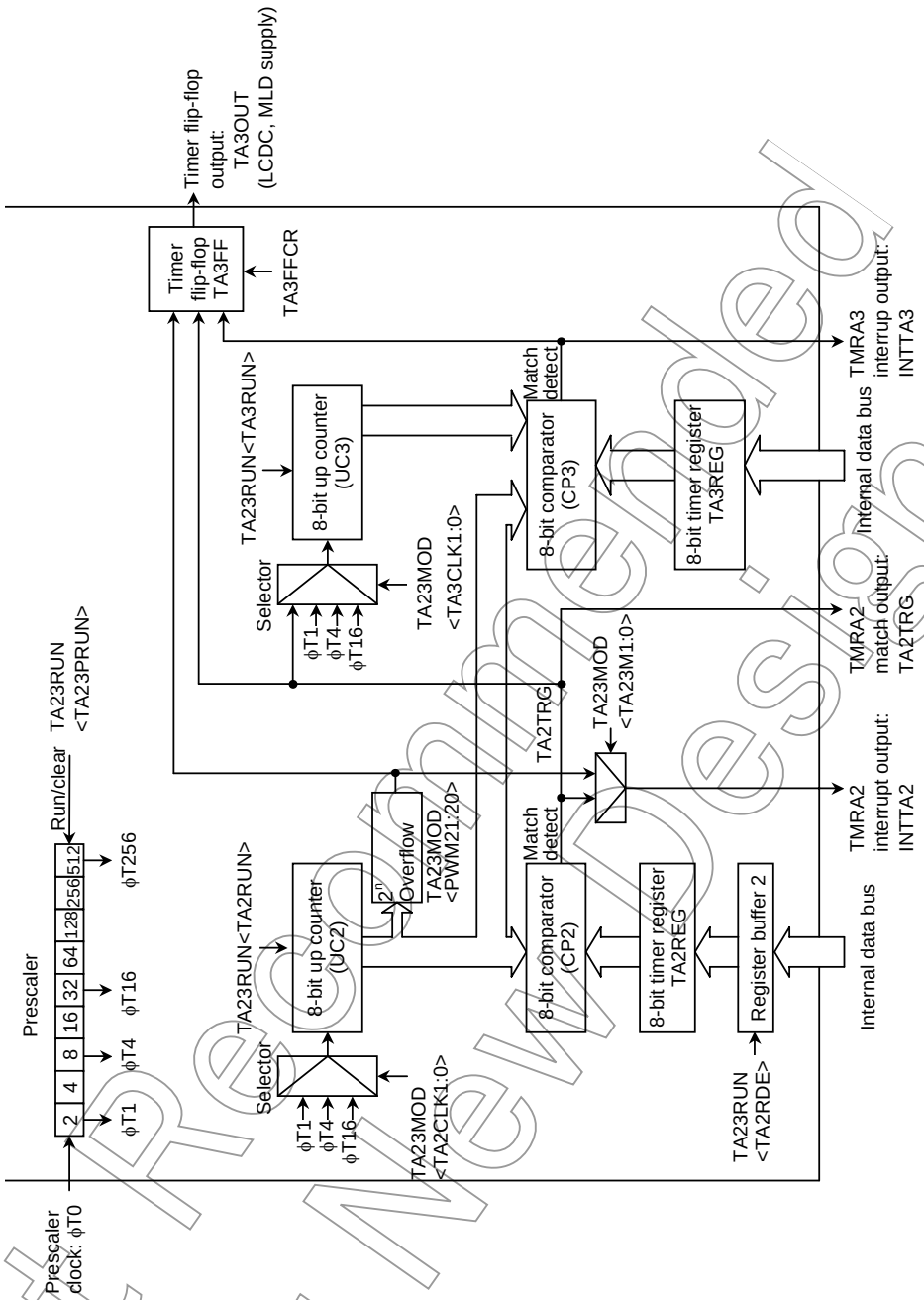


Figure 3.7.2 TMRA23 Block Diagram

3.7.2 Operation of Each Circuit

(1) Prescaler

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

The "ϕT0" as the input clock to pre-scaler is a clock divided by 4 which selected using the prescaler clock selection register SYSCR0<PRCK1:0>.

The prescaler's operation can be controlled using TA01RUN<TA01PRUN> in the timer control register. Setting <TA0PRUN> to 1 starts the count; setting <TA0PRUN> to 0 clears the prescaler to "0" and stops operation. Table 3.7.2 shows the various pre-scaler output clock resolutions.

Table 3.7.2 Prescaler Output Clock Resolution

at fc = 27 MHz, fs = 32.768 kHz

System Clock Selection SYSCR1 <SYSCK>	Prescaler Clock Selection SYSCR0 <PRCK1:0>	Gear Value SYSCR1 <GEAR2:0>	Prescaler Output Clock Resolution			
			ϕT1	ϕT4	ϕT16	ϕT256
1 (fs)	00 (f_{FPH})	XXX	2^3/fs (244 μs)	2^5/fs (977 μs)	2^7/fs (3.9 μs)	2^{11}/fs (62.5 μs)
0 (fc)		000 (fc)	2^3/fc (0.3 μs)	2^5/fc (1.2 μs)	2^7/fc (4.7 μs)	2^{11}/fc (75.85 μs)
		001 (fc/2)	2^4/fc (0.6 μs)	2^6/fc (2.4 μs)	2^8/fc (9.5 μs)	2^{12}/fc (151.7 μs)
		010 (fc/4)	2^5/fc (1.2 μs)	2^7/fc (4.7 μs)	2^9/fc (19.0 μs)	2^{13}/fc (303.4 μs)
		011 (fc/8)	2^6/fc (2.4 μs)	2^8/fc (9.5 μs)	2^{10}/fc (37.9 μs)	2^{14}/fc (606.8μs)
		100 (fc/16)	2^7/fc (4.7 μs)	2^9/fc (19.0 μs)	2^{11}/fc (75.9 μs)	2^{15}/fc (1214 μs)
	10 (fc/16 clock)	XXX	2^7/fc (4.7 μs)	2^9/fc (19.0 μs)	2^{11}/fc (75.9 μs)	2^{15}/fc (1214 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<TA01CLK1:0>.

The input clock for UC1 depends on the operation mode. In 16-bit timer mode, the overflow output from UC0 is used as the input clock. In any mode other than 16-bit timer mode, the input clock is selectable and can either be one of the internal clocks 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 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.

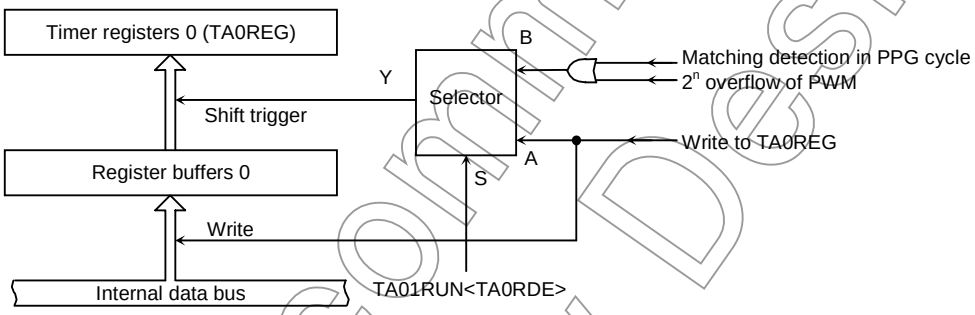


Figure 3.7.3 Configuration of TA0REG

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 "0" and an interrupt signal (INTTA0 or INTTA1) is generated. If timer flip-flop inversion is enabled, the timer flip-flop is inverted at the same time.

(5) Timer flip-flop (TA1FF)

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

Whether inversion is enabled or disabled is determined by the setting of the bit TA1FFCR<TA1FFIE> in the timer flip-flop control register.

A reset clears the value of TA1FF1 to "0".

Writing 01 or 10 to TA1FFCR<TA1FFC1:0> sets TA1FF to 0 or 1. Writing 00 to these bits inverts the value of TA1FF (This is known as software inversion).

The TA1FF signal is output via the TA1OUT pin (Concurrent with P70). 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.

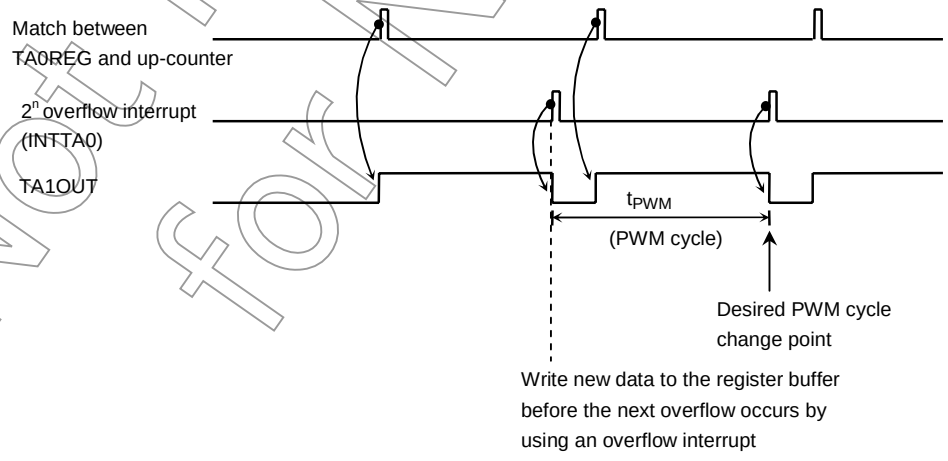
Note: When the double buffer is enabled for an 8-bit timer in PWM or PPG mode, caution is required as explained below.

If new data is written to the register buffer immediately before an overflow occurs by a match between the timer register value and the up-counter value, the timer flip-flop may output an unexpected value.

For this reason, make sure that in PWM mode new data is written to the register buffer by six cycles ($f_{SYS} \times 6$) before the next overflow occurs by using an overflow interrupt.

In the case of using PPG mode, make sure that new data is written to the register buffer by six cycles before the next cycle compare match occurs by using a cycle compare match interrupt.

Example when using PWM mode



3.7.3 SFRs

TMRA01 Run Register

TA01RUN (0100H)

	7	6	5	4	3	2	1	0
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	8-bit timer run/stop control 0: Stop and clear 1: Run (Count up)		

TA0REG double buffer control

0	Disable
1	Enable

Timer run/stop control

0	Stop and 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

TA23RUN (0108H)

	7	6	5	4	3	2	1	0
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	8-bit timer run/stop control 0: Stop and clear 1: Run (Count up)		

TA2REG double buffer control

0	Disable
1	Enable

Timer run/stop control

0	Stop and 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 10: 2^7 11: 2^8		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<TA01M1:0> ≠ 01	TA01MOD<TA01M1:0> = 01
00	Comparator output from TMRA0	Overflow output from TMRA0
01	φT1	
10	φT16	
11	φT256	(16-bit timer mode)

PWM cycle selection

00	Reserved
01	2^6 × clock source
10	2^7 × clock source
11	2^8 × 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

TA23MOD (010CH)

	7	6	5	4	3	2	1	0
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 10: 2^7 11: 2^8		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<TA23M1:0> ≠ 01	TA23MOD<TA23M1:0> = 01
00	Comparator output from TMRA2	Overflow output from TMRA2
01	φT1	
10	φT16	
11	φT256	(16-bit timer mode)

PWM cycle selection

00	Reserved
01	2^6 × clock source
10	2^7 × clock source
11	2^8 × 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

		7	6	5	4	3	2	1	0
TA1FFCR (0105H)	Bit symbol					TA1FFC1	TA1FFC0	TA1FFIE	TA1FFIS
	Read/Write					R/W		R/W	
	After reset					1	1	0	0
Read-modify-write instructions are prohibited.	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					TA3FFC1	TA3FFC0	TA3FFIE	TA3FFIS
	Read/Write					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

Timer register

		7	6	5	4	3	2	1	0
TA0REG (0102H)	bit Symbol	–							
	Read/Write	W							
	After reset	Undefined							
TA1REG (0103H)	bit Symbol	–							
	Read/Write	W							
	After reset	Undefined							
TA2REG (010AH)	bit Symbol	–							
	Read/Write	W							
	After reset	Undefined							
TA3REG (010BH)	bit Symbol	–							
	Read/Write	W							
	After reset	Undefined							

Note: The avobe registers are prohibited read-modify-write instruction.

Figure 3.7.9 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.

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

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

Example: To generate an INTTA1 interrupt every 12 μs at fc = 27 MHz, set each register as follows:

* Clock state Clock gear: 1/1

	MSB 7	6	5	4	3	2	1	LSB 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 (0.3 μs at fc = 27 MHz) as the input clock.
TA1REG	← 0	0	1	0	1	0	0	0	Set TA1REG to 12 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.

X: Don't care, –: No change

Select the input clock using 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

b. Generating a 50% duty ratio square wave pulse

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

Example: To output a 1.8-μs square wave pulse from the TA1OUT pin at fc = 27 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　　Clock gear: 1/1

		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 (0.3 μs at fc = 27 MHz) as the input clock.
TA1REG	←	0	0	0	0	0	0	1	1	Set the timer register to 1.8 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.
P7CR	←	X	X	X	X	–	–	–	1	Set P70 to function as the TA1OUT pin.
P7FC	←	–	–	–	–	–	–	–	1	
TA01RUN	←	–	X	X	X	–	1	1	–	Start TMRA1 counting.

X: Don't care, –: No change

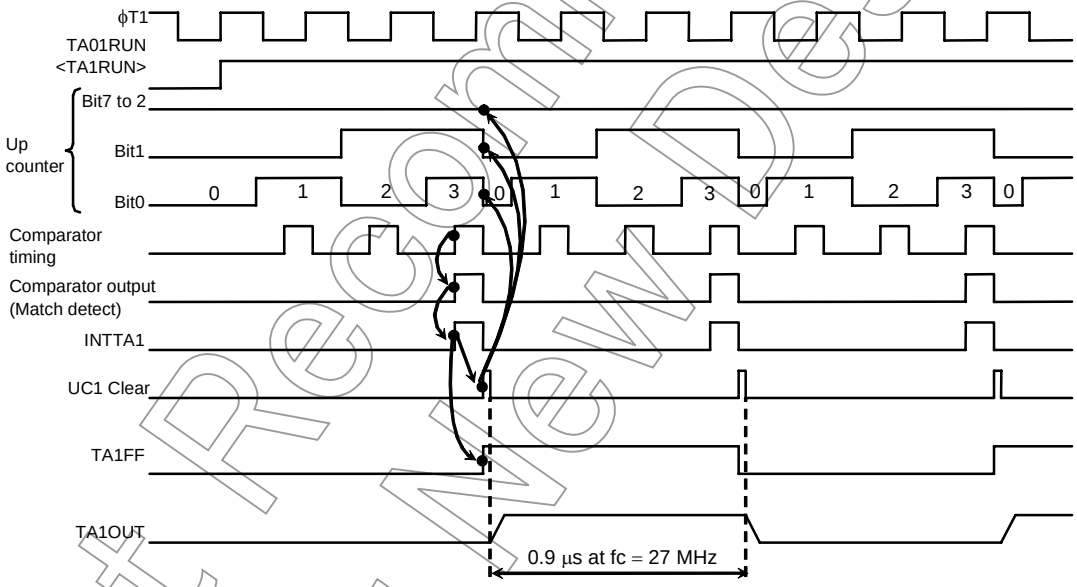


Figure 3.7.10 Square Wave Output Timing Chart (50% duty)

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

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

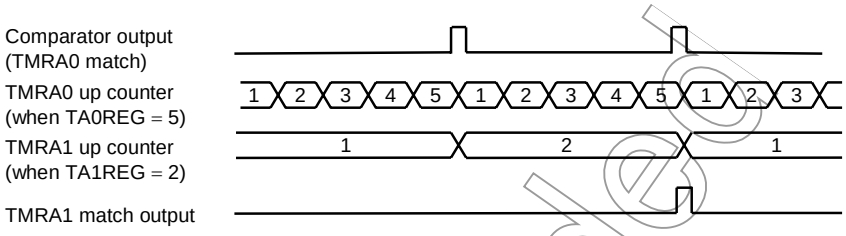


Figure 3.7.11 TMRA1 Count Up on Signal from TMRA0

(2) 16-bit timer mode

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

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

In 16-bit timer mode, the overflow output from TMRA0 is used as the input clock for TMRA1, regardless of the value set in TA01MOD<TA01CLK1:0>. 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.

Example: To generate an INTTA1 interrupt every 0.3 [s] at fc = 27 MHz, set the timer registers TA0REG and TA1REG as follows:

* Clock state Clock gear: 1/1

If ϕT16 (=(2^7/fc) s @ 27 MHz) is used as the input clock for counting, set the following value in the registers:

0.3 s ÷ (2^7/fc) s = 62500 = F424H

(e.g., set TA1REG to F4H and TA0REG to 24H).

As a result, INTTA1 interrupt can be generated every 0.29[s].

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

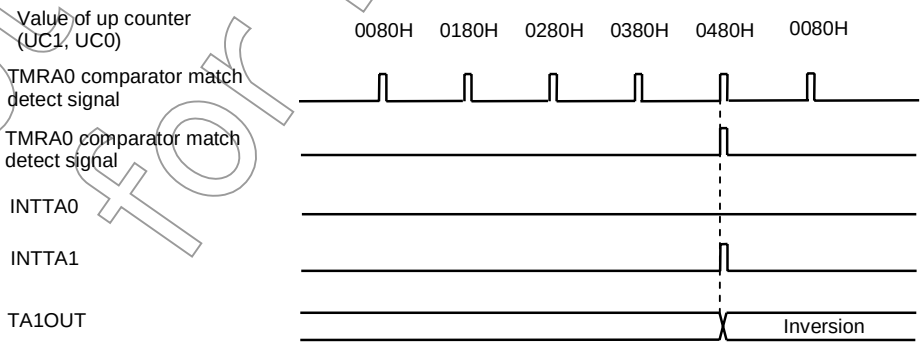


Figure 3.7.12 Timer Output by 16-Bit Timer Mode

(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.

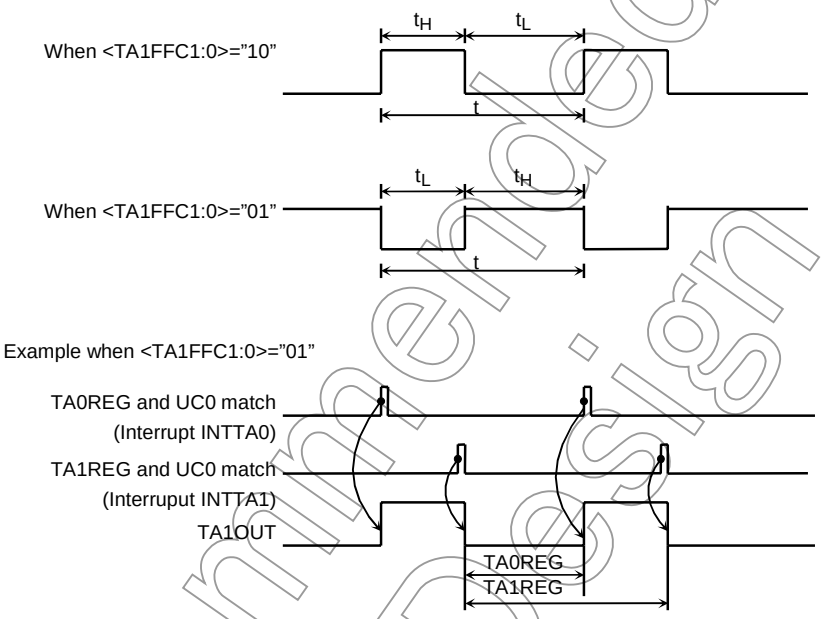


Figure 3.7.13 8-Bit PPG Output Waveforms

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 3.7.14 shows a block diagram representing this mode.

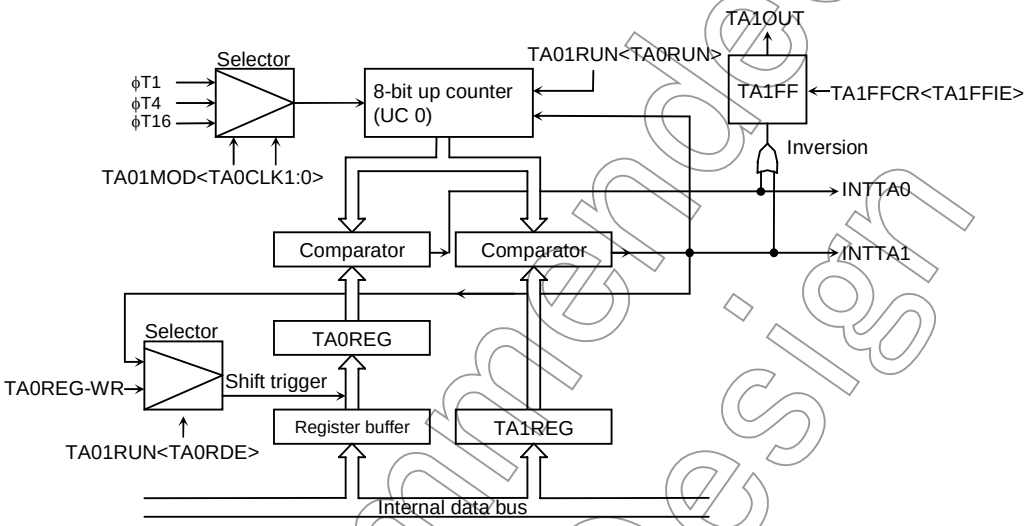


Figure 3.7.14 Block Diagram of 8-Bit PPG Output Mode

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).

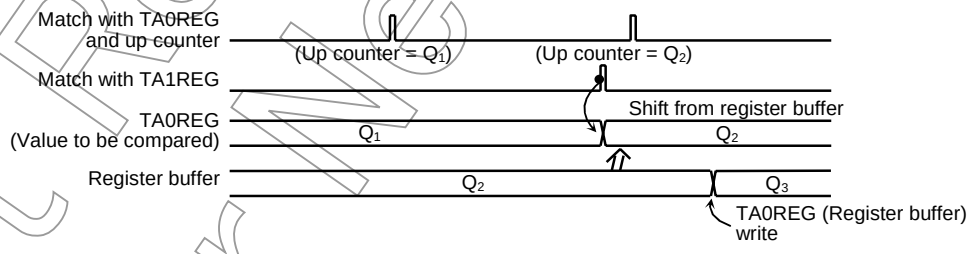
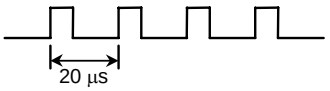


Figure 3.7.15 Operation of Register Buffer

Example: To generate 1/4-duty 50-kHz pulses (at fc = 27 MHz):



* Clock state Clock gear: 1/1

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 μs
$\phi T1 = (2^3/fc)$ s (at 27 MHz);

$20\ \mu s/(2^3/fc)\ s \approx 67$

Therefore set TA1REG to 67 (43H)
The duty is to be set to 1/4: t × 1/4 = 20 μs × 1/4 = 5 μs

$5\ \mu s/(2^3/fc)\ s \approx 17$

Therefore, set TA0REG = 17 = 11H.

		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	1	0	0	0	1	Write 11H
TA1REG	←	0	1	0	0	0	0	1	1	Write 43H
TA1FFCR	←	X	X	X	X	0	1	1	X	Set TA1FF, enabling both inversion and the double buffer.
										Writing 10 provides negative logic pulse.
P7CR	←	X	X	X	X	–	–	–	1	Set P70 as the TA1OUT pin.
P7FC	←	–	–	–	–	–	–	–	1	
TA01RUN	←	1	X	X	X	–	1	1	1	Start TMRA0 and TMRA01 counting.

X: Don't care, –: No change

(4) 8-bit PWM (Pulse width modulation) 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. 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 counter overflow occurs (n = 6, 7 or 8 as specified by TA01MOD<PWM01:00>). The up counter UC0 is cleared when 2^n 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 counter overflow
Value set in TA0REG ≠ 0

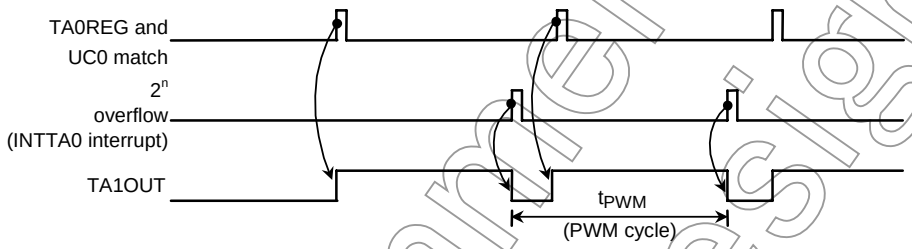


Figure 3.7.16 8-Bit PWM Waveforms

Figure 3.7.17 shows a block diagram representing this mode.

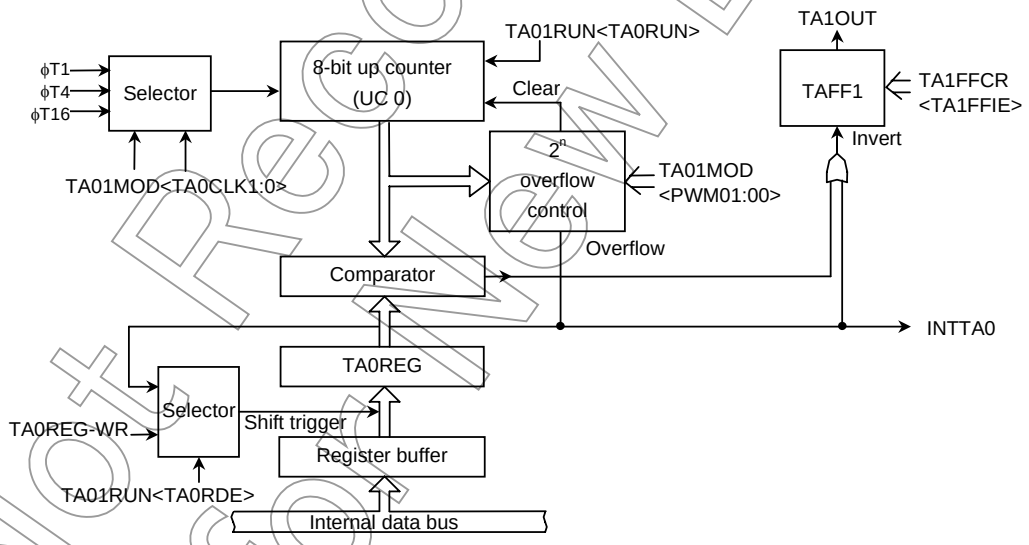


Figure 3.7.17 Block Diagram of 8-Bit PWM Mode

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

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

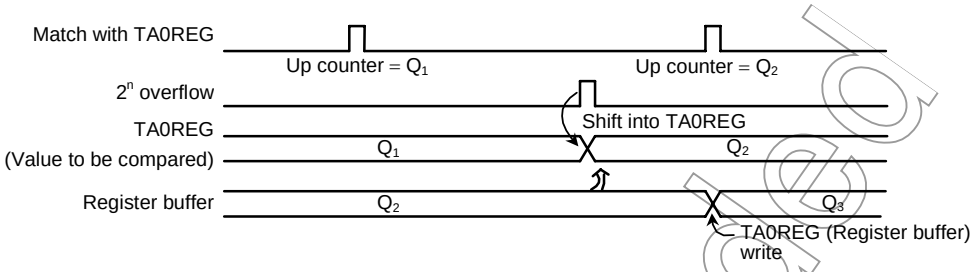
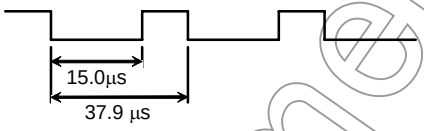


Figure 3.7.18 Register Buffer Operation

Example: To output the following PWM waves on the TA1OUT pin at fc = 27 MHz:



* Clock state [Clock gear: 1/1

To achieve a 37.9-μs PWM cycle by setting ϕT1 to $(2^3/fc)$ s (at fc = 27 MHz):

$37.9\ \mu s / (2^3/fc)\ s = 128 = 2^n$

Therefore n should be set to 7.

Since the low-level period is 15 μs when ϕT1 = $(2^3/fc)$s,

set the following value for TA0REG:

$15\ \mu s / (2^3/fc)\ s \approx 51 = 33H$

		MSB 7	6	5	4	3	2	1	LSB 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) and select ϕT1 as the input clock.
TA0REG	←	0	0	1	1	0	0	1	1	Write 33H.
TA1FFCR	←	X	X	X	X	1	0	1	X	Clear TA1FF to "0", enable the inversion and double buffer.
P7CR	←	X	X	X	X	–	–	–	1	Set P70 and the TA1OUT pin.
P7FC	←	–	–	–	–	–	–	–	1	
TA01RUN	←	1	X	X	X	–	1	–	1	Start TMRA0 counting.

X: Don't care, –: No change

Table 3.7.3 PWM Cycle

at fc = 27MHz, fs = 32.768 kHz

Select System Clock <SYSCK>	Select Prescaler Clock <PRCK1:0>	Gear Value <GEAR2:0>	PWM Cycle								
			2^6			2^7			2^8		
			φT1	φT4	φT16	φT1	φT4	φT16	φT1	φT4	φT16
1 (fs)	00 (f_{FPH})	XXX	15.6 ms	62.5 ms	250 ms	31.3 ms	125 ms	500 ms	62.5 ms	250 ms	1000 ms
0 (fc)		000 (fc)	19.0 μs	76 μs	303 μs	37.9 μs	152 μs	607 μs	76 μs	303 μs	1214 μs
		001 (fc/2)	37.9 μs	152 μs	607 μs	76 μs	303 μs	1214 μs	152 μs	607 μs	2427 μs
		010 (fc/4)	75.9 μs	303 μs	1214 μs	152 μs	607 μs	2427 μs	303 μs	1214 μs	4855 μs
		011 (fc/8)	151.7 μs	607 μs	2427 μs	303 μs	1214 μs	4855 μs	607 μs	2427 μs	9709 μs
		100 (fc/16)	303.4 μs	1214 μs	4855 μs	607 μs	2427 μs	9709 μs	1214 μs	4855 μs	19418 μs
	10 (fc/16 clock)	XXX	303.4 μs	1214 μs	4855 μs	607 μs	2427 μs	9709 μs	1214 μs	4855 μs	19418 μs

XXX: Don't care

(5) Settings for each mode

Table 3.7.4shows he SFR settings for each mode.

Table 3.7.4 Timer Mode Setting Registers

Register Name	TA01MOD				TA1FFCR
<Bit Symbol>	<TA01M1:0>	<PWM01:00>	<TA1CLK1:0>	<TA0CLK1:0>	TA1FFIS
Function	Timer Mode	PWM Cycle	Upper Timer Input Clock	Lower Timer Input Clock	Timer F/F Invert Signal Select
8-bit timer × 2 channels	00	–	Lower timer match φT1, φT16, φT256 (00, 01, 10, 11)	External clock φT1, φT4, φT16 (00, 01, 10, 11)	0: Lower timer output 1: Upper timer output
16-bit timer mode	01	–	–	External clock φT1, φT4, φT16 (00, 01, 10, 11)	–
8-bit PPG × 1 channel	10	–	–	External clock φT1, φT4, φT16 (00, 01, 10, 11)	–
8-bit PWM × 1 channel	11	2^6, 2^7, 2^8 (01, 10, 11)	–	External clock φT1, φT4, φT16 (00, 01, 10, 11)	–
8-bit timer × 1 channel	11	–	φT1, φT16, φT256 (01, 10, 11)	–	Output disabled

–: Don't care

(6) MELODY/ALARM circuit supply mode

This function can operate only TMRA3. It can use MELODY/ALARM souce clock TA3 clock generated by TMRA3. But this function is special mode, without low clock (XTIN, XTOUT), so keep the rule under below.

OPERATE

1. Clock generate by timer 3
2. Connect to LCDCLK (EMCCR4 <TA3MLDE> = 1)
3. Need setup time
4. MELODY/ALARM start to operate

STOP

1. MELODY/ALARM stop to operate
2. Clock supply cut off (<TA3MLD> = 0)

Timer Clk Supply Mode Register

		7	6	5	4	3	2	1	0
EMCCR4 (00E7H)	Bit symbol							TA3MLDE	TA3LCDE
	Read/Write							R/W	R/W
	After reset							0	0
	Function							MLD source clk 0: 32 kHz 1: TA3	LCDC source clk 0: 32 kHz 1: TA3

3.8 External Memory Extension Function (MMU)

This is MMU function which can expand program/data area to 105 Mbytes by having 4 local area.

Address pins to external memory are 2 extended address bus pins (EA24, EA25) and 5 extended chip select pins ($\overline{CS2A}$ to $\overline{CS2E}$) in addition to 24 address bus pins (A0 to A23) which are common specification of TLCS-900 family and 4 chip select pins ($\overline{CS0}$ to $\overline{CS3}$) output from CS/WAIT controller. And hook function protect program sedulity.

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 to 16 displayed as hex.

Purpose	Item	(A): For Standard Extended Memory	(B): For Many Kinds Class Extended Memory
Program ROM	Maximum memory size	2 Mbytes: common 2 + 14 Mbytes: bank (16 Mbytes × 1 pcs)	
	Used local area, bank number	Local 2 (AH = C0 – DF: 2 Mbytes × 7 banks)	
	Setting CS/WAIT	Set up AH = C0 – FF to CS2	Set up AH = 80 – FF to CS2
	Used $\overline{CS}$ pin	$\overline{CS2}$	$\overline{CS2A}$
Data ROM	Maximum memory size	64 Mbytes (64 Mbytes × 1 pcs)	64 Mbytes (16 Mbytes × 6 pcs)
	Used local area, bank number	Local 3 (AH = 80 – BF: 4 Mbytes × 16 banks)	Local 3 (AH = 80 – BF: 4 Mbytes × 16 banks)
	Setting CS/WAIT	Set up AH = 80 – BF to CS3	Set up AH = 80 – FF to CS2
	Used $\overline{CS}$ pins	$\overline{CS3}$, EA24, EA25	$\overline{CS2B}$, $\overline{CS2C}$ $\overline{CS2D}$, $\overline{CS2E}$
Option program ROM	Maximum memory size	2 Mbytes: common 1 + 14 Mbytes: bank (16 Mbytes × 1 pcs)	
	Used local area, bank number	Local 1 (AH = 40 – 5F: 2 Mbytes × 7 banks)	
	Setting CS/WAIT	Set up AH = 40 – 7F to CS1	
	Used $\overline{CS}$ pin	$\overline{CS1}$	
Data RAM (Available DRAM)	Maximum memory size	1 Mbyte: common + 7 Mbytes: bank Mbyte (8 Mbytes × 1pcs)	
	Used local area, bank number	Local 0 (AH = 10 – 1F: 1 Mbyte × 7 banks)	
	Setting CS/WAIT	Set up AH = 00 – 1F to CS0	Set up AH = 00 – 1F to CS3
	Used $\overline{CS}$ pin	$\overline{CS0}$ (Not available DRAM)	$\overline{CS3}$ (Available DRAM)
Extended memory 1	Maximum memory size		1 Mbyte (1 Mbyte × 1 pcs)
	Used local area, bank number		None
	Setting CS/WAIT		Set up AH = 20 – 2F to CS0
	Used $\overline{CS}$ pin		$\overline{CS0}$
Extended memory 2 (Direct address assigned built-in type LCD driver	Maximum memory size	256 Kbytes (64 Kbytes × 4 pcs)	
	Used local area, bank number	None	
	Setting CS/WAIT	–	
	Used $\overline{CS}$ pin	D1BSCP, D2BLP, D3BFR, DLEBCD	
Extended memory 3	Maximum memory size	1 Mbyte + 768 Kbytes	768 Kbytes
	Used local area, bank number	None	
	Setting CS/WAIT	–	
	Used $\overline{CS}$ pin	None	

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. And, a physical-address map is shown in Figure 3.8.2.

However, when memory area is less than 16 Mbytes 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.

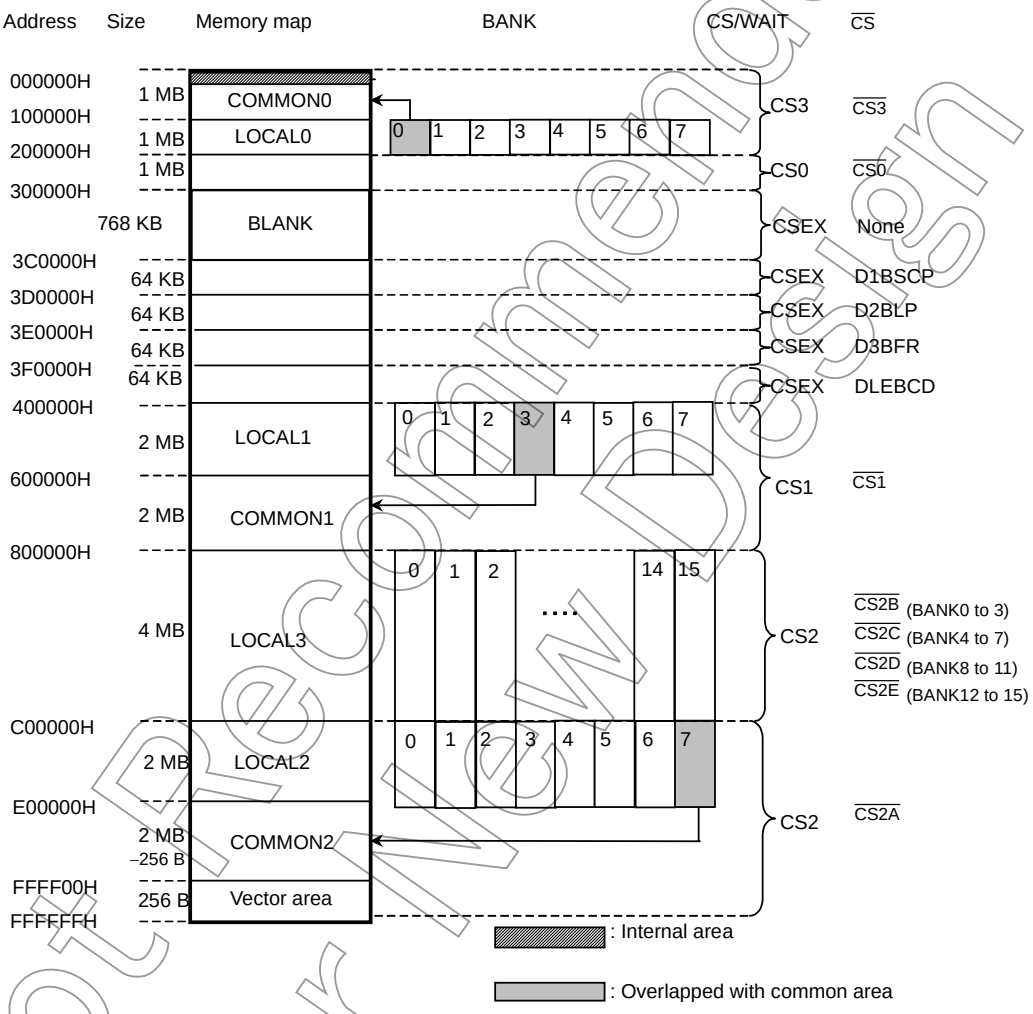


Figure 3.8.1 Logical Address Map

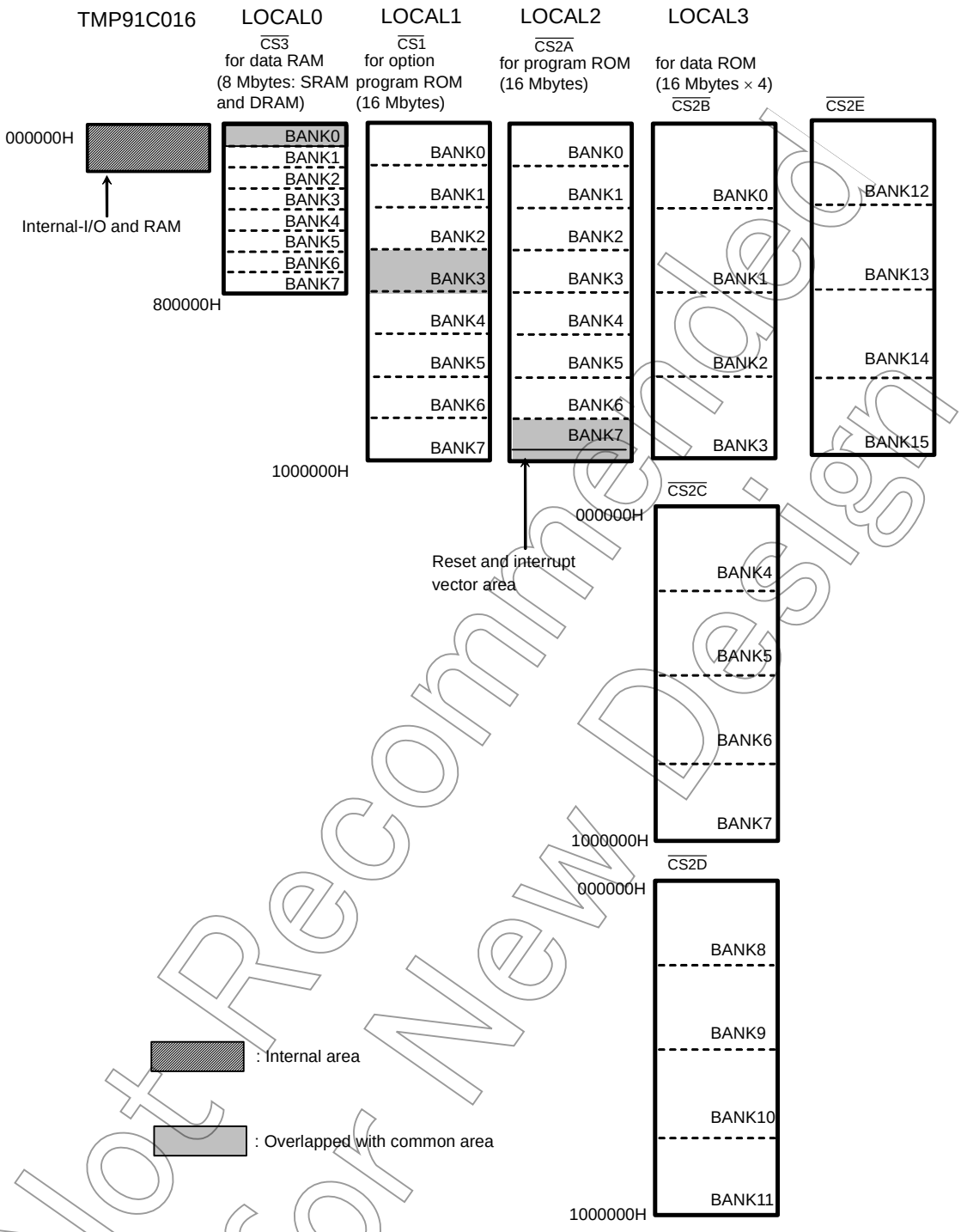


Figure 3.8.2 Physical Address Map

3.8.2 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.

