TOSHIBA

TOSHIBA Original CMOS 16-Bit Microcontroller

TLCS-900/L1 Series

TMP91C820A

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), 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

TMP91C820AF/JT5AW4-S

1. Outline and Features

TMP91C820A/JT5AW4 is a high-speed 16-bit microcontroller designed for the co mid- to large-scale equipment.

TMP91C820AF comes in a 144-pin flat package. JT5AW4-S comes in a 144-pad c

Listed below are the features.

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

Instruction mnemonics are upward compatible with TLCS-90

16 Mbytes of linear address space

General-purpose registers and register banks

16-bit multiplication and division instructions; bit

Micro DMA: 4 channels (432 ns/2 bytes at 36 MHz)

(2) Minimum instruction execution time: 111ns (at 36MHz

(3) Built-in RAM: 8 Kbytes

Built-in ROM: 8 Kbytes (However, 9999 (ROM code) has no

(4) External memory expansion

- Expandable up to 136 Mbytes (Shared program/data are
- Can simultaneously support 8- or 16-bit width extern ...Dynamic data bus syzing
- Separate bus system

(5) 8-bit timers: 4 channels

(6) 16-bit timer/event counter: 1 channel

(7) General-purpose serial interface: 3 channels

- UART/synchronous mode
- IrDA

(8) Serial bus interface: 1 channel

I²C bus mode/clock synchronous select mode

(9) LCD controller

- Shift register/built-in RAM LCD driver
- Supported 16, 8 and 4 gray levels and black and white
- Hardware blinking cursor

(10) SDRAM controller

Supported 16-M, 64-M and 128-Mbit SDRAM with 16-bit dat

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

- Based on TC8521A

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

(13) 10-bit AD converter: 8 channels

(14) Watchdog timer

(15) 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

(16) Chip select/wait controller: 4 channels

(17) MMU

- Expandable up to 136 Mbytes (4 local area/8-bank meth

(18) Interrupts: 46 interrupts

- 9 CPU interrupts: Software interrupt instruction an
- 31 internal interrupts: Seven selectable priority l
- 6 external interrupts: Seven selectable priority le

(19) Input/output ports: 77 pins (at 16-bit data bus memory)

(20) Standby function

Three HALT modes: IDLE2 (Programmable), IDLE1, STOP

(21) Hardware standby function (Power save function)

(22) Triple-clock controller

Clock doubler (DFM)

Clock gear function: Select a high-frequency clock fc t

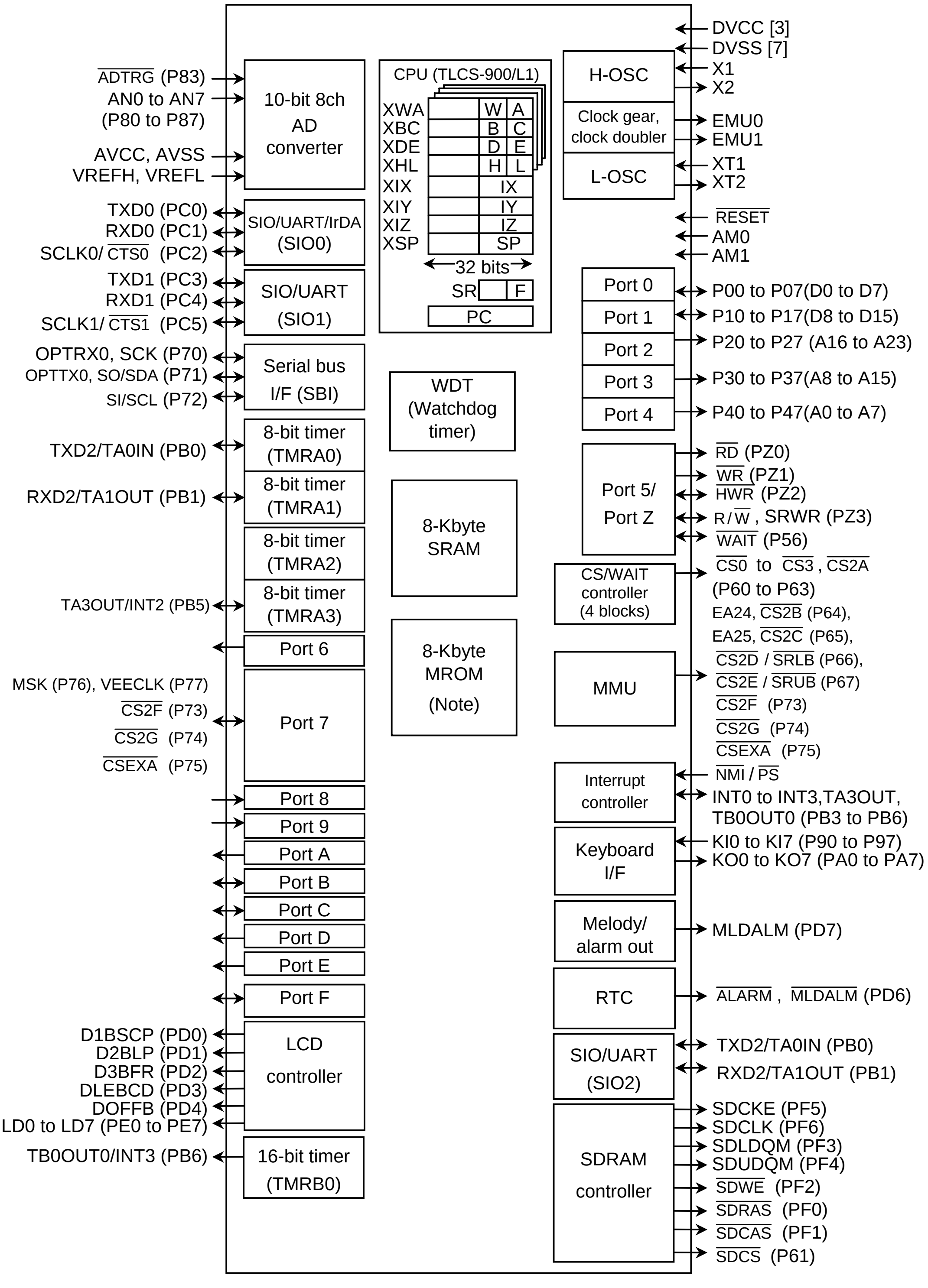
RTC (fs = 32.768 kHz)

(23) Operating voltage

- VCC = 2.7 V to 3.6 V (fc = 27 MHz)
- VCC = 3.0 V to 3.6 V (fc = 36 MHz max)

(24) Package

- 144-pin QFP: P-LQFP144-1616-0.40C
- Chip form supply also available. For details, contact your local representative.



( ): Initial function after reset

Note: When ROM code is 9999, it has no ROM.

Figure 1.1 TMP91C820A Block Diagram

2. Pin Assignment and Pin Functions

The assignment of input/output pins for the TMP91C820A follows:

2.1 Pin Assignment Diagram

Figure 2.1.1 shows the pin assignment of the TMP91C820

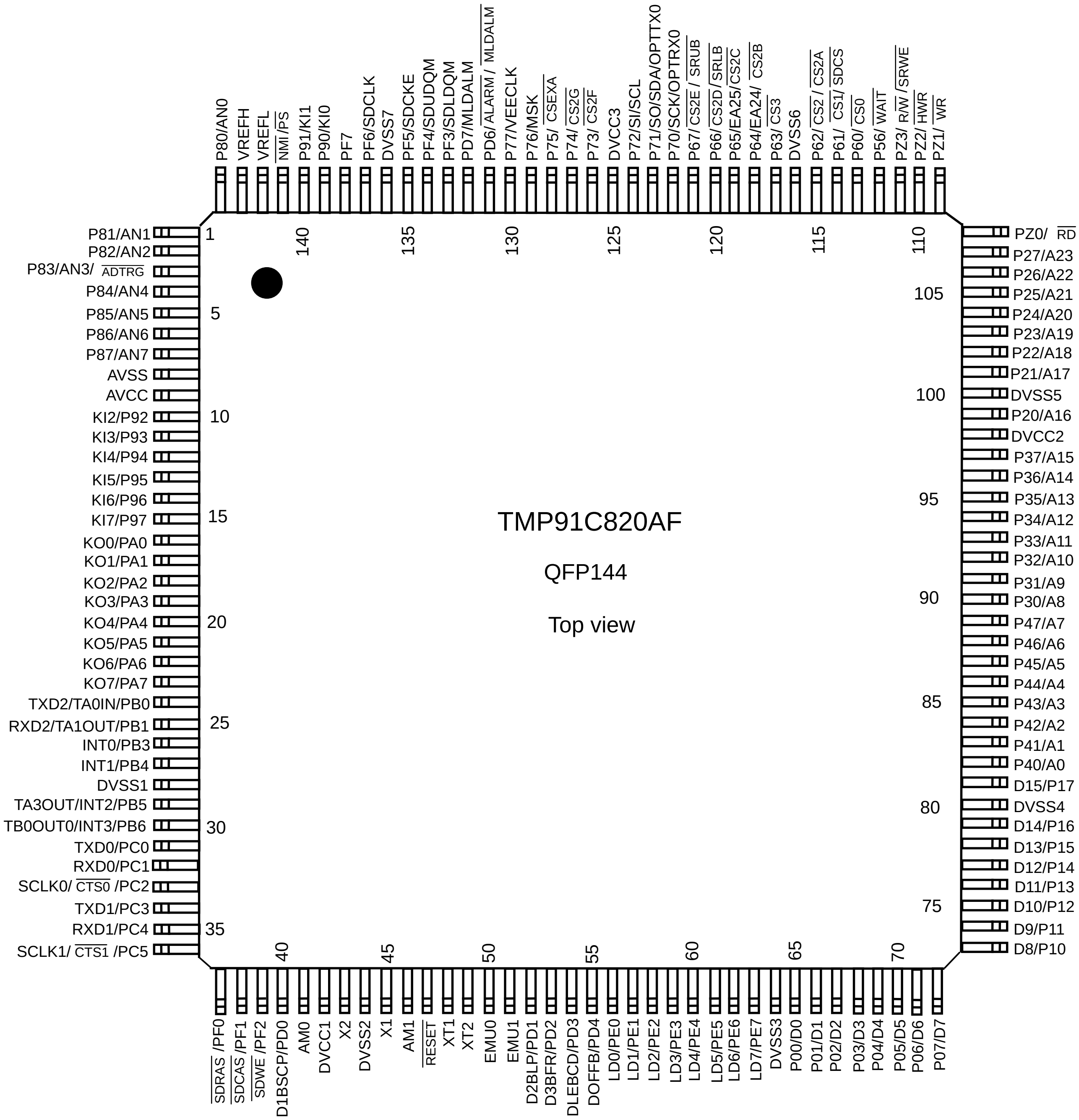


Figure 2.1.1 Pin Assignment Diagram (144-pin QFP)

2.2 PAD layout

(Chip size 5.75 mm × 5.63 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	P81	–2742	2128	49	XT2	–485	–2682	97	P37	2736	758
2	P82	–2742	2004	50	EMU0	–370	–2682	98	DVCC2	2736	872
3	P83	–2742	1888	51	EMU1	–256	–2682	99	P20	2736	986
4	P84	–2742	1774	52	PD1	–142	–2682	100	DVSS5	2736	1202
5	P85	–2742	1660	53	PD2	–28	–2682	101	P21	2736	1318
6	P86	–2742	1546	54	PD3	86	–2682	102	P22	2736	1432
7	P87	–2742	1432	55	PD4	200	–2682	103	P23	2736	1546
8	AVSS	–2742	1318	56	PE0	314	–2682	104	P24	2736	1660
9	AVCC	–2742	1204	57	PE1	428	–2682	105	P25	2736	1774
10	P92	–2742	892	58	PE2	542	–2682	106	P26	2736	1888
11	P93	–2742	778	59	PE3	656	–2682	107	P27	2736	2004
12	P94	–2742	664	60	PE4	770	–2682	108	PZ0	2736	2128
13	P95	–2742	550	61	PE5	884	–2682	109	PZ1	2188	2676
14	P96	–2742	436	62	PE6	998	–2682	110	PZ2	2062	2676
15	P97	–2742	322	63	PE7	1112	–2682	111	PZ3	1948	2676
16	PA0	–2742	208	64	DVSS3	1246	–2682	112	P56	1834	2676
17	PA1	–2742	94	65	P00	1378	–2682	113	P60	1720	2676
18	PA2	–2742	–20	66	P01	1492	–2682	114	P61	1606	2676
19	PA3	–2742	–134	67	P02	1606	–2682	115	P62	1492	2676
20	PA4	–2742	–248	68	P03	1720	–2682	116	DVSS6	1378	2676
21	PA5	–2742	–362	69	P04	1834	–2682	117	P63	1264	2676
22	PA6	–2742	–476	70	P05	1948	–2682	118	P64	1150	2676
23	PA7	–2742	–590	71	P06	2062	–2682	119	P65	1036	2676
24	PB0	–2742	–704	72	P07	2188	–2682	120	P66	922	2676
25	PB1	–2742	–818	73	P10	2736	–2134	121	P67	808	2676
26	PB3	–2742	–932	74	P11	2736	–2010	122	P70	694	2676
27	PB4	–2742	–1046	75	P12	2736	–1894	123	P71	580	2676
28	DVSS1	–2742	–1210	76	P13	2736	–1780	124	P72	382	2676
29	PB5	–2742	–1324	77	P14	2736	–1666	125	DVCC3	268	2676
30	PB6	–2742	–1438	78	P15	2736	–1552	126	P73	68	2676
31	PC0	–2742	–1552	79	P16	2736	–1438	127	P74	–46	2676
32	PC1	–2742	–1666	80	DVSS4	2736	–1318	128	P75	–160	2676
33	PC2	–2742	–1780	81	P17	2736	–1066	129	P76	–274	2676
34	PC3	–2742	–1894	82	P40	2736	–952	130	P77	–388	2676
35	PC4	–2742	–2010	83	P41	2736	–838	131	PD6	–520	2676
36	PC5	–2742	–2134	84	P42	2736	–724	132	PD7	–634	2676
37	PF0	–2194	–2682	85	P43	2736	–610	133	PF3	–748	2676
38	PF1	–2068	–2682	86	P44	2736	–496	134	PF4	–862	2676
39	PF2	–1954	–2682	87	P45	2736	–382	135	PF5	–976	2676
40	PD0	–1840	–2682	88	P46	2736	–268	136	DVSS7	–1090	2676
41	AM0	–1726	–2682	89	P47	2736	–154	137	PF6	–1204	2676
42	DVCC1	–1612	–2682	90	P30	2736	–40	138	PF7	–1318	2676
43	X2	–1410	–2682	91	P31	2736	74	139	P90	–1432	2676
44	DVSS2	–1244	–2682	92	P32	2736	188	140	P91	–1546	2676
45	X1	–1079	–2682	93	P33	2736	302	141	$\overline{\text{NMI}}$	–1660	2676
46	AM1	–963	–2682	94	P34	2736	416	142	VREFL	–1954	2676
47	$\overline{\text{RESET}}$	–849	–2682	95	P35	2736	530	143	VREFH	–2068	2676
48	XT1	–734	–2682	96	P36	2736	644	144	P80	–2194	2676