		7	6	5	4	3	2	1	0
LOCAL0 (350H)	Bit symbol	L0E					L0EA22	L0EA21	L0EA20
	Read/Write	R/W					R/W		
	After reset	0					0	0	0
	Function	Bank for LOCAL0 0: Not use 1: Use					Setting bank number for LOCAL0 000 setting is prohibited because it pretend COMMON0 area		
LOCAL1 (351H)	Bit symbol	L1E					L1EA23	L1EA22	L1EA21
	Read/Write	R/W					R/W		
	After reset	0					0	0	0
	Function	Bank for LOCAL1 0: Not use 1: Use					Setting bank number for LOCAL1 011 setting is prohibited because it pretend COMMON1 area		
LOCAL2 (352H)	Bit symbol	L2E					L2EA23	L2EA22	L2EA21
	Read/Write	R/W					R/W		
	After reset	0					0	0	0
	Function	Bank for LOCAL2 0: Disable 1: Enable					Setting bank number for LOCAL2 111 setting is prohibited because it pretend COMMON2 area		
LOCAL3 (353H)	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	Bank for LOCAL3 0: Disable 1: Enable			01000 to 01011 $\overline{CS2D}$ 00000 to 00011 $\overline{CS2B}$ 00100 to 00111 $\overline{CS2C}$		01100 to 01111: $\overline{CS2E}$ 10000 to 11111: Set prohibition		

Figure 3.8.3 MMU Control Register

Example program is as next page follows.

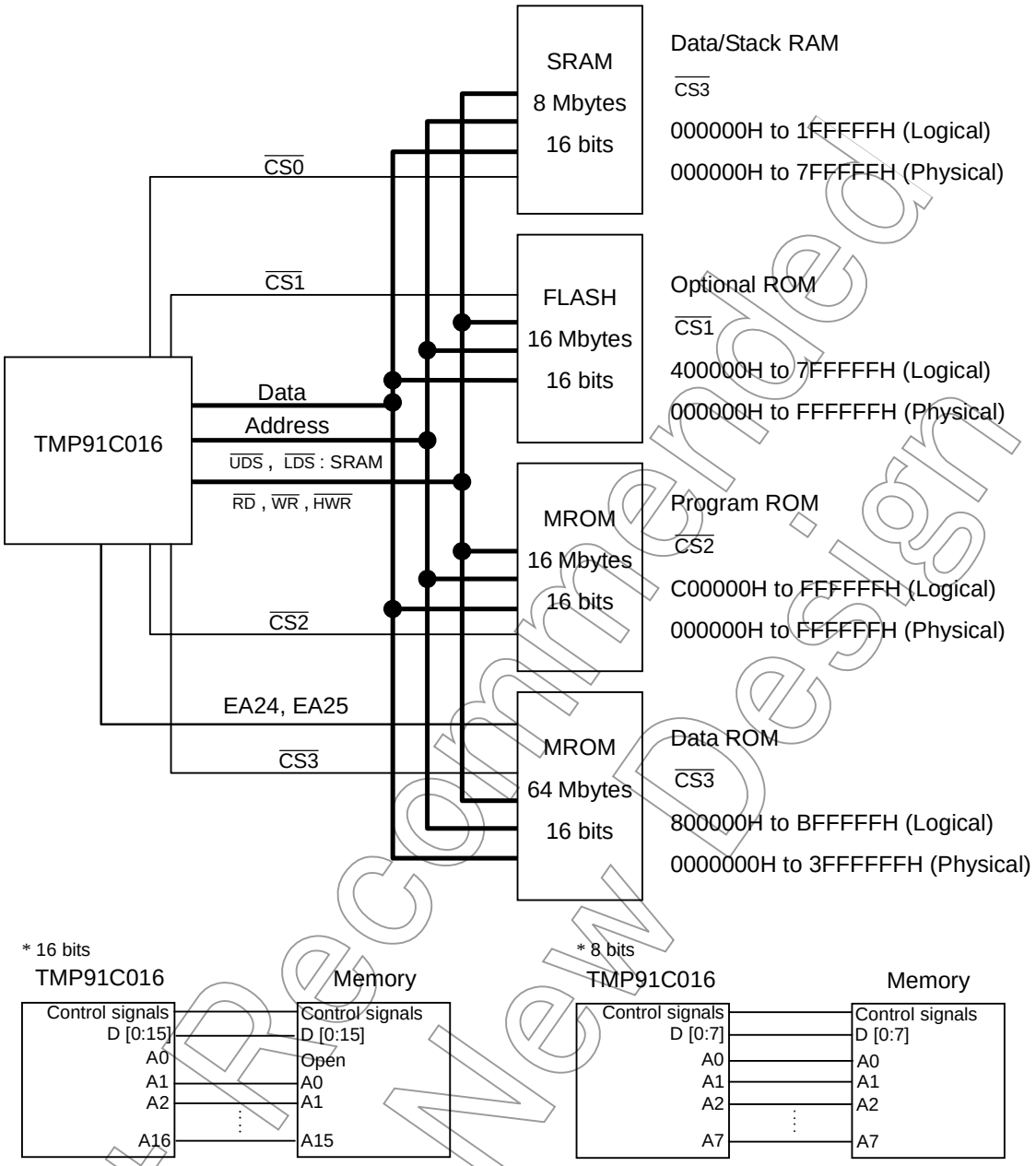


Figure 3.8.4 H/W Setting Example

At Figure 3.8.4, it shows example of connection TMP91C016 and some memories: Program ROM: MROM, 16 Mbytes, Data ROM: MROM, 64 Mbytes, Data RAM: SRAM, 8 Mbytes, Option ROM: Flash, 16 Mbytes. In case of 16-bit bus memory connection, it need to shift 1 bit address bus from TMP91C016 and 8-bit bus case, direct connection address bus from TMP91C016.

In that figure, logical address and physical address are shown. And each memory allot each chip select signal, RAM: $\overline{CS0}$, FLASH_ROM: $\overline{CS1}$, Program MROM: $\overline{CS2}$, Data MROM: $\overline{CS3}$. In case of this example, as Data MROM is 64 Mbytes, this MROM connect to EA24 and EA25.

If you want to use DRAM, it need to assign to CS3 DRAM.

At initial condition after reset, because TMP91C016 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 to 1FFFFFH
        LD      (MAMR0), 7FH    ; Logical address size: 1 Mbyte
        LD      (B0CS), 81H     ; Condition: 16 bits,1wait (8 Mbytes, SRAM)
;CS1
        LD      (MSAR1), 40H    ; Logical address area: 400000H to 7FFFFFH
        LD      (MAMR1), FFH    ; Logical address size: 4 Mbytes
        LD      (B1CS), 80H     ; Condition: 16 bits, 2 waits (16 Mbytes, Flash ROM)
;CS2
        LD      (MSAR2), C0H    ; Logical address area: C00000H to FFFFFFH
        LD      (MAMR2), 7FH    ; Logical address size: 4 Mbytes
        LD      (B2CS), C3H     ; Condition: 16 bits, 0 waits (16 Mbytes, MROM)
;CS3
        LD      (MSAR3), 80H    ; Logical address area: 800000H to BFFFFFH
        LD      (MAMR3), 7FH    ; Logical address size: 4 Mbytes
        LD      (B3CS), 83H     ; Condition: 16 bits, 3 waits (64 Mbytes, MROM)
;CSX
        LD      (BEXCS), 00H    ; Other: 16 bits, 2 waits (Don't care)
;Port
        LD      (P6FC), 3FH     ; CS0 to CS3, EA24, EA25: Port 6 setting
        LD      (P6FC2), C0H    ; LDS, UDS: Port 6 setting
~
```

Figure 3.8.5 Bank Operation S/W Example1

Secondly, it shows example of initial setting at Figure 3.8.5.

Because $\overline{CS0}$ connect to RAM: 16-bit bus, 8 Mbytes, it need to set 8-bit bus. At this example, it set 1 wait setting. In the same way $\overline{CS1}$ set to 16-bit bus and 2 waits, $\overline{CS2}$ set 16-bit bus and 0 waits, $\overline{CS3}$ set 16-bit bus and 3 waits.

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 to CS3's setting. This program example isn't used CSEX setting.

Finally pin condition is set. Port 60 to 65 set to $\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$, EA24, EA25 and $\overline{UDS}$, $\overline{LDS}$.

```
;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 to FFFFFFH
                                ; Physical address 0E00000H to 0FFFFFFH
       LD      (Local 3), 85H   ; LOCAL3 BANK5 set 14xxxxH
       LDW     HL, (800000H)    ; Load data (5555H) form BANK5 (140000H: Physical address)
                                                              of LOCAL3 (CS3 )
       LD      (Local 3), 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 3.8.6 Bank Operation S/W Example2

Here shows example of data access between one bank and other bank.Figure 3.8.6 is one software example. A dot line square area shows one memory and each dot line square shows $\overline{CS2}$'s program ROM and $\overline{CS3}$'s data ROM. Program start from E00000H address, firstly, write to bank register of Local 3 area upper 5 bit address of access point.

In case of this example, 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 800000H to BFFFFFH address: Logical LOCAL3 address, actually access to physical 1400000H to 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 BANK1 of 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 to FFFFFFH
                              ; Physical address 0E00000H to 0FFFFFFH
      LD      (Local 2), 81H  ; LOCAL2 BANK1 set 20xxxxH
      JP      C00000H         ; Jump to BANK1 (200000H: Physical address) of LOCAL2
~
ORG   E00100H
      LD      (Local 2), 83H  ; LOCAL2 BANK3 set 60xxxxH
      JP      C00000H         ; Jump to BANK3 (600000H: Physical address) of LOCAL2
~
ORG   E00200H
      LD      (Local 1), 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      (Local 1), 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 BANK of LOCAL1
      LD      (Local 1), 80H  ; LOCAL1 BANK0 set 00xxxxH
      JP      400000H         ; Jump to BANK0 (000000H: Physical address) of LOCAL1

      It's prohibit to set other bank setting in except common area
      Program run-away

ORG   FFFFFFH                 ; Program ROM: End address at BANK7 of LOCAL1
```


Figure 3.8.7 Bank Operation S/W Exapmle 3

At Figure 3.8.7, it shows example of program jump.

In the same way with before example, two dot line squares show each $\overline{CS2}$'s program ROM and $\overline{CS1}$'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 C00000H to DFFFFFH address: Logical LOCAL2 address, actually jump to physical 2000000H to 3FFFFFH address. When return to common area, it can only jump to E00000H to 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 400000H to 5FFFFFH address: Logical LOCAL1 address, actually jump to physical 8000000H to 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 modifies the bank register's data in bank area, program run-away.

3.9 Serial Channels

TMP91C016 includes 2 serial I/O channels. We call each channels, one is SIO0 and another is SIO1. SIO0 channel can selected either UART mode (Asynchronous transmission) or IrDA mode (Infrared rays transmission). And SIO1 channel can selected either UART mode (Asynchronous transmission) or I/O interface mode (Synchronous transmission).

It start to explain about SIO1 channel functions: UART mode and I/O interface mode below.

- 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 and Figure 3.9.3 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 at Table 3.9.1, hence only the operation of channel 1 is explained below.

Table 3.9.1 Differences between Channels 0 to 1

	Channel 0	Channel 1
Pin name	OPTTX0 (P71) OPTRX0 (P72)	TXD1 (PC3) RXD1 (PC4) $\overline{CTS1}$/SCLK1 (PC5)
I/O interface mode	No support	Support
IrDA mode	Support	No support

This chapter contains the following sections:

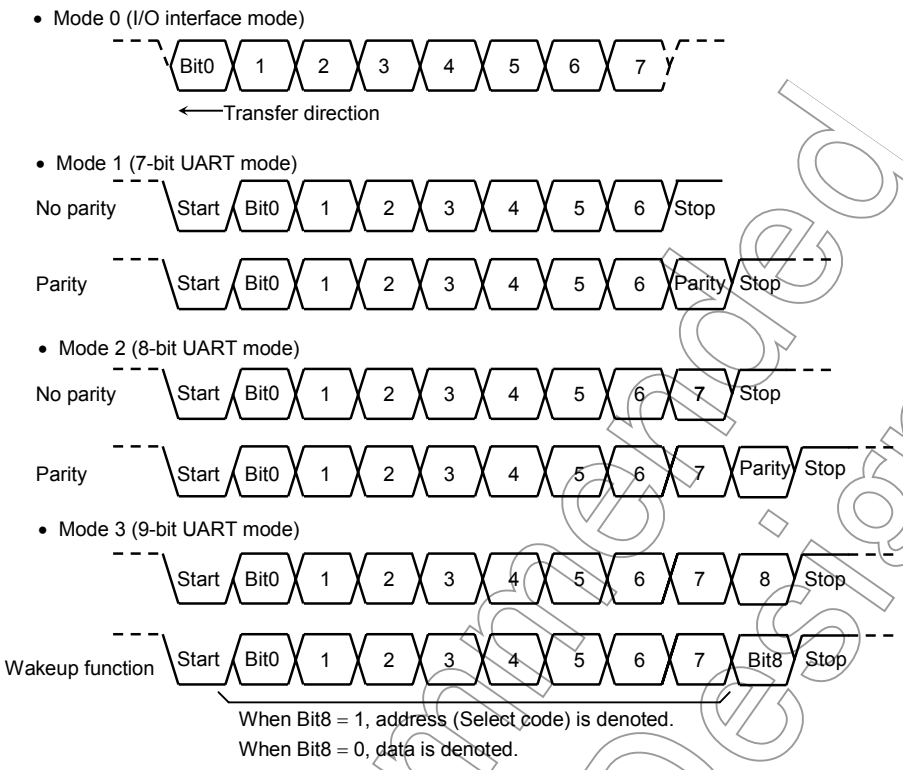


Figure 3.9.1 Data Formats

3.9.1 Block Diagrams

Figure 3.9.2 is a block diagram representing serial channel 0.

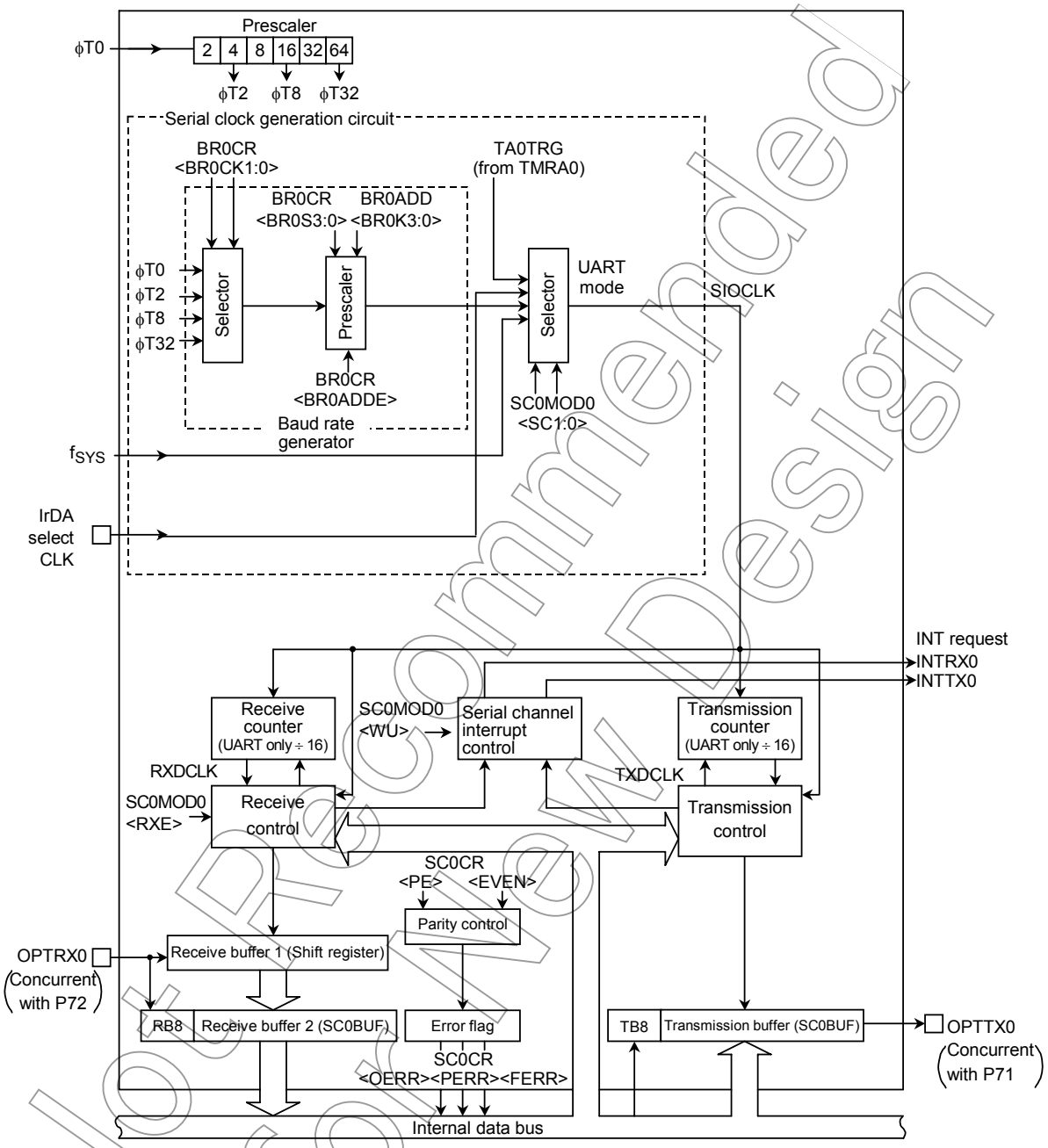


Figure 3.9.2 Block Diagram of the Serial Channel 0 (SIO0)

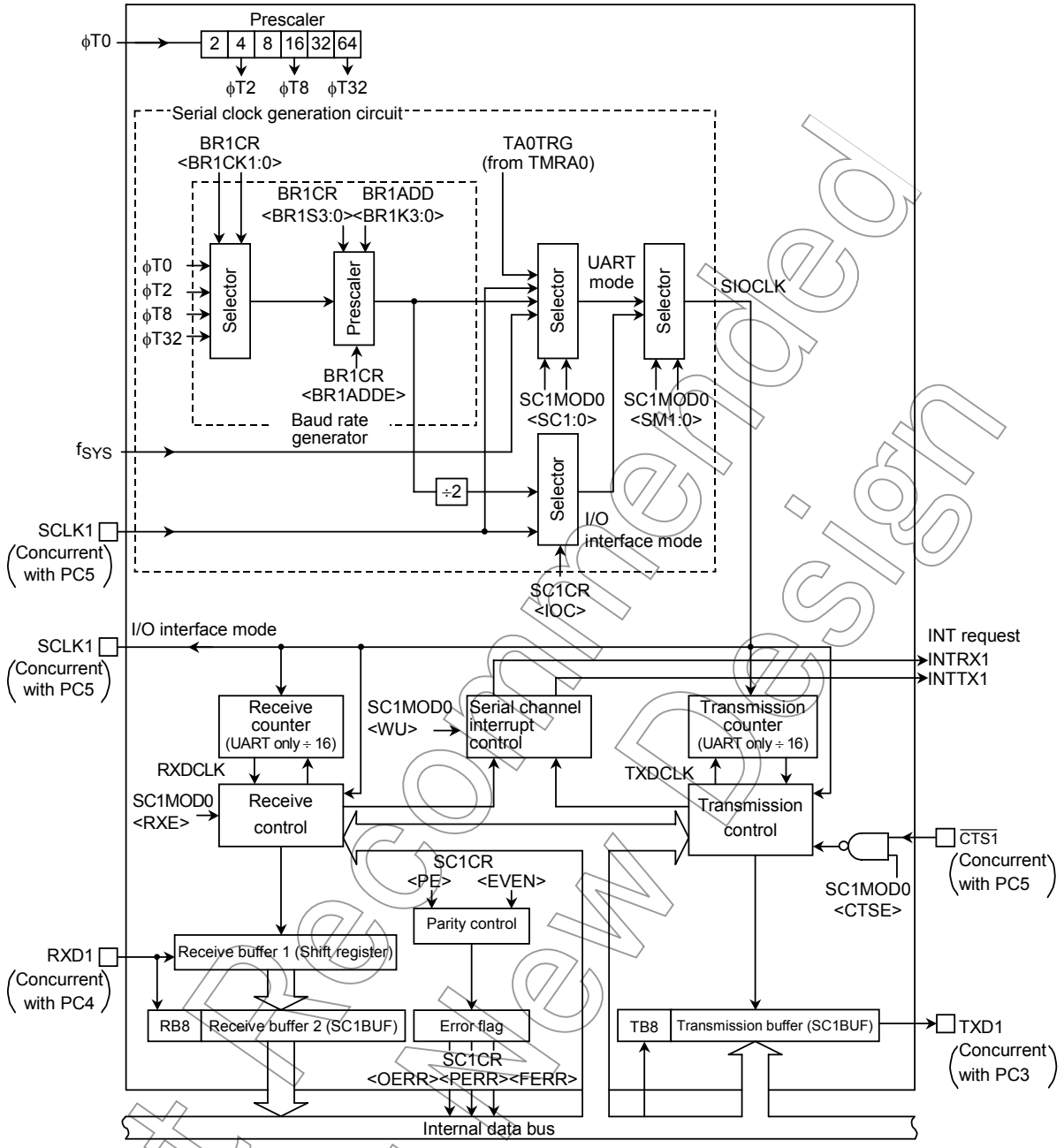


Figure 3.9.3 Block Diagram of the Serial Channel 1 (SIO1)

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

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 a 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 BR1CR<BR0CK1:0> field in the baud rate generator control register.

The baud rate generator includes a frequency divider, which divides the frequency by 1 or n + m/16 (n = 2 to 15, m = 0 to 15) to 16 values, determining the transfer rate.

The transfer rate is determined by the settings of BR1CR<BR1ADDE, BR1S3:0> and BR1ADD<BR1K3:0>.

- In UART mode

(1) When BR1CR<BR1ADDE> = 0

The settings BR1ADD<BR1K3:0> are ignored. The baud rate generator divides the selected prescaler clock by N, which is set in BR1CK<BR1S3:0>. (N = 1, 2, 3 ... 16)

(2) When BR1CR<BR1ADDE> = 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 BR1CR<BR1S3:0> (N = 2, 3 ... 15) and the value of K set in BR1ADD<BR1K3:0> (K = 1, 2, 3 ... 15)

Note: If N = 1 or N = 16, the N + (16 − K)/16 division function is disabled. Set BR1CR<BR1ADDE> to 0.

- In I/O interface mode

The N + (16 − K)/16 division function is not available in I/O interface mode. Set BR1CR<BR1ADDE> 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 (BR1CR<BR1S3:0>) = 5, and BR1CR<BR1ADDE> = 0, the baud rate in UART mode is as follows:

  * Clock state [Clock gear : 1/1

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

$$= \frac{f_C/16}{5} \div 16$$

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

  Note: The N + (16 − K)/16 division function is disabled and setting BR1ADD<BR1K3:0> is invalid.

- N + (16 − K)/16 divider (UART mode only)

  Accordingly, when the source clock frequency (fc) = 4.8 MHz, the input clock frequency = ϕT0, the frequency divider N (BR1CR<BR1S3:0>) = 7, K (BR1ADD<BR1K3:0>) = 3, and BR1CR <BR1ADDE> = 1, the baud rate in UART Mode is as follows:

  * Clock state [Clock gear : 1/1

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

$$= \frac{f_C/4}{7 + \frac{(16-3)}{16}} \div 16$$

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

  Table 3.9.3 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) ≥ 4/fc

  - 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 BR1CR<BR1ADDE> = 0)

Unit (kbps)

fc [MHz]	Input Clock / Frequency Divider (set to BR1CR<BR1S3:0>)	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	28.800	7.200	1.800
↑	3	76.800	19.200	4.800	1.200
↑	6	38.400	9.600	2.400	0.600
↑	C	19.200	4.800	1.200	0.300
19.6608	1	307.200	76.800	19.200	4.800
↑	2	153.600	38.400	9.600	2.400
↑	4	76.800	19.200	4.800	1.200
↑	8	38.400	9.600	2.400	0.600
↑	10	19.200	4.800	1.200	0.300
22.1184	3	115.200	28.800	7.200	1.800
24.576	1	384.000	96.000	24.000	6.000
↑	2	192.000	48.000	12.000	3.000
↑	4	96.000	24.000	6.000	1.500
↑	5	76.800	19.200	4.800	1.200
↑	8	48.000	12.000	3.000	0.750
↑	A	38.400	9.600	2.400	0.600
↑	10	24.000	6.000	1.500	0.375

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

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

Timer out clock (TA0TRG) can be used for source clock of UART mode only.

Calculation method the frequency of TA0TRG

Frequency of TA0TRG = Baud rate × 16

Note 1: The TMRA0 match detect signal cannot be used as the transfer clock in I/O interface mode.

(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 SC1CR<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 SC1CR<IOC> = 1, the rising edge or falling edge will be detected according to the setting of the SC1CR<SCLKS> register to generate the basic clock.

- In UART mode

  The SC1MOD0<SC1:0> setting determines whether the baud rate generator clock, the internal system clock f_{SYS}, 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 SC1CR<IOC> = 0, the RXD1 signal is sampled on the rising or falling edge of the shift clock which is output on the SCLK0 pin.

  In SCLK input mode with the setting SC1CR<IOC> = 1, the RXD1 signal is sampled on the rising or falling edge of the SCLK1 input, according to the SC1CR<SCLKS> setting.

- In UART mode

  The receiving control block has a 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 (SC1BUF); this causes an INTRX0 interrupt to be generated. The CPU only reads receiving buffer 2 (SC1BUF). Even before the CPU reads receiving buffer 2 (SC1BUF), the received data can be stored in receiving buffer 1. However, unless receiving buffer 2 (SC1BUF) 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 SC1CR<RB8> will be preserved.

SC1CR<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 SC1MOD0<WU> to 1; in this mode INTRX1 interrupts occur only when the value of SC1CR<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.

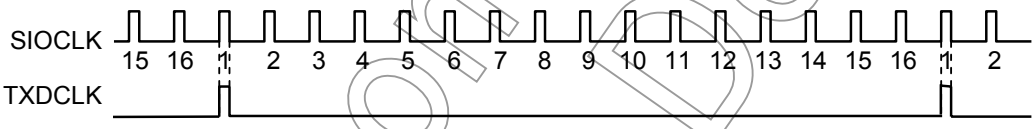


Figure 3.9.4 Generation of the Transmission Clock

(8) Transmission controller

- In I/O interface mode

  In SCLK output mode with the setting SC1CR<IOC> = 0, the data in the transmission buffer is output one bit at a time to the TXD0 pin on the rising edge or falling edge of the shift clock which is output on the SCLK1 pin.

  In SCLK input mode with the setting SC1CR<IOC> = 1, the data in the transmission buffer is output one bit at a time on the TXD1 pin on the rising or falling edge of the SCLK1 input, according to the SC1CR<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.

Handshake function

Use of $\overline{CTS1}$ 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 SC1MOD<CTSE> setting.

When the $\overline{CTS1}$ pin goes high on completion of the current data send, data transmission is halted until the $\overline{CTS1}$ pin goes low again. However, the INTTX1 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 $\overline{RTS}$ pin, a handshake function can be easily configured by setting any port assigned to be the $\overline{RTS}$ function. The $\overline{RTS}$ should be output high to request send data halt after data receive is completed by software in the RXD interrupt routine.

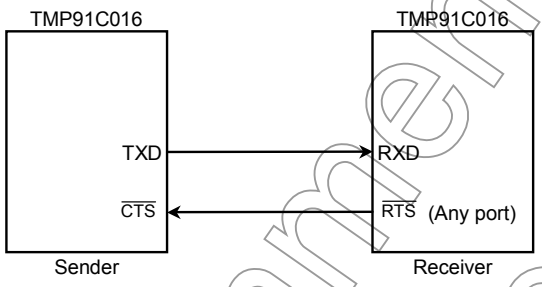
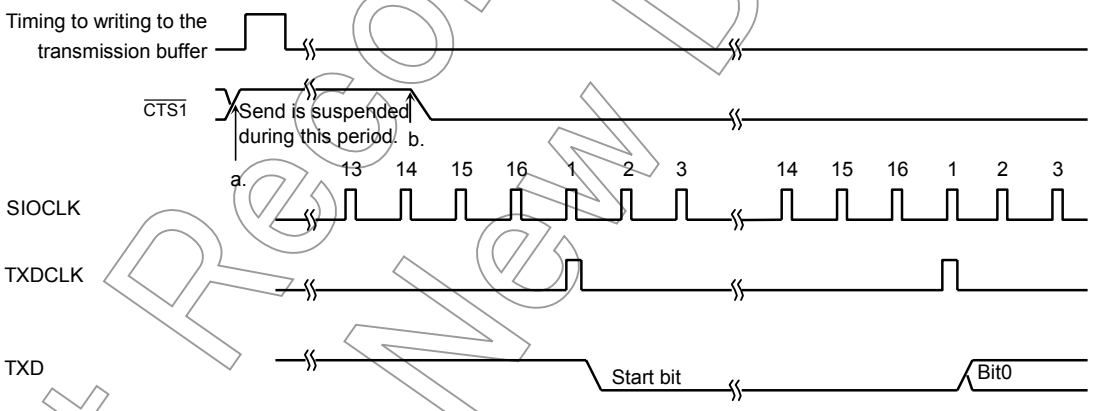


Figure 3.9.5 Handshake Function



Note 1: If the $\overline{CTS1}$ signal goes high during transmission, no more data will be sent after completion of the current transmission.

Note 2: Transmission starts on the first falling edge of the TXDCLK clock after the $\overline{CTS1}$ signal has fallen.

Figure 3.9.6 $\overline{CTS1}$ (Clear to send) Timing

(9) Transmission buffer

The transmission buffer (SC1BUF) 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 INTTX1 interrupt.

(10) Parity control circuit

When SC1CR<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 SC1CR<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 SC1BUF. The data is transmitted after the parity bit has been stored in SC1BUF<TB7> in 7-bit UART mode or in SC1MOD0<TB8> in 8-bit UART mode. SC1CR<PE> and SC1CR<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 (SC1BUF), and then compared with SC1BUF<RB7> in 7-bit UART mode or with SC1CR<RB8> in 8-bit UART mode. If they are not equal, a parity error is generated and the SC1CR<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 (SC1BUF), 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
      a) Set to disable receiving (Write 0 to SC1MOD0<RXE>)
      b) Wait to terminate current frame
      c) Read receiving buffer
      d) Read error flag
      e) Set to enable receiving (Write 1 to SC1MOD0<RXE>)
      f) Request to transmit again
   4) Other

2. Parity error <PERR>

   The parity generated for the data shifted into receiving buffer 2 (SC1BUF) is compared with the parity bit received via the RXD pin. If they are not equal, a Parity error is generated.

3. Framing error <FERR>

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

(12) Timing generation

a. In UART mode

Receiving

Mode	9 Bits	8 Bits + Parity	8 Bits, 7 Bits + Parity, 7 Bits
Interrupt timing	Center of last bit (Bit8)	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 stop bit
Overrun error timing	Center of last bit (Bit8)	Center of last bit (Parity bit)	Center of stop bit

Note: In 9-Bit and 8-Bit + parity mode, interrupts coincide with ninth bit pulse. Thus, when servicing the interrupt, it is necessary to wait for a 1-bit period (to allow the stop bit to be transfered9 to allow checking for a framing error.

Transmitting

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

b. I/O interface

Transmission interrupt timing	SCLK output mode	Immediately after the last bit. (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 (SC1BUF) (e.g. immediately after last SCLK). (See Figure 3.9.21.)
	SCLK input mode	Timing used to transfer received data to receive buffer 2 (SC1BUF) (e.g. immediately after last SCLK). (See Figure 3.9.22.)

3.9.3 SFRs

		7	6	5	4	3	2	1	0
SC0MOD0 (0202H)	Bit symbol	TB8	–	RXE	–	SM1	SM0	SC1	SC0
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Transfer data bit8	Always write 0	Receive function 0: Receive disable 1: Receive enable	Always write 0	Serial transmission mode 00: Don't set 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: IrDA clock	

Serial transmission clock source (UART)

00	Timer TMRA0 match detect signal
01	Baud rate generator
10	Internal clock f_{SYS}
11	IrDA clock

Note: In IrDA mode, "baud-rate generator" and "IrDA clock" can be selected. When "IrDA clock" is selected, set ASKCTL<IRDACK1:0> to f_{SYS}/5 clock. And in this setting, when fc is 18.432MHz, it is set to 115.2Kbps.

Serial transmission mode

00	Don't set	
01	UART mode	7-bit mode
10		8-bit mode
11		9-bit mode

Receiving function

0	Receive disabled
1	Receive enabled

Transmission data bit8

Figure 3.9.7 Serial Mode Control Register (SIO0, SC0MOD0)

SC1MOD0 (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 bit8	Handshake 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 (SCLK1 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)

Note: The Clock selection for the I/O interface mode is controllerd by the serial control register (SC1CR).

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	Don't care Other modes
0	Interrupt generated when data is received	
1	Interrupt generated only when RB8 = 1	

Receiving function

0	Receive disabled
1	Receive enabled

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

0	Disabled (Always transferable)
1	Enabled

Transmission data bit8

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		
Read/Write	R	R/W		R (Cleared to 0 when read)				
After reset	Undefined	0	0	0	0	0		
Function	Received data bit8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: Error				
				Overrun	Parity	Framing		

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	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
Read/Write	R	R/W		R (Cleared to 0 when read)			R/W	
After reset	Undefined	0	0	0	0	0	0	0
Function	Received data bit8	Parity 0: Odd 1: Even	Parity addition 0: Disable 1: Enable	1: Error			0: SCLK1 1: SCLK1	0: Baud rate generator 1: SCLK1 pin input
				Overrun	Parity	Framing		

I/O interface input clock select

0	Baud rate generator
1	SCLK1 pin input

Edge selection for SCKL pin (Input mode only)

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

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 bit8

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

BR0ADD <BR0K3:0> \ BR0CR <BR0S3:0>	BR0CR<BR0ADDE> = 1: 0000 (N = 16) or 0001 (N = 1)	BR0CR<BR0ADDE> = 1: 0010 (N = 2) to 1111 (N = 15)	BR0CR<BR0ADDE> = 0: 0001 (N = 1) (UART only) to 1111 (N = 15) 0000 (N = 16)
0000	Disable	Disable	
0001(K = 1) to 1111(K = 15)	Disable	Divided by N + (16 − K)/16	Divided by N

Note1: Availability of +(16-K)/16 division function

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

The baud rate generator can be set “1” in 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<BR0K3:0> when +(16-K)/16 division function is used. Writes to unused bits in the BR0ADD register do not affext operation, and undefined data is read from these unused bits.

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

BR0CR <BR1S3:BR1S0> / BR1ADD <BR1K3:0>	BR1CR<BR1ADDE> = 1		BR1CR<BR1ADDE> = 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	
0001 (K = 1) to 1111 (K = 15)	Disable	Disabled by N + (16 − K)/16	Divided by N

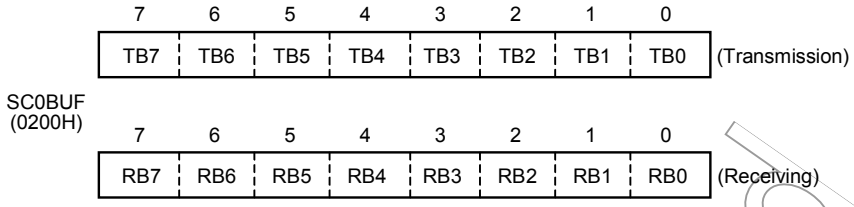
Note1: Availability of +(16-K)/16 division function

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

The baud rate generator can be set “1” in 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<BR1K3:0> when +(16-K)/16 division function is used. Writes to unused bits in the BR1ADD register do not affext operation, and undefined data is read from these unused bits.

Figure 3.9.12 Baud Rate Generator Control (SIO1, BR1CR, BR1ADD)



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)
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	I2S1	FDPX1						
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.

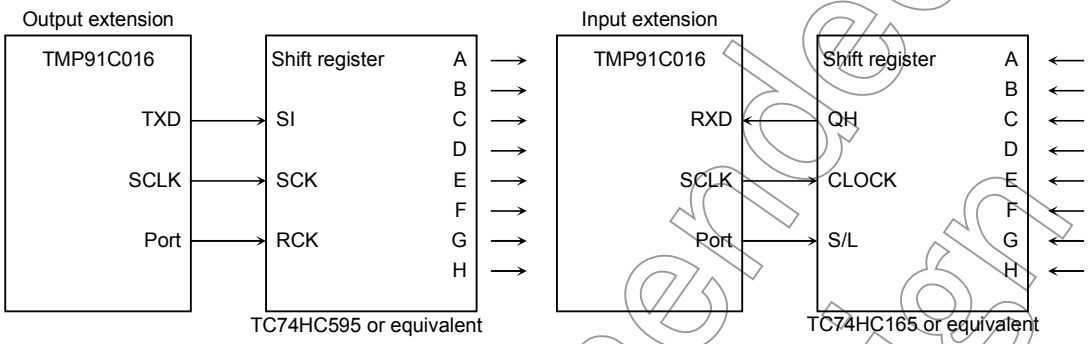


Figure 3.9.17 SCLK Output Mode Connection Example

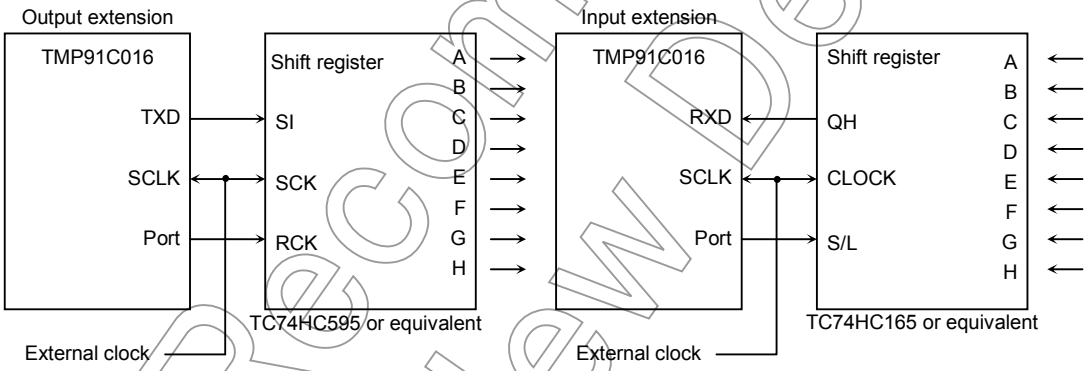


Figure 3.9.18 SCLK Input Mode Connection Example

a. Transmission

In SCLK output mode 8-bit data and a synchronous clock are output on the TXD1 and SCLK1 pins respectively each time the CPU writes the data to the transmission buffer. When all data is output, INTES1<ITX1C> will be set to generate the INTTX1 interrupt.

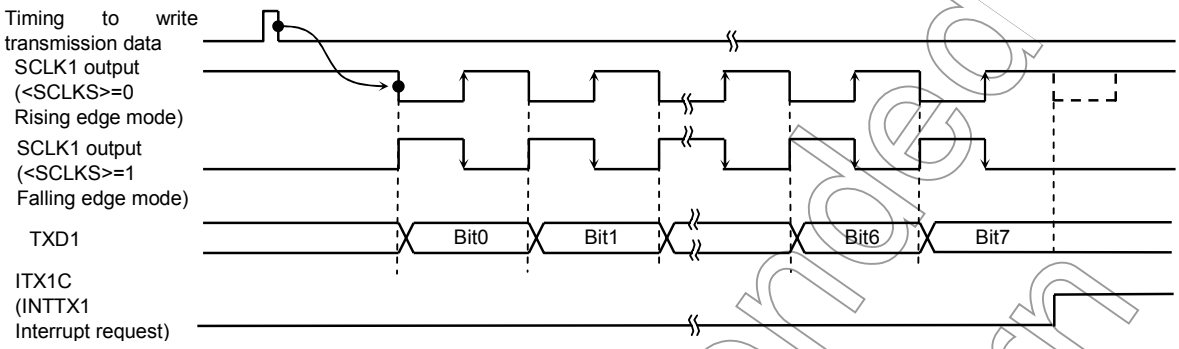


Figure 3.9.19 Transmitting Operation in I/O Interface Mode (SCLK1 output mode)

In SCLK input mode, 8-bit data is output on the TXD1 pin when the SCLK1 input becomes active after the data has been written to the transmission buffer by the CPU.

When all data is output, INTES1<ITX1C> will be set to generate INTTX1 interrupt.

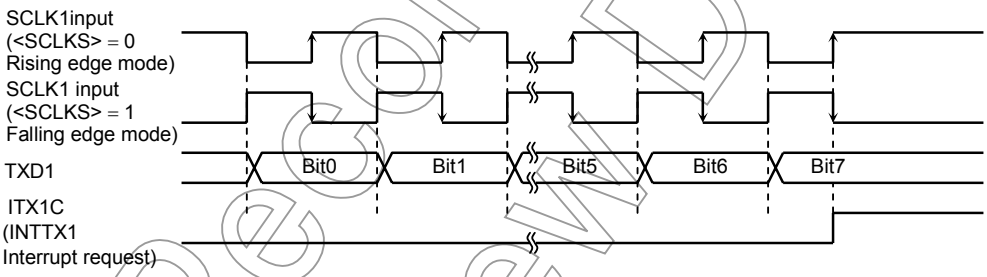


Figure 3.9.20 Transmitting Operation in I/O Interface Mode (SCLK1 input mode)

b. Receiving

In SCLK output mode, the synchronous clock is outputted from SCLK1 pin and the data is shifted to receiving buffer 1. This starts when the Receive Interrupt flag INTES1<IRX1C> is cleared by reading the received data. When 8-bit data are received, the data will be transferred to receiving buffer 2 (SC1BUF according to the timing shown below) and INTES1<IRX1C> will be set to generate INTRX1 interrupt.

The outputting for the first SCLK1 starts by setting SC1MOD0<RXE>to 1.

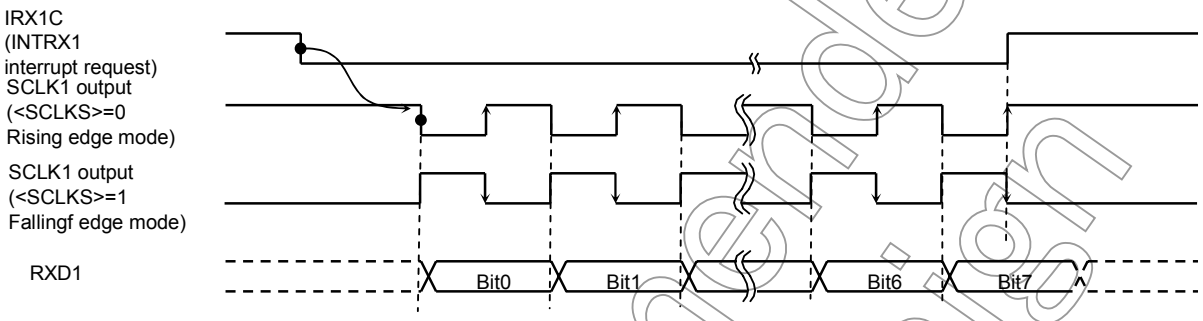


Figure 3.9.21 Receiving Operation in I/O Interface Mode (SCLK1 output mode)

In SCLK input mode, the data is shifted to receiving buffer 1 when the SCLK input becomes active after the receive interrupt flag INTES1<IRX1C> is cleared by reading the received data. When 8-bit data is received, the data will be shifted to receiving buffer 2 (SC1BUF according to the timing shown below) and INTES1<IRX1C> will be set again to be generate INTRX1 interrupt.

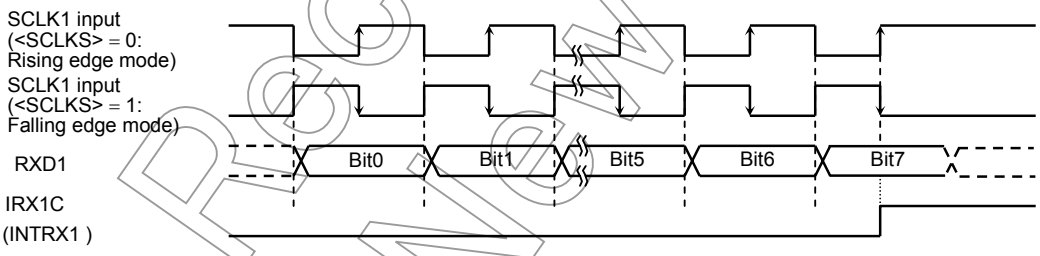


Figure 3.9.22 Receiving Operation in I/O Interface Mode (SCLK1 input mode)

Note: The system must be put in the Receive Enable state (SC1MOD0<RXE> = 1) before data can be received.

c. Transmission and receiving (Full duplex mode)

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

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

* Clock state [Clock gear: 1/1

Main routine

	7	6	5	4	3	2	1	0	
INTES1	0	0	0	1	0	0	0	0	Set the INTTX1 level to 1. Set the INTRX1 level to 0.
PCCR	–	–	1	0	1	X	X	X	
PCFC	X	X	1	X	1	X	X	X	Set PC3, PC4 and PC5 to function as the TXD1, RXD1 and SCLK1 pins respectively.
SC1MOD0	0	0	1	0	0	0	0	0	Enable receiving and select I/O interface mode.
SC1MOD1	1	1	0	0	0	0	0	0	Select full duplex mode.
SC1CR	0	0	0	0	0	0	0	0	SCLK out, transmit on negative edge mode
BR1CR	0	0	1	1	0	0	1	1	Baud rate = 9600 bps
SC1MOD0	0	0	1	0	0	0	0	0	Enable receiving
SC1BUF	*	*	*	*	*	*	*	*	Set the transmit data and start.

INTTX1 interrupt routine

Acc SC1BUF									Read the receiving buffer.
SC1BUF	*	*	*	*	*	*	*	*	Set the next transmit data.

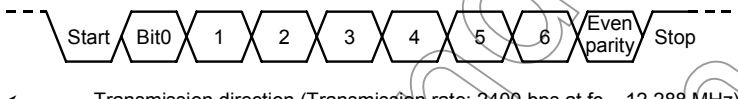
X: Don't care, –: No change

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

7-bit UART mode is selected by setting serial channel mode register SC1MOD0<SM1:0> to 01.

In this mode, a parity bit can be added. Use of a parity bit is enabled or disabled by the setting of the serial channel control register SC1CR<PE> bit; whether even parity or odd parity will be used is determined by the SC1CR<EVEN> setting when SC1CR<PE> is set to 1 (Enabled).

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 Clock gear: 1/1

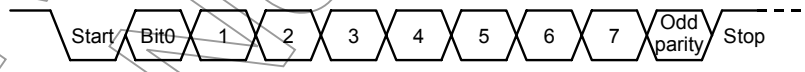
	7	6	5	4	3	2	1	0	
PCCR	← –	–	–	–	1	X	X	X	Set PC3 to function as the TXD1 pin.
PCFC	← X	X	–	X	1	X	X	X	
SC1MOD0	← X	0	–	X	0	1	0	1	Select 7-bit UART mode.
SC1CR	← X	1	1	X	X	X	0	0	Add even parity.
BR1CR	← 0	0	1	0	0	1	0	1	Set the transfer rate to 2400 bps.
INTES1	← 1	1	0	0	–	–	–	–	Enable the INTTX1 interrupt and set it to interrupt level 4.
SC1BUF	← *	*	*	*	*	*	*	*	Set data for transmission.

X: Don't care, –: No change

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

8-bit UART mode is selected by setting SC1MOD0<SM1:0> to 10. In this mode, a parity bit can be added (Use of a parity bit is enabled or disabled by the setting of SC1CR<PE>); whether even parity or odd parity will be used is determined by the SC1CR<EVEN> setting when SC1CR<PE> is set to 1 (Enabled).

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　　Clock gear: 1/1

Main settings

	7	6	5	4	3	2	1	0	
PCCR ←	–	–	–	0	–	X	X	X	Set PC4 to function as the RXD1 pin.
SC1MOD0 ←	–	0	1	X	1	0	0	1	Enable receiving in 8-bit UART mode.
SC1CR ←	X	0	1	X	X	X	0	0	Add even parity.
BR1CR ←	0	0	0	1	0	1	0	1	Set the transfer rate to 9600 bps.
INTES1 ←	–	–	–	–	1	1	0	0	Enable the INTTX1 interrupt and set it to interrupt level 4.

Interrupt processing

Acc	← SC1CR AND 00011100	Check for errors.
if Acc	≠ 0 then ERROR	
Acc	← SC1BUF	Read the received data.

X: Don't care, –: No change

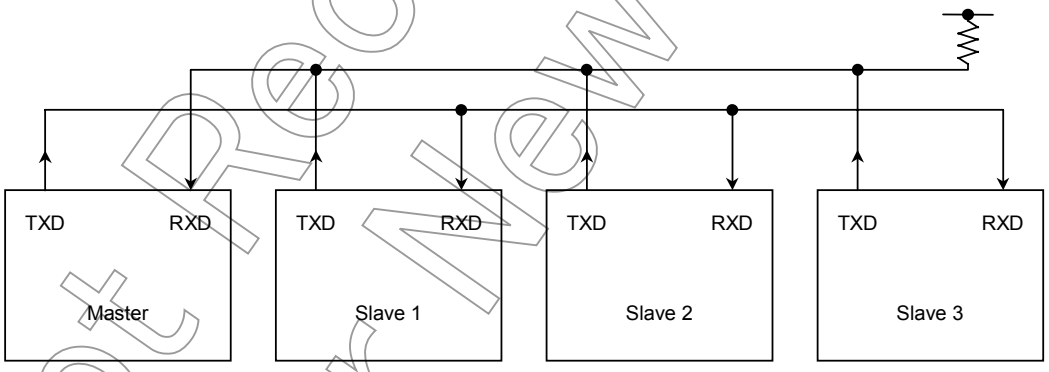
(4) Mode 3 (9-bit UART mode)

9-bit UART mode is selected by setting SC1MOD0<SM1:0> to 11. In this mode parity bit cannot be added.

In the case of transmission the MSB (9th bit) is written to SC1MOD0<TB8>. In the case of receiving it is stored in SC1CR<RB8>. When the buffer is written and read, the MSB is read or written first, before the rest of the SC1BUF data.

Wake-up function

This function is operated on only SIO1. In 9-bit UART mode, the wake-up function for slave controllers is enabled by setting SC1MOD0<WU> to 1. The interrupt INTRX1 occurs only when<RB8> = 1.



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

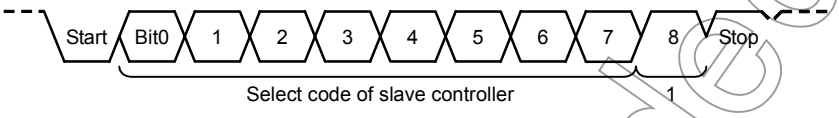
Figure 3.9.23 Serial Link using Wakeup Function

Protocol

a. Select 9-bit UART mode on the master and slave controllers.

b. Set the SC1MOD0<WU> bit on each slave controller to 1 to enable data receiving.

c. The master controller transmits one-frame data including the 8-bit select code for the slave controllers. The MSB (Bit8)<TB8> is set to 1.

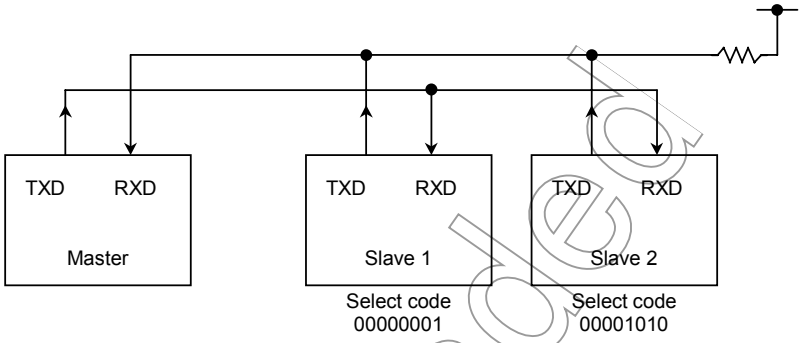


d. Each slave controller receives the above frame. Each controller checks the above select code against its own select code. The controller whose code matches clears its WU bit to 0.

e. The master controller transmits data to the specified slave controller whose SC1MOD0<WU> bit is cleared to 0. The MSB (Bit8) <TB8> is cleared to 0.

f. The other slave controllers (whose <WU> bits remain at 1) ignore the received data because their MSBs (Bit8 or <RB8>) are set to 0, disabling INTRX1 interrupts.
The slave controller (WU bit = 0) can transmit data to the master controller, and it is possible to indicate the end of data receiving to the master controller by this transmission.

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



- Setting the master controller