2.3 Pin Names and Functions

The names of the input/output pins and their functions

Table 2.3.1 Pin Names and Functions (1/4)

Pin Name	Number of Pins	I/O	Functions
P00 to P07 D0 to D7	8	I/O I/O	Port 0: I/O port that allows I/O to be selected at the bit level Data (Lower): Bits 0 to 7 of data bus
P10 to P17 D8 to D15	8	I/O I/O	Port 1: I/O port that allows I/O to be selected at the bit level (When used to the external 8-bit bus) Data (Upper): Bits 8 to15 of data bus
P20 to P27 A16 to A23	8	Output Output	Port 2: I/O port Address: Bits 16 to 23 of address bus
P30 to P37 A8 to A15	8	Output Output	Port 3: I/O port Address: Bits 8 to 15 of address bus
P40 o P47 A0 to A7	8	Output Output	Port 4: I/O port Address: Bits 0 to 7 of address bus
PZ0 $\overline{RD}$	1	Output Output	Port Z0: Output port Read: Strobe signal for reading external memory
PZ1 $\overline{WR}$	1	Output Output	Port Z1: Output port Write: Strobe signal for writing data to pins D0 to D7
PZ2 $\overline{HWR}$	1	I/O Output	Port Z2: I/O port (with pull-up resistor) High write: Strobe signal for writing data to pins D8 to D15
PZ3 R/ $\overline{W}$ $\overline{SRWR}$	1	I/O Output Output	Port Z3: I/O port (with pull-up resistor) Read/write: 1 represents read or dummy cycle; 0 represents write cycle. Write for SRAM: Strobe signal for writing data.
P56 $\overline{WAIT}$	1	I/O Input	Port 56: I/O port (with pull-up resistor) Wait: Pin used to request CPU bus wait
P60 $\overline{CS0}$	1	Output Output	Port 60: Output port Chip select 0: Outputs 0 when address is within specified address area.
P61 $\overline{CS1}$ $\overline{SDCS}$	1	Output Output Output	Port 61: Output port Chip select 1: Outputs 0 when address is within specified address area Chip select for SDRAM: Outputs 0 when address is within SDRAM address area
P62 $\overline{CS2}$ $\overline{CS2A}$	1	Output Output Output	Port 62: Output port Chip select 2: Outputs 0 when address is within specified address area Expand chip select 2A: Outputs 0 when address is within specified address area
P63 $\overline{CS3}$	1	Output Output	Port 63: Output port Chip select 3: Outputs 0 when address is within specified address area
P64 EA24 $\overline{CS2B}$	1	Output Output	Port 64: Output port Chip select 24: Outputs 0 when address is within specified address area Expand chip select 2B: Outputs 0 when address is within specified address area
P65 EA25 $\overline{CS2C}$	1	Output Output Output	Port 65: Output port Chip select 25: Outputs 0 when address is within specified address area Expand chip select 2C: Outputs 0 when address is within specified address area
P66 $\overline{CS2D}$ $\overline{SRLB}$	1	Output Output Output	Port 66: Output port Expand chip select 2D: Outputs 0 when address is within specified address area Lower byte enable for SRAM: Outputs 0 when lower data is enable.
P67 $\overline{CS2E}$ $\overline{SRUB}$	1	Output Output Output	Port 67: Output port Expand chip select 2E: Outputs 0 when address is within specified address area Upper byte enable for SRAM: Outputs 0 when upper data is enable.

Table 2.3.2 Pin Names and Functions (2/4)

Pin Name	Number of Pins	I/O	Functions
P70 SCK OPTRX0	1	I/O I/O Input	Port 70: I/O port Serial bus interface clock I/O data at SIO mode Serial 0 recive data
P71 S0 SDA OPTRX0	1	I/O Output I/O Output	Port 71: I/O port Serial bus interface send data at SIO mode Serial bus interface send/recive data at I^{2}C bus mode Open-drain output mode by programmable Serial 0 send data
P72 SI SCL	1	I/O Input I/O	Port 72: I/O port Serial bus interface recive data at SIO mode Serial bus interface clock I/O data at I^{2}C bus mode Open-drain output mode by programmable
P73 $\overline{\text{CS2F}}$	1	I/O Output	Port 73: I/O port Expand chip select 2F: Outputs 0 when address is within specified address area
P74 $\overline{\text{CS2G}}$	1	I/O Output	Port 74: I/O port Expand chip select 2G: Outputs 0 when address is within specified address area
P75 $\overline{\text{CSEXA}}$	1	I/O Output	Port 75: I/O port Expand chip select EXA: Outputs 0 when address is within specified address area
P76 MSK	1	I/O Input	Port 76: I/O port Mask: Use for disable to output VEECLK for LCD driver
P77 VEECLK	1	I/O Output	Port 77: I/O port Output 32.768 kHz clock to LCD driver. (Can be disabled by MSK pin.)
P80 to P87 AN0 to AN7 $\overline{\text{ADTRG}}$	8	Input Input Input	Port 80 to 87: Pin used to input ports Analog input 0 to 7: Pin used to input to AD conveter AD trigger: Signal used to request AD start (with used to P83)
P90 to P97 KI0 to KI7	8	Input Input	Port 90 to 97: Pin used to input ports Key input 0 to 7: Pin used of key-on wakeup 0 to 7 (Schmitt input, with pull-up resistor)
PA0 to PA7 KO0 to KO7	8	Output Output	Port A0 to A7: Pin used to output ports Key output 0 to 7: Pin used of key-scan strobe 0 to 7
PB0 TA0IN TXD2	1	I/O Input Output	Port B0: I/O port 8-bit timer 0 input: Timer 0 input Serial 2 send data: Open-drain output pin by programmable
PB1 TA1OUT RXD2	1	I/O Output Input	Port B1: I/O port 8-bit timer 1 output: Timer 1 output Serial 2 receive data
PB3 INT0	1	I/O Input	Port B3: I/O port Interrupt request pin0: Interrupt request pin with programmable level/rising/falling edge
PB4 INT1	1	I/O Input	Port B4: I/O port Interrupt request pin1: Interrupt request pin with programmable rising/falling edge

Table 2.3.3 Pin Names and Functions (3/4)

Pin Name	Number of Pins	I/O	Functions
PB5	1	I/O	Port B5: I/O port
INT2		Input	Interrupt request pin2: Interrupt request pin with programmable rising/falling edge
TA3OUT		Output	8-bit timer 3 output: Timer 3 output
PB6	1	I/O	Port B6: I/O port
INT3		Input	Interrupt request pin3: Interrupt request pin with programmable rising/falling edge
TB0OUT0		Outout	Timer B0 output
PC0	1	I/O	Port C0: I/O port
TXD0		Output	Serial 0 send data: Open-drain output pin by programmable
PC1	1	I/O	Port C1: I/O port
RXD0		Input	Serial 0 receive data
PC2	1	I/O	Port C2: I/O port
SCLK0		I/O	Serial 0 clock I/O
$\overline{\text{CTS0}}$		Input	Serial 0 data send enable (Clear to send)
PC3	1	I/O	Port C3: I/O port
TXD1		Output	Serial 1 send data
			Open-drain output pin by programmable
PC4	1	I/O	Port C4: I/O port
RXD1		Input	Serial 1 receive data
PC5	1	I/O	Port C5: I/O port
SCLK1		I/O	Serial 1 clock I/O
$\overline{\text{CTS1}}$		Input	Serial 1 data send enable (Clear to send)
PD0	1	Output	Port D0: Output port
D1BSCP		Output	LCD driver output pin
PD1	1	Output	Port D1: Output port
D2BLP		Output	LCD driver output pin
PD2	1	Output	Port D2: Output port
D3BFR		Output	LCD driver output pin
PD3	1	Output	Port D3: Output port
DLEBCD		Output	LCD driver output pin
PD4	1	Output	Port D4: Output port
DOFFB		Output	LCD driver output pin
PD6	1	Output	Port D6: Output port
$\overline{\text{ALARM}}$		Output	RTC alarm output pin
$\overline{\text{MLDALM}}$		Output	Melody/alarm output pin (Inverted)
PD7	1	Output	Port D7: Output port
MLDALM		Output	Melody/alarm output pin
PE0 to PE7	8	I/O	Port E0 to E7: I/O port
LD0 to LD7		Output	Data bus for LCD driver
PF0	1	I/O	Port F0: Output port
$\overline{\text{SDRAS}}$		Output	Row address storobe for SDRAM: Outputs 0 when address is within SDRAM address area
PF1	1	I/O	Port F1: Output port
$\overline{\text{SDCAS}}$		Output	Column address storobe for SDRAM: Outputs 0 when address is within SDRAM address area

Table 2.3.4 Pin Names and Functions (4/4)

Pin Name	Number of Pins	I/O	Functions
PF2 $\overline{\text{SDWE}}$	1	Output Output	Port F2: Output port Write enable for SDRAM
PF3 SDLDQM	1	Output Output	Port F3: Output port Lower data enable for SDRAM
PF4 SDUDQM	1	Output Output	Port F4: Output port Upper data enable for SDRAM
PF5 SDCKE	1	Output Output	Port F5: Output port Clock enable for SDRAM
PF6 SDCLK	1	Output Output	Port F6: Output port Clock for SDRAM
PF7	1	Output	Port F7: Output port
$\overline{\text{PS}}$ $\overline{\text{NMI}}$ (Note)	1	Input Input	Power save mode setting terminal Non-maskable interrupt request: Interrupt request pin with programmable falling edge level or with both edge levels programmable
AM0 to AM1	2	Input	Operation mode: Fixed to AM1 = 1, AM0 = 1 when using internal ROM (when ROM code is 9999, setting is prohibitted). Fixed to AM1 = 0, AM0 = 1 when using external ROM by 16-bit external bus, or 8- or 16-bit dynamic sizing. Fixed to AM1 = 0, AM0 = 0 when using external ROM by 8-bit external bus.
EMU0	1	Output	Open pin
EMU1	1	Output	Open pin
$\overline{\text{RESET}}$	1	Input	Reset: Initializes TMP91C820A (with pull-up resistor).
VREFH	1	Input	Pin for reference voltage input to AD converter (H)
VREFL	1	Input	Pin for reference voltage input to AD converter (L)
X1/X2	2	I/O	High-frequency oscillator connection pins
XT1/XT2	2	I/O	Low-frequency oscillator connection pins
AVCC	1		Power supply pin for AD converter
AVSS	1		GND pin for AD converter (0 V)
DVCC	3		Power supply pins (All VCC pins should be connecyed with the power supply pin).
DVSS	7		GND pins (All pins should be connected with GND (0 V).)

Note: Please input 1 into $\overline{\text{NMI}}/\overline{\text{PS}}$ pin, because $\overline{\text{NMI}}/\overline{\text{PS}} = 0$ means power save mode after reset.

3. Operation

This following describes block by block the functions ar
Notes and restrictions for eatch book are outlined in 6 “
end of this manual.

3.1 CPU

The TMP91C820A incorporates a high-performance 16-bit CPU (The 900/L1 CP
operation, see the “TLCS-900/L1 CPU”.

The following describe the unique function of the CPU
functions are not covered in the TLCS-900/L1 CPU sectio

3.1.1 Reset

When resetting the TMP91C820A microcontroller, ensure that the power sup
is within the operating voltage range, and that the in
stabilized. Then hold the RESET input to low level for at least 10 sys
MHz).

Thus, when turn on the switch, be sure that the power supply voltage is withi
voltage range, and that the internal high-frequency o
RESET input to low level at least for 10 system clocks.

Clock gear is initialized 1/16 mode by reset. It means that the
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 accord
  at address FFFF00H to FFFF02H:

  PC<7:0> ← Value at FFFF00H address

  PC<15:8> ← Value at FFFF01H address

  PC<23:16> ← Value at FFFF02H address

- Sets the stack pointer (XSP) to 100H.
- Sets bits <IFF2:0> of the status register (SR) to
  register to level 7).
- Sets the <MAX> bit of the status register to 1 (MAX
  (Note: As this product does not support MIN mode,
- Clears bits <RFP2:0> of the status register to 000

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

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

Initializes the internal I/O registers.

Sets the port pins, including the pins that also act as internal I/O, to ge
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 of the TMP91C820A-9