```
Main
PCCR    ← – – – 0 1 X X X   }  Set PC3 and PC4 to function as the TXD1 and RXD1 pins
PCFC    ← X X – X 1 X X X   }  respectively.
INTES1  ← 1 1 0 0 1 1 0 1      Enable the INTTX1nterrupt and set it to interrupt level 4.
                               Enable the INTRX1interrupt and set it to interrupt level 5.
SC1MOD0 ← 1 0 1 0 1 1 1 0      Set fSYS as the transmission clock for 9-bit UART mode.
SC1UF   ← 0 0 0 0 0 0 0 1      Set the select code for slave controller 1.

INTTX1interrupt
SC1MOD0 ← 0 – – – – – – –      Set TB8 to 0.
SC1UF   ← * * * * * * * *      Set data for transmission.
```

- Setting the slave controller

```
Main
PCCR    ← – – – 0 1 X X X   }
PCFC    ← X X – X 1 X X X   }  Set PC3 to TXD1 (Open-drain output) and PC4 to RXD1.
PCUDOE  ← X X 1 0 0 0 0 0   }
INTES1  ← 1 1 0 1 1 1 1 0      Enable INTRX1and INTTX1
SC1MOD0 ← 0 0 1 1 1 1 1 0      Set <WU> to 1 in 9-bit UART transmission mode using fSYS as
                               the transfer clock.

INTRX1interrupt
Acc ← SC1BUF
if Acc = select code
Then SC1MOD0 ← - - - 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.

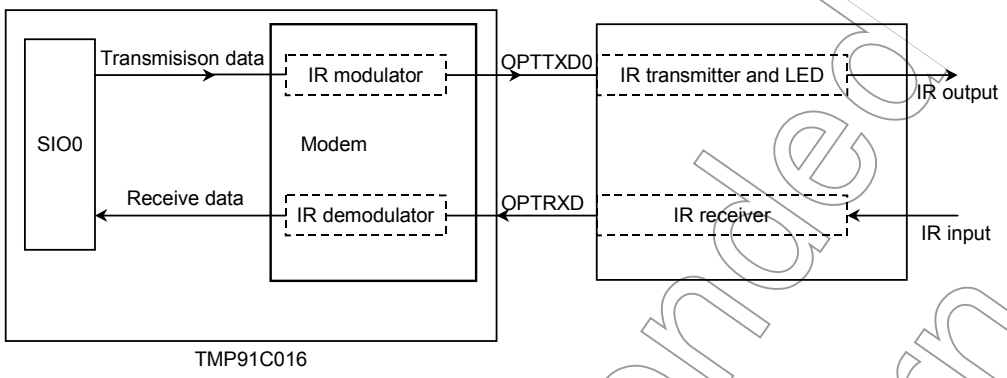


Figure 3.9.24 IrDA Block Diagram

(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.

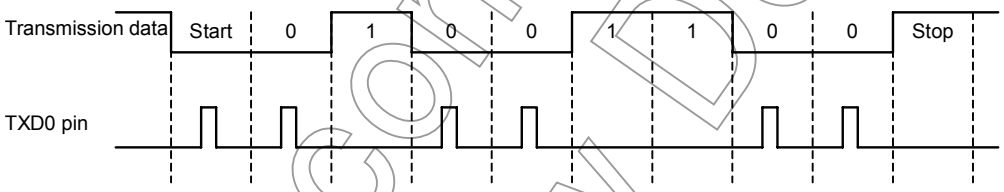


Figure 3.9.25 Modulation Example of Transfer Data

(2) Demodulation 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>.

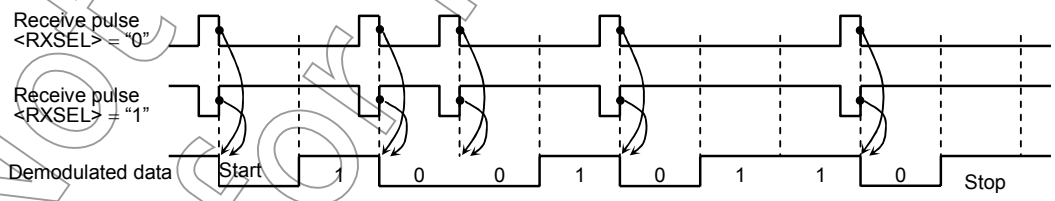


Figure 3.9.26 Demodulation Example of Receive Data

(3) Data format

The data format is fixed as follows:

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

It can't guarantee the correct operation in any other setting.

(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.
↓
4) Start transmission and receiving for SIO0 ; The modem operates as follows:
• 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 3.9.4 Baud Rate and Pulse Width Specifications

Baud Rate	Modulation	Rate Tolerance (% of rate)	Pulse Width (min)	Pulse Width (typ.)	Pulse Width (max)
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 × 3/16 or 1.6 μs (1.6 μs is equal to 3/16 pulse width when baud rate is 115.2 kbps).

The TMP91C016 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 38.4 kbps and 115.2 kbps, the output pulse width should not be set to T × 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 3.9.5 shows Baud rate and pulse width for (16 – K)/16 division function.

Table 3.9.6 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 $2x \times x$ (value + 1) + 100ns Can be set: 1 to 14 Can not be set: 0, 15			

Select receive pulse width

Formula: Effective pulse width

$\geq 2x \times x$ (value + 1) + 100ns

$x = 1/f_{FPH}$

0000	Cannot be set
0001	Equal or more than 4x + 100 ns
to	
1110	Equal or more than 30x + 100 ns
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

Note: If a pulse width complying with the IrDA1.0 standard (1.6μs min.) can be guaranteed with a low baud rate, setting this bit to "1" shortens the duration of infrared ray activation, resulting in reduced power dissipation.

Figure 3.9.27 IrDA Control Register

3.10 DRAM Controller

TMP91C016 incorporates a 1-channel DRAM controller for interface with × 8-/16-bit DRAM. The DRAM controller consists of a control circuit to refresh the DRAM, an access circuit for reading and writing, and a row/column address multiplexer.

1) Refresh mode
   $\overline{CAS}$ before $\overline{RAS}$ refreshing
2) Refresh interval
   Programmable (31 to 2700 states)
3) Refresh cycle width
   Programmable (2 to 9 states)
4) Mapping areas
   $\overline{CS3}$ area
5) Address mapping size
   $\overline{CS3}$ areas: 32 kbytes-8 Mbytes
6) Memory access mode
   2CAS mode
7) Memory access address length
   8 to 11 bits selectable
8) Wait control
   In according with CS/WAIT controller setting
9) Arbitration of refresh/access contention
   Refresh has higher priority. Wait states are automatically inserted in the access cycle.

DREFCR1 Register

		7	6	5	4	3	2	1	0
DREFCR (0430H)	Bit symbol	DMI	RS2	RS1	RS0	RW2	RW1	RW1	RC
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Dummy cycle 0: Disable 1: Dummy cycle	Refresh cycle insertion interval 000: 31 states 001: 110 states 010: 220 states 011: 450 states 100: 900 states 101: 1200 states 110: 1800 states 111: 2700 states			Refresh cycle width 000: 2 states 001: 3 states 010: 4 states 011: 5 states 100: 6 states 101: 7 states 110: 8 states 111: 9 states			Refresh cycle 0: Disable 1: Enable

Refresh cycle control

0	No refresh cycle
1	Refresh cycle

Refresh cycle width control

RW2	RW1	RW0	Refresh cycle width
0	0	0	2 states
0	0	1	3 states
0	1	0	4 states
0	1	1	5 states
1	0	0	6 states
1	0	1	7 states
1	1	0	8 states
1	1	1	9 states

See Table 3.10.3 “Refresh Cycle Insertion Intervals”

Dummy cycle control

0	No dummy cycle
1	Dummy cycle

Figure 3.10.1 Refresh Control Register

		7	6	5	4	3	2	1	0
DMEMCR (0431H)	Bit symbol	SRFC	-	-	MACM	MUXE	MUXW	MUXW	MAC
	Read/Write	W	R/W						
	After reset	1	0	0	0	0	0	0	0
Read -modify -write instruction is prohibited	Function	Self-refresh 0: Self-refresh 1: Self-refresh release	Always Write 0	Always write 0	Memory access control 0: Normal 1: Slow	Address multiplexing 0: Disable 1: Enable	Multiplex address length 00: 8 bits 01: 9 bits 10: 10 bits 11: 11 bits		Memory access control 0: Disable 1: Enable

Memory access control

0	Disable
1	Enable

Multiplex address length control
(Valid only when MUXE = 1)

MUXW	MUXW	Multiplex address length
0	0	8 bits
0	1	9 bits
1	0	10 bits
1	1	11 bits

Address multiplex control

0	Disable
1	Enable

Memory access speed control

0	Normal access mode
1	Slow access mode

Self-refresh mode control

0	Self-refresh
1	Self-refresh release

Figure 3.10.2 DRAM Memory Access Control Register

3.10.1 Description of Operation

TMP91C016 has a one-channel internal DRAM controller. This channel is normally linked to CS3 of the CS/WAIT controller. The DRAM controller generates the DRAM access cycle. The DRAM signals share pins with port 6 and port 7 (for details on setting the pins to DRAM pins, see 3.5.4, Port 6 and 3.5.5 Port 7)

(1) Memory access control

Setting DMEMCR<MAC> to 1 enables access control. If the area set as the $\overline{CS3}$ area in the CS/WAIT controller is accessed when access control is enabled, a valid signal is output to DRAM in accordance with the DRAM memory access control register setting. The access cycle (Bus cycle, number of waits) at this time depends on the $\overline{CS3}$ area setting in the CS/WAIT controller.

If the bus size is 16-bits, the specified area is accessed using the $\overline{2CAS}$ ($\overline{RAS}$, $\overline{UCAS}$ $\overline{LCAS}$ and $\overline{WE}$), depending on the DMEMCR<MACS> setting. When the bus size is 8 bits, the specified area is accessed by the $\overline{RAS}$, $\overline{CAS}$ and $\overline{WE}$ signals regardless of the <MACS> setting.

To facilitate the connection with low-speed DRAM, the DRAM controller accelerates the rising of $\overline{RAS}$ signal when some waits are inserted, and extends the $\overline{RAS}$ pre-charge time (RAS high width). Slow access mode is set by DMEMCR<MACM>. A reset clears <MACH> to 0 and sets NORMAL mode.

The internal address multiplexer outputs the row/column address from A0 to A11 during the access cycle. The DMEMCR<MUXE> bit specifies whether or not to multiplex addresses, and DMEMCR<MUX0:1> specifies the multiplexed address width. Note, however, that the multiplexed address lines depend on the bus size: 8 bits or 16 bits.

Table 3.10.1 DRAM Pins

Pin Name \ Mode	8-Bit Bus	16-Bit Bus
P63 ($\overline{CS3}$, $\overline{RAS}$)	$\overline{RAS}$	$\overline{RAS}$
P74 ($\overline{CAS}$, $\overline{WE}$)	$\overline{CAS}$	$\overline{WE}$
P67 ($\overline{LCAS}$, $\overline{LDS}$, $\overline{REFOUT}$)	$\overline{REFOUT}$	$\overline{LCAS}$
P66 ($\overline{UCAS}$, $\overline{UDS}$, $\overline{WE}$)	$\overline{WE}$	$\overline{UCAS}$
P73 ($\overline{DRAMOE}$, $\overline{EXRD}$, $\overline{NMI}$)	$\overline{DRAMOE}$	$\overline{DRAMOE}$

Table 3.10.2 Address Multiplexing (–: Don't care)

Row Address	Column Address							
	8 Bits		9 Bits		10 Bits		11 Bits	
	8	16	8	16	8	16	8	16
A0	A8	–	A9	–	A10	–	A11	–
A1	A9	A9	A10	A10	A11	A11	A12	A12
A2	A10	A10	A11	A11	A12	A12	A13	A13
A3	A11	A11	A12	A12	A13	A13	A14	A14
A4	A12	A12	A13	A13	A14	A14	A15	A15
A5	A13	A13	A14	A14	A15	A15	A16	A16
A6	A14	A14	A15	A15	A16	A16	A17	A17
A7	A15	A15	A16	A16	A17	A17	A18	A18
A8	–	A16	A17	A17	A18	A18	A19	A19
A9	–	–	–	A18	A19	A19	A20	A20
A10	–	–	–	–	–	A20	A21	A21
A11	–	–	–	–	–	–	–	A22

Multiplex address length

Access bus size (Set in the CS/WAIT controller)

(2) Refresh control block

TMP91C016 outputs the $\overline{RAS}$, $\overline{CAS}$ ($\overline{LCAS}$, $\overline{UCAS}$) signals, which can be used for refreshing DRAM. When using an 8-bit bus, the device also outputs state signal $\overline{REFOUT}$ to indicate a refresh cycle.

As the output cycle and pulse width of the $\overline{RAS}$, $\overline{CAS}$ ($\overline{LCAS}$, $\overline{UCAS}$) output can be set by program, the DRAM refresh is easily realized. The refresh controller block has the following features.

- Refresh mode: $\overline{CAS}$ -before- $\overline{RAS}$ interval refresh mode, $\overline{CAS}$ -before- $\overline{RAS}$ self-refresh mode
- Refresh interval: 31 to 2700 states (Programmable)
- Refresh cycle width: 2 to 9 states (Programmable)
- Dummy cycles can be generated.
- The refresh cycle is asynchronous the CPU operating cycle.

$\overline{CAS}$ -before- $\overline{RAS}$ interval refresh mode

The refresh interval and the refresh cycle width in the $\overline{CAS}$ -before- $\overline{RAS}$ interval refresh mode vary according to the DRAM being used.

The refresh interval and the refresh cycle width in TMP91C016 can be set in accordance with the system clock and type of DRAM used, by modifying the value of the refresh control register.

a. Refresh cycle insertion interval

3 bits of the DREFCR<RS2:0> register is used to set insertion interval in accordance with the system clock used.

Example: When using the system clock at 25 MHz, set these bits to 111 to set the DRAM refresh cycle to 216 μs.

Table 3.10.3 Refresh Cycle Insertion Interval

(Unit: μs)

Refresh Cycle			Insertion Interval (States)	Frequency (f_{OSCH})						
RS2	RS1	RS0		8 MHz	10 MHz	12.5 MHz	14 MHz	16 MHz	20 MHz	25 MHz
0	0	0	31	7.55	6.2	4.96	4.43	3.88	3.1	2.48
0	0	1	110	27.5	22	17.6	15.7	13.75	11	8.80
0	1	0	220	55	44	35.2	31.4	27.5	22	17.6
0	1	1	450	112.5	90	72	64.3	56.25	45	36
1	0	0	900	225	180	144	128.6	112.5	90	72
1	0	1	1200	300	240	192	171.4	150	120	96
1	1	0	1800	450	360	288	257.1	225	180	144
1	1	1	2700	675	540	432	385.7	337.5	270	216

b. Refresh cycle width

3 bits of the DREFCR<RW2:0> register can vary the refresh cycle width ($\overline{RAS}$, $\overline{CAS}$, low output width).

c. Refresh cycle control

Manipulating the bits of the DREFCR<RC> register enables or disables the refresh cycle.

$\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ self-refresh mode

This mode is used when the clock supplied to the DRAM controller is stopped by a HALT instruction (IDLE, STOP) while refreshing using the $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ interval refresh mode.

To refresh DRAM in $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ self-refresh mode, first, set DRAM to $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ interval refresh mode. Then, before entering the HALT instruction, set DMEMCR<SRFC> to 0 to execute a single $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ interval refresh. Then the $\overline{\text{CAS}}$ and $\overline{\text{RAS}}$ pins maintain their low levels, and $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ self-refresh mode starts. When the halt is released and the clock is supplied to the DRAM controller, DMEMCR<SRFC> is automatically set to 1 and $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ self-refresh mode is released. After the release, be sure to execute a single $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ interval refresh to return to interval refresh mode. (Note that when a halt is released by a reset, the I/O registers are initialized; therefore, the $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ interval refresh is not executed.)

After setting DMEMCR<SRFC> to 0, execute any instruction, such as a NOP instruction, then execute a HALT instruction.

In case of resetting release HALT condition, register is cleared, too, refresh operation can not be moved. After reset, $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ ($\overline{\text{LCAS}}$, $\overline{\text{UCAS}}$) pins become to High-Z mode on TMP91C016.

If it need data protection after reset condition, it need external pull-down resistor to those pins.

(3) DRAM initialization

The DRAM controller can generate the continuous $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ dummy cycles required when using DRAM. Setting the DREFCR<DMI> bit to 1 generates the dummy cycles. Dummy cycle generation is released by writing 0 to <DMI> (Including a write due to reset), by enabling refresh cycle insertion (DREFCR<RC> = 1), or by enabling access control (DMEMCR<MAC> = 1).

When dummy cycle generation is released by enabling refresh cycle insertion or by enabling access control, the <DMI> bit is not cleared to 0. The dummy cycle width is fixed to 4 states; the interval, to 6 states.

3.10.2 Priorities

As the DRAM refresh cycle is asynchronous to the CPU operating cycle, the refresh cycle may overlap with DRAM read and write cycles. If an overlap occurs, the DRAM controller gives priority to the cycle that started first. In case of CPU access first, refresh cycle occurs after CPU access, and in case of refresh cycle first, DRAMC automatically insert to WAIT to CPU until to finish that refresh cycle.

3.10.3 Connection Example

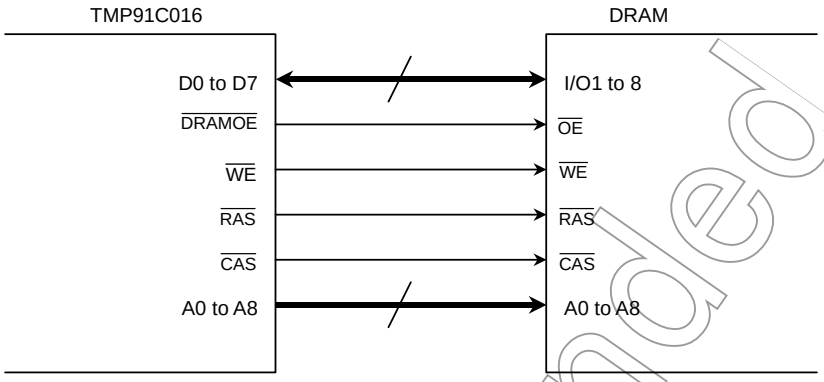


Table 3.10.4 8-Bit Bus Configuration

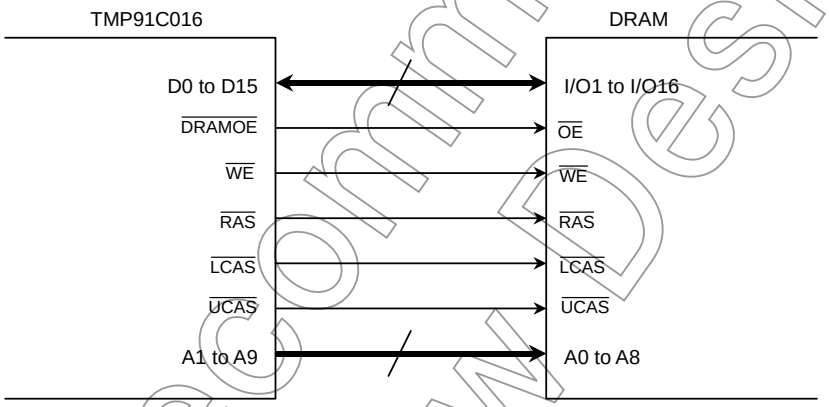


Table 3.10.5 16-Bit Bus Configuration

3.11 Watchdog Timer (Runaway detection timer)

The TMP91C016 features a watchdog timer for detecting runaway.

The watchdog timer (WDT) is used to return the CPU to normal state when it detects that the CPU has started to malfunction (Runaway) due to causes such as noise.

When the watchdog timer detects a malfunction, it generates a non-maskable interrupt INTWD to notify the CPU. Connecting the watchdog timer output to the reset pin internally forces a reset. (The level of external $\overline{\text{RESET}}$ pin is not changed)

3.11.1 Configuration

Figure 3.11.1 is a block diagram of he watchdog timer (WDT).

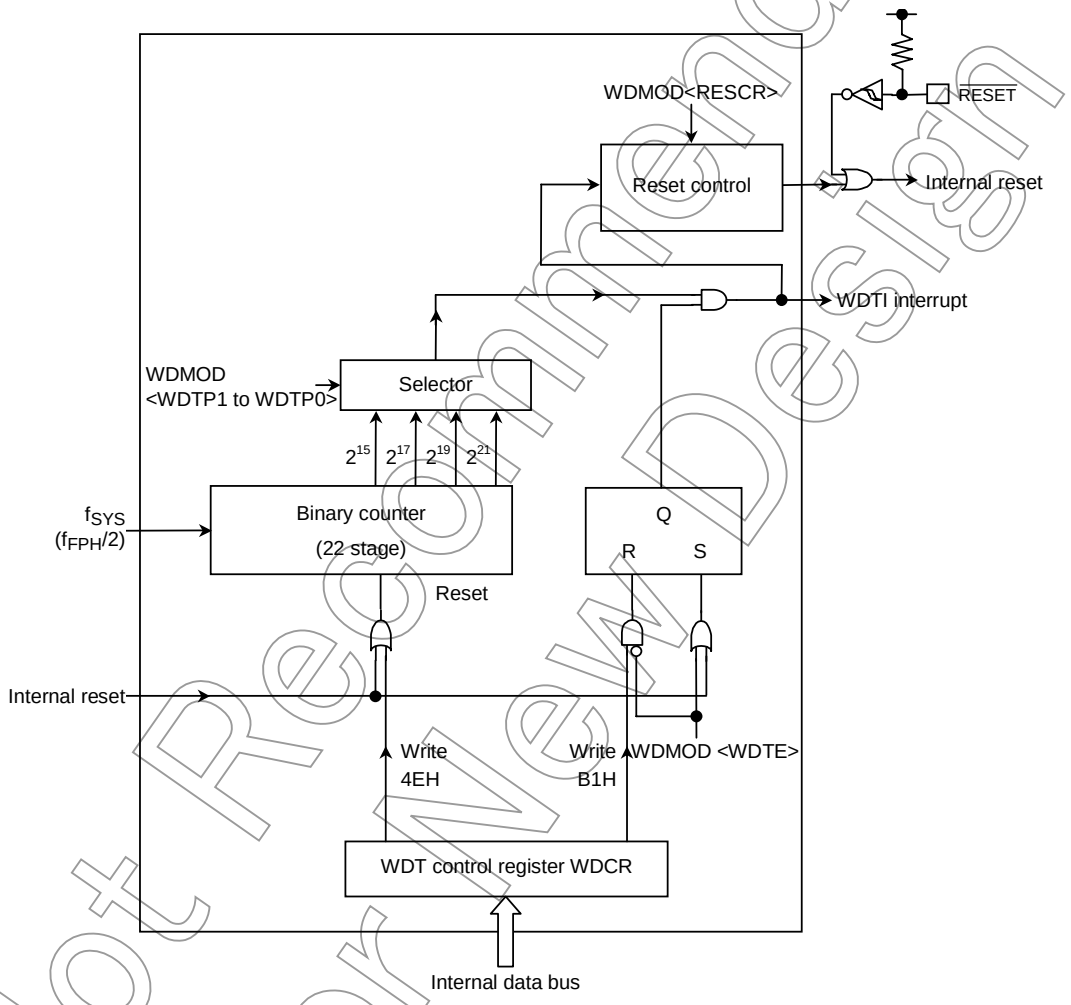


Figure 3.11.1 Block Diagram of Watchdog Timer

Note: It needs to care designing the total machine set, because watchdog timer can't operate completely by external noise.

The watchdog timer consists of a 22-stage binary counter which uses the system clock (f_{SYS}) as the input clock. The binary counter can output $f_{SYS}/2^{15}$, $f_{SYS}/2^{17}$, $f_{SYS}/2^{19}$ and $f_{SYS}/2^{21}$.

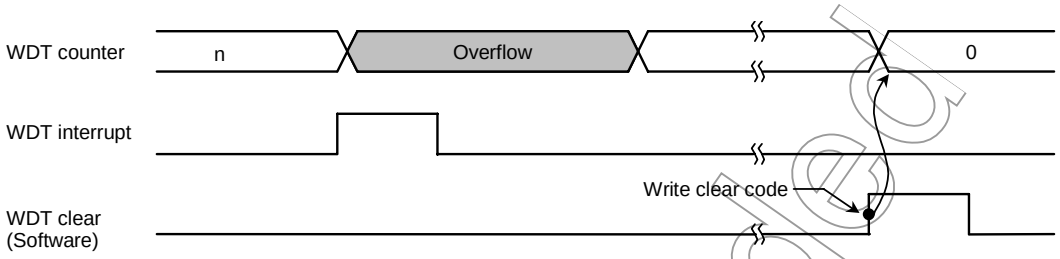


Figure 3.11.2 NORMAL Mode

The runaway is detected when an overflow occurs, and the watchdog timer can reset device. In this case, the reset time will be between 22 and 29 states (26.1 to 34.4 μs at f_{OSCH} = 1 state)is $f_{FPH}/2$, where f_{FPH} is generated by dividing the high-speed oscillator clock (f_{OSCH}) by sixteen through the clock gear function.

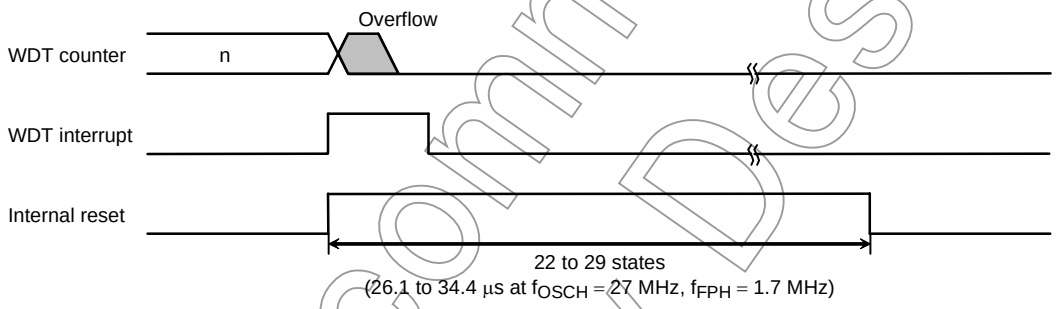


Figure 3.11.3 RESET Mode

3.11.2 Control Registers

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

(1) Watchdog timer mode register (WDMOD)

a. Setting the detection time for the watchdog timer in <WDTP1:0>

This 2-bit register is used for setting the watchdog timer interrupt time used when detecting runaway. After reset, this register is initialized to WDMOD<WDTP1:0> = 00.

The detection times for WDT are shown in Figure 3.11.4.

b. Watchdog timer enable/disable control register <WDTE>

After Reset, WDMOD<WDTE> is initialized to 1, enabling the watchdog timer.

To disable the watchdog timer, it is necessary to set this bit to 0 and to write the disable code (B1H) to the watchdog timer control register WDCR. This makes it difficult for the watchdog timer to be disabled by runaway.

However, it is possible to return the watchdog timer from the disabled state to the enabled state merely by setting <WDTE> to 1.

c. Watchdog timer out reset connection <RESCR>

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

(2) Watchdog timer control register (WDCR)

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

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

```
WDMOD   ← 0 – – – – – –     Clear WDMOD<WDTE> to 0.
WDCR    ← 1 0 1 1 0 0 0 1   Write the disable code (B1H).
```

- Enable control

  Set WDMOD<WDTE> to 1.

- Watchdog timer clear control

  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).
```

Note1: If it is used disable control, set the disable code (B1H) to WDCR after write the clear code (4EH) once. (Please refer to setting example.)

Note2: If it is changed Watchdog timer setting, change setting after set to disable condition once.

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

Watchdog timer out control

0	–
1	Connects WDT out to a reset

IDLE2 Control

0	Stop
1	Operation

Watchdog timer detection time

at fc = 27 MHz, fs = 32.768 kHz

SYSCR1 System Clock Selection <SYSCK>	SYSCR1 Gear Value <GEAR2:0>	Watchdog Timer Detection Time WDMOD<WDTP1:0> 00	01	10	11
1 (fs)	XXX	2.0 s	8.0 s	32.0 s	128.0 s
0 (fc)	000 (fc)	2.43 ms	9.71 ms	38.84 ms	155.34 ms
	001 (fc/2)	4.85 ms	19.42 ms	77.67 ms	310.69 ms
	010 (fc/4)	9.71 ms	38.84 ms	155.34 ms	621.38 ms
	011 (fc/8)	19.42 ms	77.67 ms	310.69 ms	1242.76 ms
	100 (fc/16)	38.84 ms	155.34 ms	621.38 ms	2485.51 ms

Watchdog timer enable/disable control

0	Disabled
1	Enabled

Figure 3.11.4 Watchdog Timer Mode Register

WDCR (0301H)

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

Read -modify -write instructions are prohibited

→ Disable/Clear WDT

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

Figure 3.11.5 Watchdog Timer Control Register

3.11.3 Operation

The watchdog timer generates an INTWD interrupt when the detection time set in the WDMOD<WDTP1:0> has elapsed. The watchdog timer must be cleared 0 by software before an INTWD interrupt will be generated. If the CPU malfunctions (e.g., 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-multifunction program. By connecting the watchdog timer out pin to a peripheral device's reset input, the occurrence of a CPU malfunction can also be relayed to other devices.

The watchdog timer works immediately after reset.

The watchdog 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: a. Clear the binary counter.

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

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