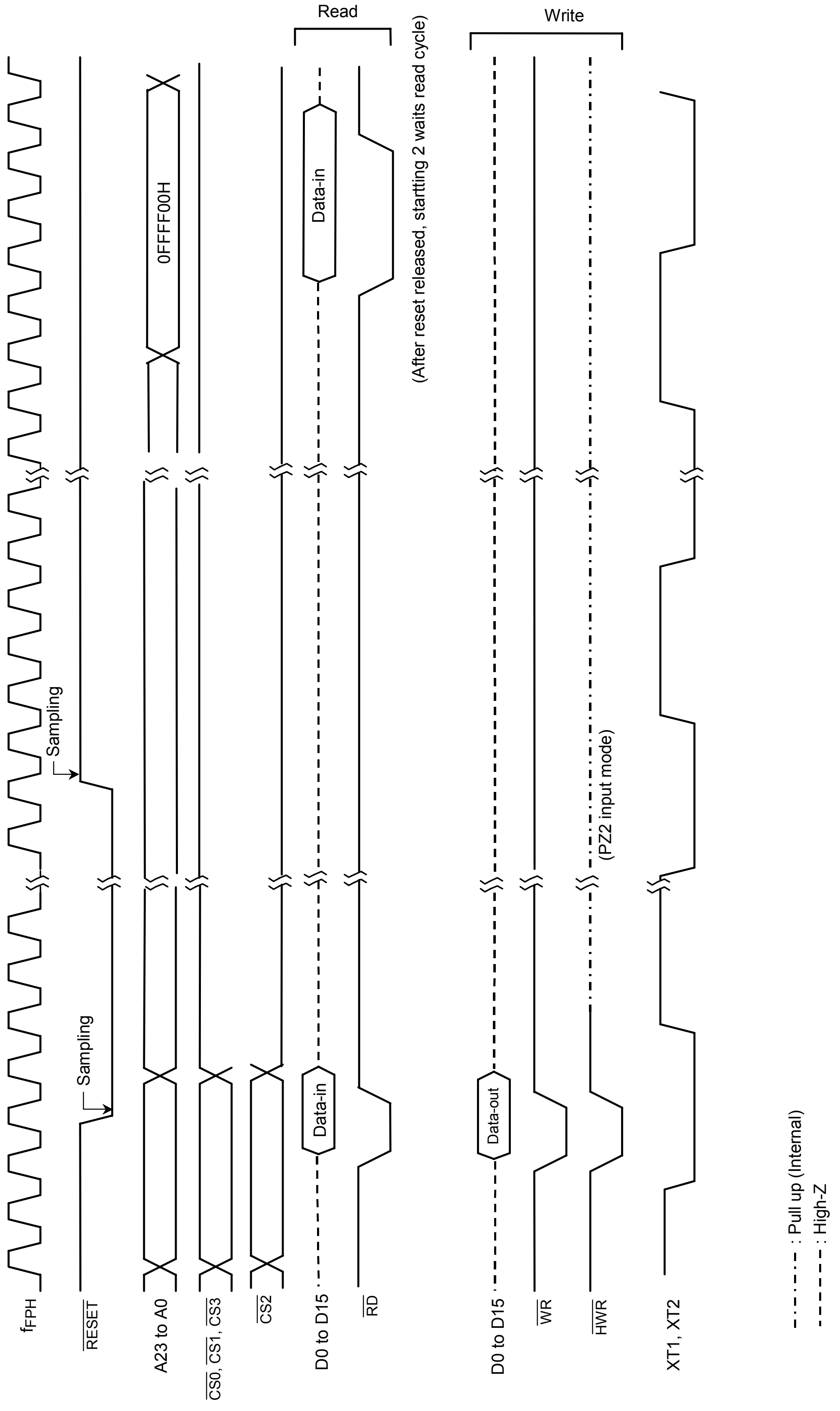


Figure 3.1.1 TMP91C820A-9999 Reset Timing Example (The case of using external ROM)

3.2 Memory Map

Figure 3.2.1 is a memory map of the TMP91C820A.

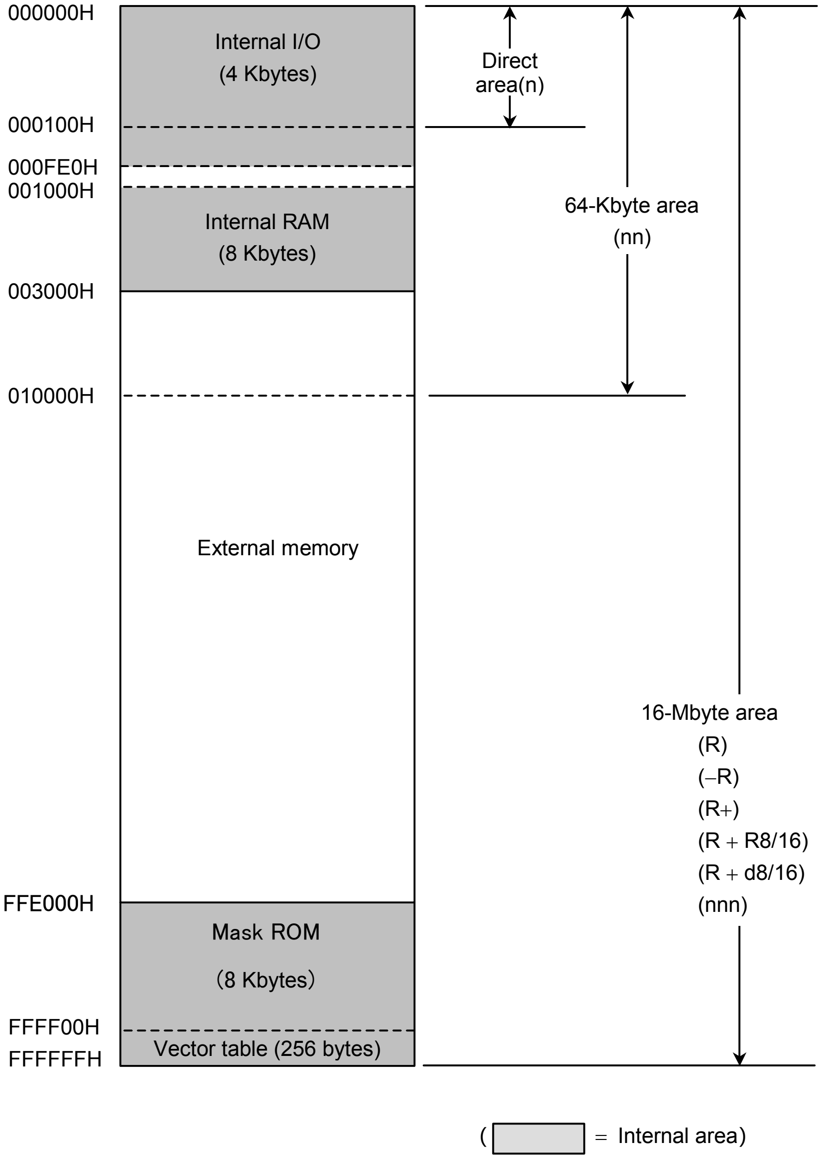


Figure 3.2.1 Memory Map

Note: Address 000FE0H to 000FFFH is assigned for the external memory area of built-in RAM type LCD driver.
And when ROM code is 9999, internal mask ROM area also defines external memory area.

3.3 Triple Clock Function and Standby Function

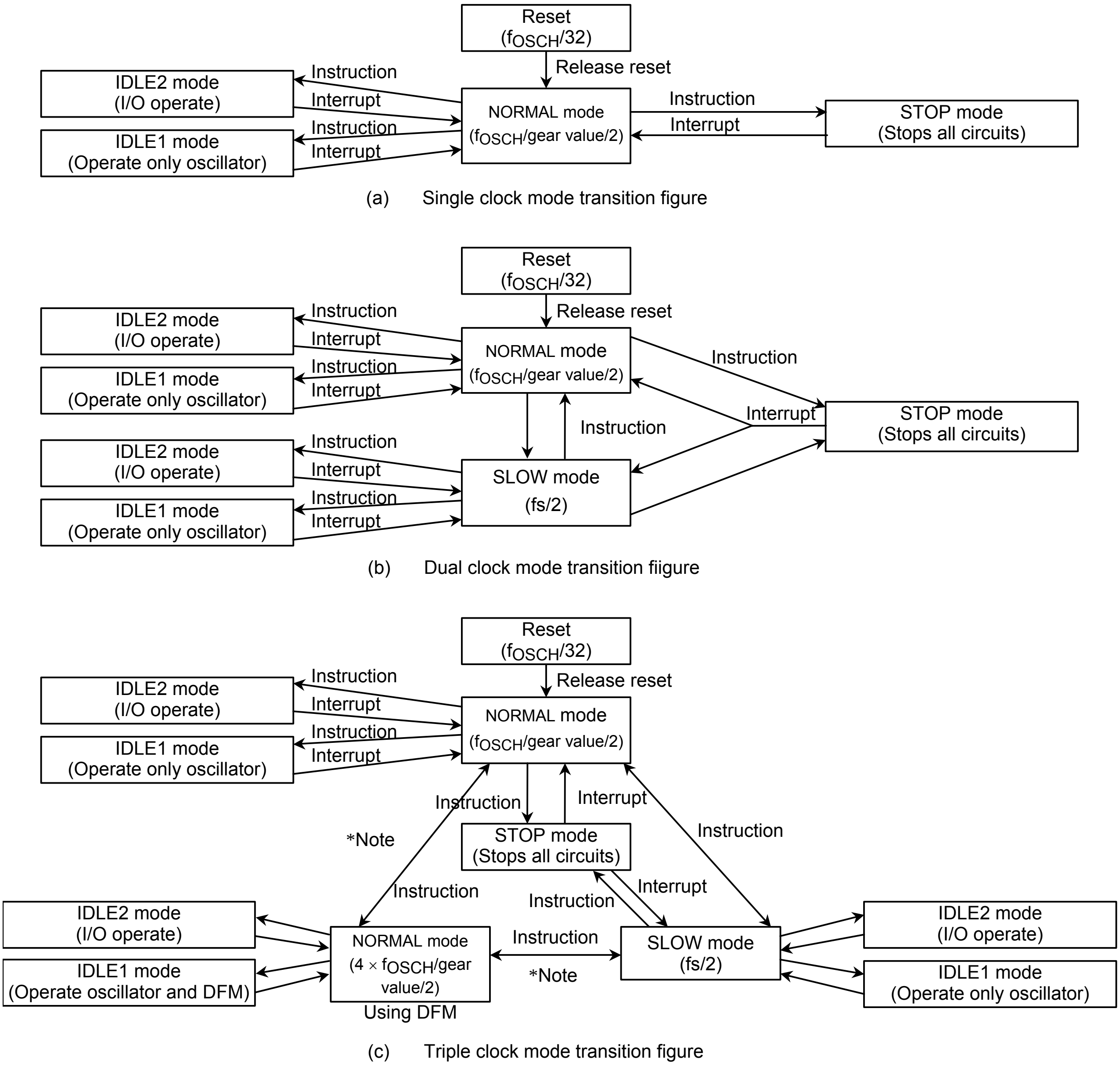
TMP91C820A contains (1) Clock gear, (2) Clock doubler
(4) Noise-reducing circuit. It is used for low-power, l

This chapter is organized as follows:

3.3.1 Block Diagram of System Clock
3.3.2 SFRs
3.3.3 System Clock Controller
3.3.4 Prescaler Clock Controller
3.3.5 Clock Doubler (DFM)
3.3.6 Noise Reduction Circuits
3.3.7 Standby Controller

The clock operating modes are as follows: (a) Single clock mode (X1, X2 pins on
clock mode (X1, X2, XT1 and XT2 pins) and (c) Triple clock mode (The X1, X2, XT1
and DFM).

Figure 3.3.1 shows a transition figure.



(a) Single clock mode transition figure

(b) Dual clock mode transition fiigure

(c) Triple clock mode transition figure

Note 1: It's prohibited to control DFM in SLOW mode when shifting from SLOW mode to NORMAL mode with use of DFM. (DFM start up/stop/change write to DFMCR0<ACT1:0> register.)

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

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 f
from the XT1 and XT2 pins is called fs. The clock frequer
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 defined to as one state.

3.3.1 Block Diagram of System Clock

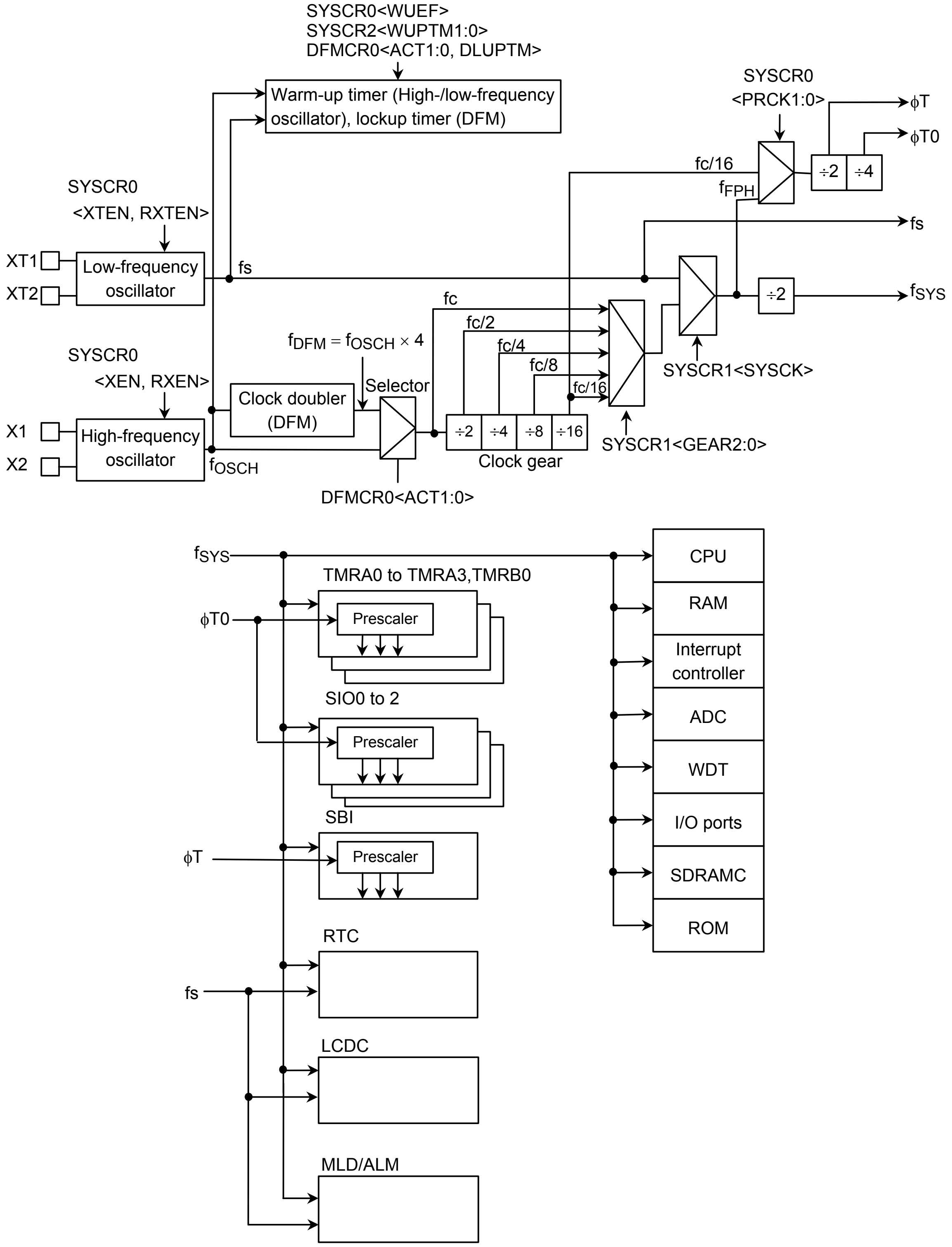


Figure 3.3.2 Block Diagram of System Clock

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 (Note 1)	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} (Note 2) 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	PSENV		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	1:Disable 0:Power save mode enable (Note 3)		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 1: STOP 0: IDLE1 (Note 4)	Pin state control in STOP/IDLE1 mode 0: I/O off 1: Remains the state before halt

Note 1: By reset, low-frequency oscillator is enabled.

Note 2: It's prohibit to use to fc/16 prescaler clock when SBI block use. (I^2C bus and clock synchronous.)