```
WDMOD ← 1 0 1 – – – – –
```

c. Disable the watchdog timer.

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

3.12 Real Time Clock (RTC)

3.12.1 Function Description for RTC

1) Clock function (second, minute, hour, day, month, leap year)
2) Auto Calender function
3) 24 or 12-hour (AM/PM) clock function
4) ±30 second adjustment function (by software)
5) Alarm output 1Hz/16Hz (from $\overline{\text{ALARM}}$ pin)
6) Interrupt generate by Alarm output 1Hz/16Hz

3.12.2 Block Diagram

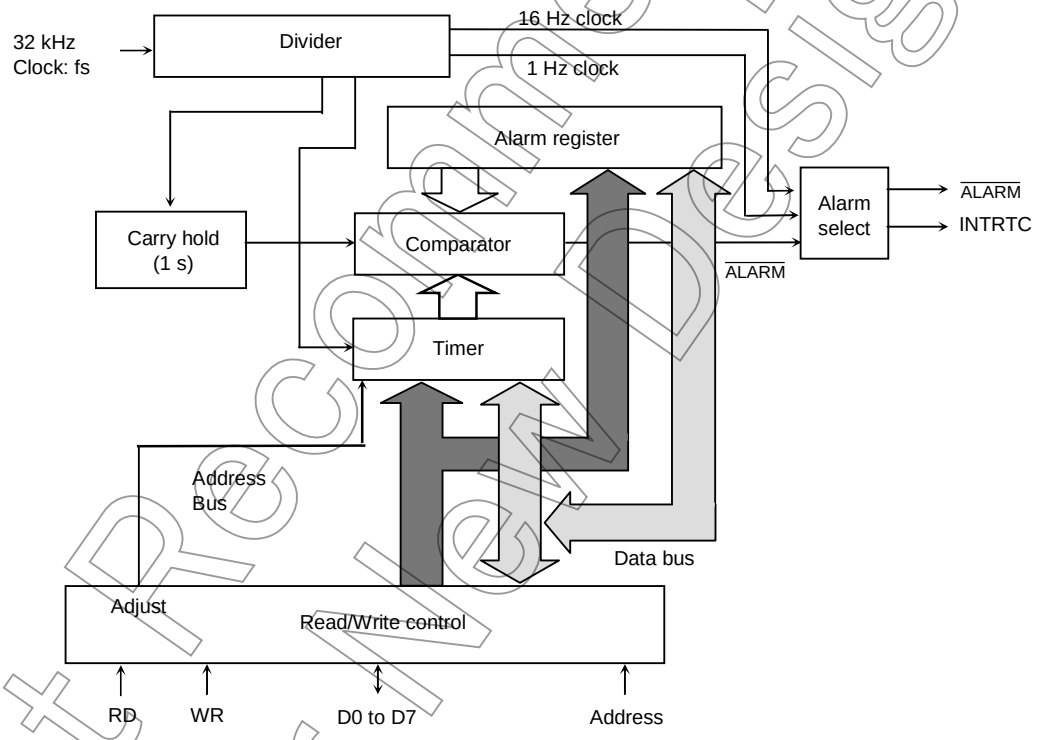


Figure 3.12.1 Block Diagram

Note 1: 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.

Note 2: 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.

3.12.3 Control Registers

Table 3.12.1 Page 0 (Timer function) Registers

Symbol	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function	Read/Write
SECR	0320H		40 s	20 s	10 s	8 s	4 s	2 s	1 s	Scound column	R/W
MINR	0321H		40 min	20 min	10 min	8 min	4 min	2 min	1 mon	Minute column	R/W
HOURR	0322H			20 /PM/AM	10 hours	8 hours	4 hours	2 hours	1 hours	Hour column	R/W
DAYR	0323H						W2	W1	W0	Day of the weel 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.	Deb.	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	Interrupt enable			Adjust-me nt function	Clock enable	Alarm enable		PAGE setting	PAGE register	W, R/W
RESTR	0328H	1HZ enable	16HZ enable	Clock reset	Alarm reset	Always write “0”				Reser register	W

Note1: As for SECR, MINR, HOURR, DAYR, MONTHR, YEARR of PAGE0, current state is read when read it.

Table 3.12.2 Page 1 (Alarm function) Registers

Symbol	Address	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0	Function	Read/Write
SECR	0320H										R/W
MINR	0321H		40 min	20 min	10 min	8 min	4 min	2 min	1 mon	Minute column	R/W
HOURR	0322H			20 /PM/AM	10 hours	8 hours	4 hours	2 hours	1 hours	Hour column	R/W
DAYR	0323H						W2	W1	W0	Day of the weel column	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							LEAP1	LEAP0	Leap-year mode	R/W
PAGER	0327H	Interrupt enable				Clock enable	Alarm enable		PAGE setting	PAGE register	W, R/W
RESTR	0328H	1HZ enable	16HZ enable	Clock reset	Alarm reset	Always write “0”				Reset register	W

Note2: As for SECR, MINR, HOURR, DAYR, MONTHR, YEARR of PAGE1, current state is read when read it.

3.12.4 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 Page 0 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 s column	20 s column	10 s column	8 s column	4 s column	2 s column	1 s column

0	0	0	0	0	0	0	0 s
0	0	0	0	0	0	1	1 s
0	0	0	0	0	1	0	2 s
0	0	0	0	0	1	1	3 s
0	0	0	0	1	0	0	4 s
0	0	0	0	1	0	1	5 s
0	0	0	0	1	1	0	6 s
0	0	0	0	1	1	1	7 s
0	0	0	1	0	0	0	8 s
0	0	0	1	0	0	1	9 s
0	0	1	0	0	0	0	10 s

0	0	1	1	0	0	1	19 s
0	1	0	0	0	0	0	20 s

0	1	0	1	0	0	1	29 s
0	1	1	0	0	0	0	30 s

0	1	1	1	0	0	1	39 s
1	0	0	0	0	0	0	40 s

1	0	0	1	0	0	1	49 s
1	0	1	0	0	0	0	50 s

1	0	1	1	0	0	1	59 s

Note: Do not set the data other than showing above.

(2) Minute column register (for Page 0/1)

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	20 min column	10 min column	8 min column	4 min column	2 min column	1 min 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

Note: Do not set the data other than showing above.

(3) Hour column register (for Page 0/1)

a. In case of 24-hour clock mode (MONTHR<MO0> = 1) of Page 1

HOURR (0322H)

	7	6	5	4	3	2	1	0
Bit symbol			HO5	HO4	HO3	HO2	HO1	HO0
Read/Write			R/W					
After reset			Undefined					
Function	0 is read.		20 h column	10 h column	8 h column	4 h column	2 h column	1 h 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

Note: Do not set the data other than showing above.

b. In case of 12-hour clock mode (MONTHR<MO0> = 0) of Page 1

HOURR (0322H)

	7	6	5	4	3	2	1	0
Bit symbol			HO5	HO4	HO3	HO2	HO1	HO0
Read/Write			R/W					
After reset			Undefined					
Function	0 is read.		PM/AM	10 h column	8 h column	4 h column	2 h column	1 h 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

Note: Do not set the data other than showing above.

(4) Day of the week column register (for Page 0/1)

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.					2 week	1 week	0 week

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

Note: Do not set the data other than showing above.

(5) Day column register (for Page 0/1)

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.		20 d	10 d	8 d	4 d	2 d	1 d

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

Note1: Do not set the data other than showing above.

Note2: Do not set the day which is not existed. (ex: 30th Feb)

(6) Month column register (for Page 0 only)

MONTHR (0325H)

	7	6	5	4	3	2	1	0
Bit symbol				MO4	MO3	MO2	MO1	MO0
Read/Write				R/W				
After reset				Undefined				
Function	0 is read.			10 month	8 month	4 month	2 month	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

Note: Do not set the data other than showing above.

(7) Select 24-hour clock or 12-hour clock (for Page 1 only)

MONTHR (0325H)

	7	6	5	4	3	2	1	0
Bit symbol								MO0
Read/Write								R/W
After reset								Undefined
Function	0 is read.							1: 24 h 0: 12 h

(8) Year column register (for Page 0 only)

YEARR (0326H)

	7	6	5	4	3	2	1	0
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 years

1	0	0	1	1	0	0	1	99 years
0	0	0	0	0	0	0	0	00 years
0	0	0	0	0	0	0	1	01 years
0	0	0	0	0	0	1	0	02 years
0	0	0	0	0	0	1	1	03 years
0	0	0	0	0	1	0	0	04 years
0	0	0	0	0	1	0	1	05 years

1	0	0	1	1	0	0	1	99 y

Note: Do not set the data other than showing above.

(9) Leap-year register (for Page 1 only)

YEARR (0326H)

	7	6	5	4	3	2	1	0
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 Page 0/1)

PAGER (0327H)

Read-modify write instruction are proibited

	7	6	5	4	3	2	1	0
Bit symbol	INTENA			ADJUST	ENATMR	ENAALM		PAGE
Read/Write	R/W			W	R/W	R/W		R/W
After reset	0			Undefined	Undefined	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

Note: Pleas keep the setting order below and don't set same time.
(Set difference time to Clock/Alarm setting and interrupt setting)

(Example) Clock setting/Alarm setting

ld (pager), 0ch : Clock, Alarm enable

ld (pager), 8ch : Interrupt enable

PAGE	0	Select Page0
	1	Select Page1

ADJUST	0	Don't care
	1	Adjust sec. counter. When set 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". Output Adjust signal during 1 cycle of f_{SYS}. After being adjusted once, Adjust is released automatically. (PAGE0 only)

(11) Reset register setting (for Page 0/1)

RESTR (0328H)

Read-modify write instruction are proibited

	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: 1 Hz	0: 16 Hz	1: Timer reset	1: Alarm reset	Always write "0"			

RSTALM	0	Unused
	1	Reset alarm register

RSTTMR	0	Unused
	1	Reset timer register

<DIS1HZ>	<DIS1HZ>	(PAGER) <ENAALM>	Source signal
1	1	1	Alarm
0	1	0	1Hz
1	0	0	16Hz
Others			Output "0"

3.12.5 Operational Description

(1) Reading timer data

a. 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.

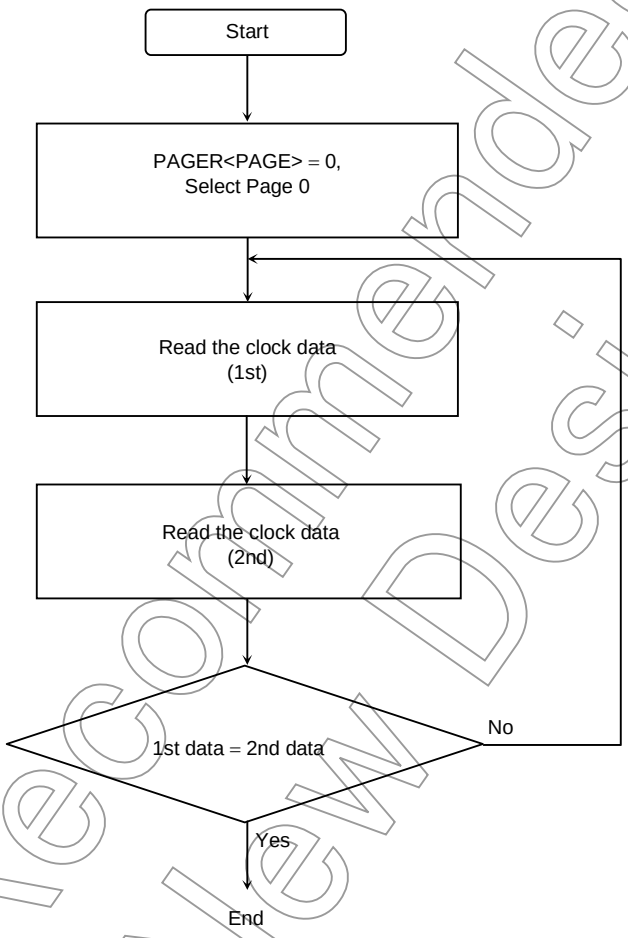


Figure 3.12.2 Flowchart of Timer Data Read

(2) Timing of INTRTC and Clock data

When time is read by interrupt, read clock data within 0.5s(s) after generating interrupt. This is because count up of clock data occurs by rising edge of 1Hz pulse cycle.

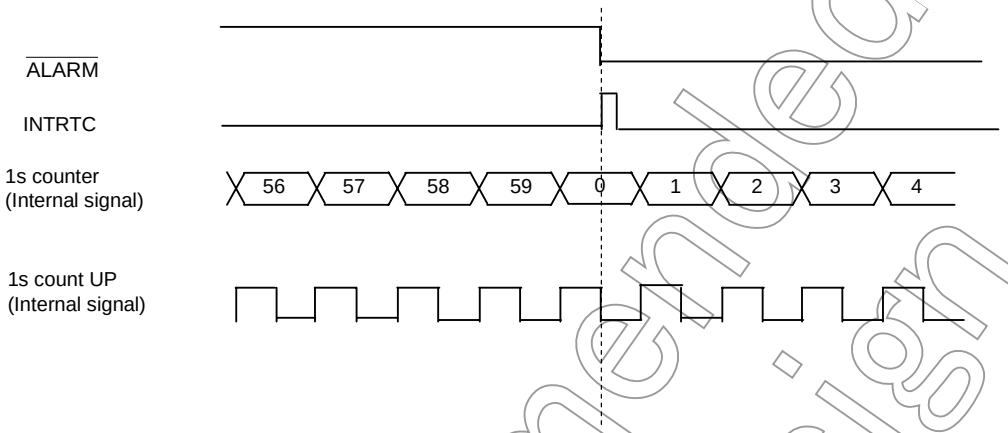


Figure 3.12.3 Timing of INTRTC and Clock data

(3) 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.

a. Reset for a divider

Inside of RTC, there is 15-stage divider which generates 1 Hz clock from 32.768 kHz. Carry of a timer is not done for one second when reset this divider. So write in data during this interval.

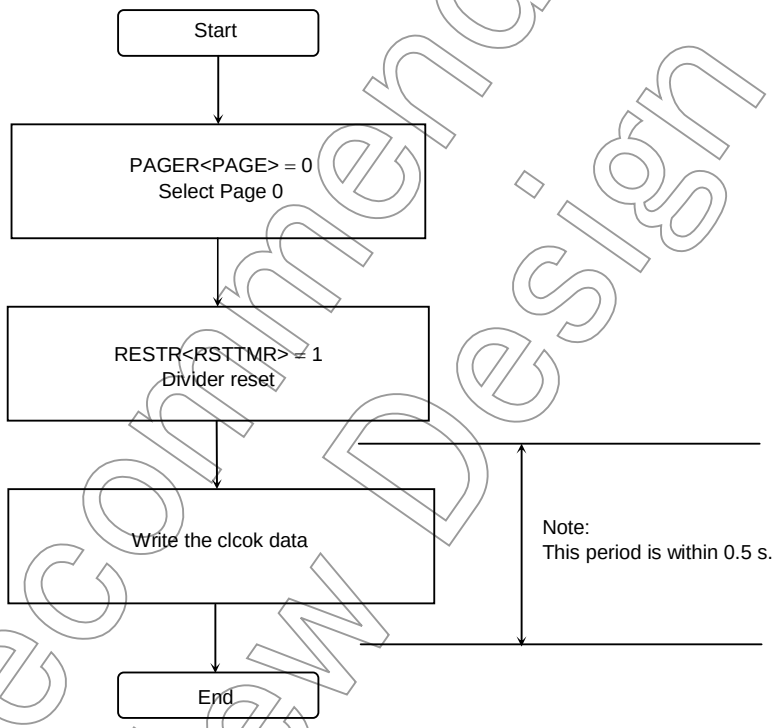


Figure 3.12.4 Flowchart of Data Write

b. Disabling the timer

Carry of a timer is prohibited when write "0" to PAGER<ENATMR> and can prevent malfunction by 1s Carry hold circuit. During a timer prohibited, 1s Carry hold circuit 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.

Since clock hold circuit is not initialized by external $\overline{\text{RESET}}$, a second counter may added 1 or 2 sec at the case of only after power supply is on. To avoid it, the below is recommended setting flow.

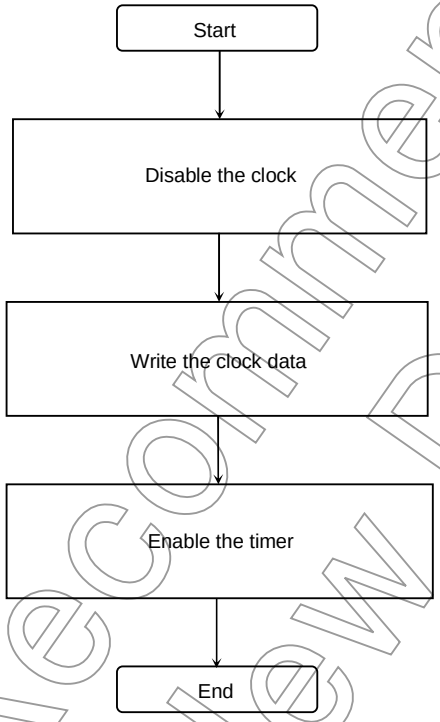


Figure 3.12.5 Flowchart of Timer Disable

3.12.6 Explanation of the Alarm Function

Can use alarm function by setting of register of PAGE1 and output either of three signal from $\overline{\text{ALARM}}$ pin as follows by write "1" to PAGER<PAGE>. INTRTC outputs 1shot pulse when the falling edge is detected. RTC is not initializes by RESET. Therefore, when clock or alarm function is used, clear interrupt request flag in INTC (interrupt controller).

(1) In accordance of alarm register and the timer, output "0".
(2) Output clock of 1 Hz.
(3) Output clock of 16 Hz.

(1) In accordance of alarm register and a timer, output 0.

When value of a clock of PAGE0 accorded with alarm register of PAGE1 with a state of PAGER<ENAALM>= "1", output "0" to $\overline{\text{ALARM}}$ 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 clock and request INTRTC interrupt if PAGER<ENAALM> is "1".

Setting alarm min., alarm hour, alarm day and alarm the day week are done by writing in data at each register of PAGE1.

When all setting contents accorded, RTC generates INTRTC interrupt, if PAGER<INTENA><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 set to Don't care.

The following is an example program for outputting alarm from $\overline{\text{ALARM}}$ -pin at noon (PM12:00) every day.

```
LD      (PAGER), 09H        ;   Alarm disable, setting PAGE1
LD      (RESTR), D0H        ;   Alarm initialize
LD      (DAYR), 01H         ;   W0
LD      (DATAR),01H         ;   1 day
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 when you use only internal interruption.

(2) When output clock of 1 Hz

RTC outputs clock of 1 Hz to $\overline{\text{ALARM}}$ 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 16 Hz

RTC outputs clock of 16 Hz to $\overline{\text{ALARM}}$ pin by setting up PAGER<ENAALM> = 0, RESTR<DIS1HZ> = 1, <DIS16HZ> = 0. And RTC generates INTRTC interrupt by falling edge of the clock.

3.13 LCD Driver Controller (LCDC)

The TMP91C016 incorporates two types liquid crystal display driving circuit for controlling LCD driver LSI.

One circuit handles a RAM build-in type LCD driver that can store display data in the LCD driver in itself, and the other circuit handles a shift-register type LCD driver that must serially transfer the display data to LCD driver for each display picture.

Shift-register type LCD driver control mode (SR mode)

Set the mode of operation, start address of source data save memory and LCD size to control register before setting start register. After set start register LCDC outputs bus release request to CPU and read data from source memory. After that LCDC transmits data of volume of LCD size to external LCD driver through data bus. At this time, control signals (DIBSCP etc.) connected LCD driver output specified waveform synchronize with data transmission. After finish data transmission, LCDC cancels the bus release request and CPU will re-start.

RAM built-in type LCD driver control mode (RAM mode)

Data transmission to LCD driver is executed by move instruction of CPU.

After setting mode of operation to control register, when move instruction of CPU is executed LCDC outputs chip select signal to LCD driver connected to the outside from control pin (D1BSCP etc.). Therefore control of data transmission numbers corresponding to LCD size is controlled by instruction of CPU.

Special mode

It is assigned <TA3LCDE> at bit0 and <TA3MLDE> at bit1, of EMCCR4 register (00E7hex). These bits are used when you want to operate LCDD and MELODY circuit without low frequency clock (XTIN, XTOUT). After reset these two bits set to 0 and low clock is supplied each LCDD and MELODY circuit. If you write these bits to 1, TA3 (Generate by timer 3) is supplied each LCDD and MELODY circuit. In this case, you should set 32 kHz timer 3 frequency. For detail, look AC specification characteristics.

This section is constituted as follows.

3.13.1 Feature of LCDC of Each Mode

Each feature and operation of pin is as follows.

Table 3.13.1 Feature of LCDC of Each Mode

		Shift- Register Type LCD Driver Control Mode	RAM Built-in Type LCD Driver Control Mode
The number of picture elements can be handled		Common (Row): 64, 68, 80, 100, 120, 128, 144, 160, 200, 240 Segment (Column): 32, 64, 80, 120, 128, 160, 240, 320, 360	There is not a limitation
Receiver data bus width		8 bits,16 bits selectable	8 bits,16 bits, selectable (Depend on CPU command)
Transfer data bus width		8 bits, 4 bits,1 bit selectable	8 bits fixed
Transfer rate (at f_{FPH} = 16 [MHz])		250 ns/1 byte at Byte mode 375 ns/1 byte at Nibble mode 1125 ns/1 byte at Bit mode	Equal to memory cycle
External pins	Data bus: (D7 to D0)	Data bus; Connect with DI pin of column driver. Upper 7 pins do not use in Bit mode and upper 4 pins do not use in Nibble mode.	Data bus; Connect with DB pin of column/row driver.
	Write strobe: ($\overline{WR}$)	Not used	Write strobe; Connect with $\overline{WR}$ pin of column/row driver.
	Address bus: (A0)	Not used	Address 0; Connect with D/I pin of column driver. When A0 = 1 data bus value means display data, when A0 = 0 data bus means instruction data.
	Shift clock pulse: (D1BSCP)	Shift clock pulse; Connect with SCP pin of column driver. LCD driver latches data bus value by falling edge of this pin.	Chip enable for column driver 1; Connect with $\overline{CE}$ pin of column driver 1.
	Latch pulse: (D2BLP)	Latch pulse output; Connect with LP/EIO1 pin of column/row driver. Display data is latched in output buffer in LCD driver by rising edge of this pin.	Chip enable for column driver 2; Connect with $\overline{CE}$ pin of column driver 2.
	Frame: (D3BFR)	LCD frame output; Connect with FR pin of column/row driver.	Chip enable for column driver 3; Connect with $\overline{CE}$ pin of column driver 3.
	Cascade pulse: (DLEBCD)	Cascade pulse output; Connect with DIO1 pin of row driver. This pin outputs 1 shot pulse by every D3BFR pin changes.	Chip enable for row driver; Connect with $\overline{LE}$ pin of row driver.
	Display OFF: ($\overline{DOFF}$)	Display off output; Connect with $\overline{DSPOF}$ terminal of column/row driver. L means display off and H means display on.	

3.13.2 Block Diagram

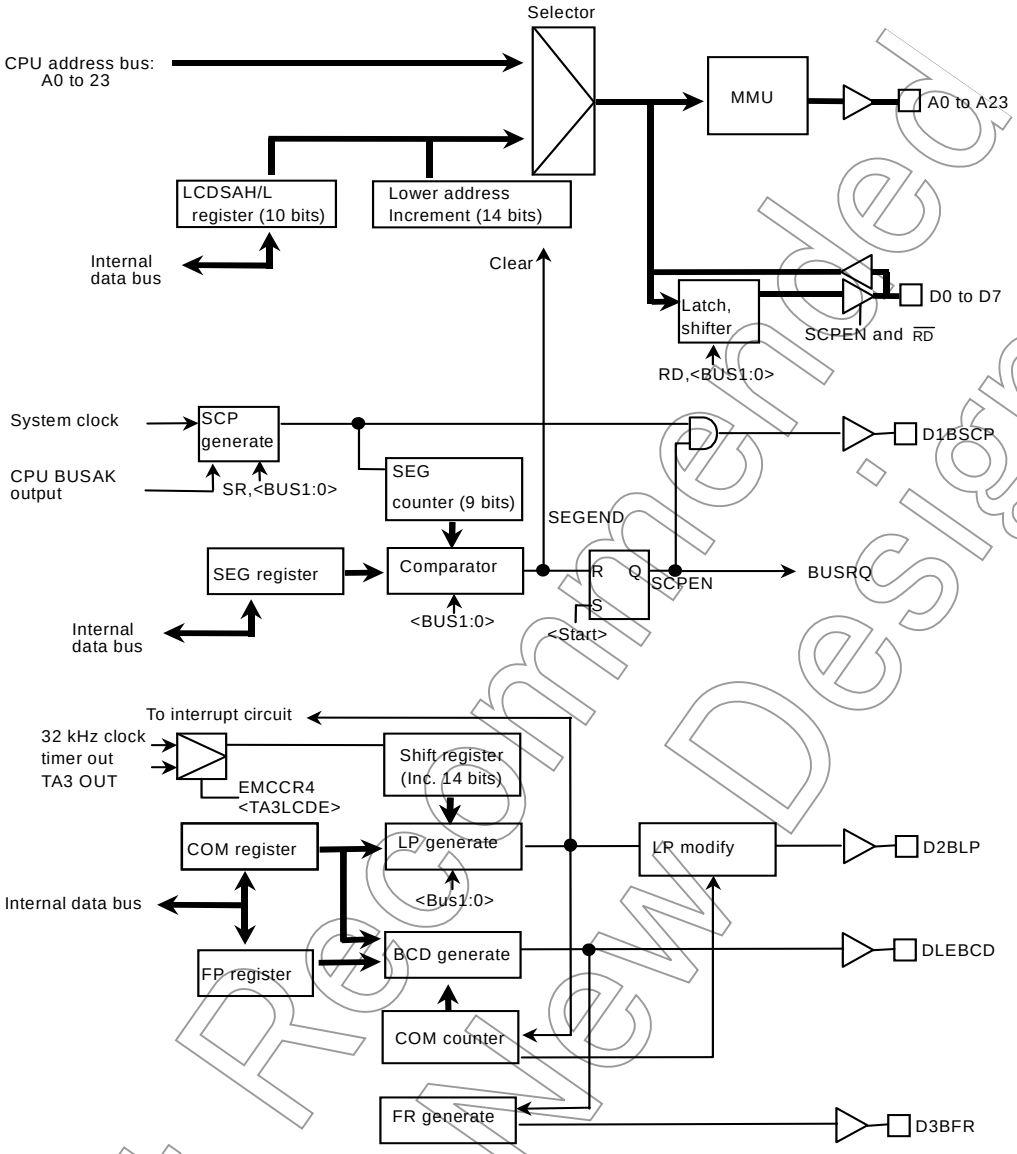


Figure 3.13.1 LCDC Block Diagram

3.13.3 Control Registers

LCDSAL Register

	7	6	5	4	3	2	1	0
Bit symbol	SAL15	SAL14	SAL13	SAL12		–	–	MODE
Read/Write	R/W	R/W	R/W	R/W		R/W	R/W	R/W
After reset	0	0	0	0		0	0	0
Function	SR mode Display memory address (Low: A15 to A12)					Always write 0	Always write 0	Mode select 0: RAM 1: SR

LCDSAL (0360H)

LCDSAH Register

	7	6	5	4	3	2	1	0
Bit symbol	SAL23	SAL22	SAL21	SAL20	SAL19	SAL18	SAL17	SAL16
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	SR mode Display memory address (High: A23 to A16)							

LCDSAH (0361H)

LCDSIZE Register

	7	6	5	4	3	2	1	0
Bit symbol	COM3	COM2	COM1	COM0	SEG3	SEG2	SEG1	SEG0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	LCD common number (SR mode) 0000: 64 0101:128 0001: 68 0110:144 0010: 80 0111:160 0011:100 1000:200 0100:120 1001:240 Other: Reserved				LCD segment number (SR mode) 0000: 32 0101:160 0001: 64 0110:240 0010: 80 0111:320 0011:120 1000:360 0100:128 Other: Reserved			

LCDSIZE (0362H)

Note: Bit mode can not select in 240 common number.

LCDCTL Register

	7	6	5	4	3	2	1	0
Bit symbol	LCDON	-	-	BUS1	BUS0	MMULCD	FP8	START
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	$\overline{DOFF}$ (SR, RAM mode) 0: OFF 1: ON	Always write 0	Always write 0	Data bus width (SR mode) 00: 8 bits (Byte mode) 01: 4 bits (Nibble mode) 10: 1 bit (Bit mode)		Setting direct RAM 0: OFF 1: ON	Setting bit 8 for f_{FP}	Start control (SR mode) 0: Stop 1: Start

LCDCTL (0363H)

Note 1: There is a limitation about to set LCDSAH and LCDSAL start address.
It prohibit to set A13 carry to A14 by all 1-frame data transmit.
Ex.: In case 240 (Row)×360 (Column): 2a30 bytes
Start address of LCDC: SAL15 to SAL12 = 0000 or 0001;

Note 2: Initial incriminator's address (LSB 14 bits) for LCDC DMA is 0000 (Hex).

LCDFFP Register

		7	6	5	4	3	2	1	0
LCDFFP (0364H)	Bit symbol	FP7	FP6	FP5	FP4	FP3	FP2	FP1	FP0
	Read/Write	R/W							
	After reset	0							
	Function	Setting bit 7 to 0 for f_{FP}							

LCDCTR2 Register

		7	6	5	4	3	2	1	0
LCDCTL2 (0366H)	Bit symbol	–	–	–			RAMBUS	AC1	AC0
	Read/Write	R/W	R/W	R/W			R/W	R/W	R/W
	After reset	0	0	0			0	0	0
	Function	Always write to 111 (Note)					0: Byte 1: Word	00: Type A 01: Type B 10: Type C 11: Reserve	

Note: Please write bit<7:5> to 111, even if you use <RAMBUS>,<AC1> and <AC0> as initilal setting.

Figure 3.13.2 LCDC Register

LCDC0L/LCDC0H/LCDC1L/LCDC1H/LCDC2L/LCDC2H/LCDR0L/LCDR0H Register

	7	6	5	4	3	2	1	0
Bit symbol	D7	D6	D5	D4	D3	D2	D1	D0
Read/Write	Depend on the specification of external LCD driver							
After reset	Depend on the specification of external LCD driver							
Function	Depend on the specification of external LCD driver							

These registers do not exist on TMP91C016. These are image for instruction registers and display registers of external RAM built-in sequential access type(Note) LCD driver.

Address as follows is assigned to these registers, and the following chip enable pin becomes active when accesses corresponding address.

And, the area of these address is external area, so $\overline{RD}$, $\overline{WR}$ terminal becomes active by external access.Table 3.13.3 shows the address map in the case of controlling RAM built-in random access type(Note) LCD driver.

This selection is performed by LCDCTL<MMULCD>.

Register	Address	Purpose		Chip Enable Terminal	A0 Terminal
LCDC1L	0FE0H	RAM built-in type column driver 1	Instruction	D1BSCP	0
LCDC1H	0FE1H		Display data		1
LCDC2L	0FE2H	RAM built-in type column driver 2	Instruction	D2BLP	0
LCDC2H	0FE3H		Display data		1
LCDC3L	0FE4H	RAM built-in type column driver 3	Instruction	D3BFR	0
LCDC3H	0FE5H		Display data		1
LCDR1L	0FE6H	RAM built-in type row driver	Instruction	DLEBCD	0
LCDR1H	0FE7H		Display data		1

Figure 3.13.3 Memory Mapping for Built-in RAM Sequential Access Type

Address	Purpose	Chip Enable Terminal
3C0000H to 3CFFFFH	RAM built-in type driver 1	D1BSCP
3D0000H to 3DFFFFH	RAM built-in type driver 2	D2BLP
3E0000H to 3EFFFFH	RAM built-in type driver 3	D3BFR
3F0000H to 3FFFFFH	RAM built-in type driver 4	DLEBCD

Figure 3.13.4 Memory Mapping for Built-in RAM Random Access Type

Note: We call built-in RAM sequential access type LCD driver that use register to access to display ram without address pin.
We call built-in RAM random access type LCD driver that is same method to access to SRAM with address pin.

3.13.4 Operation Explanation of Each Mode

3.13.4.1 Shift-register Type LCD Driver Control Mode (SR mode)

Set the mode of operation, start address of source data save memory and LCD size to control registers before setting start register. After set start register LCDC outputs bus release request to CPU and read data from source memory. After that LCDC transmits data of volume of LCD size to external LCD driver through data bus. At this time, control signals (D1BSCP etc.) connected LCD driver output specified waveform synchronize with data transmission. After finish data transmission, LCDC cancels the bus release request and CPU will restart.

LCD controller uses the clock (LCDCK) different from f_{SYS} to make D3BFR, DLEBCD and D2BLP signal.

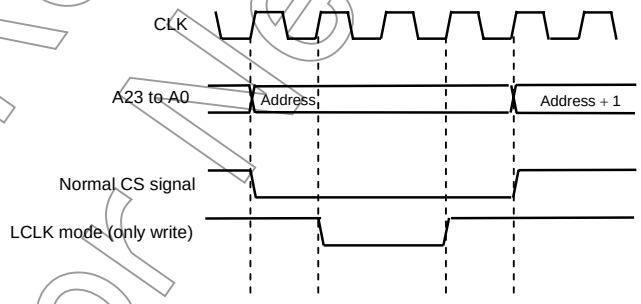
LCDCK can be selected from the low frequency oscillator (fs: 32.768kHz) or timer out (TA3OUT) outputs from internal 8bit timer circuit (TMRA23) by EMCCR0<TA3LCDE>. After reset, this bit is cleared to "0" and low frequency oscillator is selected.

LCDC timing figure in the case of 240 seg × 120 com and BYTE mode is shown in Figure 3.13.6, Figure 3.13.7.

The table of t_{LP} (D2BLP pin cycle) by the number of segments and the common number and CPU stop timer (t_{STOP})/stop ratio are shown in Table 3.13.2 and f_{FP} (Frame frequency) by the common number is shown in Table 3.13.3 and Table 3.13.4.

The example of a 240 seg × 120 com LCD connection circuit is shown Figure 3.13.8.

The circuit that can correspond without especially adding an external circuit outside is built into even when the command for LCDD is written (Read is prohibited). Please refer to Figure 3.13.5. When these signals are outputted from CS0, set P63FC3<P60F3>, and when these signals are outputted from CS2C, set P6FC3<P65F3>. Please refer the section of "Port 6".



Note: When LCLK mode selected, CS signal out OR gate (Original CS signal and $\overline{WR}$ / $\overline{HWR}$ signal). CS signal is not ouptut when read.

3.13.4.2 Settlement to frame frequency function

TMP91C016 defines so-called frame period (Refresh interval for LCD panel) by the value set in f_{FP} [8:0]. DLEBCD pin outputs pulse every frame period. D3BFR pin usually outputs the signal inverts polarity every frame period.

Basic frame period; DLEBCD signal, is made according to the resister f_{FP} [8:0] setting mentioned before. However this f_{FP} [8:0] setting is generally equal to common number, frame period can be corrected by increasing f_{FP} [8:0] with ease.

The equation can calculate frame period.

Frame period = LCDCK/(D × f_{FP}) [Hz] D: constant for each common (Table 3.13.3)
FFP: setting of f_{FP} [8:0] resister
LCDCK: source clock of LCD
(Low clock is usually selected)

Please select the value of f_{FP} [8:0] as the frame period you want to set in the Table 3.13.3.

Note: Please make the value set to f_{FP} [8:0] into the following range.
COM(common number) ≤ f_{FP} ≤ 320

Example1: In the case where frame period is set to 72.10 Hz by 240 coms.
f_{FP} = 240 (COM) + 63 = 303 = 12FH (by Table 3.13.3)
Therefore, LCDCTL<FP8> = 1 and LCDFFP<FP7:0> = 2FH are set up.

LCDCTL Register

		7	6	5	4	3	2	1	0
LCDCTL (0363H)	Bit symbol	LCDON	–	–	BUS1	BUS0	MMULCD	FP8	START
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	0	0
	Function	$\overline{DOFF}$ (SR, RAM mode) 0: OFF 1: ON	Always write 0	Always write 0	Data bus width (SR mode) 00: 8 bits (Byte mode) 01: 4 bits (Nibble mode) 10: 1 bit (Bit mode)		Setting direct RAM 0: OFF 1: ON	Setting bit 8 for f_{FP}	Start control (SR mode) 0: Stop 1: Start

LCDFFP Register

		7	6	5	4	3	2	1	0
LCDFFP (0364H)	Bit symbol	FP7	FP6	FP5	FP4	FP3	FP2	FP1	FP0
	Read/Write	R/W							
	After reset	0							
	Function	Setting bit 7 to 0 for f_{FP}							

3.13.4.3 Timer out LCDCK

LCD source clock (LCDCK) can select low frequency (XT1, XT2: 32.768 [kHz]) or timer out (TA3OUT) outputs from internal TMRA23.

Example2: Here indicates the method that frame period is set 70 [Hz] by selecting TA3OUT for source clock of LCD. (fc = 6 [MHz], 120COM)

The next equation calculates frame period.

Frame period = $1/(t_{LP} \times f_{FP})$ [Hz] t_{LP}: The period of D2BLP

Source clock for LCDC defines as XT [Hz] and then this t_{LP} represents

$t_{LP} = D/XT$ D: the value is 3.5 at 120 COM

Therefore if you set the frame period at 70 [Hz] under 120 COM,

$XT = 120 \times 3.5 \times 70$

$= 29400$ [Hz]

XT should be above value.

In order to make XT = 29400 [Hz] under fc = 6 [MHz] with ϕT1 of timer3,

$1/XT = (TA3REG) \times 2 \times 8/fc$ [s] (TA3REG): the value of timer register in short, $XT = fc/(TA3REG) \times 2 \times 8)$ [Hz]

However (TA3REG) is 12.75 after calculate, it's impossible to set the value under a decimal point.

So if (TA3REG) is set 0CH, XT = 31250 [Hz]. And because of D = 3.5,

Frame period = $31250/(120 \times 3.5)$

$= 74.404$ [Hz]

Further if f_{FP} is 127 (COM + 7) with correction,

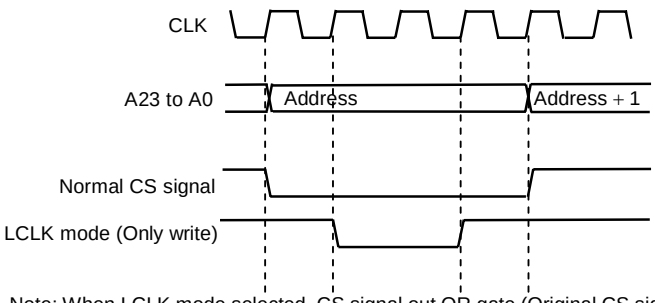
Frame period = $31250/(127 \times 3.5)$

$= 70.30 \ldots$ [Hz]

Reference: To maintain quality for display, please refer to following value for each gray scale.

(You have to use settlement of frame frequency function, frame invert adjustment function and timer out LCDCK.)

Monochrome: Frame period = 70 [Hz]



Note: When LCLK mode selected, CS signal out OR gate (Original CS signal and $\overline{WR}$ / $\overline{HWR}$ signal).
CS signal is not ouptut when read.

Figure 3.13.5 LCLK Mode Timing Chart

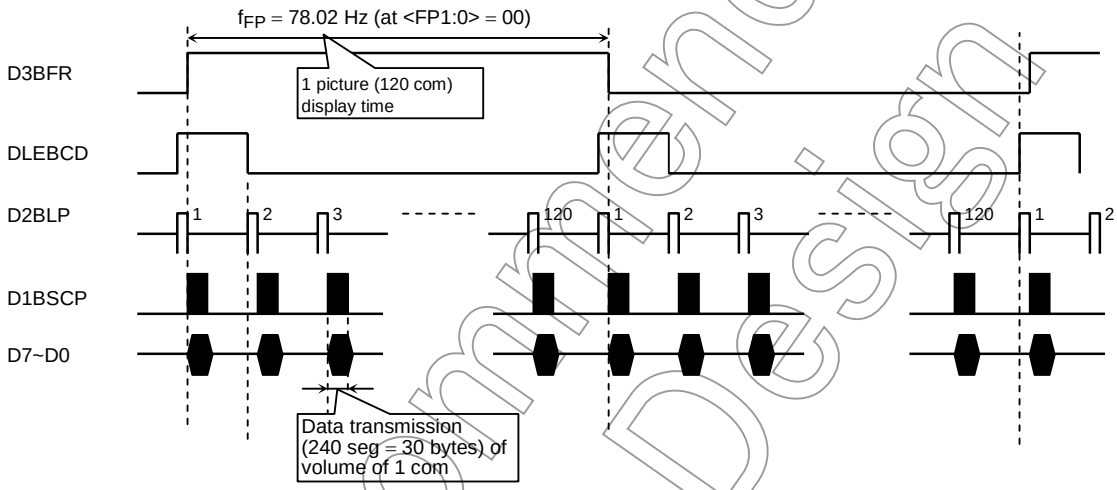


Figure 3.13.6 Timing Diagram for SR Mode

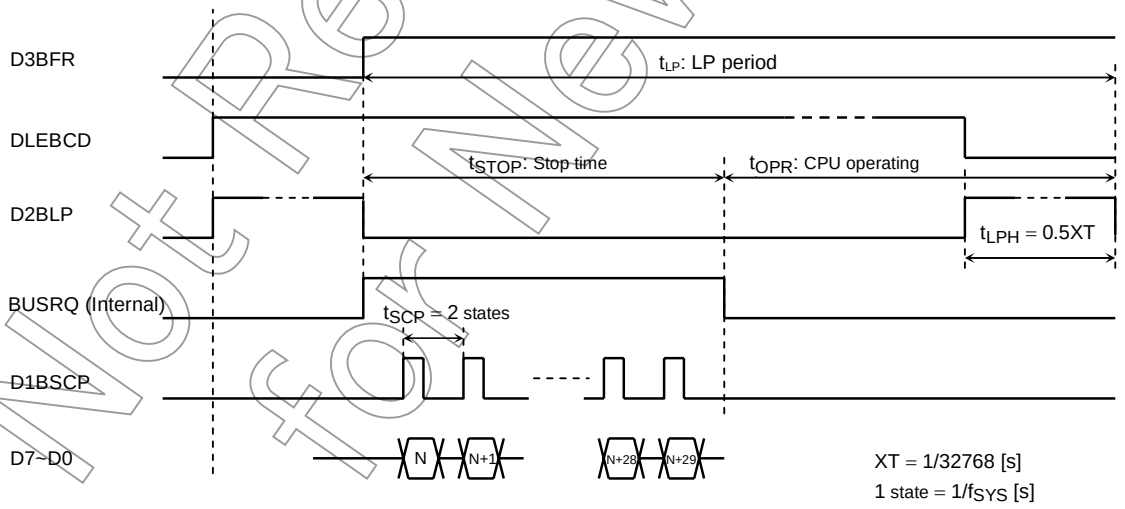


Figure 3.13.7 Timing Diagram for SR Mode (Detail)

Table 3.13.2 Performance Listing for Each Segment and Common Number

		64 com	68 com	80 com	100 com	120 com	128 com	144 com	160 com	200 com	240 com	Unit
XT number of counts for t_{LP} making: D		6.5	6.0	5.0	4.0	3.5	3.0	2.5	2.5	2.0	1.5	–
T_{LP}		198.4	183.1	152.6	122.1	106.8	91.6	76.3	76.3	61.0	45.8	μs
32 seg	T_{STOP}	0.6										μs
	CPU stop rate	0.3	0.3	0.4	0.5	0.6	0.6	0.8	0.8	1.0	1.3	%
64 seg	T_{STOP}	1.2										μs
	CPU stop rate	0.6	0.6	0.8	1.0	1.1	1.3	1.6	1.6	1.9	2.6	%
80 seg	T_{STOP}	1.5										μs
	CPU stop rate	0.7	0.8	1.0	1.2	1.4	1.6	1.9	1.9	2.4	3.2	%
120 seg	T_{STOP}	2.2										μs
	CPU stop rate	1.1	1.2	1.5	1.8	2.1	2.4	2.9	2.9	3.6	4.9	%
128 seg	T_{STOP}	2.4										μs
	CPU stop rate	1.2	1.3	1.6	1.9	2.2	2.6	3.1	3.1	3.9	4.9	%
160 seg	T_{STOP}	3.0										μs
	CPU stop rate	1.5	1.6	1.9	2.4	2.8	3.2	3.9	3.9	4.9	6.5	%
240 seg	T_{STOP}	4.4										μs
	CPU stop rate	2.2	2.4	2.9	3.6	4.2	4.9	5.8	5.8	7.3	9.7	%
320 seg	T_{STOP}	5.9										μs
	CPU stop rate	3.0	3.2	3.9	4.9	5.5	6.5	7.8	7.8	9.7	12.9	%
360 seg	T_{STOP}	6.7										μs
	CPU stop rate	3.4	3.6	4.4	5.5	6.2	7.3	8.7	8.7	10.9	14.6	%

Note 1: The above time distance are value which used f_{FPH} = 27 [MHz], f_S = 32.768 [kHz].

Note 2: CPU stop time t_{STOP}: A value is value when reading a transmitting memory by 0 waits in the byte write/byte read mode. The value becomes × 1.5 in Nibble write mode and × 4.5 in Bit write mode. Details, see the “state/cycle” is each type timing table.
The time required to the transmission start accompanied by bus opening demand is not included in the above-mentioned numerical value.

Note 3: t_{LP} can be calculated in the following formulas.
t_{LP} = D/32768 [s]
(Example) In case of 240 com, t_{LP} = 1.5/32768 = 45.8 [μs] because of D = 1.5

Table 3.13.3 f_{FP} Table for Each Common Number (1/2)

D	6.5	6.0	5.0	4.0	3.5	3.0	2.5	2.5	2.0	1.5
COM	64	68	80	100	120	128	144	160	200	240
COM + 0	78.77	80.31	81.92	81.92	78.02	85.33	91.02	81.92	81.92	91.02
COM + 1	77.56	79.15	80.91	81.11	77.37	84.67	90.39	81.41	81.51	90.64
COM	76.38	78.02	79.92	80.31	76.74	84.02	89.78	80.91	81.11	90.27
COM	75.24	76.92	78.96	79.53	76.12	83.38	89.16	80.41	80.71	89.90
COM	74.14	75.85	78.02	78.77	75.50	82.75	88.56	79.92	80.31	89.53
COM	73.06	74.81	77.10	78.02	74.90	82.13	87.97	79.44	79.92	89.16
COM	72.02	73.80	76.20	77.28	74.30	81.51	87.38	78.96	79.53	88.80
COM	71.00	72.82	75.33	76.56	73.72	80.91	86.80	78.49	79.15	88.44
COM	70.02	71.86	74.47	75.85	73.14	80.31	86.23	78.02	78.77	88.09
COM	69.06	70.93	73.64	75.16	72.58	79.73	85.67	77.56	78.39	87.73
COM + 10	68.12	70.02	72.82	74.47	72.02	79.15	85.11	77.10	78.02	87.38
COM	67.22	69.13	72.02	73.80	71.47	78.58	84.56	76.65	77.65	87.03
COM	66.33	68.27	71.23	73.14	70.93	78.02	84.02	76.20	77.28	86.69
COM	65.47	67.42	70.47	72.50	70.39	77.47	83.49	75.76	76.92	86.35
COM	64.63	66.60	69.72	71.86	69.87	76.92	82.96	75.33	76.56	86.01
COM	63.81	65.80	68.99	71.23	69.35	76.38	82.44	74.90	76.20	85.67
COM	63.02	65.02	68.27	70.62	68.84	75.85	81.92	74.47	75.85	85.33
COM	62.24	64.25	67.56	70.02	68.34	75.33	81.41	74.05	75.50	85.00
COM	61.48	63.50	66.87	69.42	67.84	74.81	80.91	73.64	75.16	84.67
COM	60.74	62.77	66.20	68.84	67.35	74.30	80.41	73.22	74.81	84.34
COM + 20	60.01	62.06	65.54	68.27	66.87	73.80	79.92	72.82	74.47	84.02
COM	59.31	61.36	64.89	67.70	66.40	73.31	79.44	72.42	74.14	83.70
COM	58.62	60.68	64.25	67.15	65.93	72.82	78.96	72.02	73.80	83.38
COM	57.95	60.01	63.63	66.60	65.47	72.34	78.49	71.62	73.47	83.06
COM	57.29	59.36	63.02	66.06	65.02	71.86	78.02	71.23	73.14	82.75
COM	56.64	58.72	62.42	65.54	64.57	71.39	77.56	70.85	72.82	82.44
COM	56.01	58.10	61.83	65.02	64.13	70.93	77.10	70.47	72.50	82.13
COM	55.40	57.49	61.25	64.50	63.69	70.47	76.65	70.09	72.18	81.82
COM	54.80	56.89	60.68	64.00	63.26	70.02	76.20	69.72	71.86	81.51
COM	54.21	56.30	60.12	63.50	62.83	69.57	75.76	69.35	71.55	81.21
COM + 30	53.63	55.73	59.58	63.02	62.42	69.13	75.33	68.99	71.23	80.91
COM	53.07	55.16	59.04	62.53	62.00	68.70	74.90	68.62	70.93	80.61
COM	52.51	54.61	58.51	62.06	61.59	68.27	74.47	68.27	70.62	80.31
COM	51.97	54.07	58.00	61.59	61.19	67.84	74.05	67.91	70.32	80.02
COM	51.44	53.54	57.49	61.13	60.79	67.42	73.64	67.56	70.02	79.73
COM	50.92	53.02	56.99	60.68	60.40	67.01	73.22	67.22	69.72	79.44
COM	50.41	52.51	56.50	60.24	60.01	66.60	72.82	66.87	69.42	79.15
COM	49.91	52.01	56.01	59.80	59.63	66.20	72.42	66.53	69.13	78.86
COM	49.42	51.52	55.54	59.36	59.25	65.80	72.02	66.20	68.84	78.58
COM + 39	48.94	51.04	55.07	58.94	58.88	65.41	71.62	65.87	68.55	78.30

Note: f_{FP} can be calculated in the following formulas.

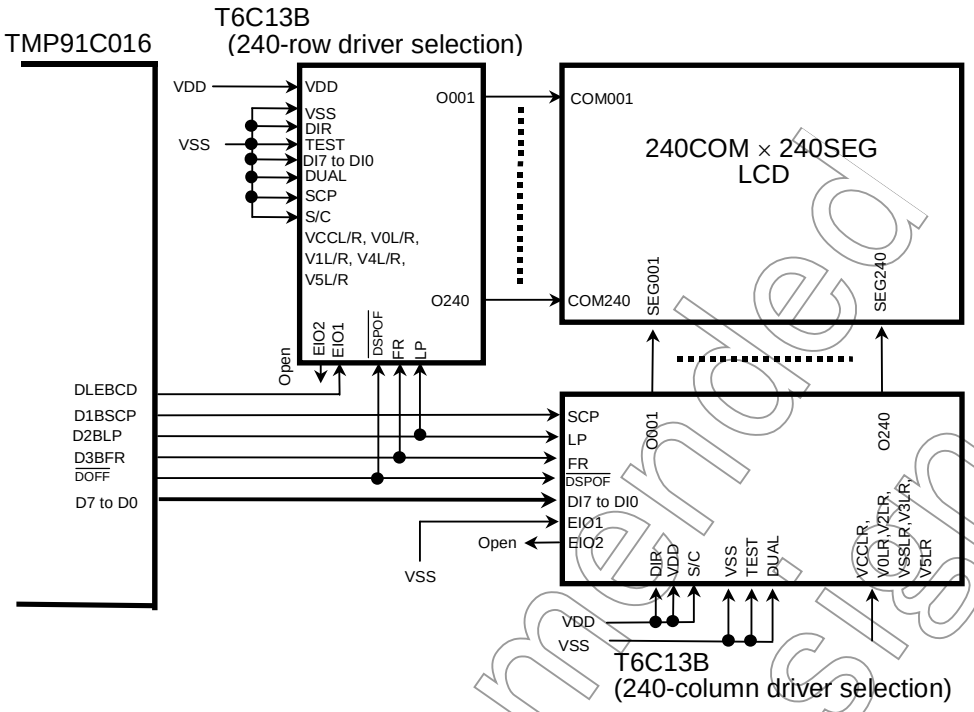
$f_{FP} = 32768/(D \times FP)$ [Hz]

(Ex) In case of 120 com, <FP8:0> = 131,

$f_{FP} = 32768/(3.5 \times 131) = 71.5$ [Hz]

Table 3.13.4 f_{FP} Table for Each Common Number (2/2)

D	6.5	6.0	5.0	4.0	3.5	3.0	2.5	2.5	2.0	1.5
COM	64	68	80	100	120	128	144	160	200	240
COM + 40	48.47	50.57	54.61	58.51	58.51	65.02	71.23	65.54	68.27	78.02
COM	48.01	50.10	54.16	58.10	58.15	64.63	70.85	65.21	67.98	77.74
COM	47.56	49.65	53.72	57.69	57.79	64.25	70.47	64.89	67.70	77.47
COM	47.11	49.20	53.28	57.29	57.44	63.88	70.09	64.57	67.42	77.19
COM	46.68	48.76	52.85	56.89	57.09	63.50	69.72	64.25	67.15	76.92
COM	46.25	48.33	52.43	56.50	56.74	63.14	69.35	63.94	66.87	76.65
COM	45.83	47.91	52.01	56.11	56.40	62.77	68.99	63.63	66.60	76.38
COM	45.42	47.49	51.60	55.73	56.06	62.42	68.62	63.32	66.33	76.12
COM	45.01	47.08	51.20	55.35	55.73	62.06	68.27	63.02	66.06	75.85
COM	44.61	46.68	50.80	54.98	55.40	61.71	67.91	62.71	65.80	75.59
COM + 50	44.22	46.28	50.41	54.61	55.07	61.36	67.56	62.42	65.54	75.33
COM	43.84	45.89	50.03	54.25	54.75	61.02	67.22	62.12	65.27	75.07
COM	43.46	45.51	49.65	53.89	54.43	60.68	66.87	61.83	65.02	74.81
COM	43.09	45.13	49.28	53.54	54.12	60.35	66.53	61.54	64.76	74.56
COM	42.72	44.77	48.91	53.19	53.81	60.01	66.20	61.25	64.50	74.30
COM	42.36	44.40	48.55	52.85	53.50	59.69	65.87	60.96	64.25	74.05
COM	42.01	44.04	48.19	52.51	53.19	59.36	65.54	60.68	64.00	73.80
COM	41.66	43.69	47.84	52.18	52.89	59.04	65.21	60.40	63.75	73.55
COM	41.32	43.34	47.49	51.85	52.60	58.72	64.89	60.12	63.50	73.31
COM	40.99	43.00	47.15	51.52	52.30	58.41	64.57	59.85	63.26	73.06
COM + 60	40.66	42.67	46.81	51.20	52.01	58.10	64.25	59.58	63.02	72.82
COM	40.33	42.34	46.48	50.88	51.73	57.79	63.94	59.31	62.77	72.58
COM	40.01	42.01	46.15	50.57	51.44	57.49	63.63	59.04	62.53	72.34
COM	39.69	41.69	45.83	50.26	51.16	57.19	63.32	58.78	62.30	72.10
COM	39.38	41.37	45.51	49.95	50.88	56.89	63.02	58.51	62.06	71.86
COM	39.08	41.06	45.20	49.65	50.61	56.59	62.71	58.25	61.83	71.62
COM	38.78	40.76	44.89	49.35	50.33	56.30	62.42	58.00	61.59	71.39
COM	38.48	40.45	44.58	49.05	50.07	56.01	62.12	57.74	61.36	71.16
COM	38.19	40.16	44.28	48.76	49.80	55.73	61.83	57.49	61.13	70.93
COM	37.90	39.86	43.98	48.47	49.54	55.45	61.54	57.24	60.91	70.70
COM + 70	37.62	39.57	43.69	48.19	49.28	55.16	61.25	56.99	60.68	70.47
COM	37.34	39.29	43.40	47.91	49.02	54.89	60.96	56.74	60.46	70.24
COM	37.07	39.01	43.12	47.63	48.76	54.61	60.68	56.50	60.24	70.02
COM	36.80	38.73	42.83	47.35	48.51	54.34	60.40	56.25	60.01	69.79
COM	36.53	38.46	42.56	47.08	48.26	54.07	60.12	56.01	59.80	69.57
COM	36.27	38.19	42.28	46.81	48.01	53.81	59.85	55.78	59.58	69.35
COM	36.01	37.93	42.01	46.55	47.77	53.54	59.58	55.54	59.36	69.13
COM	35.75	37.66	41.74	46.28	47.52	53.28	59.31	55.30	59.15	68.91
COM	35.50	37.41	41.48	46.02	47.28	53.02	59.04	55.07	58.94	68.70
COM	35.25	37.15	41.22	45.77	47.05	52.77	58.78	54.84	58.72	68.48
COM + 80	35.01	36.90	40.96	45.51	46.81	52.51	58.51	54.61	58.51	68.27



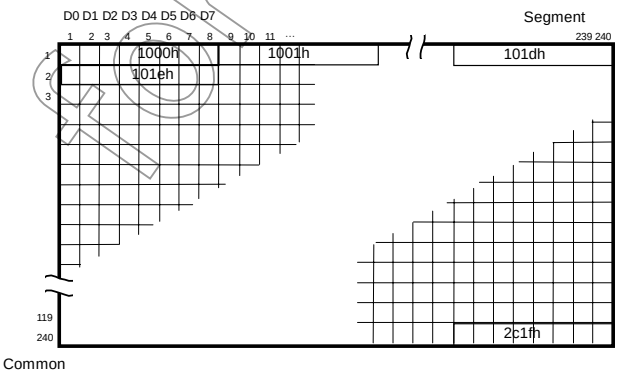
Note: Other circuit is necessary for LCD drive power supply for LCD driver display.

Figure 3.13.8 Interface Example for Shift Register Type LCD Driver

a. Setting example: In case of use 240 seg × 240 com, 8-bit bus width LCD driver.

In case of store 7200 bytes transfer data to LCD driver in built-in RAM (1000H to 2c1FH).

```
LD      (PDCR), 1FH      ; Setting control terminal
LD      (LCDSAL), 11H    ; Select SR mode
LD      (LCDSAH), 00H    ; Source start address = 1000H
LD      (LCDSIZE), 96H   ; 240 seg × 240 com
LD      (LCDFFP), 308    ; fFP = 70.93 Hz
LD      (LCDCTL), 81H    ; Byte mode FP = 70.93 Hz,
                         ; LCDON, Transfer start
```



Relation display panel and display memory (in case of above setting)

b. Transfer time by data bus width

Data bus width of LCD driver can be selected either of byte/nibble/bit by LCDCTL<BUS1:0>. And that cycle is selectable, type A, type B and type C. Each type have each timing, for detail, look for timing table.

Readout bus width of source is selectable 8 bits or 16 bits, without concern to bus width of LCD driver.

WAIT number of the read cycle is 0 waits in case of built-in RAM and works by setting value of CS/WAIT controller in case of external RAM.

c. LCDC operation in HALT mode

When LCDC is working, CPU executes HALT instruction and changes in HALT mode, LCDC continue operation if CPU in IDLE2 mode. But LCDC stops in case of IDLE1, STOP mode.

Note: It need to set the same bus width setting of display RAM, CS/WAIT controller and LCDCTL2<RAMBUS>.

Table 3.13.5 Timing Table Each Type

Read Bus Width	Type	Write Mode	Set Up Time	Hold Time	D1BSCP Pulse Width	D1BSCP Cycle	State/Cycle
Byte	A	Byte	0.5x	1.0x	1.5x	4.0x	4.0x
		Nibble	0.5x	1.0x	1.0x	2.0x	6.0x
		Bit	0.5x	1.0x	1.0x	2.0x	18.0x
	B	Byte	1.0x	0.5x	2.0x	4.0x	4.0x
		Nibble	1.0x	0.5x	1.0x	2.0x	6.0x
		Bit	1.0x	0.5x	1.0x	2.0x	18.0x
	C	Byte	1.0x	2.5x	1.5x	6.0x	6.0x
		Nibble	1.0x	1.5x	2.5x	5.0x	10.0x
		Bit	1.0x	1.0x	1.0x	2.0x	20.0x
Word	A	Byte	0.5x	1.0x	1.0x	2.0x	6.0x
		Nibble	0.5x	1.0x	1.0x	2.0x	10.0x
		Bit	No support. Please use byte read mode				
	B	Byte	1.0x	0.5x	1.0x	2.0x	6.0x
		Nibble	1.0x	0.5x	1.0x	2.0x	10.0x
		Bit	No support. Please use byte read mode				
	C	Byte	1.0x	1.5x	1.5x	3.0x	8.0x
		Nibble	1.0x	1.5x	2.5x	5.0x	20.0x
		Bit	No support. Please use byte read mode				

Note: Number in above table shows f_{FPH} clock cycle, for example, in case of 27 MHz frequency Xin-Xout, 1.00 equal 37 ns.

Above table doesn't show to guarantee the time, it shows outline. For details, look for AC TIMING at after page.

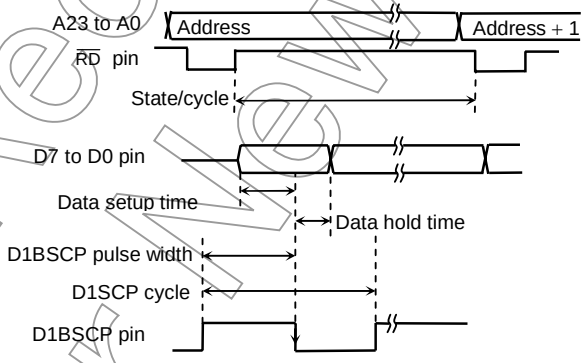
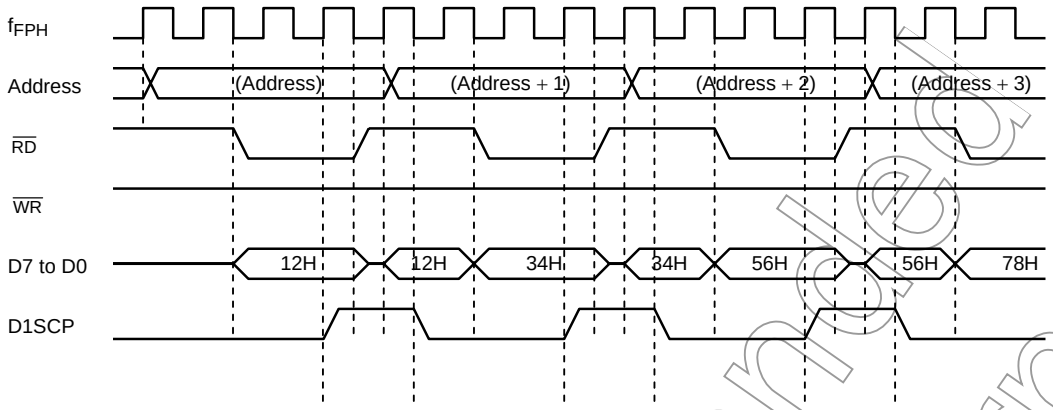
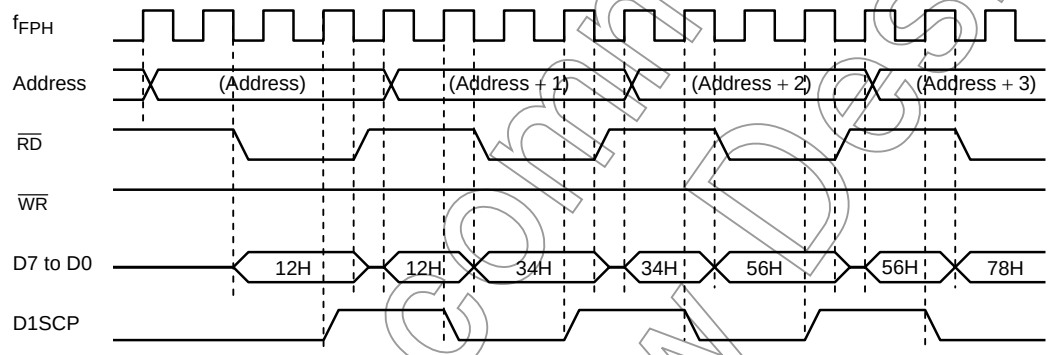


Figure 3.13.9 Definition of Specification

Type	Read	Trance	RAM
A	8 bits	8 bits	SRAM



Type	Read	Trance	RAM
B	8 bits	8 bits	SRAM



Type	Read	Trance	RAM
C	8 bits	8 bits	SRAM

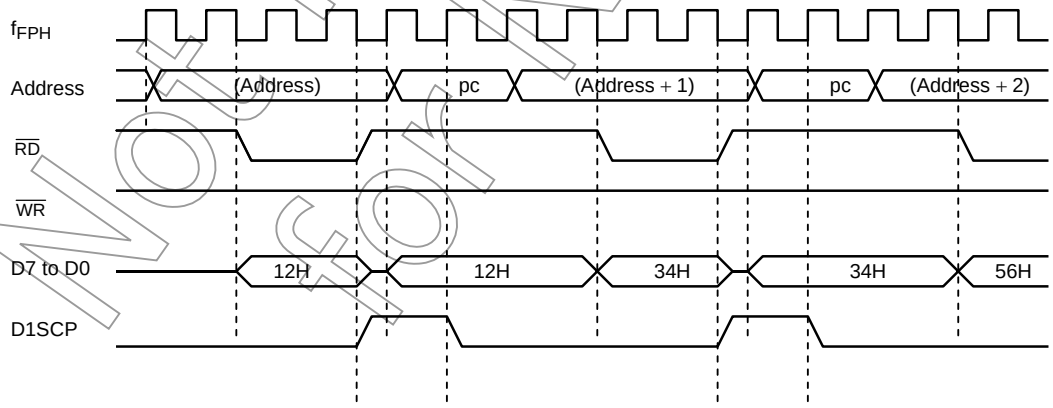
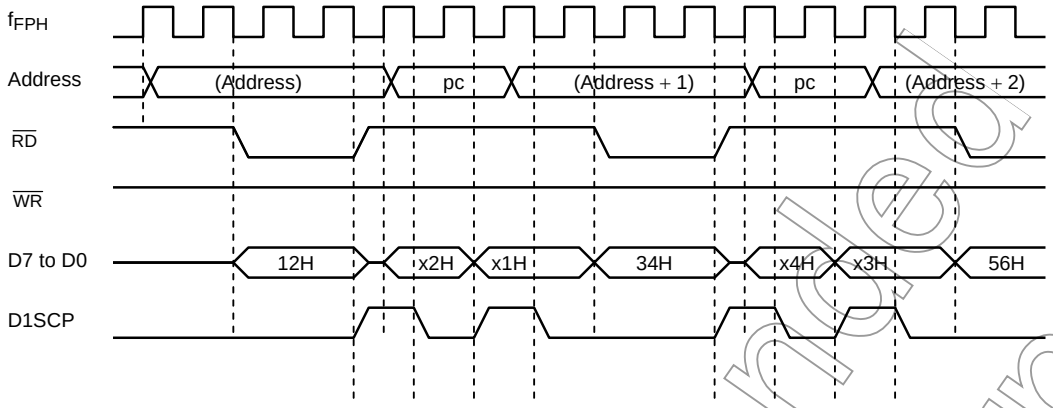
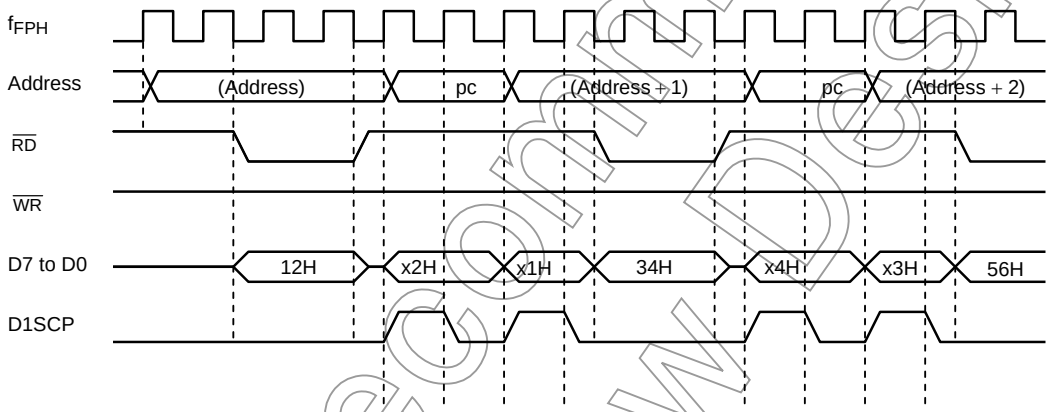


Figure 3.13.10 Byte Read from RAM and Byte Write to LCDD

Type	Read	Trance	RAM
A	8 bits	4 bits	SRAM



Type	Read	Trance	RAM
B	8 bits	4 bits	SRAM



Type	Read	Trance	RAM
C	8 bits	4 bits	SRAM

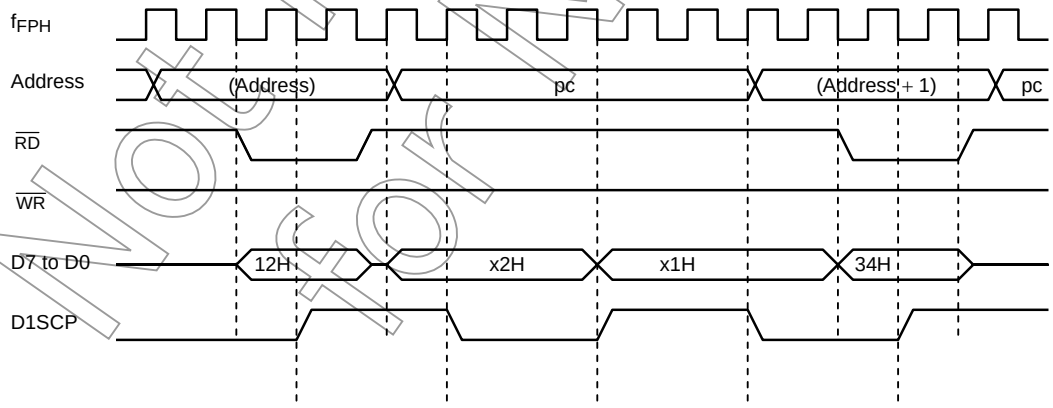
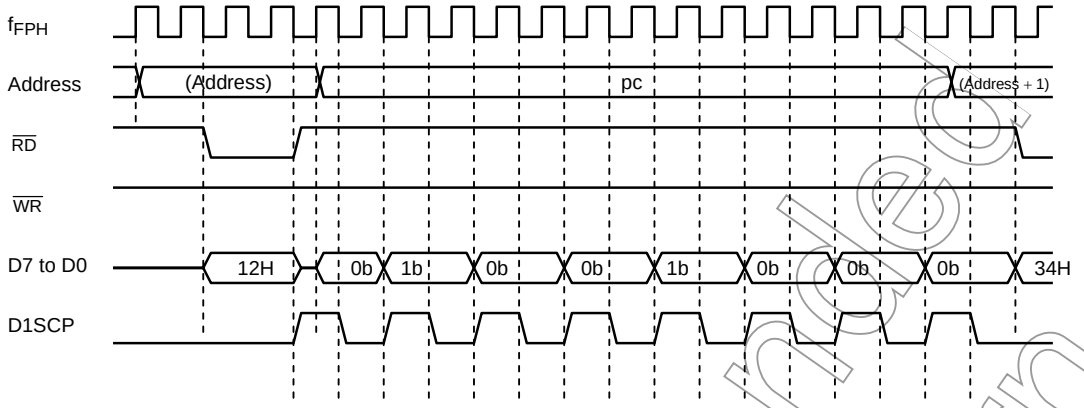
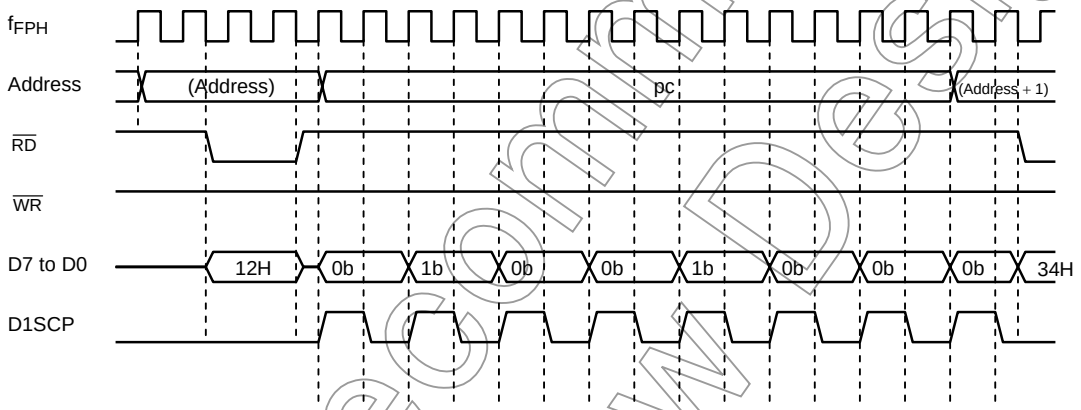


Figure 3.13.11 Byte Read from RAM and Nibble Write to LCDD

Type	Read	Trance	RAM
A	8 bits	1 bit	SRAM



Type	Read	Trance	RAM
B	8 bits	1 bit	SRAM



Type	Read	Trance	RAM
C	8 bits	1 bit	SRAM

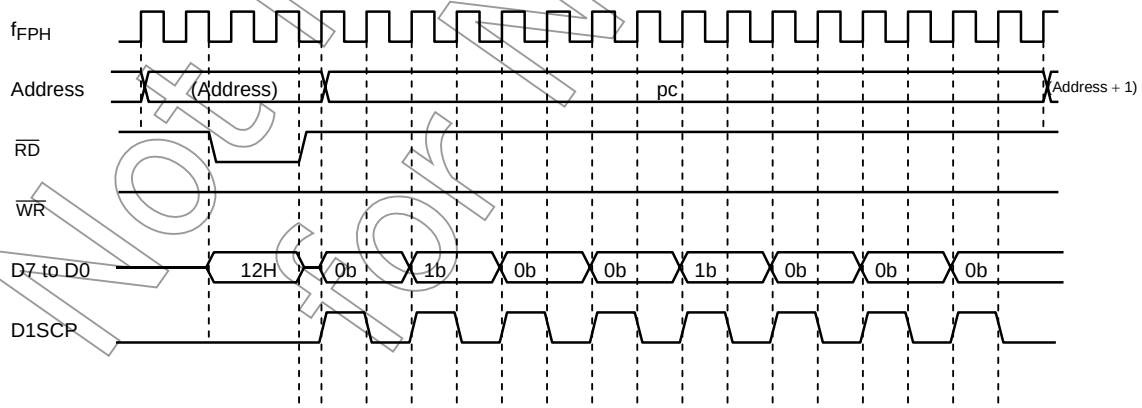
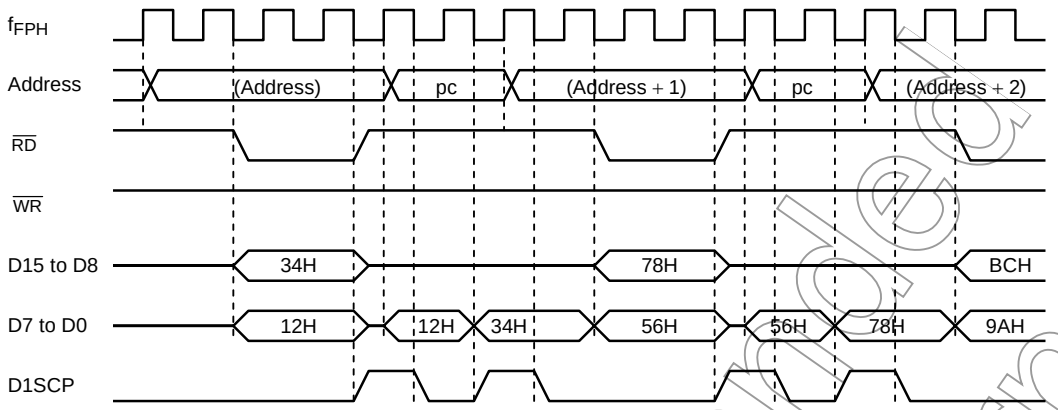
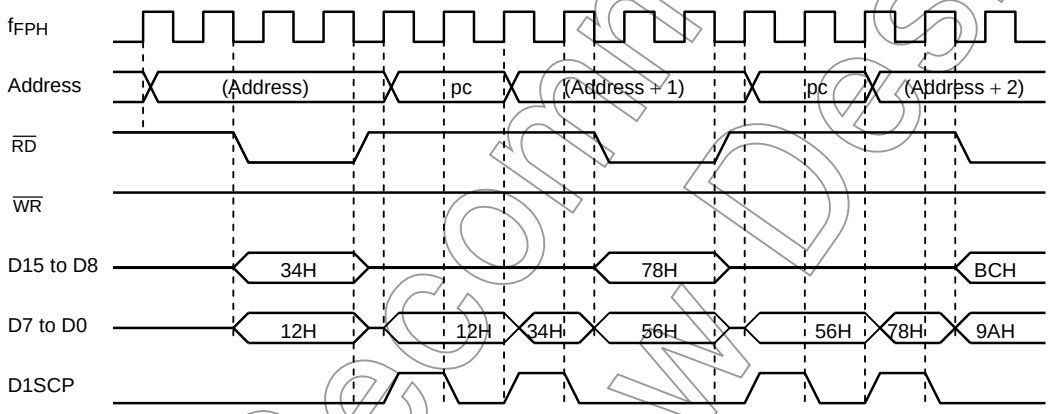


Figure 3.13.12 Byte Read from RAM and Bit Write to LCDD

Type	Read	Trance	RAM
A	16 bits	8 bits	SRAM



Type	Read	Trance	RAM
B	16 bits	8 bits	SRAM



Type	Read	Trance	RAM
C	16 bits	8 bits	SRAM

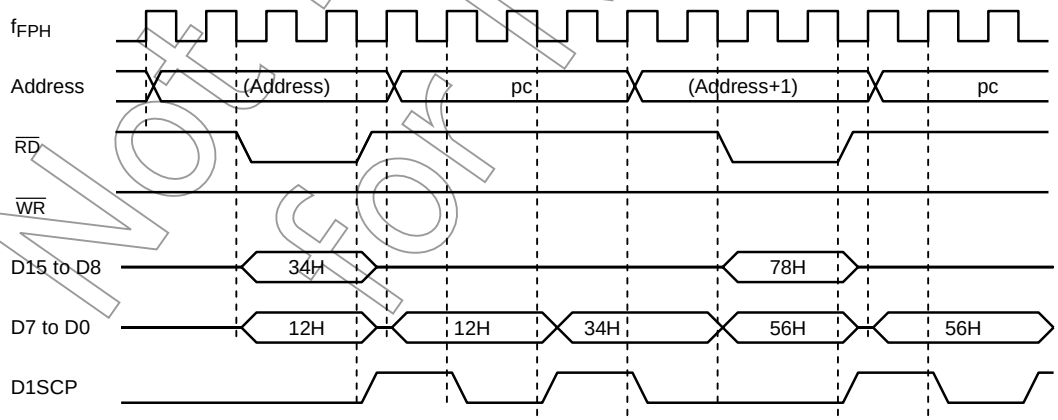


Figure 3.13.13 Word Read from RAM and Byte Write to LCDD

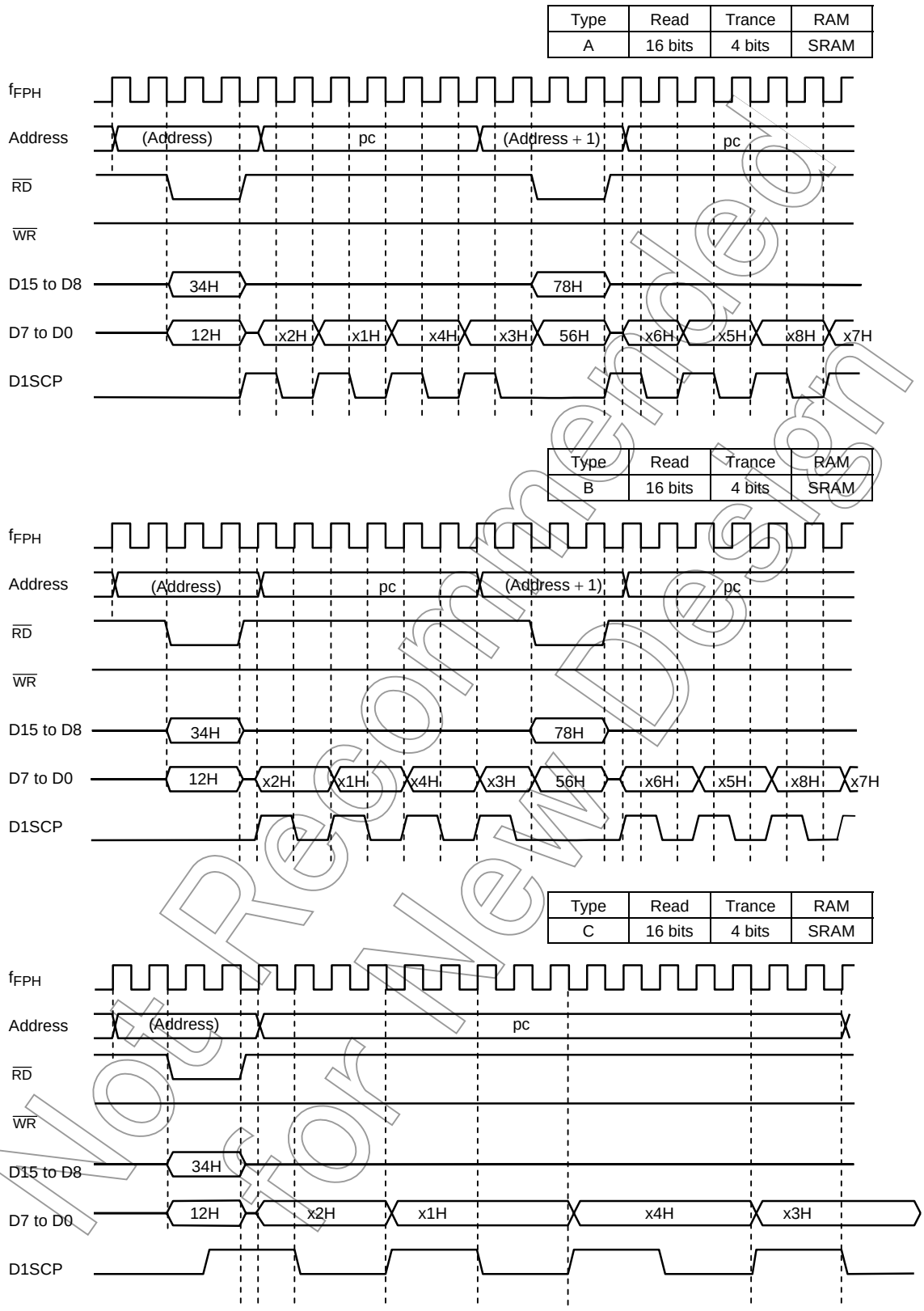


Figure 3.13.14 Word Read from RAM and Nibble Write to LCDD

3.13.4.4 RAM Built-in Type LCD Driver Control Mode (RAM mode)

Data transmission to LCD driver is executed by move instruction of CPU.

After setting mode of operation to control register, when move instruction of CPU is executed LCDC outputs chip select signal to LCD driver connected to the outside from control pin (D1BSCP etc.). Therefore control of data transmission numbers corresponding to LCD size is controlled by instruction of CPU. There are 2 kinds of addresses of LCD driver in this case, and which is chosen determines by LCDCTL <MMULCD> register.

It corresponds to LCD driver which has every 1 byte of instruction register and display data register in LCD driver at the time of <MMULCD> = 0. Please make the transmission place address at this time into either of FE0H to FE7FH. (Figure 3.13.2 references)

It corresponds to address direct writing type LCD driver at the time of <MMULCD> = 1.

The transmission place address at this time can also assign the memory area of 3C0000H to 3FFFFF to four area for every 64 Kbytes. (Figure 3.13.2 references)

The example of a setting is shown as follows and connection example is shown in Figure 3.13.6 (1) at the time below. [<MMULCD> = 0]

a. Setting example: In case of use 80 seg × 65 com LCD driver.

Assign external column driver to LCDC0 and row driver to LCDR0.

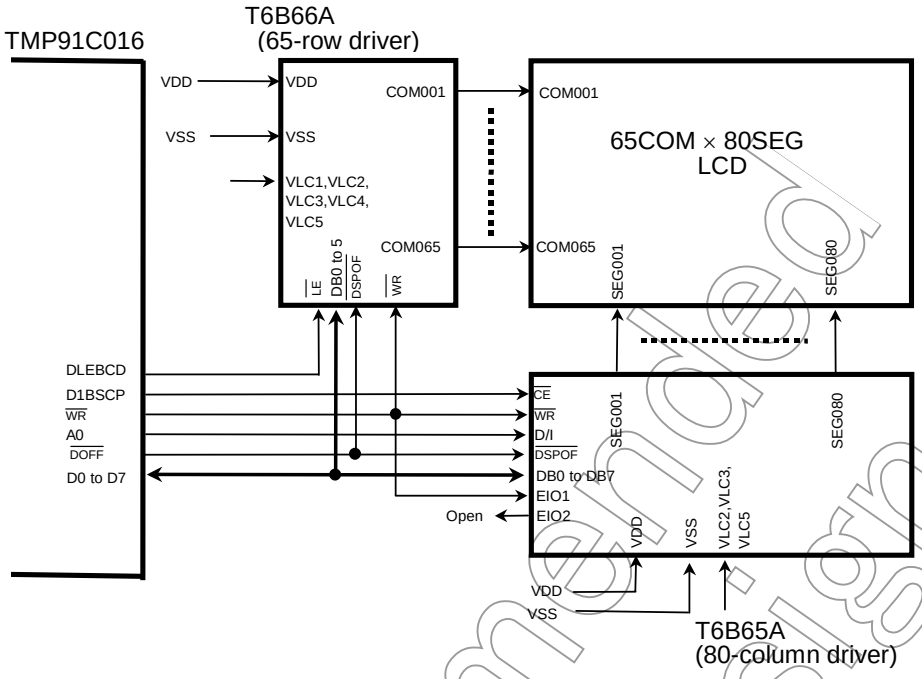
This example used LD instruction in setting of instruction and used burst function of micro DMA by soft start in setting of display data.

In case of store 650 bytes transfer data to LCD driver in built-in RAM (1000H to 1289H).

```
; Setting external terminal
   LD      (PDCR), 19H       ; CE for LCDC0: D1BSCP,
                             ; LE for LCDR0: DLEBCD,
                             ; Setting for DOFF
                             ; Setting for LCDC
   LD      (LCDSAL), 00H     ; Select RAM mode
   LD      (LCDCTL), 80H     ; LCDON

; Setting for mode of LCDC0/LCDR0
   LD      (LCDC0L), XX      ; Setting instruction for LCDC0
   LD      (LCDR0L), XX      ; Setting instruction for LCDR0

; Setting for micro DMA and INTTC (ch0)
   LD      A, 08H            ; Source address INC mode
   LDC     DMAM0, A          ;
   LD      WA,650            ; Count = 650
   LDC     DMAC0, WA         ;
   LD      XWA, 1000H        ; Source address = 1000H
   LDC     DMAS0, XWA        ;
   LD      XWA, 0FE1H        ; Destination address = FE1H (LCDC0H)
   LDC     DMAD0, XWA        ;
   LD      (INTETC01), 06H   ; INTTC0 level = 6
   EI      6                 ;
   LD      (DMAB), 01H       ; Burst  mode
   LD      (DMAR), 01H       ; Soft start
```



Note: Other circuit is necessary for LCD drive power supply for LCD driver display.

Figure 3.13.15 Interface Example for RAM Built-in Type LCD Driver

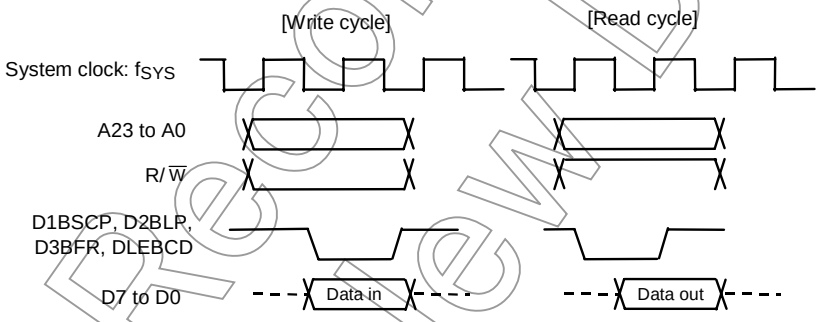


Figure 3.13.16 Example of Access Timing for RAM Built-in Type LCD Driver (Wait = 0)

3.14 Melody/Alarm Generator (MLD)

TMP91C016 incorporates melody function and alarm function, both of which are output from the MLDALM pin. Five 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 (4 Hz to 5461 Hz) based on low-speed clock (32.768 kHz) 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 eight kinds of alarm waveform having a modulation frequency (4096 Hz) determined by the low-speed clock (32.768 kHz). 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 five kinds of fixed cycle (1 Hz, 2 Hz, 64 Hz, 512 Hz, 8192 Hz) interrupts are generated by the free-run counter which is used for alarm generator.

- Special mode

  It is assigned <TA3LCDE> at bit0 and <TA3MLDE> at bit1, of EMCCR4 register (00E7hex). These bits are used when you want to operate LCDD and MELODY circuit without low frequency clock (XTIN, XTOUT). After reset these two bits set to 0 and low clock is supplied each LCDD and MELODY circuit. If you write these bits to 1, TA3 (Generate by timer 3) is supplied each LCDD and MELODY circuit. In this case, you should set 32 kHz timer 3 frequency. For detail, look AC specification characteristics.

This section is constituted as follows.

3.14.1 Block Diagram

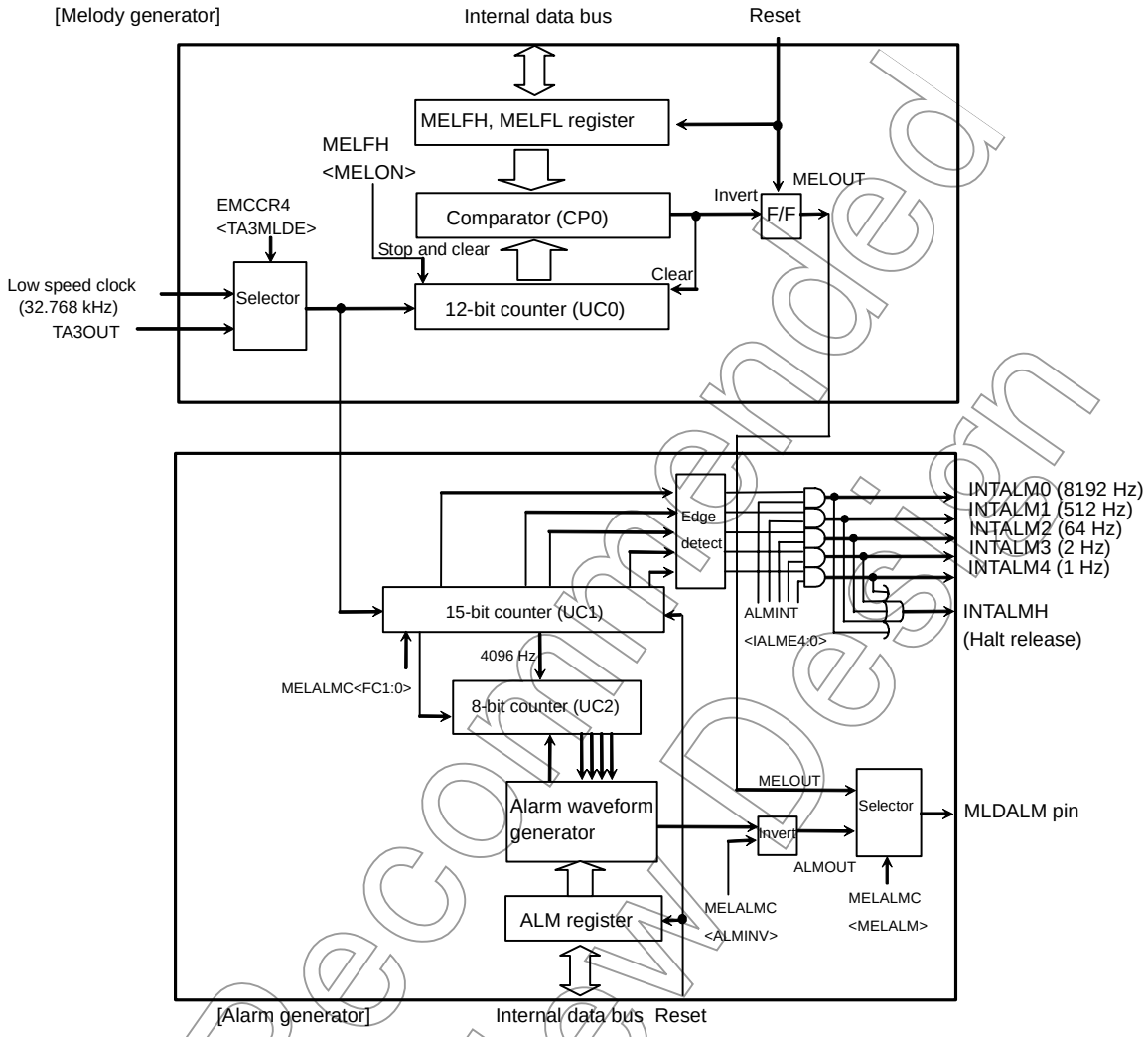


Figure 3.14.1 MLD Block Diagram

3.14.2 Control Registers

ALM R Register

ALM (0330H)

	7	6	5	4	3	2	1	0
Bit symbol	AL8	AL7	AL6	AL5	AL4	AL3	AL2	AL1
Read/Write	R/W							
After reset	0							
Function	Setting alarm pattern							

MELALMC Register

MELALMC (0331H)

	7	6	5	4	3	2	1	0
Bit symbol	FC1	FC0	ALMINV	-	-	-	-	MELALM
Read/Write	R/W		R/W	R/W	R/W	R/W	R/W	R/W
After reset	0		0	0	0	0	0	0
Function	Free-run counter control 00: Hold 01: Restart 10: Clear 11: Clear and start		Alarm Waveform invert 1: Invert	Always write 0				Output waveform select 0: Alarm 1: Melody

Note 1: MELALMEC<FC1> is read always 0.