Note 3: When use $\overline{NMI}$ / $\overline{PS}$ pin as $\overline{NMI}$ function, set <PSENV> to 1.

Note 4: 0 means IDLE1, and 1 means STOP. Please be careful because this setting is sometimes different from others.

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} 00 STOP STOP f_{OSCH} 01 RUN RUN f_{OSCH} 10 RUN STOP f_{DFM} 11 RUN STOP f_{OSCH}		Lockup status flag 0: End 1: Not end	Lockup time 0: $2^{12}/f_{OSCH}$ 1: $2^{10}/f_{OSCH}$				
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 9 MHz (at 2.7 V to 3.6 V): Write 0BH							

Figure 3.3.4 SFR for DFM

Limitation point on the use of DFM

1. It's prohibited to execute DFM enable/disable contr
(Write to DFMCR0<ACT1> = "0") You should control DFM in the N

2. If you stop DFM operation during using DFM (DFMCR0<ACT1:0> = "10"), you shou
execute that change the clock f_{DFM} to f_{OSCH} and stop the DFM at the same tim
the above executions should be separated into two pro

```
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 oscil

Please refer to 3.3.5 "Double clock (DFM)" for the details

		7	6	5	4	3	2	1	0
EMCCR0 (00E3H)	Bit symbol	PROTECT	TA3LCDE	AHOLD	TA3MLDE	−	EXTIN	DRVOSCH	DRVOSCL
	Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	0	0	0	0	1	1
	Function	Protect flag 0: OFF 1: ON	LCDC Source clock 0: 32 kHz 1: TA3OUT	Address hold (Note) 0: Disable 1: Enable	Melody/alarm source clock 0: 32 kHz 1: TA3OUT	Always fixed to “0”.	1: External clock	fc oscillator driver ability 1: Normal 0: Weak	fs oscillator driver ability 1: Normal 0: Weak
EMCCR1 (00E4H)	Bit symbol	Switching the protect ON/OFF by write to following 1st-KEY, 2nd-KEY 1st-KEY: EMCCR1 = 5AH, EMCCR2 = A5H in succession write 2nd-KEY: EMCCR1 = A5H, EMCCR2 = 5AH in succession write							
	Read/Write								
	After reset								
	Function								
EMCCR2 (00E5H)	Bit symbol								
	Read/Write								
	After reset								
	Function								
EMCCR3 (00E6H)	Bit symbol		ENFROM	ENDROM	ENPROM		FFLAG	DFLAG	PFLAG
	Read/Write		R/W	R/W	R/W		R/W	R/W	R/W
	After reset		0	0	0		0	0	0
	Function		CS1A area detect control 0: Disable 1: Enable	CS2B 2G area detect control 0: Disable 1: Enable	CS2A area detect control 0: Disable 1: Enable		CS1A write operation flag	CS2B 2G write operation flag	CS2A write operation flag
							When reading 0: Not written 1: Written When writing 0: Clear flag		

Note: When getting access to the logic address 000000H to 000FDFH, 001000H to 002FFFH and FFE000H to FFFFFFH, A0 to A23 holds the previous address of external access.

Figure 3.3.5 SFR for Noise Reducing

3.3.3 System Clock Controller

The system clock controller generates the system clock signal (f_{SYS}) for the CPU core a
internal I/O. It contains two oscillation circuits and a clock gear circuit fo
(fc) operation. The register SYSCR1<SYSCK> changes
SYSCR0<XEN> and SYSCR0<XTEN> control enabling and d
and SYSCR1<GEAR0:2> sets the high-frequency clock gear to either 1, 2, 4, 8 o
fc/4, fc/8 or fc/16). These functions can reduce the p
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 reset.

For example, f_{SYS} is set to 0.5 MHz when the 16-MHz oscillato
and X2 pins.

(1) Switching from NORMAL mode to SLOW mode

When the resonator is connected to the X1 and X2 pin
the warm-up timer can be used to change the operation frequency a
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 s
examples 1 and 2.

Table 3.3.1 shows the warm-up times.

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.

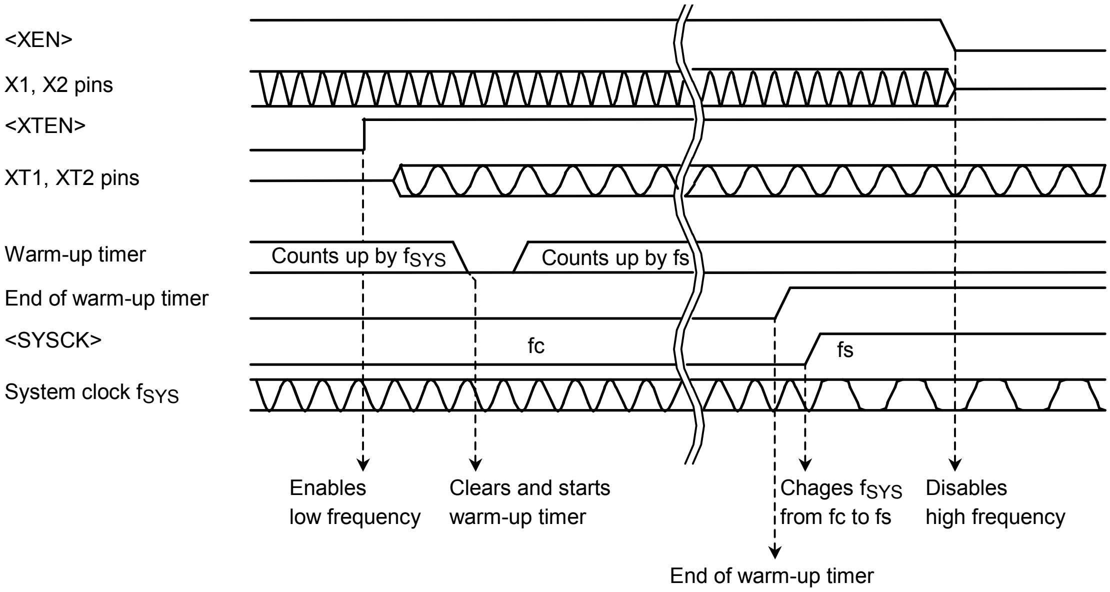
Table 3.3.1 Warm-up Times

Warm-up Time SYSCR2 <WUPTM1:0>	Change to NORMAL Mode	Change to SLOW Mode
01 (2^8/frequency)	16 [μs]	7.8 [ms]
10 (2^{14}/frequency)	1.024 [ms]	500 [ms]
11 (2^{16}/frequency)	4.096 [ms]	2000 [ms]

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

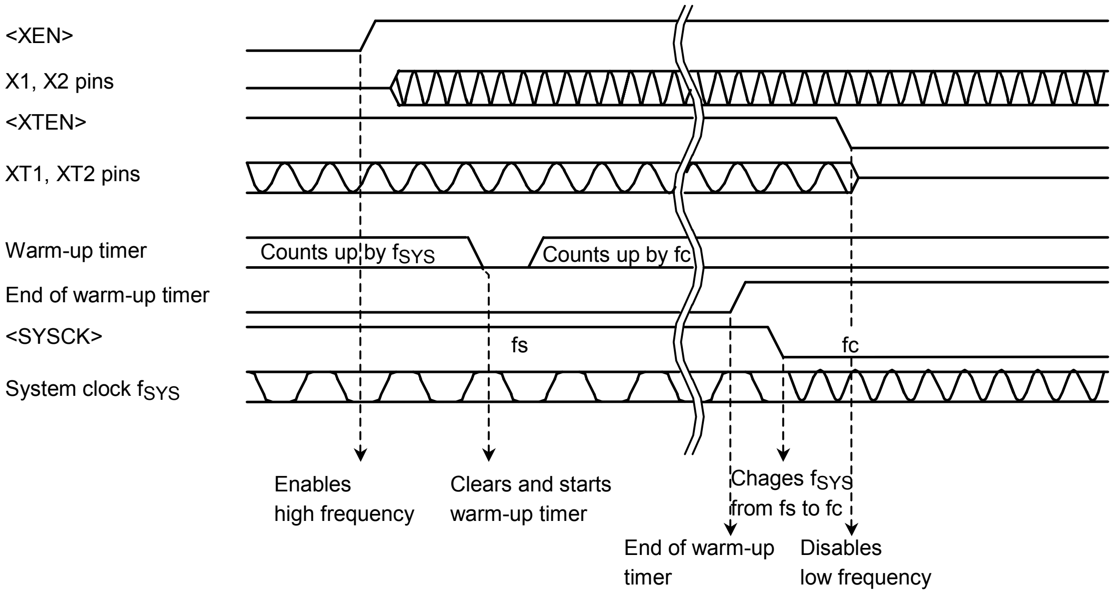
```
 Example 1:   Setting the clock
              Changing from high frequency (fc) to low frequency (fs).
SYSCR0        EQU     00E0H
SYSCR1        EQU     00E1H
SYSCR2        EQU     00E2H
              LD      (SYSCR2), −X11− − − −B   ;    Sets warm-up time to 2^16/fs.
              SET     6, (SYSCR0)              ;    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), −X10− − − −B    ;   Sets warm-up time to 2^14/fc.
            SET     7, (SYSCR0)               ;   Enables high-frequency oscillation.
            SET     2, (SYSCR0)               ;   Clears and starts warm-up timer.
WUP:        BIT     2, (SYSCR0)               ; ⎫
            JR      NZ, WUP                   ; ⎭ Detects stopping of warm-up timer.
            RES     3, (SYSCR1)               ;   Changes fSYS from fs to fc.
            RES     6, (SYSCR0)               ;   Disables low-frequency oscillation.
```

X: Don’t care, −: No change



(2) Clock gear controller

When the high-frequency clock fc is selected by setting SYSCR1<SYS = 0, f_{FPH} is set according to the contents of the clock gear select register SYSCR1 either fc, fc/2, fc/4, fc/8 or fc/16. Using the clock gear to select a lower f_{FPH} lowe 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 t necessary the warm-up time until changing after writing the register valu

There is the possibility that the instruction next to the clock gear changing executed by the clock gear before changing. To execute switching instruction by the clock gear after changing, input the dummy instruc (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.
```