Note 2: When setting MELALMC register except <FC1:0> during the free-run counter is running, <FC1:0> is kept 01.

MELFL Register

MELFL (0332H)

	7	6	5	4	3	2	1	0
Bit symbol	ML7	ML6	ML5	ML4	ML3	ML2	ML1	ML0
Read/Write	R/W							
After reset	0							
Function	Setting melody frequency (Lower 8 bits)							

MELFH Register

MELFH (0333H)

	7	6	5	4	3	2	1	0
Bit symbol	MELON				ML11	ML10	ML9	ML8
Read/Write	R/W				R/W			
After reset	0				0			
Function	Control melody counter 0: Stop and clear 1: Start				Setting melody frequency (Upper 4 bits)			

ALMINT Register

ALMINT (0334H)

	7	6	5	4	3	2	1	0
Bit symbol			-	IALM4E	IALM3E	IALM2E	IALM1E	IALM0E
Read/Write			R/W	R/W				
After reset			0	0				
Function			Always write 0	1: Interrupt enable for INTALM4 to INTALM0				

3.14.3 Operational Description

3.14.3.1 Melody Generator

The Melody function generates signals of any frequency (4 Hz to 5461 Hz) based on low-speed clock (32.768 kHz) 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)

at fs = 32.768 [kHz]

Melody output waveform f_{MLD} [Hz] = 32768/(2 × N + 4)

Setting value for melody N = (16384/f_{MLD}) − 2

(Note: N = 1 to 4095 (001H to FFFH), 0 is not acceptable)

(Example program)

In case of outputting "La" musical scale (440 Hz)

```
LD    (MELALMC), 11X00001B    ; 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 4096 Hz determined by the low-speed clock (32.768 kHz). And this waveform is reversible by setting a value to a register.

By connecting a loud speaker outside, Alarm tone can sound easily.

Five kinds of fixed cycle (1 Hz, 2 Hz, 64 Hz, 512 Hz, 8 kHz) interrupts are generate 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 "10" be set on MELALMC<FC1:0> register, and clear internal counter. Finally alarm pattern has to be set on 8-bit register of ALM. If it is inverted output-data, set <ALMINV> as invert.

Followings are example program, setting value of alarm pattern and waveform of each setting value

(Setting value of alarm pattern)

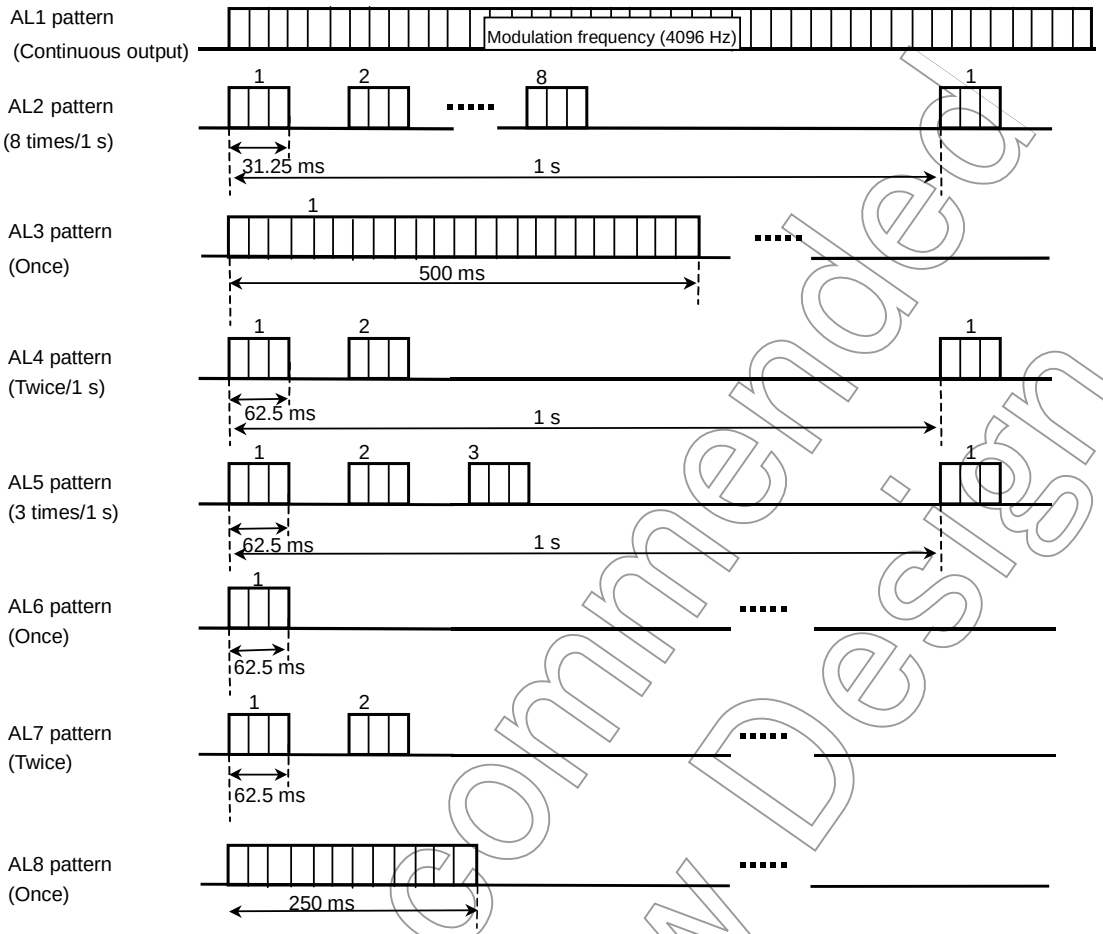
Setting Value for ALM Register	Alarm Waveform
00H	0 fixed
01H	AL1 pattern
02H	AL2 pattern
04H	AL3 pattern
08H	AL4 pattern
10H	AL5 pattern
20H	AL6pattern
40H	AL7 pattern
80H	AL8 pattern
Other	Undefined (Do not set)

(Example program)

In case of outputting AL2 pattern (31.25 ms/8 times/1 s)

```
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)



3.15 Voltage Level Detector

This function has 3-channel input voltage and reference voltage. Each channel can set own some voltage level and also have interrupt generator. These voltage level compare circuit (Voltage detector) are included in this LSI.

It shows Figure 3.15.1,Figure 3.15.2 and Figure 3.15.3 block diagram of 3-channel voltage level detector (VLD0 to VLD2).

These 3-channel VLD input can use also general purpose I/O port (Port B).

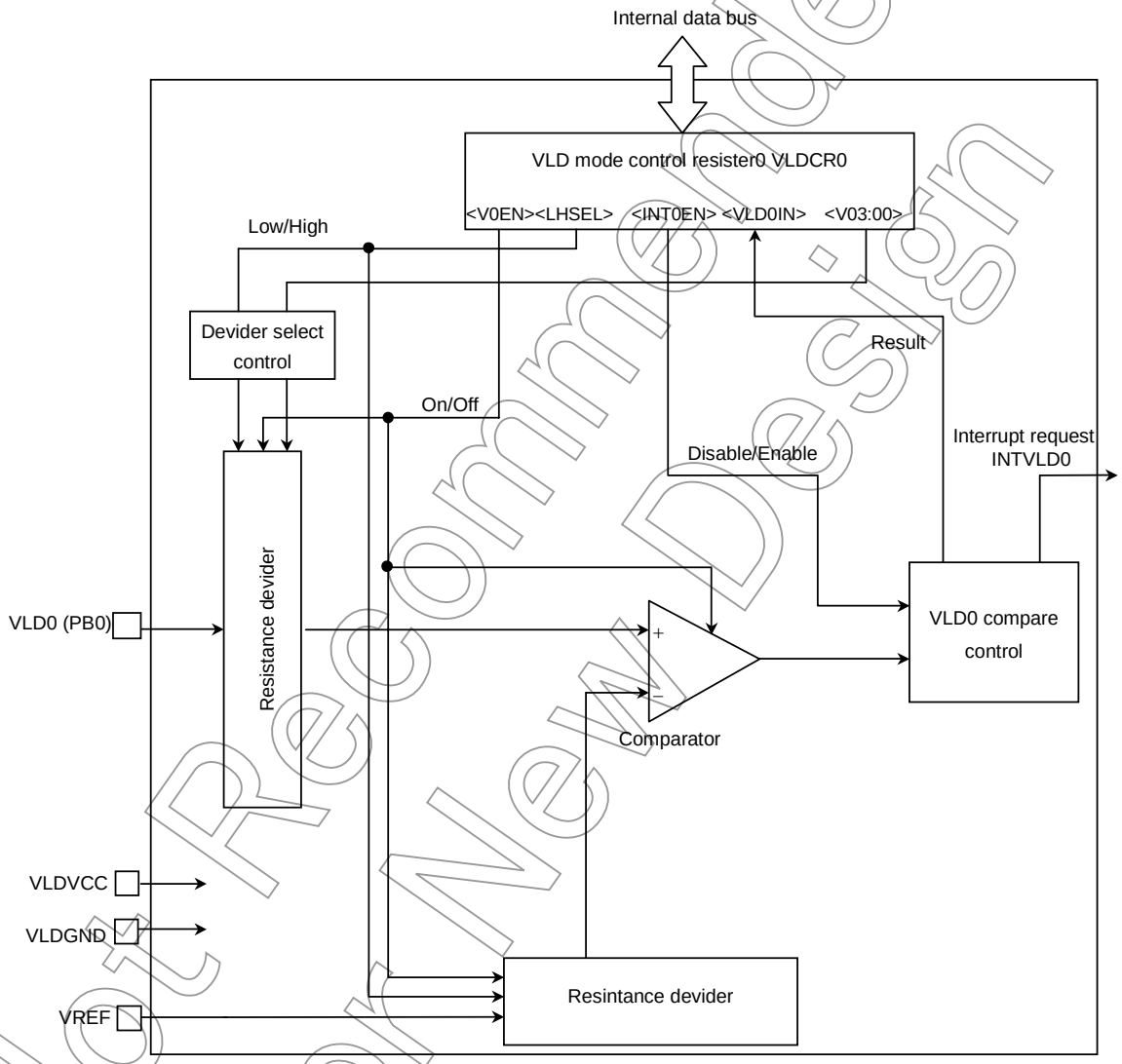


Figure 3.15.1 Block Diagram of Voltage Level Detector 0 (VLD0)

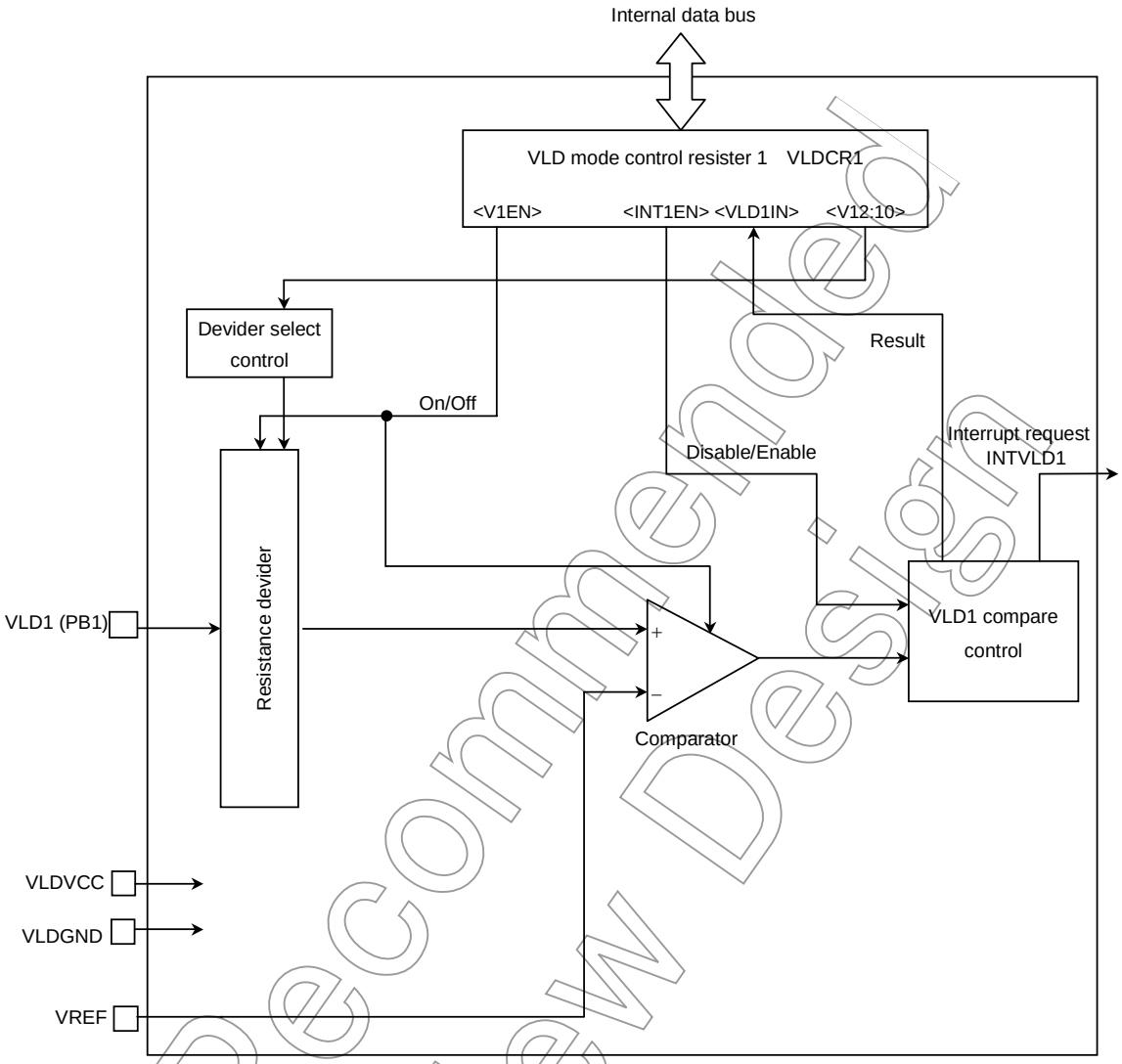


Figure 3.15.2 Block Diagram of Voltage Level Detector 1 (VLD1)

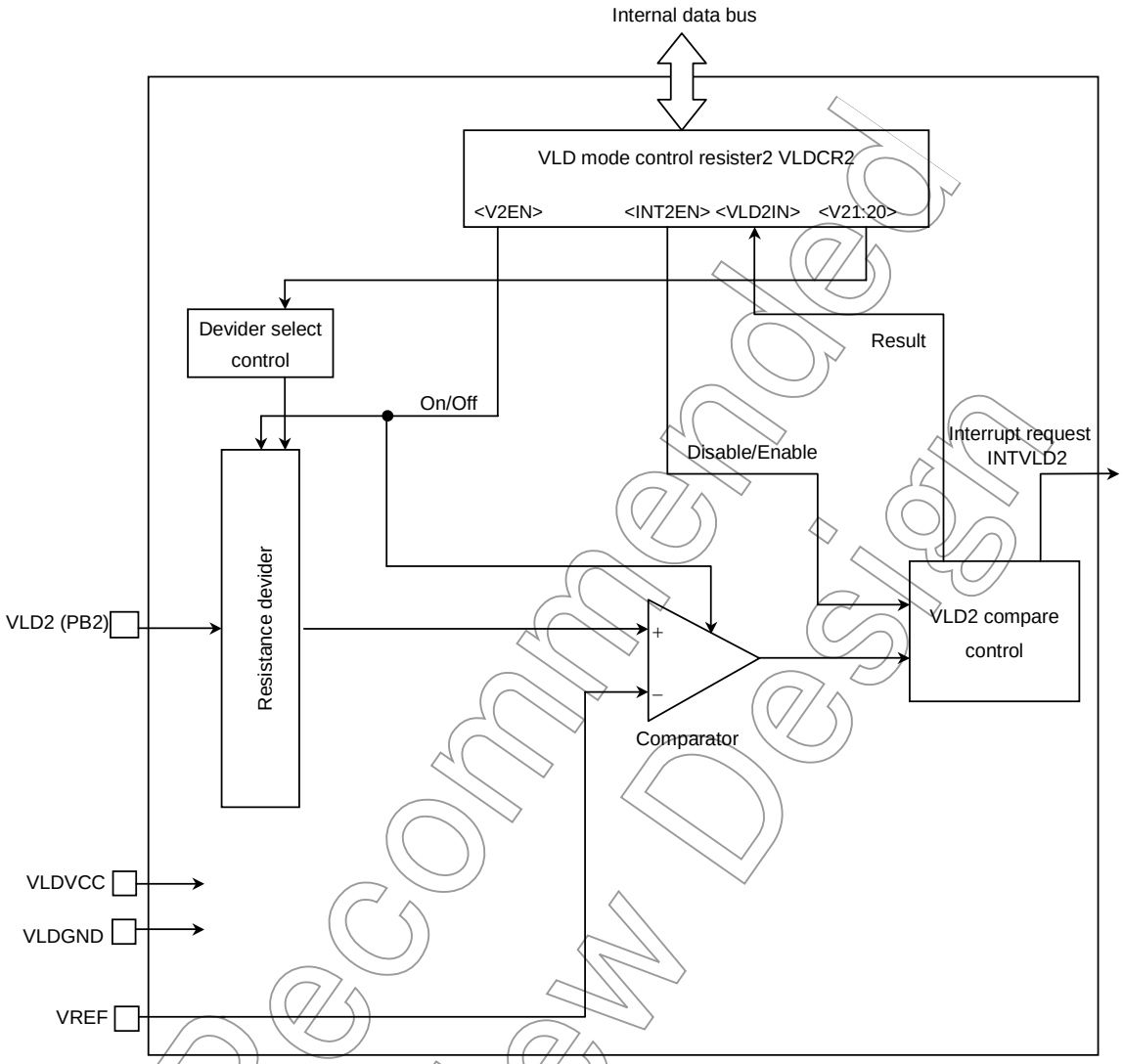


Figure 3.15.3 Block Diagram of Voltage Level Detector 2 (VLD2)

3.15.1 SFR

Voltage level detector are controlled 3 registers: VLDCR0, VLDCR1 and VLDCR2. And the interruption can be controlled by voltage compare result.

It shows Figure 3.15.4 VLDCR0 register.

VLD Mode Control Register 0

		7	6	5	4	3	2	1	0
VLDCR0 (0440H)	Bit symbol	V0EN	LHSEL	INT0EN	VLD0IN	V03	V02	V01	V00
	Read/Write	R/W		R/W	R/W	R/W			
	After reset	0	0	0	0	0	0	0	0
	Function	Voltage detect start flag 0: Disable 1: Enable	Detect level select flag 0: 1.5 V more 1: 1.4 V less	Interrupt enable flag 0: Disable 1: Enable	Detect voltage result and clear write 0: Normal 1: Voltage low	Detect Level setting register 00H to 0BH 12 steps (LHSEL = 0) + 5 steps (LHSEL = 1) = 18 steps			

VLD0 detector voltage setting

0000	1.5 V (LHSEL = 0), 0.9 V (LHSEL = 1)
0001	1.6 V (LHSEL = 0), 1.0 V (LHSEL = 1)
0010	1.7 V (LHSEL = 0), 1.1 V (LHSEL = 1)
0011	1.8 V (LHSEL = 0), 1.2 V (LHSEL = 1)
0100	1.9 V (LHSEL = 0), 1.3 V (LHSEL = 1)
0101	2.0 V (LHSEL = 0), 1.4 V (LHSEL = 1)
0110	2.1 V (LHSEL = 0), – (LHSEL = 1)
0111	2.2 V (LHSEL = 0), – (LHSEL = 1)
1000	2.3 V (LHSEL = 0), – (LHSEL = 1)
1001	2.4 V (LHSEL = 0), – (LHSEL = 1)
1010	2.5 V (LHSEL = 0), – (LHSEL = 1)
1011	2.6 V (LHSEL = 0), – (LHSEL = 1)
11**	– (LHSEL = 0), – (LHSEL = 1)

Detect voltage compare result (Only 0 data write OK)

0	Input voltage > Detect voltage
1	Input voltage < Detect voltage

Interrupt enable/disable

0	Interrupt disable
1	Interrupt enable

Detect voltage flag setting

0	Detect voltage area 1.5 V to 2.6 V
1	Detect voltage area 0.9 V to 1.4 V

Voltage detect start

0	Voltage compare turn off
1	Voltage compare turn on

Note: This register can't read and modify and write, because <VLD0IN> bit have different means between write data and read data.

Figure 3.15.4 VLD Mode Control 0 Register

VLD Mode Control Register

		7	6	5	4	3	2	1	0
VLDCR1 (0441H)	Bit symbol	V1EN	-	INT1EN	VLD1IN	-	V12	V11	V10
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W		
	After reset	0	0	0	0	0	0	0	0
	Function	Voltage detect start flag 0: Disable 1: Enable	Always write 0	Interrupt enable flag 0: Disable 1: Enable	Detect voltage result and clear write 0: Normal 1: Voltage low	Always write 0	Detect level setting register 00H to 04H 5 steps		

VLD1 detector voltage setting

000	2.2 V
001	2.3 V
010	2.4 V
011	2.5 V
100	2.6 V
101 to 111	Prohibition

Detect voltage compare result (Only 0 data write OK)

0	Input voltage > Detect voltage
1	Input voltage < Detect voltage

Interrupt enable/disable

0	Interrupt disable
1	Interrupt enable

Voltage detect start

0	Voltage compare turn off
1	Voltage compare turn on

Note: This register can't read and modify and write, because <VLD1IN> bit have different means between write data and read data.

Figure 3.15.5 VLD Mode Control 1 Register

VLD Mode Control Register 2

		7	6	5	4	3	2	1	0
VLDCR2 (0442H)	Bit symbol	V2EN	-	INT2EN	VLD2IN	-	-	V21	V20
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	After reset	0	0	0	0	0	0	0	0
	Function	Voltage detect start flag 0: Disable 1: Enable	Always write 0	Interrupt enable flag 0: Disable 1: Enable	Detect voltage result and clear write 0: Normal 1: Voltage low	Always write 0		Detect Level setting register 00H to 02H 3 steps	

VLD2 detector voltage setting

00	1.7 V
01	2.6 V
10	2.9 V
11	Prohibition

Detect voltage compare result (Only 0 data write OK)

0	Input voltage > Detect voltage
1	Input voltage < Detect voltage

Interrupt enable/disable

0	Interrupt disable
1	Interrupt enable

Voltage detect start

0	Voltage compare turn off
1	Voltage compare turn on

Note: This register can't read and modify and write, because <VLD2IN> bit have different means between write data and read data.

Figure 3.15.6 VLD Mode Control 2 Register

VLD Control Register 2

		7	6	5	4	3	2	1	0
VLDCTL (0449H)	Bit symbol					XT1VSEL	VLD2USE	VLD1USE	VLD0USE
	Read/Write					W	R/W	R/W	R/W
	After reset					0	0	0	0
	Function					0: Vcc 1: Vref	0: VLD not use 1: VLD use	0: VLD not use 1: VLD use	0: VLD not use 1: VLD use

VLD2 function control

0	VLD function don't use
1	VLD function use
Note: If even one of VLD0 to 2 use VLD function, others port function cant use output mode. It can only input mode.	

Low frequency oscillator's (XT1, XT2) power source

0	Vcc
1	Vref

Figure 3.15.7 VLD Control Register

3.15.2 Explanation of Function

Preferences

It select that does not use whether PB port is used as VLD with a register of low rank 3 bits of VLDCTR.

(1) Comparison reference voltage

Firstly, It supplies on VREF pin 1.5 V, reference voltage. Each voltage level detector compare with the reference voltage and the voltage input from each VLD terminal. Setting of detect level is decided by doing a partial pressure of the voltage input from VLD terminal. And only VLD0 can set reference voltage to 0.9 V, and can compare the voltage value too to 0.9 V to 1.4 V.

It can OFF with detector about the voltage comparison device and resistor divider circuit and a switch between VLDGND by writing in 0 at each VLDCR* <V*EN> bit. And, if it start from disable condition of VLD circuit, it must need first write <V*EN> to 1 and next wait about 1ms set-up time (no-related with system clock frequency) and next write VLDCR* <INT*EN> to 1, or first read VLDCR* <VLD*IN> data and use of detect result.

* (Asterisk) shows 0, 1 and 2 (3 channels)

(2) A selection of voltage level detector

A selection of three voltage level detector is different from next setting voltage detection level by a purpose of use.

Main battery voltage detection (VLDCR0<V0EN> = 1)

Detection voltage range is 0.9 V to 2.6 V. The voltage comparison of totaled 18 level is possible by 0.1 V step.

Sub-battery voltage detection for back-up (VLDCR1<V1EN> = 1)

Detection voltage range is 2.2 V to 2.6 V. The voltage comparison of totaled 5 level is possible by 0.1 V step.

CPU-power source battery (VLDCR2<V2EN> = 1)

Detection voltage points are 1.7 V, 2.6 V and 2.9 V.

(3) The voltage comparison start

At first, It set detect level of VLD, and movement starts the voltage comparison by establishing 1 in VLDCR* <V*EN>. VLDCR* <INT*EN> can know comparison result afterwards after) progress more than (1mS between fixed time whether I establish 1 and wait for the interrupt input by leading VLDCR* <VLD*IN>.

It maintain the result by the comparison result control circuit after I became less than detect level that established it once, and having detect the voltage fall. It establish 0 → 1 in VLDCR* <INT*EN> when let interrupt reflect the next comparison update result, and update becomes possible in clearing current maintenance result. It need to check result by all means when do not clear detect level and need to confirm current search result. In particular VLDCR* <INT*EN> establishes 1 already, and interrupt does not occur when detecting the voltage fall from starting detection when does not execute the above-mentioned clear (0: Off → 1: On light of interrupt flag) once.

* (Asterisk) shows 0, 1 and 2 (3 channels)

(4) The voltage level comparison and comparison result interrupt

Next 3 are prepared in interrupt generated by comparison result of three VLD. INTVLD0, INTVLD1 and INTVLD2 can mask own interrupt at source level, but at interrupt circuit, these interruptions are recognized as non-maskable interruption. Because it is the non-maskable interruption entirely, interrupt level is fixed in 7. Besides, as non-maskable interruption, there are NMI terminal and watchdog timer. And I accept interrupt according to default priority when interrupt request of same level occurred simultaneously. Please refer to the control of interrupt controller in detail.

(5) VLD comparison time

Comparison state per 1 channel is 8064 states (1 ms at f_{FPH} = 16 MHz).

(6) Housing and readout of VLD comparison result

VLD voltage comparison result is stored in <VLD*IN>: bit4 of VLDCR0 to VLDCR2. It is stored away successively from the moment that established 1 in <INT*EN> to <VLD*IN> after movement started it by establishing 1 in <V*EN> of VLD mode control register.

VLD comparison result housing flag <VLD*IN> shows VLD comparison result. When the voltage falls than setting detect value the input voltage from VLD* terminal this flag, 1 is led, and 0 is led when higher than setting detect value.

And this comparison result leads the output result of VLD. It is updated during data comparison movement at any time, and data will change, but the contents which data changed into last when comparison movement was stopped are maintained. On this account I can clear these data. In other words a write of 0 data becomes possible (Impossible a write of 1 data).

This signal comes to demand interrupt for CPU and, as for the interrupt, it is done edge interrupt request with a signal after it was controlled with a gate for interrupt permission flag.

I ask for the voltage setting, movement, interrupt to establish it by the following order.

When setting detect voltage .....

- Mask to interruption
- Disable VLD operation
- Change value
- Enable VLD operation
- Need set up time (More 1 ms)
- Clear write <VLD*IN>
- Release interruption mask

Note: * shows 0, 1 and 2 (3 channels)

Setting example

a. In case of setting that seems to jump to VLD0 interrupt (INTVLD0) handling routine, compare the analog input voltage of VLD0 terminal the voltage, and fall than detect voltage which the result analog input voltage established

Main routine setting

		7	6	5	4	3	2	1	0	
VLDCR0	←	1	0	0	1	0	0	0	0	VLD0 turn ON, detect voltage 1.5 V set, interrupt flag clear of VLD (Comparison start)
										<set-up time>
VLDCR0	←	–	–	–	1	0	–	–	– –	Enable interruption. Don't change detect voltage (1.5 V)

b. In case of setting the voltage comparison result of analog input voltage of VLD1 terminal is led, and VLD1 cuts in by handling routine to continue, and (INTVLD1) is validated, and to wait for interrupt outbreak from comparison result

Main routine setting

		7	6	5	4	3	2	1	0	
VLDCR1	←	1	X	0	1	X	0	1	0	VLD1 turn ON, detect voltage 2.4 V set, comparison start
										<set-up time>
VLDCR1	→	–	X	–	–	X	–	–	–	Read result of comparison
										Don't change the detect voltage and other setting
VLDCR1	←	–	X	–	1	X	–	–	–	Clear interrupt
VLDCR1	←	–	X	1	0	X	–	–	–	Enable interruption. Don't change detect voltage (2.4 V)

X: Don't care, –: No change

3.15.3 Special Function Explanation of VLD

VLD circuit is different from the usual voltage search, and a special function is included. This circuit is called interval operation function, and it operates the following movement.

It is movement to repeat movement and standstill by the interval when interval operation function established each VLD. Without utilizing CPU and timer, VLD movement that reduced consumption electric current can come true.

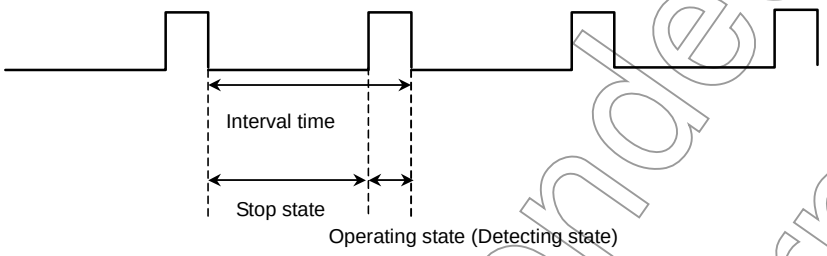


Figure 3.15.8 Interval Operation Example

Clearing the flag in the stop state of interval operation function should be executed after voltage detect start flag (VLDCRx <VxEN>) is disabled same as normal operation.

HPCTST1 (0445H)

	7	6	5	4	3	2	1	0
Bit symbol		-	TIM21	TIM20	TIM11	TIM10	TIM01	TIM00
Read/Write		R/W						
After reset		0	0	0	0	0	0	0
Function		Always write 0	VLD2 (for CPU) Sampling time		VLD1 (for back-up) Sampling time		VLD0 (for main) Sampling time	

VLD0, VLD1, VLD2 interval time setting

TIM*1.TIM*0	Detect time	Interval time
00	1.95 ms	(1.96 × 8) ms
01	1.95 ms	(1.96 × 16) ms
10	1.95 ms	(1.96 × 32) ms
11	1.95 ms	(1.96 × 64) ms

Figure 3.15.9 VLD Special Function Register 1

HPCTST2 (0446H)

	7	6	5	4	3	2	1	0
Bit symbol	-	-	SAM_2	SAM_1	SAM_0	-	-	-
Read/Write	R/W	R/W	R/W					
After reset	0	0	0	0	0	0	0	0
Function	Always write 0		VLD2 0: Normal 1: Interval	VLD1 0: Normal 1: Interval	VLD0 0: Normal 1: Interval	Always write 0		

VLD0, VLD1, VLD2 continue/interval control

0	Normal (Continues mode)
1	Interval

Figure 3.15.10 VLD Special Function Register 2

3.16 Data Horizontal and Vertical Conversion Circuit

This LSI built in data horizontal and vertical conversion (HVC) circuit.

Horizontal and vertical can convert data of maximum 8*8 bit into. Horizontal and vertical of data of character ROM are the functions that a burden of software is lightened in case converted into.

It shows Figure 3.16.1 block diagram HVC.

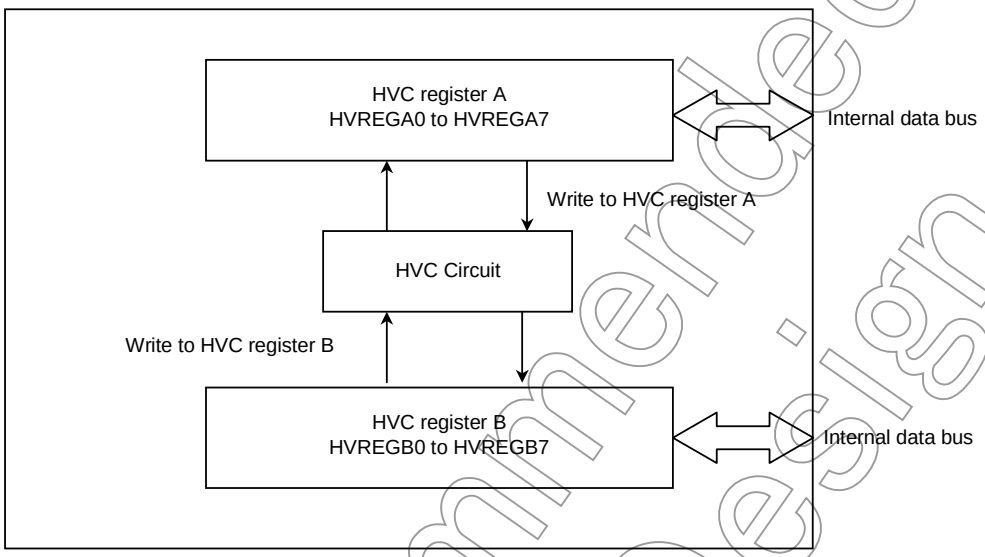


Figure 3.16.1 Block Diagram HVC

3.16.1 SFR

There is each H/V conversion register A for H/V converter to store away H/V conversion data (HVREGA0 to HVREGA7), H/V conversion register B (HVREGB0 to HVREGB7) 8.

It show Figure 3.16.2 to Figure 3.16.5 HVC registers.

HVC Register A 0

		7	6	5	4	3	2	1	0
HVREGA0 (0450H)	Bit symbol	HVRA07	HVRA06	HVRA05	HVRA04	HVRA03	HVRA02	HVRA01	HVRA00
	Read/Write	R/W							
	After reset	0							
	Function	HVC data housing							

HVC Register A 1

		7	6	5	4	3	2	1	0
HVREGA1 (0451H)	Bit symbol	HVRA17	HVRA16	HVRA15	HVRA14	HVRA13	HVRA12	HVRA11	HVRA10
	Read/Write	R/W							
	After reset	0							
	Function	HVC data housing							

HVC Register A 2

		7	6	5	4	3	2	1	0
HVREGA2 (0452H)	Bit symbol	HVRA27	HVRA26	HVRA25	HVRA24	HVRA23	HVRA22	HVRA21	HVRA20
	Read/Write	R/W							
	After reset	0							
	Function	HVC data housing							

HVC Register A 3

		7	6	5	4	3	2	1	0
HVREGA3 (0453H)	Bit symbol	HVRA37	HVRA36	HVRA35	HVRA34	HVRA33	HVRA32	HVRA31	HVRA30
	Read/Write	R/W							
	After reset	0							
	Function	HVC data housing							

Figure 3.16.2 HVC Register 1

HVC Register A 4

HVREGA4 (0454H)

	7	6	5	4	3	2	1	0
Bit symbol	HVRA47	HVRA46	HVRA45	HVRA44	HVRA43	HVRA42	HVRA41	HVRA40
Read/Write	R/W							
After reset	0							
Function	HVC data housing							

HVC Register A 5

HVREGA5 (0455H)

	7	6	5	4	3	2	1	0
Bit symbol	HVRA57	HVRA56	HVRA55	HVRA54	HVRA53	HVRA52	HVRA51	HVRA50
Read/Write	R/W							
After reset	0							
Function	HVC data housing							

HVC Register A 6

HVREGA6 (0456H)

	7	6	5	4	3	2	1	0
Bit symbol	HVRA67	HVRA66	HVRA65	HVRA64	HVRA63	HVRA62	HVRA61	HVRA60
Read/Write	R/W							
After reset	0							
Function	HVC data housing							

HVC Register A 7

HVREGA7 (0457H)

	7	6	5	4	3	2	1	0
Bit symbol	HVRA77	HVRA76	HVRA75	HVRA74	HVRA73	HVRA72	HVRA71	HVRA70
Read/Write	R/W							
After reset	0							
Function	HVC data housing							

Figure 3.16.3 HVC Register 2

HVC Data Register B 0

HVREGB0 (0458H)

	7	6	5	4	3	2	1	0
Bit symbol	HVRB07	HVRB06	HVRB05	HVRB04	HVRB03	HVRB02	HVRB01	HVRB00
Read/write	R/W							
After reset	0							
Function	HVC data housing							

HVC Data Register B 1

HVREGB1 (0459H)

	7	6	5	4	3	2	1	0
Bit symbol	HVRB17	HVRB16	HVRB15	HVRB14	HVRB13	HVRB12	HVRB11	HVRB10
Read/Write	R/W							
After reset	0							
Function	HVC data housing							

HVC Data Register B 2

HVREGB2 (045AH)

	7	6	5	4	3	2	1	0
Bit symbol	HVRB27	HVRB26	HVRB25	HVRB24	HVRB23	HVRB22	HVRB21	HVRB20
Read/Write	R/W							
After reset	0							
Function	HVC data housing							

HVC Data Register B 3

HVREGB3 (045BH)

	7	6	5	4	3	2	1	0
Bit symbol	HVRB37	HVRB36	HVRB35	HVRB34	HVRB33	HVRB32	HVRB31	HVRB30
Read/Write	R/W							
After reset	0							
Function	HVC data housing							

Figure 3.16.4 HVC Register 3

HVC Data Register B 4

HVREGB4 (045CH)

	7	6	5	4	3	2	1	0
Bit symbol	HVRB47	HVRB46	HVRB45	HVRB44	HVRB43	HVRB42	HVRB41	HVRB40
Read/Write	R/W							
After reset	0							
Function	HVC data housing							

HVC Data Register B 5

HVREGB5 (045DH)

	7	6	5	4	3	2	1	0
Bit symbol	HVRB57	HVRB56	HVRB55	HVRB54	HVRB53	HVRB52	HVRB51	HVRB50
Read/Write	R/W							
After reset	0							
Function	HVC data housing							

HVC Data Register B 6

HVREGB6 (045EH)

	7	6	5	4	3	2	1	0
Bit symbol	HVRB67	HVRB66	HVRB65	HVRB64	HVRB63	HVRB62	HVRB61	HVRB60
Read/Write	R/W							
After reset	0							
Function	HVC data housing							

HVC Data Register B 7

HVREGB7 (045FH)

	7	6	5	4	3	2	1	0
Bit symbol	HVRB77	HVRB76	HVRB75	HVRB74	HVRB73	HVRB72	HVRB71	HVRB70
Read/Write	R/W							
After reset	0							
Function	HVC data housing							

Figure 3.16.5 HVC Register 4

3.16.2 Operation Explanation

Conversion result is stored away by HVREGB register when did a light of data to do H/V conversion to HVREGA register. The data which did a light begin to be read when led HVREGA register then. However, It is different from the data which a light did even if HVREGA register was led when did a light of HVREGB register after having done a light of HVREGA register. It operate the same movement about HVREGB register.

It shows Table 3.16.1 "Relation of HVC Data".

Table 3.16.1 Relation of HVC Data

Bit	7	6	5	4	3	2	1	0
HVREGB7	HVRA77	HVRA67	HVRA57	HVRA47	HVRA37	HVRA27	HVRA17	HVRA07
HVREGB6	76	66	56	46	36	26	16	06
HVREGB5	75	65	55	45	35	25	15	05
HVREGB4	74	64	54	44	34	24	14	04
HVREGB3	73	63	53	43	33	23	13	03
HVREGB2	72	62	52	42	32	22	12	02
HVREGB1	71	61	51	41	31	21	11	01
HVREGB0	70	60	50	40	30	20	10	00
	↑	↑	↑	↑	↑	↑	↑	↑
	HVREGA7< >	HVREGA6< >	HVREGA5< >	HVREGA4< >	HVREGA3< >	HVREGA2< >	HVREGA1< >	HVREGA0< >

4. Electrical Characteristics

4.1 Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Power supply voltage	Vcc	−0.5 to 4.0	V
Input voltage	VIN	−0.5 to Vcc + 0.5	
Output current	IOL	2	mA
Output current	IOH	−2	
Output current (total)	ΣIOL	80	
Output current (total)	ΣIOH	−80	
Power dissipation (Ta = 85°C)	PD	600	mW
Soldering temperature (10 s)	TSOLDER	260	°C
Storage temperature	TSTG	−65 to 150	
Operating temperature	TOPR	−10 to 70	

Note: The absolute maximum ratings are rated values which must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, a device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products which include this device, ensure that no absolute maximum rating value will ever be exceeded.

Solderability of lead free products

Test parameter	Test condition	Note
Solderability	Use of Sn-37Pb solder Bath Solder bath temperature = 230°C, Dipping time = 5 seconds The number of times = one, Use of R-type flux	Pass: solderability rate until forming ≥ 95%
	Use of Sn-3.0Ag-0.5Cu solder bath Solder bath temperature = 245°C, Dipping time = 5 seconds The number of times = one, Use of R-type flux (use of lead free)	

4.2 DC Characteristics (1/2)

Parameter		Symbol	Condition		Min	Typ. (Note 1)	Max	Unit
Power supply voltage (AVCC = DVCC) (AVSS = DVSS = 0 V)		VCC	fc = 2 to 27 MHz	fs = 30 to 34 kHz	2.7		3.6	V
			fc = 2 to10 MHz		1.8			
Power supply voltage		VREF	3.6 ≥ Vcc ≥ 1.8 V		–	1.5	–	
Input low voltage	D0 to D15	VIL	Vcc ≥ 2.7 V		–0.3		0.6	
			Vcc < 2.7 V				0.2 Vcc	
	P52 to PD7 (except $\overline{\text{RESET}}$, P52, P72, P74, P9, PB3, PB4, PB5, PC4, PC5)	VIL1	Vcc ≥ 2.7 V				0.3 Vcc	
			Vcc < 2.7 V				0.2 Vcc	
	$\overline{\text{RESET}}$, P52, P72, P74, P9, PB3, PB4, PB5, PC4, PC5	VIL2	Vcc ≥ 2.7 V				0.25 Vcc	
			Vcc < 2.7 V				0.15 Vcc	
	AM0 to AM1	VIL3	Vcc ≥ 2.7 V				0.3	
			Vcc < 2.7 V				0.3	
	X1	VIL4	Vcc ≥ 2.7 V				0.2 Vcc	
			Vcc < 2.7 V				0.1 Vcc	
Input high voltage	D0 to D15	VIH	3.6 ≥ Vcc ≥ 3.3 V		2.4		Vcc + 0.3	
			3.3 > Vcc ≥ 2.7 V		2.0			
			Vcc < 2.7 V		0.7 Vcc			
	P52 to PD7 (except $\overline{\text{RESET}}$, P52, P72, P74, P9, PB3, PB4, PB5, PC4, PC5)	VIH1	Vcc ≥ 2.7 V		0.7 Vcc			
			Vcc < 2.7 V		0.8 Vcc			
	$\overline{\text{RESET}}$, P52, P72, P74, P9, PB3, PB4, PB5, PC4, PC5	VIH2	Vcc ≥ 2.7 V		0.75 Vcc			
			Vcc < 2.7 V		0.85 Vcc			
	AM0 to AM1	VIH3	Vcc ≥ 2.7 V		Vcc – 0.3			
			Vcc < 2.7 V		Vcc – 0.3			
	X1	VIH4	Vcc ≥ 2.7 V		0.8 Vcc			
			Vcc < 2.7 V		0.9 Vcc			
Output low voltage		VOL	IOL = 1.6 mA	Vcc ≥ 2.7 V			0.45	
			IOL = 0.4 mA	Vcc < 2.7 V			0.15 Vcc	
Output high voltage		VOH	IOH = –400 μA	Vcc ≥ 2.7 V	Vcc – 0.3			
			IOH = –200 μA	Vcc < 2.7 V	0.8 Vcc			

4.2 DC Characteristics (2/2)

Parameter	Symbol	Condition	Min	Typ. (Note 1)	Max	Unit
Input leakage current	ILI	0.0 ≤ VIN ≤ Vcc		0.02	±5	μA
Output leakage current	ILO	0.2 ≤ VIN ≤ Vcc − 0.2		0.05	±5	
$\overline{\text{RESET}}$ pull-up resistor	RRST	2.7 ≤ Vcc ≤ 3.6 V	80		400	kΩ
		Vcc = 2 V ± 10%	200		1000	
Pin capacitance	CIO	fc = 1 MHz			10	pF
Schmitt width ($\overline{\text{RESET}}$, INT3, OPTRX0, $\overline{\text{NMI}}$, KI0 to KI7, INT0, INT1, INT2, RXD1, SCLK1/ $\overline{\text{CTS1}}$)	VTH	Vcc ≥ 2.7 V	0.4	0.9		V
		Vcc < 2.7 V	0.3	0.7		
Programmable pull-up resistor (P53, P56, P60 toP67, P70 to P71, P73, PD0 to P7)	RKH1	2.7 ≤ Vcc ≤ 3.6 V	80	200	400	kΩ
		Vcc = 2 V ± 10%	200		1000	
Programmable pull-up resistor (P90 to P97, PB0 to PB2, PB4 to PB5, P52, P72, PC4 to PC5)	RKH2	2.7 ≤ Vcc ≤ 3.6 V	60	180	350	
		Vcc = 2 V ± 10%	180		900	
Programmable pull-up resistor (PB3 at Vcc)	RKH3	2.7 ≤ Vcc ≤ 3.6 V	50	167	280	
		Vcc = 2 V ± 10%	120		900	
Programmable pull-up resistor (PB3 at Vss)		2.7 ≤ Vcc ≤ 3.6 V	400	1000	2000	
		Vcc = 2 V ± 10%	800		4500	
Programmable pull-down resistor (P72, PB4 to PB5, PC4 to PC5)	RKL	2.7 ≤ Vcc ≤ 3.6 V	80	200	600	
		Vcc = 2 V ± 10%	200		1000	
NORMAL (Note 2)	Icc	2.7 ≤ Vcc ≤ 3.6 V fc = 27 MHz		11.0	15.0	mA
IDLE2				4.5	6.7	
IDLE1				1.5	2.9	
NORMAL (Note 2)		Vcc = 2 V ± 10% fc = 10 MHz (Typ.: Vcc = 2.0 V)		2.5	3.5	mA
IDLE2				1.0	1.4	
IDLE1				0.3	0.6	
SLOW (Note 2)		2.7 ≤ Vcc ≤ 3.6 V fs = 32.768 kHz		15.0	30.0	μA
IDLE2				6.0	23.0	
IDLE1				2.5	20	
SLOW (Note 2)		Vcc = 2V ± 10% fs = 32.768 kHz (Typ.: Vcc = 2.0 V)		9.0	20	μA
IDLE2				4.0	15	
IDLE1				1.0	10	
STOP		1.8 ≤ Vcc ≤ 3.6 V		0.3	10	μA
XT: VREF power operation	Icc Iref	VREF = 1.5 V		0.8	1.2	μA

Note 1: Typical values are for when Ta = 25°C and Vcc = 3.3 V unless otherwise noted.

Note 2: Icc measurement conditions (Normal, Slow):
All functions are operational; output pins are open and input pins are fixed. Data and address bus CL = 30 pF loaded.

Note 3: All Icc specifications are VREF = 1.5 V and fs power = VREF condition.

4.3 AC Characteristics

(1) Vcc = 2.7 to 3.6 V

No.	Parameter	Symbol	Variable		f_{FPH} = 27 MHz		Unit
			Min	Max	Min	Max	
1	f_{FPH} period (= x)	t_{FPH}	37.0	31250	37.0		ns
2	A0 to A23 valid → $\overline{RD}$ / $\overline{WR}$ fall	t_{AC}	x – 23		14		ns
	SR mode (LCDC DMA case: READ only)		1.5x – 13		32		ns
3	$\overline{RD}$ rise → A0 to A23 hold	t_{CAR}	0.5x – 13		5		ns
4	$\overline{WR}$ rise → A0 to A23 hold	t_{CAW}	x – 13		24		ns
	$\overline{DS}$ rise → A0 to A23 hold		x – 13		24		ns
5	A0 to A23 valid → D0 to D15 input	t_{AD}		3.5x – 24		105	ns
6	$\overline{RD}$ fall → D0 to D15 input	t_{RD}		2.5x – 24		68	ns
	SR mode (LCDC DMA case)			2.0x – 24		50	ns
7	$\overline{RD}$ low width	t_{RR}	2.5x – 15		77		ns
	SR mode (LCDC DMA case)		2.0x – 15		59		ns
8	$\overline{RD}$ rise → D0 to A15 hold	t_{HR}	0		0		ns
9	$\overline{WR}$ low width	t_{WW}	2.0x – 15		59		ns
	$\overline{DS}$ Low Width		2.0x – 15		59		ns
10	D0 to D15 valid → $\overline{WR}$ rise	t_{DW}	1.5x – 35		20		ns
	D0 to D15 valid → $\overline{DS}$ rise		1.5x – 35		20		ns
11	$\overline{WR}$ rise → D0 to D15 hold	t_{WD}	x – 25		12		ns
	$\overline{DS}$ rise → D0 to D15 hold		x – 25		12		ns
12	A0 to A23 valid → $\overline{WAIT}$ input (1 + N) WAIT mode	t_{AW}		3.5x – 60		69	ns
13	$\overline{RD}$ / $\overline{WR}$ fall → $\overline{WAIT}$ hold (1 + N) WAIT mode	t_{CW}	2.5x + 0		92		ns
	SR mode (LCDC DMA case: READ only)		2.0x + 0		74		ns
14	A0 to A23 valid → Port input	t_{APH}		3.5x – 89		40	ns
15	A0 to A23 valid → Port hold	t_{APH2}	3.5x		129		ns
16	A0 to A23 valid → Port valid	t_{APO}		3.5x + 60		189	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.

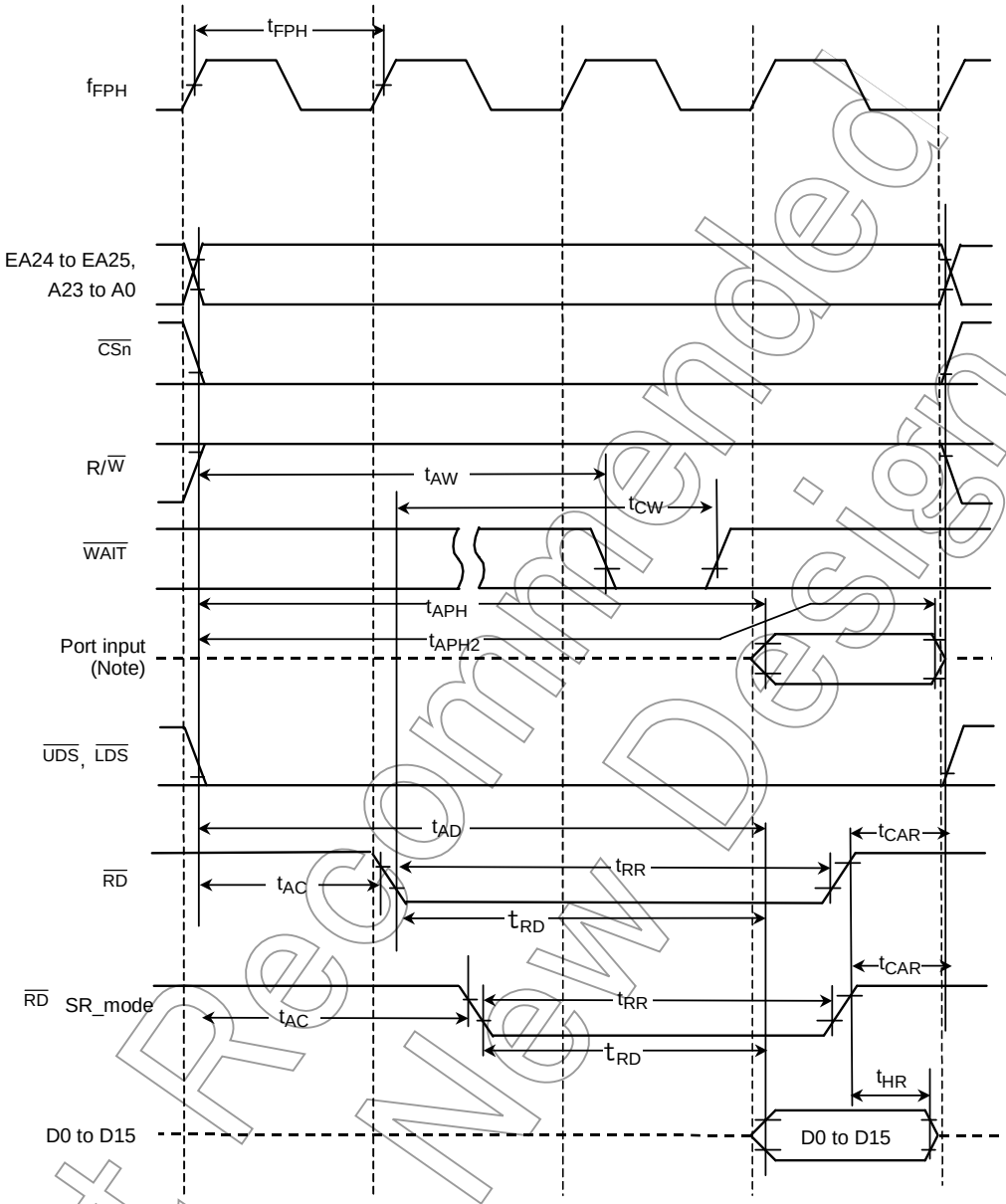
(2) Vcc = 2.0 V ± 10%

No.	Parameter	Symbol	Variable		f_{FPH} = 10 MHz		Unit
			Min	Max	Min	Max	
1	f_{FPH} period (= x)	t_{FPH}	100	31250	100		ns
2	A0 to A23 valid → $\overline{RD}$ / $\overline{WR}$ fall	t_{AC}	x – 46		54		ns
	SR mode (LCDC DMA case: READ only)		1.5x – 46		104		ns
3	$\overline{RD}$ rise → A0 to A23 hold	t_{CAR}	0.5x –30		20		ns
4	$\overline{WR}$ rise → A0 to A23 hold	t_{CAW}	x – 26		74		ns
	$\overline{DS}$ rise → A0 to A23 hold		x – 26		74		ns
5	A0 to A23 valid → D0 to D15 input	t_{AD}		3.5x – 48		302	ns
6	$\overline{RD}$ fall → D0 to D15 input	t_{RD}		2.5x – 48		202	ns
	SR mode (LCDC DMA case)			2.0x – 48		152	ns
7	$\overline{RD}$ low width	t_{RR}	2.5x – 30		220		ns
	SR mode (LCDC DMA case)		2.0x – 30		170		ns
8	$\overline{RD}$ rise → D0 to A15 hold	t_{HR}	0		0		ns
9	$\overline{WR}$ low width	t_{WW}	2.0x – 30		170		ns
	$\overline{DS}$ low width		2.0x – 30		170		ns
10	D0 to D15 valid → $\overline{WR}$ rise	t_{DW}	1.5x – 70		80		ns
	D0 to D15 valid → $\overline{DS}$ rise		1.5x – 70		80		ns
11	$\overline{WR}$ rise → D0 to D15 hold	t_{WD}	x – 50		50		ns
	$\overline{DS}$ rise → D0 to D15 hold		x – 50		50		ns
12	A0 to A23 valid → $\overline{WAIT}$ input (1 + N) WAIT mode	t_{AW}		3.5x – 120		230	ns
13	$\overline{RD}$ / $\overline{WR}$ fall → $\overline{WAIT}$ hold (1 + N) WAIT mode	t_{CW}	2.5x + 0		250		ns
	SR mode (LCDC DMA case: READ only)		2.0x + 0		200		ns
14	A0 to A23 valid → Port input	t_{APH}		3.5x – 178		172	ns
15	A0 to A23 valid → Port hold	t_{APH2}	3.5x		350		ns
16	A0 to A23 valid → Port valid	t_{APO}		3.5x + 120		470	ns

AC measuring conditions

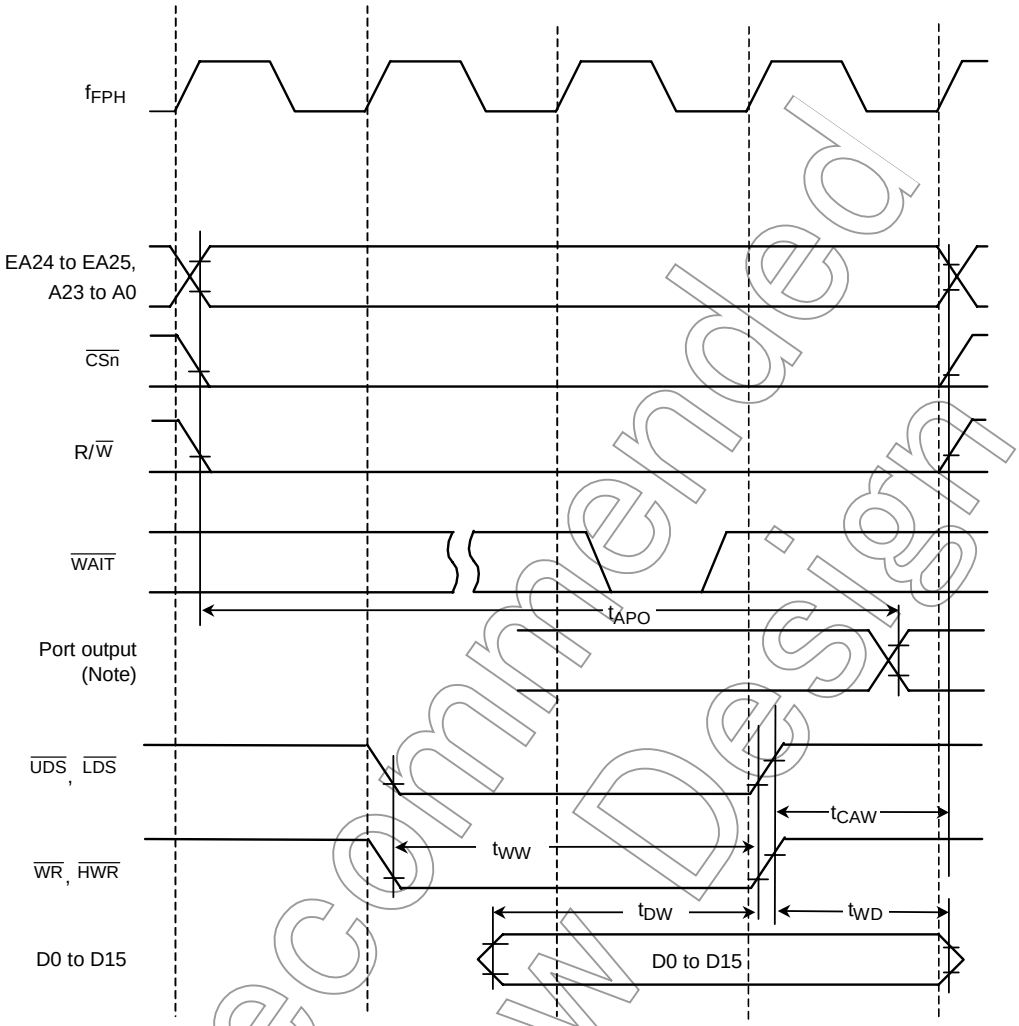
- Output level: High = 0.7 V, Low = 0.3 V, CL = 50 pF
- Input level: High = 0.9 V, Low = 0.1 V

(3) Read cycle



Note: Since the CPU accesses the internal area to read data from a port, the control signals of external pins such as $\overline{RD}$ and $\overline{CS}$ are not enabled. Therefore, the above waveform diagram should be regarded as depicting internal operation. Please also note that the timing and AC characteristics of port input/output shown above are typical representation. For details, contact your local Toshiba sales representative.

(4) Write cycle



Note: Since the CPU accesses the internal area to write data to a port, the control signals of external pins such as $\overline{WR}$ and $\overline{CS}$ are not enabled. Therefore, the above waveform diagram should be regarded as depicting internal operation. Please also note that the timing and AC characteristics of port input/output shown above are typical representation. For details, contact your local Toshiba sales representative.

(5) Vcc = 3.0 to 3.6 V

No.	Parameter	Symbol	Variable		27 MHz		Unit
			Min	Max	Min	Max	
1	$\overline{RAS}$ cycle time	t_{RC}	4.0x		148		ns
2	$\overline{RAS}$ access time	t_{RAC}		3.0x – 35		76	ns
3	$\overline{CAS}$ access time	t_{CAC}		1.5x – 30		26	ns
4	Column address access time	t_{AA}		2.5x – 45		48	ns
5	After $\overline{UCAS}$, $\overline{LCAS}$ data hold time	t_{OFF1}	0		0		ns
6	$\overline{RAS}$ pre-charge time	t_{RP}	1.5x – 4		52		ns
7	$\overline{RAS}$ pulse width	t_{RAS}	2.5x – 20		73		ns
8	$\overline{RAS}$ hold time	t_{RSH}	1.0x – 15		22		ns
9	$\overline{CAS}$ hold time	t_{CSH}	3.0x – 35		76		ns
10	$\overline{CAS}$ pulse width	t_{CAS}	1.5x – 15		41		ns
11	$\overline{RAS}$ – $\overline{CAS}$ delay time	t_{RCD}	1.5x – 30	1.5x	26	55	ns
12	$\overline{RAS}$ column address delay time	t_{RAD}	0.5x – 3	0.5x + 20	16	38	ns
13	$\overline{CAS}$ – $\overline{RAS}$ pre-charge time	t_{CRP}	1.0x – 25		12		ns
14	$\overline{CAS}$ pre-charge time	t_{CPD}	2.5x – 35		58		ns
15	Row address setup time	t_{ASR}	0.5x – 15		4		ns
16	Row address hold time	t_{RAH}	0.5x – 7		12		ns
17	Column address setup time	t_{ASC}	1.0x – 25		12		ns
18	Column address hold time	t_{CAH}	2.0x – 50		24		ns
19	Column address $\overline{RAS}$ read time	t_{RAL}	2.0x – 30		44		ns
20	Write command $\overline{CAS}$ read time	t_{CWL}	2.0x – 35		39		ns
21	Data output setup time	t_{DS}	0.5x – 17		2		ns
22	Data output hold time	t_{DH}	2.0x – 35		39		ns
23	Write command setup time	t_{WCS}	0.5x – 18		0		ns
24	$\overline{CAS}$ hold time ($\overline{CAS}$ before $\overline{RAS}$)	t_{CHR*1}	2.0x – 50		24		ns
25	$\overline{RAS}$ pre-charge $\overline{CAS}$ active time	t_{RPC}	1.5x – 30		26		ns
26	$\overline{CAS}$ setup time ($\overline{CAS}$ before $\overline{RAS}$)	t_{CSR*}	0.5x – 2		17		ns
27	$\overline{RAS}$ pre-charge time (Self refresh)	t_{RPS*2}	4.0x – 16		132		ns
28	$\overline{CAS}$ hold time (Self refresh)	t_{CHS*2}	0		0		ns
29	Refresh setup time	t_{CFL*}	1.0x – 10		27		ns
30	Refresh hold time	t_{CFH*}	1.0x – 15		22		ns
31	Write command pulse width	t_{WP}	2.0x – 40		34		ns
32	Write command hold time	t_{WCH}	1.5x – 35		21		ns
33	$\overline{OE}$ access time 1	t_{OAC1}		2.5x – 50		43	ns
	$\overline{OE}$ access time 2	t_{OAC2}		2.0x – 40		34	ns
34	After $\overline{OE}$ input data hold time	t_{OFF2}	0		0		ns

AC measuring conditions

- Output level: High = 0.7 V, Low = 0.3 V, CL = 50 pF
- Input level: High = 0.9 V, Low = 0.1 V

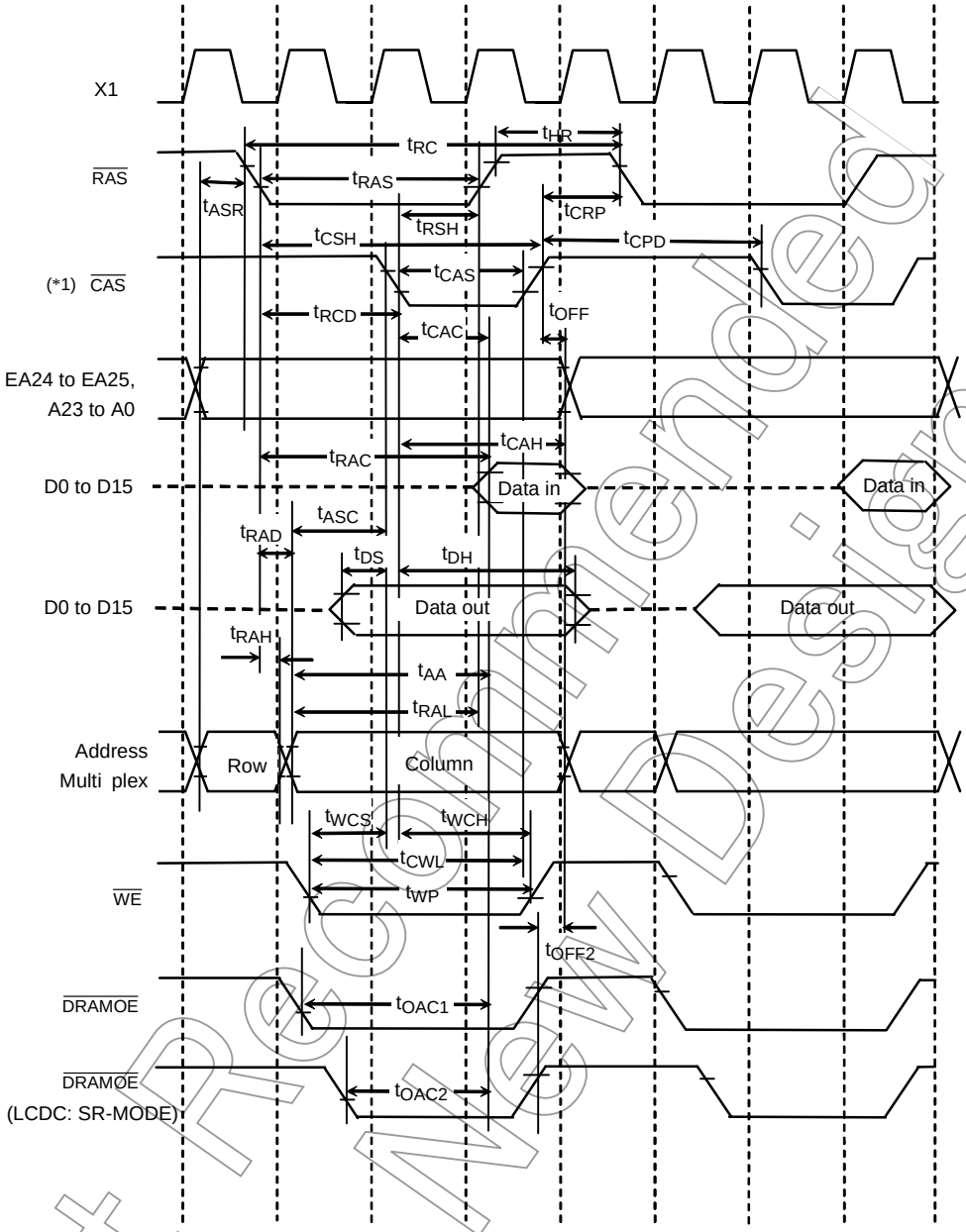
(6) Vcc = 2.7 to 3.6 V

No.	Parameter	Symbol	Variable		27 MHz		Unit
			Min	Max	Min	Max	
1	$\overline{RAS}$ cycle time	t_{RC}	4.0x		148		ns
2	$\overline{RAS}$ access time	t_{RAC}		3.0x – 38		73	ns
3	$\overline{RAS}$ access time	t_{CAC}		1.5x – 38		23	ns
4	Column address access time	t_{AA}		2.5x – 48		45	ns
5	After $\overline{UCAS}$, $\overline{LCAS}$ data hold time	t_{OFF1}	0		0		ns
6	$\overline{RAS}$ pre-charge time	t_{RP}	1.5x – 6		50		ns
7	$\overline{RAS}$ pulse width	t_{RAS}	2.5x – 22		71		ns
8	$\overline{RAS}$ hold time	t_{RSH}	1.0x – 18		19		ns
9	$\overline{CAS}$ hold time	t_{CSH}	3.0x – 33		74		ns
10	$\overline{CAS}$ pulse width	t_{CAS}	1.5x – 13		39		ns
11	$\overline{RAS}$ – $\overline{CAS}$ delay time	t_{RCD}	1.5x – 32	1.5x	24	53	ns
12	$\overline{RAS}$ column address delay time	t_{RAD}	0.5x – 5	0.5x + 20	13	36	ns
13	$\overline{CAS}$ – $\overline{RAS}$ pre-charge time	t_{CRP}	1.0x – 27		10		ns
14	$\overline{CAS}$ pre-charge time	t_{CPD}	2.5x – 37		56		ns
15	Row address setup time	t_{ASR}	0.5x – 16		3		ns
16	Row address hold time	t_{RAH}	0.5x – 8		10		ns
17	Column address setup time	t_{ASC}	1.0x – 27		10		ns
18	Column address hold time	t_{CAH}	2.0x – 52		22		ns
19	Column address $\overline{RAS}$ read time	t_{RAL}	2.0x – 32		42		ns
20	Write command $\overline{CAS}$ read time	t_{CWL}	2.0x – 37		37		ns
21	Data output setup time	t_{DS}	0.5x – 17		2		ns
22	Data output hold time	t_{DH}	2.0x – 37		37		ns
23	Write command setup time	t_{WCS}	0.5x – 18		0		ns
24	$\overline{CAS}$ hold time ($\overline{CAS}$ before $\overline{RAS}$)	t_{CHR*1}	2.0x – 52		22		ns
25	$\overline{RAS}$ pre-charge $\overline{CAS}$ active time	t_{RPC}	1.5x – 31		24		ns
26	$\overline{CAS}$ setup time ($\overline{CAS}$ before $\overline{RAS}$)	t_{CSR*}	0.5x – 2		17		ns
27	$\overline{RAS}$ pre-charge time (Self refresh)	t_{RPS*2}	4.0x – 18		130		ns
28	$\overline{CAS}$ hold time (Self refresh)	t_{CHS*2}	0		0		ns
29	Refresh setup time	t_{CFL*}	1.0x – 10		27		ns
30	Refresh hold time	t_{CFH*}	1.0x – 17		20		ns
31	Write command pulse width	t_{WP}	2.0x – 42		32		ns
32	Write command hold time	t_{WCH}	1.5x – 36		20		ns
33	$\overline{OE}$ access time1	t_{OAC1}		2.5x – 53		40	ns
	$\overline{OE}$ access time2	t_{OAC2}		2.0x – 43		31	ns
34	After $\overline{OE}$ input data hold time	t_{OFF2}	0		0		ns

AC measuring conditions

- Output level: High = 0.7 V, Low = 0.3 V, CL = 50 pF
- Input level: High = 0.9 V, Low = 0.1 V

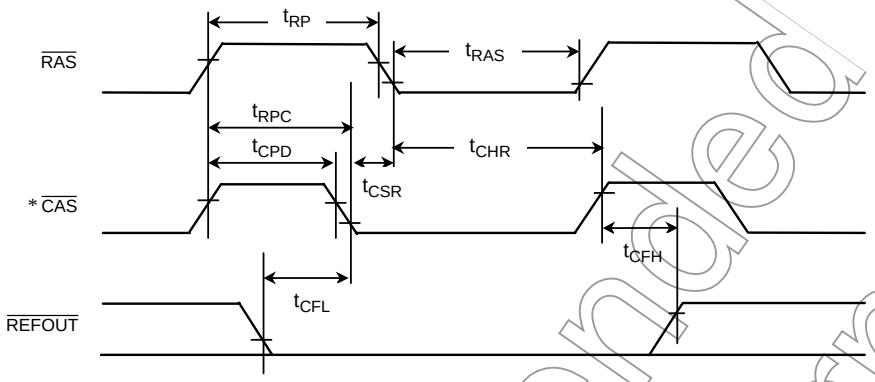
(7) DRAM read/write cycle



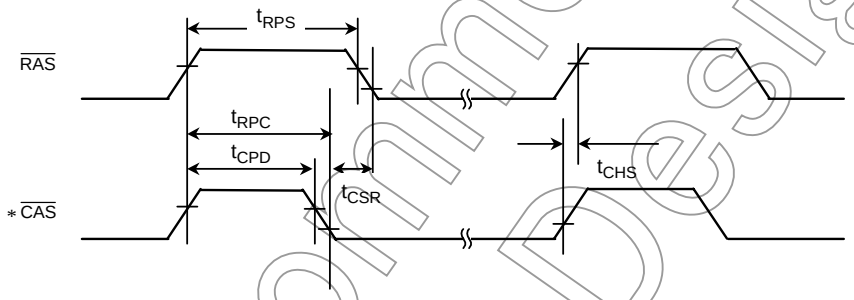
Note: CAS wave form in above figure, shows both of LCAS and UCAS.

(8) DRAM refresh cycle

$\overline{CAS}$ before $\overline{RAS}$ interval refresh cycle



$\overline{CAS}$ before $\overline{RAS}$ self refresh cycle



Note: $\overline{CAS}$ wave form in above figure, shows both of $\overline{LCAS}$ and $\overline{UCAS}$.

4.4 VLD Detect Characteristics

VLDVcc = Vcc, VLDGND = Vss

Parameter	Symbol	Conditions	Min	Typ.	Max	Unit
VREF current (Note 4)	IREF	3.6 V ≥ Vcc ≥ 2.7 V VREF = 1.5V		0.2	1	μA
Detect voltage accuracy (VLD0) (Note 1)	ADCTV0	3.6 V ≥ Vcc ≥ 2.7 V Vcc ≥ VLD0 ≥ VLDGND, VREF = 1.5 V (Note 2)	VLD0 × 0.98		VLD0 × 1.02	V
Not-detect voltage accuracy (VLD0) (Note1)	NADCTV0		VLD0 × 0.98		VLD0 × 1.02	V
VLD0 current (Note 3)	IVLD0			0.3	1	μA
Detect voltage accuracy (VLD1) (Note 1)	ADCTV1	3.6 V ≥ Vcc ≥ 2.7 V Vcc ≥ VLD0 ≥ VLDGND, VREF = 1.5 V (Note 2)	VLD0 × 0.98		VLD0 × 1.02	V
Not-detect voltage accuracy (VLD1) (Note 1)	NADCTV1		VLD0 × 0.98		VLD0 × 1.02	V
VLD1 current (Note 3)	IVLD1			0.3	1	μA
Detect voltage accuracy (VLD2) (Note 1)	ADCTV2	3.6 V ≥ Vcc ≥ 2.7 V Vcc ≥ VLD0 ≥ VLDGND, VREF = 1.5 V (Note 2)	VLD0 × 0.98		VLD0 × 1.02	V
Not-detect voltage accuracy (VLD2) (Note1)	NADCTV2		VLD0 × 0.98		VLD0 × 1.02	V
VLD2 current (Note 3)	IVLD2			0.3	1	μA

Note 1: "Detect voltage accuracy" means accuracy of voltage down, "Not-detect voltage accuracy" means accuracy of voltage rise-up.

Note 2: It is prohibit that setting over the Vcc voltage. (Example: Vcc = 2.7 V, Detect Voltage = 2.9 V)

Note 3: It shows highest detect voltage setting by each channel

Note 4: In case detecting voltage only for VLD2 (Vcc = VLD2), the setting "Detecting voltage = 2.6V" is possible.

Note 5: XT (Low-frequency oscillator) operate by Vcc and Vss swing

4.5 Serial Channel Timing (I/O internal mode)

(1) SCLK input mode

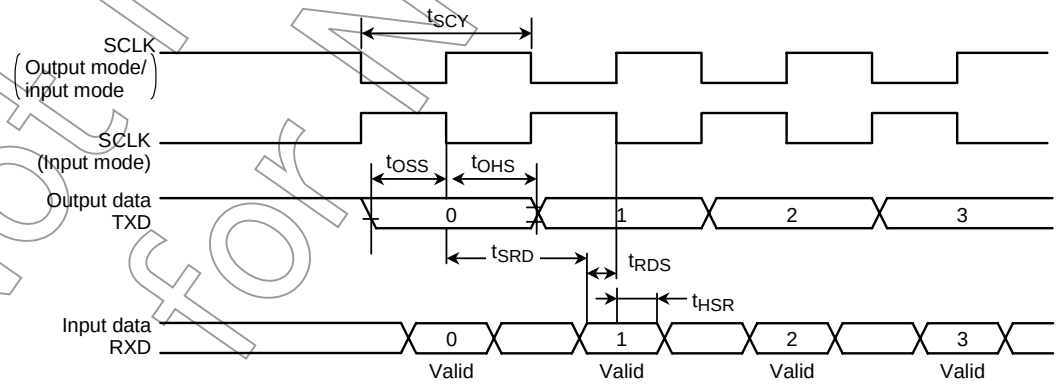
Parameter	Symbol	Variable		27 MHz		10 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK period	T_{SCY}	16X		0.59		1.6		μs
Output data → SCLK Rising/falling edge*	T_{OSS}	$t_{SCY}/2 - 4X - 110$ (Vcc = 3 V ± 10%)		334		290		ns
		$t_{SCY}/2 - 4X - 180$ (Vcc = 2 V ± 10%)		–		220		ns
SCLK rising/falling edge* → Output data hold	T_{OHS}	$t_{SCY}/2 + 2X + 0$		370		1000		ns
SCLK rising/falling edge* → Input data hold	T_{HSR}	3X + 10		121		310		ns
SCLK rising/falling edge* → Valid data input	T_{SRD}		$t_{SCY} - 0$		592		1600	ns
Valid data input → SCLK rising/falling edge*	T_{RDS}	0		0		0		ns

(2) SCLK output mode

Parameter	Symbol	Variable		10 MHz		27 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK period (Programmable)	T_{SCY}	16X	8192X	1.6	819	0.59	303	μs
Output data →SCLK rising edge	T_{OSS}	$t_{SCY}/2 - 40$		760		256		ns
SCLK rising edge → Output data hold	T_{OHS}	$t_{SCY}/2 - 40$		760		256		ns
SCLK rising edge → Input data hold	T_{HSR}	0		0		0		ns
SCLK rising edge → Valid data input	T_{SRD}		$t_{SCY}/2 - 1X - 180$		1320		375	ns
Valid data input → SCLK rising/falling edge*	T_{RDS}	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.

Value of 27 MHz and 10 MHz in above table, are that one on t_{SCY} = 16X case



4.6 Interrupt, Capture

(1) $\overline{NMI}$, INT0 to INT3 interrupts

Parameter	Symbol	Variable		10 MHz		27 MHz		Unit
		Min	Max	Min	Max	Min	Max	
$\overline{NMI}$, INT0 to INT3 low level width	t_{INTAL}	4X + 40		440		188		ns
$\overline{NMI}$, INT0 to INT3 high level width	t_{INTAH}	4X + 40		440		188		ns

4.7 SCOUT Pin AC Characteristics

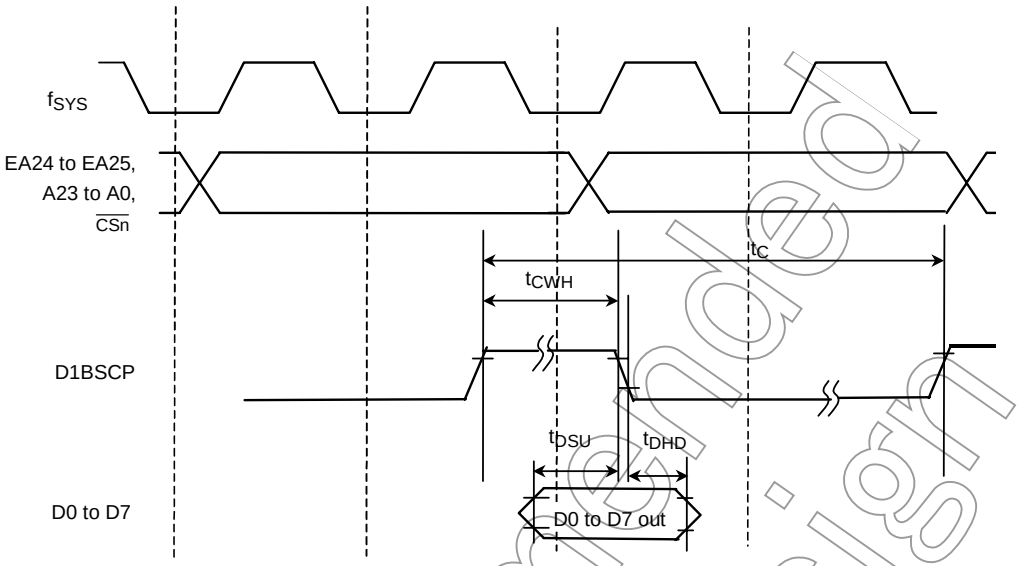
Parameter	Symbol	Variable		10 MHz		27 MHz		Condition	Unit
		Min	Max	Min	Max	Min	Max		
Low level width	t_{SCL}	0.5T – 10		40		8		Vcc ≥ 2.7 V	ns
		0.5T – 30		20		–		Vcc < 2.7 V	
High level width	t_{SCH}	0.5T – 10		40		8		Vcc ≥ 2.7 V	ns
		0.5T – 30		20		–		Vcc < 2.7 V	

Note: T = period of SCOUT

Measuring conditions

- Output level: High = 0.7 V, Low = 0.3 V, CL = 10 pF

4.8 LCD Controller (SR mode)



Read Bus Width	Type	Write Mode	Set Up Time (t_{DSU})	Hold Time (t_{DHD})	Clock High Width (t_{CWH})	Cycle (tc)	State/Cycle
Byte	A	Byte	0.5x – α	1.0x – β	1.5x – γ	4.0x	4.0x
		Nibble	0.5x – α	1.0x – β	1.0x – γ	2.0x	6.0x
		Bit	0.5x – α	1.0x – β	1.0x – γ	2.0x	18.0x
	B	Byte	1.0x – α	0.5x – β	2.0x – γ	4.0x	4.0x
		Nibble	1.0x – α	0.5x – β	1.0x – γ	2.0x	6.0x
		Bit	1.0x – α	0.5x – β	1.0x – γ	2.0x	18.0x
	C	Byte	1.0x – α	2.5x – β	1.5x – γ	6.0x	6.0x
		Nibble	1.0x – α	1.5x – β	2.5x – γ	5.0x	10.0x
		Bit	1.0x – α	1.0x – β	1.0x – γ	2.0x	20.0x
Word	A	Byte	0.5x – α	1.0x – β	1.0x – γ	2.0x	6.0x
		Nibble	0.5x – α	1.0x – β	1.0x – γ	2.0x	10.0x
		Bit	No support. Please use byte read mode.				
	B	Byte	1.0x – α	0.5x – β	1.0x – γ	2.0x	6.0x
		Nibble	1.0x – α	0.5x – β	1.0x – γ	2.0x	10.0x
		Bit	No support. Please use byte read mode				
	C	Byte	1.0x – α	1.5x – β	1.5x – γ	3.0x	8.0x
		Nibble	1.0x – α	1.5x – β	2.5x – γ	5.0x	20.0x
		Bit	No support. Please use byte read mode.				

* Value of alpha, beta and gamma are showed next page.

No.	Parameter	Symbol	Variable		27 MHz		10 MHz		Condition	Unit
			Min	Max	Min	Max	Min	Max		
1	D1BSCP rising-up → Data set up time	t_{DSU}	0.50x – 8		10		42		3.6 V ≥ Vcc ≥ 2.7 V	ns
			0.50x – 20		–		30		Vcc = 2.0 V ± 10%	
			1.00x – 8		29		92		3.6 V ≥ Vcc ≥ 2.7 V	
			1.00x – 20		–		80		Vcc = 2.0 V ± 10%	
2	D1BSCP falling down → Data hold time	t_{DHD}	0.50x – 8		10		42		3.6 V ≥ Vcc ≥ 2.7 V	
			0.50x – 20		–		30		Vcc = 2.0 V ± 10%	
			1.00x – 8		32		92		3.6 V ≥ Vcc ≥ 2.7 V	
			1.00x – 20		–		80		Vcc = 2.0 V ± 10%	
			1.50x – 8		50		142		3.6 V ≥ Vcc ≥ 2.7 V	
			1.50x – 20		–		130		Vcc = 2.0 V ± 10%	
			2.50x – 8		87		242		3.6 V ≥ Vcc ≥ 2.7 V	
			2.50x – 20		–		230		Vcc = 2.0 V ± 10%	
3	D1BSCP → High width	t_{CWH}	1.00x – 5		32		95		3.6 V ≥ Vcc ≥ 2.7 V	
			1.00x – 15		–		85		Vcc = 2.0 V ± 10%	
			1.50x – 5		50		145		3.6 V ≥ Vcc ≥ 2.7 V	
			1.50x – 15		–		135		Vcc = 2.0 V ± 10%	
			2.00x – 5		69		195		3.6 V ≥ Vcc ≥ 2.7 V	
			2.00x – 15				185		Vcc = 2.0 V ± 10%	
			2.50x – 5		87		245		3.6 V ≥ Vcc ≥ 2.7 V	
			2.50x – 15		–		235		Vcc = 2.0 V ± 10%	
4	D1BSCP → Clock cycle	t_C	2.00x		74		200		3.6 V ≥ Vcc ≥ 2.7 V	
			2.00x		–		200		Vcc = 2.0 V ± 10%	
			3.00x		111		300		3.6 V ≥ Vcc ≥ 2.7 V	
			3.00x		–		300		Vcc = 2.0 V ± 10%	
			4.00x		148		400		3.6 V ≥ Vcc ≥ 2.7 V	
			4.00x		–		400		Vcc = 2.0 V ± 10%	
			5.00x		185		500		3.6 V ≥ Vcc ≥ 2.7 V	
			5.00x		–		500		Vcc = 2.0 V ± 10%	
			6.00x		222		600		3.6 V ≥ Vcc ≥ 2.7 V	
			6.00x		–		600		Vcc = 2.0 V ± 10%	

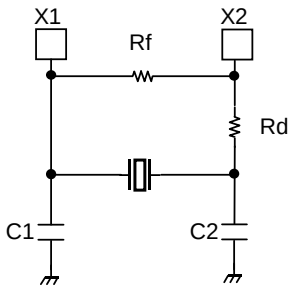
Note: The reading characteristics of display data from the memory which does not define above table, is same as 4.3 “AC Characteristics”.

4.9 Recommended Crystal Oscillation Circuit

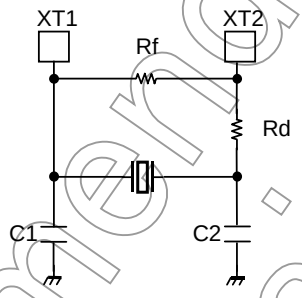
TMP91C016 is evaluated by below oscillator vender. When selecting external parts, make use of this information.

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) TMP91C016 recommended ceramic oscillator: Murata Manufacturing. 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 [°C]
TMP91C016	2.00	CSTLS2M00G56-B0	(47)	(47)	Open	0	1.8 to 2.2	−40 to +85
	2.50	CSTLS2M50G56-B0	(47)	(47)	Open	0		
	10.00	CSTLS10M0G53-B0	(15)	(15)	Open	0		
	12.50	CSALA12M5T55093-B0	30	30	Open	0		
		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 [°C]
TMP91C016	4.00	CSTLS4M00G56-B0	(47)	(47)	Open	0	2.7 to 3.6	−40 to +85
	6.750	CSTLS6M75G56-B0	(47)	(47)	Open	0		
	12.50	CSALA12M5T55-B0	30	30	Open	0		
		CSTLA12M5T55-B0	(30)	(30)	Open	0		
	20.00	CSALS20M0X53-B0	5	5	Open	0		
		CSTLS20M0X51-B0	(5)	(5)	Open	0		
	27.00	CSALS27M0X51-B0	Open	Open	10K	0		
	32.00	CSALA32M0X51-B0	3	3	Open	0		

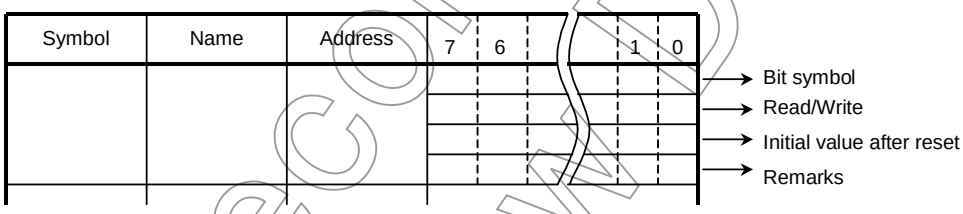
- The valves enclosed blackest in C1 and C2 columns apply to condenser built-in type.
- The product numbers and specifications of the resonators by Murata Manufacturing Co., Ltd. are subject to change.
  For up-to-date information, please refer to the following URL:
  http://www.murata.co.jp/search/index.html

5. Table of SFRs

The SFRs (Special function registers) include the I/O ports and peripheral control registers allocated to the 4-Kbyte address space from 000000H to 000FFFH.

(1) I/O ports
(2) I/O port control
(3) Interrupt control
(4) Chip select/wait control
(5) Clock gear
(6) DFM control
(7) 8-bit timer
(8) UART/SIO channel
(9) DRAM controller
(10) Watchdog timer
(11) RTC (Real time clock)
(12) Melody/alarm generator
(13) MMU
(14) LCD control
(15) HVC (Horizontal and vertical converter)
(16) HPLT, VLD

Table layout



Note: "Prohibit RMW" in the table means that you cannot use RMW instructions on these registers.

Example: When setting bit0 only of the register PxCR, the instruction "SET 0, (PxCR)" cannot be used. The LD (Transfer) instruction must be used to write all eight bits.

Read/Write

R/W: Both read and write are possible.

R: Only read is possible.

W: Only write is possible.

W*: Both read and write are possible (when this bit is read as "1")

Prohibit RMW: Read-modify-write instructions are prohibited. (The EX, ADD, ADC, BUS, SBC, INC, DEC, AND, OR, XOR, STCF, RES, SET, CHG, TSET, RLC, RRC, RL, RR, SLA, SRA, SLL, SRL, RLD and RRD instruction are read-modify-write instructions.)

R/W*: Read-modify-write instructions are prohibited when controlling the pull-up resistor.

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	P5FC2
DH	P5
EH	P5UDE
FH	

Address	Name
0010H	P6FC3
1H	
2H	P6
3H	P7
4H	P6CR
5H	P6FC
6H	P7CR
7H	P7FC
8H	P6UE
9H	P9
AH	
BH	P6FC2
CH	P7FC2
DH	P9FC
EH	P9UE
FH	P7UDE

Address	Name
0022H	PBUDE
1H	
2H	PB
3H	PC
4H	PBCR
5H	PBFC
6H	PCCR
7H	PCFC
8H	PCUDOE
9H	PD
AH	PDFC
BH	PDCR
CH	PDUE
DH	
EH	
FH	

[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	INTE0
1H	INTE12
2H	INTE3ALM4
3H	INTEALM01
4H	INTEALM23
5H	INTETA01
6H	INTETA23
7H	INTERTCKEY
8H	INTES0
9H	INTES1
AH	INTELCD
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	EMCCR0
4H	EMCCR1
5H	EMCCR2
6H	EMCCR3
7H	EMCCR4
8H	DFMCR0
9H	DFMCR1
AH	
BH	
CH	
DH	
EH	
FH	

Note: Do not access to the unnamed addresses, e.g., 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	TA1FFCR
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	SC0MOD1
6H	
7H	SIRCR
8H	SC1BUF
9H	SC1CR
AH	SC1MOD0
BH	BR1CR
CH	BR1ADD
DH	SC1MOD1
EH	
FH	

[9] DRAMC

Address	Name
0430H	DREFCR
1H	DMEMCR
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[10] WDT

Address	Name
0300H	WDMOD
1H	WDCR
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[11] RTC

Address	Name
0320H	SECR
1H	MINR
2H	HOURR
3H	DAYR
4H	DATER
5H	MONTHR
6H	YEARR
7H	PAGER
8H	RESTR
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[12] MLD

Address	Name
0330H	ALM
1H	MELALMC
2H	MELFL
3H	MELFH
4H	ALMINT
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Note: Do not access to the unnamed addresses, e.g., addresses to which no register has been allocated.

Table 5.3 Address Map SFRs

[13] MMU

Address	Name
0350H	LOCAL0
1H	LOCAL1
2H	LOCAL2
3H	LOCAL3
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[14] LCD

Address	Name
0360H	LCDSAL
1H	LCDSAH
2H	LCDSIZE
3H	LCDCTL
4H	LCDFFP
5H	
6H	LCDCTL2
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[15] VLD

Address	Name
0440H	VLDCR0
1H	VLDCR1
2H	VLDCR2
3H	
4H	
5H	HPCTST1
6H	HPCTST2
7H	
8H	
9H	VLDCTL
AH	
BH	
CH	
DH	
EH	
FH	

[16] HVC

Address	Name
0450H	HVREGA0
1H	HVREGA1
2H	HVREGA2
3H	HVREGA3
4H	HVREGA4
5H	HVREGA5
6H	HVREGA6
7H	HVREGA7
8H	HVREGB0
9H	HVREGB1
AH	HVREGB2
BH	HVREGB3
CH	HVREGB4
DH	HVREGB5
EH	HVREGB6
FH	HVREGB7

Note: Do not access to the unnamed addresses, e.g., addresses to which no register has been allocated.

(1) I/O ports

Symbol	Name	Address	7	6	5	4	3	2	1	0
P1	Port 1	01H	P17	P16	P15	P14	P13	P12	P11	P10
			R/W							
			Data from external port (Output latch register is cleared to 0)							
P2	Port 2	06H	P27	P26	P25	P24	P23	P22	P21	P20
			R/W							
			1	1	1	1	1	1	1	1
P5	Port 5	0DH		P56			P53	P52		RDE
				R/W			R/W	R/W		R/W
			Data from external port (Output latch register is set to 1)							
P6	Port 6	12H	P67	P66	P65	P64	P63		P61	P60
			R/W							
			Data from external port (Output latch register is set to 1)							
P7	Port 7	13H				P74	P73	P72	P71	P70
						R/W				
						Data from external port (Output latch register is set to 1)				
P9	Port 9	19H	P97	P96	P95	P94	P93	P92	P91	P90
			R							
			Data from external port							
PB	Port B	22H			PB5	PB4	PB3	PB2	PB1	PB0
					R/W					
					Data from external port (Output latch register is set to 1)					
PC	Port C	23H	PC7	PC6	PC5	PC4	PC3			
			R/W	R/W	R/W	R/W	R/W			
			Data from external port (Output latch register is set to 1)							
PD	Port D	29H	PD7	PD6		PD4	PD3	PD2	PD1	PD0
			R/W	R/W		R/W	R/W	R/W	R/W	R/W
			Data from external port (Output latch register is set to 1)							

(2) I/O port control (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P1CR	Port 1 control	04H (Prohibit RMW)	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
			W							
			0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
			0: Input 1: Output							
P2FC	Port 2 function	09H (Prohibit RMW)	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	Port 5 control	0AH (Prohibit RMW)		P56C			P53C	P52C		
				W			W	W		
				0			0	0		
				0: Input 1: Output						
P5FC	Port 5 function	0BH (Prohibit RMW)		P56F			P53F	P52F		
				W			W	W		
				0			0	0		
				0: Port 1: R/$\overline{W}$			0: Port 1: $\overline{EXWR}$	0: Port 1: $\overline{HWR}$		
P5FC2	Port 5 function 2	0CH (Prohibit RMW)		P56F2				P52F2		
				W				W		
				0				0		
				MSK Logic select 0: Clk by 1 1: Clk by 0				0: <P52F> 1: INT3		
P5UDE	Port 5 resister	0EH (Prohibit RMW)		P56U			P53U	UDEP52	P52UD	
				W			W	W	W	
				1			1	1	0	
				Pull up 0: Disable 1: Enable			Pull up 0: Disable 1: Enable	Pull up/down 0: Disable 1: Enable	Resistance control 0: Pull up 1: Pull down	
P6CR	Port 6 control	14H (Prohibit RMW)	P67C	P66C	P65C	P64C	P63C		P61C	P60C
			W						W	
			0	0	0	0	0		0	0
			0: Input 1: Output						0: Input 1: Output	
P6FC	Port 6 function	15H (Prohibit RMW)	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
			W							
			0	0	0	0	0	0	0	0
			0: Port 1: $\overline{LCAS}$ or $\overline{REFOUT}$	0: Port 1: $\overline{UCAS}$ or $\overline{WE}$	0: PORT 1: EA25	0: Port 1: EA24	0: Port 1: $\overline{CS3}$ or $\overline{RAS}$	0: $\overline{CS2}$ 1: $\overline{CS2A}$	0: Port 1: $\overline{CS1}$	0: Port 1: $\overline{CS0}$
P6FC2	Port 6 function 2	1BH (Prohibit RMW)	P67F2	P66F2	P65F2	P64F2	-	P65F2D		
			W				W	W		
			0				0	0		
			0:<P67F> 1: $\overline{LDS}$	0: <P66F> 1: $\overline{UDS}$	0: <P65F> 1: $\overline{CS2C}$	0: <P64F> 1: $\overline{CS2B}$	Always write 0	0: <P65F2> 1: VEECLK		
P6UE	Port 6 pull-up control	18H (Prohibit RMW)	P67U	P66U	P65U	P64U	P63U		P61U	P60U
			W						W	
			0	0	1	1	0		1	1
			Pull up 0: Disable 1: Enable	Pull up 0: Disable 1: Enable	Pull up 0: Disable 1: Enable	Pull up 0: Disable 1: Enable	Pull up 0: Disable 1: Enable		Pull up 0: Disable 1: Enable	Pull up 0: Disable 1: Enable
P6FC3	Port 6 function 3	10H (Prohibit RMW)			P65F3				-	P60F3
					W				W	W
					0				0	0
					0: Normal 1: LCLK2				Always write 0	0: Normal 1: LCLK0

I/O port control (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P7CR	Port 7 control	16H (Prohibit RMW)				P74C	P73C	P72C	P71C	P70C
						W				
						0	0	0	0	0
						0 : Input 1 : Output				
P7FC	Port 7 function	17H (Prohibit RMW)				P74F	P73F	P72F	P71F	P70F
						W				
						0	0	0	0	0
						0: Port 1: $\overline{NMI}$	0: Port 1: $\overline{EXRD}$	0: Port 1: $\overline{CS2E}$	0: Port 1: $\overline{CS2D}$	0: Port 1: TA1OUT
P7FC2	Port 7 function 2	1CH (Prohibit RMW)				P74F2	P73F2		P71F2	P70F2
						W				
						0	0		0	0
						0: <P74F> 1: $\overline{WE}$ or $\overline{CAS}$	0: <P73F> 1: $\overline{DRAMOE}$		0: <P71F> 1: OPTTX0	0: <P70F> 1: SCOUT
P7UDE	Port 7 pull up/down control	1FH (Prohibit RMW)			P72UD	P74U	P73U	UDEP72	P71U	P70U
					W					
					0	1	1	0	0	0
					Resistance control 0: Pull up 1: Pull down	Pull up 0: Disable 1: Enable	Pull up 0: Disable 1: Enable	Resistance control 0: Pull up 1: Pull down	Pull up 0: Disable 1: Enable	Pull up 0: Disable 1: Enable
P9FC	Port 9 function	1DH (Prohibit RMW)	P97F	P96F	P95F	P94F	P93F	P92F	P91F	P90F
			W							
			0	0	0	0	0	0	0	0
			0: Key-in disable 1:Key-in enable							
P9UE	Port 9 pull up control	1EH (Prohibit RMW)	P97U	P96U	P95U	P94U	P93U	P92U	P91U	P90U
			W							
			1	1	1	1	1	1	1	1
			Pull up 0: Disable 1: Enable	Pull up 0: Disable 1: Enable	Pull up 0: Disable 1: Enable	Pull up 0: Disable 1: Enable	Pull up 0: Disable 1: Enable	Pull up 0: Disable 1: Enable	Pull up 0: Disable 1: Enable	Pull up 0: Disable 1: Enable
PBCR	Port B control	24H (Prohibit RMW)			PB5C	PB4C	PB3C	PB2C	PB1C	PB0C
					W					
					0	0	0	0	0	0
					0: Input 1: Output					
PBFC	Port B function	25H (Prohibit RMW)			PB5F	PB4F	PB3F			
					W					
					0	0	0			
					0: Port 1: INT2	0: Port 1: INT1	0: Port 1: INT0			
PBUDE	Port B pull up/down control	20H (Prohibit RMW)	PB5UD	PB4UD	UDEPB5	UDEPB4	PB3U	PB2U	PB1U	PB0U
			W							
			0	0	0	0	1	0	0	0
			Resistance control 0: Pull up 1: Pull down	Resistance control 0: Pull up 1: Pull down	Resistance control 0: Pull up 1: Pull down	Resistance control 0: Pull up 1: Pull down	Pull-up resistance 0: Disable 1: Enable	Pull-up resistance 0: Disable 1: Enable	Pull-up resistance 0: Disable 1: Enable	Pull-up resistance 0: Disable 1: Enable

I/O port control (3/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
PCCR	Port C control	26H	PC7C	PC6C	PC5C	PC4C	PC3C			
			W							
		(Prohibit RMW)	1	1	0	0	0			
			0: Input		1: Output					
PCFC	Port C function	27H			PC5F		PC3F			
					W		W			
		(Prohibit RMW)			0		0			
					0: Port 1: SCLK1		0: Port 1: TXD1			
PCUDOE	Port C open drain	28H			ODEPC3	PC5UD	PC4UD	UDEPC5	UDEPC4	PC3U
					W					
		(Prohibit RMW)			0	0	0	0	0	0
					0: 3 states 1: Open drain	Resistance control 0: Pull up 1: Pull down	Resistance control 0: Pull up 1: Pull down	Resistance control 0: Disable 1: Enable	Resistance control 0: Disable 1: Enable	Pull up 0: Disable 1: Enable
PDFC	Port D function	2AH	PD7F	PD6F		PD4F	PD3F	PD2F	PD1F	PD0F
			W	W		W	W	W	W	W
		(Prohibit RMW)	0	0		0	0	0	0	0
			0: Port 1: MLDALM	0: Port 1: $\overline{\text{ALARM}}$ at <PD6> = 1 1: $\overline{\text{MLDALM}}$ at <PD6> = 0		0: Port 1: DOFFB	0: Port 1: DLEBCD	0: Port 1:D3BFR	0: Port 1: D2BLP	0: Port 1: D1BSCP
PDCR	Port D control	2BH	PD7C	PD6C		PD4C	PD3C	PD2C	PD1C	PD0C
			W	W		W	W	W	W	W
		(Prohibit RMW)	0	0		0	0	0	0	0
			0: Input	1: Output		0: Input		1: Output		
PDUDE	Port D pull up/down control	2CH	PD7U	PD6U	-	PD4D	PD3U	PD2U	PD1U	PD0U
			W							
		(Prohibit RMW)	0	0	0	1	1	1	1	1
			Pull up 0: Disable 1: Enable	Pull up 0: Disable 1: Enable	Always write 0	Pull up 0: Disable 1: Enable	Pull up 0: Disable 1: Enable	Pull up 0: Disable 1: Enable	Pull up 0: Disable 1: Enable	Pull up 0: Disable 1: Enable

(3) Interrupt control (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE0	Interrupt enable 0	90H					INT0			
							I0C	I0M2	I0M1	I0M0
							R	R/W		
							0	0	0	0
							1: INT0	Interrupt level		
INTE12	Interrupt enable INT2/1	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 INT3 and 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 RTCKEY	Interrupt enable RTC and KEY	97H	INTKEY				INTRTC			
			IKC	IKM2	IKM1	IKM0	IRC	IRM2	IRM1	IRM0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTKEY	Interrupt level			1: INTRTC	Interrupt level		

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		
INTE LCD	Interrupt enable LCD	9AH	INT LCD							
			ILCD1C	ILCDM2	ILCDM1	ILCDM0				
			R	R/W						
			0	0	0	0				
			1: INTLCD	Interrupt level						
INTETC 01	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
INTETC 23	Interrupt enable TC2/3	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	Interrupt enable PC0/1	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

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					
					0	0	0	0	0	0
					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	W
			0	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 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		B0OM1	B0OM0	B0BUS	B0W2	B0W1	B0W0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			0: Disable 1: Enable		00: ROM/SRAM 01: 10: Reserved 11:		Data bus width 0: 16 bits 1: 8 bits	000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits		100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits
B1CS	Block 1 CS/WAIT control register	C1H (Prohibit RMW)	B1E		B1OM1	B1OM0	B1BUS	B1W2	B1W1	B1W0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			0: Disable 1: Enable		00: ROM/SRAM 01: 10: Reserved 11:		Data bus width 0: 16 bits 1: 8 bits	000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits		100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits
B2CS	Block 2 CS/WAIT control register	C2H (Prohibit RMW)	B2E	B2M	B2OM1	B2OM0	B2BUS	B2W2	B2W1	B2W0
			W	W	W	W	W	W	W	W
			1	0	0	0	0	0	0	0
			0: Disable 1: Enable	0: 16 M Area 1: Area setting	00: ROM/SRAM 01: 10: Reserved 11:		Data bus width 0: 16 bits 1: 8 bits	000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits		100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits
B3CS	Block 3 CS/WAIT control register	C3H (Prohibit RMW)	B3E		B3OM1	B3OM0	B3BUS	B3W2	B3W1	B3W0
			W		W	W	W	W	W	W
			0		0	0	0	0	0	0
			0: Disable 1: Enable		00: ROM/SRAM 01: Reserved 10: DRAMC 11: Reserved		Data bus width 0: 16 bits 1: 8 bits	000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits		100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits
BEXCS	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 bits 1: 8 bits	000: 2 waits 001: 1 wait 010: (1 + N) waits 011: 0 waits		100: Reserved 101: 3 waits 110: 4 waits 111: 8 waits
MSAR0	Memory start address register 0	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 register 0	C9H	V20	V19	V18	V17	V16	V15	V14 to 9	V8
			R/W							
			1	1	1	1	1	1	1	1
			CS0 area size 0: Enable to address comparison							
MSAR1	Memory start address register 1	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 register 1	CBH	V21	V20	V19	V18	V17	V16	V15 to 9	V8
			R/W							
			1	1	1	1	1	1	1	
			CS1 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 register 2	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 register 2	CDH	V22	V21	V20	V19	V18	V17	V16	V15
			R/W							
			1	1	1	1	1	1	1	1
			CS2 area size 0: Enable to address comparison							
MSAR3	Memory start address register 3	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 register 3	CFH	V22	V21	V20	V19	V18	V17	V16	V15
			R/W							
			1	1	1	1	1	1	1	1
			CS3 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 register 0	E0H	XEN	XTEN	RXEN	RXTEN	RSYSCK	WUEF	PRCK1	PRCK0
			R/W							
			1	1	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	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	SELDRV	DRVE
				R/W	R/W	R/W	R/W	R/W	R/W	R/W
				0	1	0	1	1	0	0
				0: fs 1: f_{FPH}	Warming-up time 00: Reserved 01: 2^8/input frequency 10: 2^{14}/input frequency 11: 2^{16}/input frequency		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

Clock gear (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
EMCCR0	EMC control register 0	E3H	PROTECT	-	-	-	-	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	Always write 0	Always write 1	Always write 0	Always write 0	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	Set protection ON/OFF by 1st-key, 2nd-key Protection ON: Set continually EMCCR1 = 5AH, EMCCR2 = A5H in1st-key Protection OFF: Set continually EMCCR1 = A5H, EMCCR2 = 5AH in 2nd-key							
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 detect 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		
EMCCR4	EMC control register 4	E7H							TA3MLDE3	TA3LCDE
									R/W	R/W
									0	0
									MLD clock selection 0: 32 kHz 1: Timer 3	LCD clock selection 0: 32 kHz 1: Timer 3

Note: EMCCR1/2

If protection is on,the following SFRs can't be rewrite

1. CS/WAIT Controller
   B0CS, B1CS, B2CS, B3CS, BEXCS,
   MSAR0/1/2/3, MAMR0/1/2/3
2. MMU
   LOCAL0/1/2/3
3. Clock gear (EMCCR1, EMCCR2 can be written to)
   SYSCR0, SYSCR1, SYSCR2, EMCCR0, EMCCR3
4. DFM
   DFMCR0, DRMCR1
5. PORT
   P2FC, P5CR, P5FC, P5FC2, P6CR, P6FC, P6FC2
   P7CR, P7FC, P7FC2, PDCR, PDFC
6. DRAMC
   DMEMCR, DREFCR

(6) DFM control

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}		Lockup flag 0: Out of LUP 1: In LUP	Lockup time 0: $2^{12}/f_{OSCH}$ 1: $2^{10}/f_{OSCH}$				
DFMCR0	DFM control register 1	E9H	D7	D6	D5	D4	D3	D2	D1	D0
			R/W							
			0	0	0	1	0	0	1	1
			DFM correction Input frequency 4 to 6.75 MHz (at 2.7 V to 3.6 V): Write 0BH Input frequency 2 to 2.5 MHz (at 2 V ± 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: Run	8-bit timer run/stop control 0: Stop and 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 and 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 PWM cycle 10: 2^7 11: 2^8		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)					TA1FFC1	TA1FFC0	TA1FFIE	TA1FFIS
							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: Run	8-bit timer run/stop control 0: Stop and 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 and 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: 2^6PWM cycle 10: 2^7 11: 2^8		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)					TA3FFC1	TA3FFC0	TA3FFIE	TA3FFIS
							R/W		R/W	
							1	1	0	0
							00: Invert TA3FF 01: Set TA3FF 10: Clear TA3FF 11: Don't care		1: TA3FF invert enable	0: TMRA2 1: TMRA3 inverision

(8) UART/SIO 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 (Prohibit RMW)	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
			R (Receiving)/W (Transmission)							
			Undefined							
SC0CR	Serial channel 0 control	201H	RB8	EVEN	PE	OERR	PERR	FERR		
			R	R/W		R (Cleared to 0 by reading)				
			Undefined	0	0	0	0	0		
			Receiveing data bit 8	Parity 0: Odd 1: Even	1: Parity enable	1: Error Overrun	Parity	Framing		
SC0 MOD0	Serial channel 0 mode0	202H	TB8	-	RXE	-	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Transfer data bit 8	Always write 0	Receive function 0: Receive disable 1: Receive enable	Always write 0	00: I/O Interface 01: UART 7 bits 10: UART 8 bits 11: UART 9 bits		00: TA0TRG 01: Baud rate generater 10: Internal clock f_{SYS} 11: IrDA clock	
BR0CR	Baud rate control	203H	–	BR0ADDE	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
			R/W							
			0	0	0		0	0	0	0
			Always write 0	1: (16 – K)/16 divded enable	00: φT0 01: φT2 10: φT8 11: φT32		Setting the dividied frequency "N" (0 to F)			
BR0 ADD	Serial channel 0 K setting register	204H					BR0K3	BR0K2	BR0K1	BR0K0
							R/W			
							0	0	0	0
							Sets the frequency divisor "K" (Divided by N + (16 – K)/16)			
SC0 MOD1	Serial channel 0 mode1	205H	I2S0	FDPX0						
			R/W	R/W						
			0	0						
			IDLE2 0: Stop 1: Run	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 effective SIRRxD pulse width Pulse width more than 2x × (Set value + 1) + 100ns Possibale: 1 to 14 Impossible: 0, 15			

UART/SIO channel (2/2)

(8-3) UART/SIO channel 1

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC1BUF	Serial channel 1 buffer	208H (Prohibit RMW)	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
			R (Receiving)/W (Transmission)							
			Undefined							
SC1CR	Serial channel 1 control	209H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Cleared to 0 by reading)			R/W	
			Undefined	0	0	0	0	0	0	0
			Receiving data bit 8	Parity 0: Odd 1: Even	1: Parity enable	1: Error Overrun	1: Error Parity	1: Error Framing	0: SCLK1↑ 1: SCLK1↓	1: SCLK1 Pin
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 bit8	1: CTS enable	1: Receive enable	1: Wake up enable	00: I/O Interface 01: UART 7 bits 10: UART 8 bits 11: UART 9 bits		00: TA0TRG 01: Baud rate generator 10: Internal clock f_{SYS} 11: External clock SCLK1	
BR1CR	Baud rate control	20BH	–	BR1ADDE	BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
			R/W							
			0	0	0	0	0	0	0	0
			Always write 0	1: (16 – K)/16 divded enable	00: φT0 01: φT2 10: φT8 11: φT32		Setting the dividied frequency “N” (0 to F)			
BR1 ADD	Serial channel 1 K setting register	20CH					BR1K3	BR1K2	BR1K1	BR1K0
							R/W			
							0	0	0	0
							Sets the frequency divisor “K” (Divided by N + (16 – K)/16)			
SC1 MOD1	Serial channel 1 mode1	20DH	I2S1	FDPX1						
			R/W	R/W						
			0	0						
			IDLE2 0: Stop 1: Run	I/O interface mode 1: Full duplex 0: Half duplex						

(9) DRAM control

Symbol	Name	Address	7	6	5	4	3	2	1	0
DREFCR	DRAM function control	430H	DMI	RS2	RS1	RS0	RW2	RW1	RW0	RC
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			Dummy cycle 0: Disable 1: Dummy cycle	Refresh-cycle insetsion interval 000: 31 states 001: 110 states 010: 220 states 011: 450 states 100: 900 states 101: 1200 states 110: 1800 states 111: 2700 states			Refresh-cycle width 000: 2 states 001: 3 states 010: 4 states 011: 5 states 100: 6 states 101: 7 states 110: 8 states 111: 9 states			Refresh-cycle 0: Disable 1: Enable
DMEMCR	DRAM memory control	431H (Prohibit RMW)	SRFC	-	-	MACM	MUXE	MUXW1	MUXW0	MAC
			W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			1	0	0	0	0	0	0	0
			Self -refresh 0: Self -refresh 1: Release	Always write 0	Always write 0	Memory access control 0: Normal 1: Slow	Multiplex address 0: Disable 1: Enable	Multiplex address length 00: 8 bits 01: 9 bits 10: 10 bits 11: 11 bits		Memory access control 0: Disable 1: Enable

(10) Watchdog timer

Symbol	Name	Address	7	6	5	4	3	2	1	0
WDMOD	WDT MODE register	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: Stop 1: Run	1: RESET connect internally WDT out pin to RESET pin	Always write 0
WDCR	WD control	301H (Prohibit RMW)	–							
			W							
			–							
			B1H: WDT disable 4EH: WDT clear							

(11) RTC (Real time clock)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SECR	Second register	320H		SE6	SE5	SE4	SE3	SE2	SE1	SE0
				R/W						
				Undefined						
			0 is read	40 s	20 s	10 s	8 s	4 s	2 s	1 s
MINR	Minute register	321H		MI6	MI5	MI4	MI3	MI2	MI1	MI0
				R/W						
				Undefined						
			0 is read	40 min	20 min	10 min	8 min	4 min	2 min	1 min
HOURR	Hour register	322H			HO5	HO4	HO3	HO2	HO1	HO0
					R/W					
					Undefined					
			0 is read		20 h/ (PM/AM)	10 h	8 h	4 h	2 h	1 h
DAYR	Day register	323H						WE2	WE1	WE0
								R/W		
								Undefined		
			0 is read					W2	W1	W0
DATER	Date register	324H			DA5	DA4	DA3	DA2	DA1	DA0
					R/W					
					Undefined					
			0 is read		20 d	10 d	8 d	4 d	2 d	1 d
MONTHR	Month register	325H				MO4	MO3	MO2	MO1	MO0
						R/W				
						Undefined				
		Page 0	0 is read			10 month	8 month	4 month	2 month	1 month
		Page 1	0 is read							0: Indicator for 12 hour 1: Indicator for 24hour
YEARR	Year register	326H	YE7	YE6	YE5	YE4	YE3	YE2	YE1	YE0
			R/W							
			Undefined							
		Page 0	80 y	40 y	20 y	10 y	8 y	4 y	2 y	1 y
		Page 1	0 is read						Leap year setting	
PAGER	Page register	327H (Prohibit RMW)				ADJUST	ENATMR	ENAALM		PAGE
						W	R/W			R/W
						Undefined				Undefined
			0 is read			Adjust	Timer enable	Alarm enable	0 is read	Page setting
RESTR	Reset register	328H (Prohibit RMW)	DIS1HZ	DIS16HZ	RSTTMR	RSTALM	RE3	RE2	RE1	RE0
			W							
			Undefined							
			0: 1 Hz	0: 16 Hz	1: Reset timer	1: Reset alarm	Always write 0			

(12) Melody/alarm generator

Symbol	Name	Address	7	6	5	4	3	2	1	0
ALM	Alarm-pattern register	330H	AL8	AL7	AL6	AL5	AL4	AL3	AL2	AL1
			R/W							
			0							
			Alarm-pattern setting							
MELALMC	Melody/alarm control register	331H	FC1	FC0	ALMINV	-	-	-	-	MELALM
			R/W		R/W					
			0		0	0	0	0	0	0
			Free run counter control 00: Hold 01: Restart 10: Clear 11: Clear and start		Alarm wave invert 1: Invert	Always write 0				Output wave 0: Alarm 1: Melody
MELFL	Melody frequency register-L	332H	ML7	ML6	ML5	ML4	ML3	ML2	ML1	ML0
			R/W							
			0							
			Melody frequency setting (low 8 bits)							
MELFH	Melody frequency register-H	333H	MELON				ML11	ML10	ML9	ML8
			R/W				R/W			
			0				0			
			Melody counter control 0: Stop and Clear 1: Start				Melody frequency setting (High 4 bits)			
ALMINT	Alarm interrupt enable register	334H			-	IALM4E	IALM3E	IALM2E	IALM1E	IALM0E
					R/W	R/W				
					0	0				
					Always write 0	INTALM4 to INTALM0 interrupt output enable				

(13) MMU

Symbol	Name	Address	7	6	5	4	3	2	1	0
LOCAL0	LOCAL0 control register	350H	L0E					L0EA22	L0EA21	L0EA20
			R/W					R/W		
			0					0		
			Bank for LOCAL0 0: Disable 1: Enable					Setting bank number for LOCAL0 000 Setting is prohibited because it predetend common0 area		
LOCAL1	LOCAL1 control register	351H	L1E					L1EA23	L1EA22	L1EA21
			R/W					R/W		
			0					0		
			Bank for LOCAL1 0: Disable 1: Enable					Setting bank number for LOCAL0 011 Setting is prohibited because it predetend common1 area		
LOCAL2	LOCAL2 control register	352H	L2E					L2EA23	L2EA22	L2EA21
			R/W					R/W		
			0					0		
			Bank for LOCAL2 0: Disable 1: Enable					Setting bank number for LOCAL0 111 Setting is prohibited because it predetend common2 area		
LOCAL3	LOCAL3 control register	353H	L3E			L3EA26	L3EA25	L3EA24	L3EA23	L3EA22
			R/W					R/W		
			0					0		
			Bank for LOCAL3 0: Disable 1: Enable			01000 to 01011 : $\overline{CS2D}$ 00000 to 00011 : $\overline{CS2B}$ 00100 to 00111 : $\overline{CS2C}$		01100 to 01111 : $\overline{CS2E}$ 10000 to 1111 : Set prohibition		

(14) LCD control

Symbol	Name	Address	7	6	5	4	3	2	1	0
LCDSAL	LCD start address register low	360H	SAL15	SAL14	SAL13	SAL12		-	-	MODE
			R/W					R/W	R/W	R/W
			0					0	0	0
			SR mode start address A15 to A12					Always write 0	Always write 0	Mode 0: RAM 1: SR
LCDSAH	LCD start address register high	361H	SAL23	SAL22	SAL21	SAL20	SAL19	SAL18	SAL17	SAL16
			R/W							
			0							
			SR mode start address A23 to A16							
LCDSIZE	LCD size register	362H	COM3	COM2	COM1	COM0	SEG3	SEG2	SEG1	SEG0
			R/W							
			0							
			SR mode: LCD common setting 0000: 64 0101: 128 0001: 68 0110: 144 0010: 80 0111: 160 0011: 100 1000: 200 0100: 120 1001: 240 Other: Reserved				SR mode LCD common setting 0000: 32 0101: 160 0001: 64 0110: 240 0010: 80 0111: 320 0011: 120 1000: 360 0100: 128 Other: Reserved			
LCDCTL	LCD control register	363H	LCDON	-	-	BUS1	BUS0	MMULCD	FP8	START
			R/W							
			0							
			$\overline{DOFF}$ pin 0: Off 1: On	Always write 0	Always write 0	SR mode data bus width select 00: Byte 01: Nibble 10: Bit		RAM type setting 0: Off 1: On	f_{FP} set value bit8	SR mode start address 1: Start
LCDFFP	LCD frame frequency register	364H	FP7	FP6	FP5	FP4	FP3	FP2	FP1	FP0
			R/W							
			0							
			f_{FP} set value bit7 to 0							
LCDCTL2	LCD control register 2	366H	-	-	-			RAMBUS	AC1	AC2
			R/W	R/W	R/W			R/W	R/W	R/W
			0	0	0			0	0	0
			Always write 111					0: Byte 1: Word	00: Type A 01: Type B 10: Type C 11: Reserved	

(15) HVC (Horizontal and vertical converter) (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
HVREGA0	H/V converter register A0	450H	HVRA07	HVRA06	HVRA05	HVRA04	HVRA03	HVRA02	HVRA01	HVRA00
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			H/V conversion data hangar							
HVREGA1	H/V converter register A1	451H	HVRA17	HVRA16	HVRA15	HVRA14	HVRA13	HVRA12	HVRA11	HVRA10
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			H/V conversion data hangar							
HVREGA2	H/V converter register A2	452H	HVRA27	HVRA26	HVRA25	HVRA24	HVRA23	HVRA22	HVRA21	HVRA20
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			H/V conversion data hangar							
HVREGA3	H/V converter register A3	453H	HVRA37	HVRA36	HVRA35	HVRA34	HVRA33	HVRA32	HVRA31	HVRA30
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			H/V conversion data hangar							
HVREGA4	H/V converter register A4	454H	HVRA47	HVRA46	HVRA45	HVRA44	HVRA43	HVRA42	HVRA41	HVRA40
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			H/V conversion data hangar							
HVREGA5	H/V converter register A5	455H	HVRA57	HVRA56	HVRA55	HVRA54	HVRA53	HVRA52	HVRA51	HVRA50
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			H/V conversion data hangar							
HVREGA6	H/V converter register A6	456H	HVRA67	HVRA66	HVRA65	HVRA64	HVRA63	HVRA62	HVRA61	HVRA60
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			H/V conversion data hangar							
HVREGA7	H/V converter register A7	457H	HVRA77	HVRA76	HVRA75	HVRA74	HVRA73	HVRA72	HVRA71	HVRA70
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			H/V conversion data hangar							

HVC (Horizontal and vertical converter) (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
HVREGB0	H/V converter register B0	458H	HVRB07	HVRB06	HVRB05	HVRB04	HVRB03	HVRB02	HVRB01	HVRB00
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			H/V conversion data hangar							
HVREGB1	H/V converter register B1	459H	HVRB17	HVRB16	HVRB15	HVRB14	HVRB13	HVRB12	HVRB11	HVRB10
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			H/V conversion data hangar							
HVREGB2	H/V converter register B2	45AH	HVRB27	HVRB26	HVRB25	HVRB24	HVRB23	HVRB22	HVRB21	HVRB20
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			H/V conversion data hangar							
HVREGB3	H/V converter register B3	45BH	HVRB37	HVRB36	HVRB35	HVRB34	HVRB33	HVRB32	HVRB31	HVRB30
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			H/V conversion data hangar							
HVREGB4	H/V converter register B4	45CH	HVRB47	HVRB46	HVRB45	HVRB44	HVRB43	HVRB42	HVRB41	HVRB40
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			H/V conversion data hangar							
HVREGB5	H/V converter register B5	45DH	HVRB57	HVRB56	HVRB55	HVRB54	HVRB53	HVRB52	HVRB51	HVRB50
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			H/V conversion data hangar							
HVREGB6	H/V converter register B6	45EH	HVRB67	HVRB66	HVRB65	HVRB64	HVRB63	HVRB62	HVRB61	HVRB60
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			H/V conversion data hangar							
HVREGB7	H/V converter register B7	45FH	HVRB77	HVRB76	HVRB75	HVRB74	HVRB73	HVRB72	HVRB71	HVRB70
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			H/V conversion data hangar							

(16) HPLT, VLD

Symbol	Name	Address	7	6	5	4	3	2	1	0
VLDCR0	VLD mode control register 0	440H (Prohibit RMW)	V0EN	LHSEL	INT0EN	VLD0IN	V03	V02	V01	V00
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			Voltage detection start flag 0: Detection off 1: Detection on	Detect level flag 0: Over 1.5 V 1: Below 1.4 V	Interrupt permission flag 0: Interrupt off 1: Interrupt on	Comparison result of voltage detection (Read-clear-write) 0: Voltage normal 1: Voltage decline	The register of voltage detection level LHSEL = 0: 12 levels (00H to 0BH) LHSEL = 1: 6 levels 18 levels (Total) can be set			
VLDCR1	VLD mode control register 1	441H (Prohibit RMW)	V1EN	-	INT1EN	VLD1IN	-	V12	V11	V10
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			Voltage detection start flag 0: Detection off 1: Detection on	Always write 0	Interrupt permission flag 0: Interrupt off 1: Interrupt on	Comparison result of voltage detection (Read-clear-write) 0: Voltage normal 1: Voltage decline	Always write 0	The register setting of voltage detection level 5 levels (00H to 04H) can be set		
VLDCR2	VLD mode control register 2	442H (Prohibit RMW)	V2EN	-	INT2EN	VLD2IN	-	-	V21	V20
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			Voltage detection start flag 0: Detection off 1: Detection on	Always write 0	Interrupt permission flag 0: Interrupt off 1: Interrupt on	Comparison result of voltage detection (Read-clear-write) 0: Voltage normal 1: Voltage decline	Always write 0	Always write 0	The register setting of voltage detection level 2 levels (00H to 02H) can be set	
VLDCTL	VLD control register	449H					XT1SEL	VLD2USE	VLD1USE	VLD0USE
							R/W	R/W	R/W	R/W
							0	0	0	0
							0: Vcc drive 1: Vref drive	0: VLD no use 1: VLD use	0: VLD no use 1: VLD use	0: VLD no use 1: VLD use
HPCTST1	HPLT function register 1	445H		-	TIM21	TIM20	TIM11	TIM10	TIM01	TIM00
				R/W	R/W	R/W	R/W	R/W	R/W	R/W
				0	0	0	0	0	0	0
				Always write 0	VLD2 sampling time		VLD1 sampling time		VLD0 sampling time	
HPCTST2	HPLT function register 2	446H	-		SAM_2	SAM_1	SAM_0	-	-	-
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			Always write 0	Always write 0	VLD2 0: Run continually 1: Run intermit-tently	VLD1 0: Run continually 1: Run intermit-tently	VLD0 0: Run continually 1: Run intermit-tently	Always write 0	Always write 0	Always write 0

6. Points of Note and Restrictions

(1) Notation

a) 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.

b) 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)

c) 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<ACT1:0> 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) EMU0and EMU1

Open pins.

c) Reserved address areas

The TMP91C016 does not have any reserved areas.

d) HALT mode (IDLE1)

When IDLE1 mode is used (in which oscillator operation only occurs), set RTCCR<RTCRUN> to 0 stop the timer for the real-time clock before the HALT instructions is executed.

e) 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.

f) 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., P5) are used to turn the pull-up/pull-down resistors on/off. Consequently read-modify-write instructions are prohibited.

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) 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)).

i) Undefined SFR

The value of an undefined bit in an SFR is undefined when read.

j) POP SR instruction

Please execute the POP SR instruction during DI condition.

k) Releasing the HALT mode by requesting an interruption

Usually, interrupts can release all halts status. However, the interrupts ($\overline{\text{NMI}}$, INT0 to INT3, INTKEY, INTRTC, INTALM0 to INTALM4, INTVLD0 to INTVLD2) which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 5 clocks of f_{FPH}) with IDLE1 or STOP mode (IDLE2 is not applicable to this case). (In this case, an interrupt request is kept on hold internally.)

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

7. Package Dimensions

LQFP100-P-1414-0.50F

Unit: mm