3.3.4 Prescaler Clock Controller

For the internal I/O (TMRAt0o1 StIoOT1M,RSAB2I3), tShIeOr0e is a pre
can divide the clock.

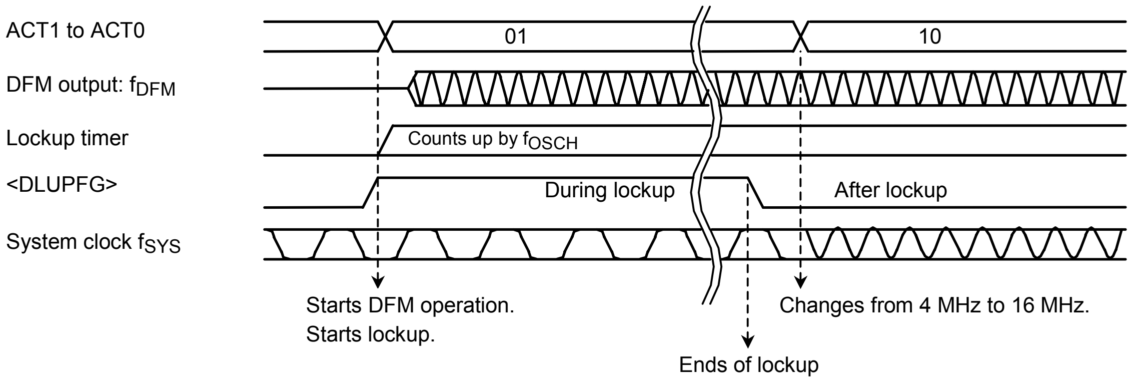
TheφT clock input to the prescaFPHledrivisdeeidt bhye 2 ohretcth eocclko
divided by 2. The setting of the SYSCR0<PRCK0:1> regi
is input. When it's useidt i,n<tPeRrCnKa1l:S0nB>urseitgbriessteer to 00.

3.3.5 Clock Doubler (DFM)

DFM outputsDFMtchleofck signal, which is fOSCHour lttimaerns uasseftahse
low-frequency oscillator, even though the internal c
A reset initializes DFM to stop status, setting to DF
Like an oscillator, thitsocsitracbuiiltizeequTihriessitsicmaelled t
The following example shows how DFM is used.

```
DFMCR0    EQU    00E8H
DFMCR1    EQU    00E9H
          LD     (DFMCR1), 0BH          ;    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 MHz to 9 MHz (Vcc = 2.7 V to 3.6 V): Write 0BH to DFMCR1.

Limitation point on the use of DFM

1. It's prohibited to execute DFM enable/disable contr
   (Write to DFMCR0<ACT1:0> = "10") You should control DFM in the N

2. If you stop DFM operation during using DFM (DFMCR0< = "10"), you shou
   execute the commands that change the clock f_{DFM} to f_{OSCH} and stop the DFM at the
   time. Therefore the above execution should be separa
   below.

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

3. If you stop high-frequency oscillator during using = "10"), you
   should stop DFM before you stop high-frequency oscil

Examples of settings are below.

(1) Start up/change control

(OK) Low-frequency oscillator operation mode (fs
High-frequency oscillator start up → High-frequency oscillator oper (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)           ; }
        JR      NZ, WUP               ; }Check for the flag of lockup end.
        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)           ; }
        JR      NZ, LUP               ; }Check for the flag of lockup end.
        LD      (DFMCR0), 10−0−−−−B   ;   Change the system clock.
```

(OK) Low-frequency oscillator operation mode
operation) → High-frequency oscillator (f_{OSCH}) → DFM start up → DFM use mode (f
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)           ; }
        JR      NZ, LUP               ; }Check for the flag of lockup end.
        LD      (DFMCR0), 10−0−−−−B   ;   Change the system clock fOSCH to fDFM.
```

(Error) Low-frequency oscillator operation mode
→ High-frequency osc → 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)           ; }
        JR      NZ, WUP               ; }Check for the flag of lockup end.
        LD      (DFMCR0), 01−0−−−−B   ;   DFM start-up/lockup start.
LUP:    BIT     5, (DFMCR0)           ; }
        JR      NZ, LUP               ; }Check for the flag of lockup end.
        LD      (DFMCR0), 10-0−−−−B   ;   Change the internal clock fOSCH to fDFM.
        LD      (SYSCR1), −−−−0−−−B   ;   Change the system clock fS to fDFM.
```

(2) Change/stop control

(OK) DFM use mode (f_{DFM}) → High-frequency oscillator (f_{OSCH}) → opera
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      (SYSCR0), 0–––––––B   ;   High-frequency oscillator stop.
```

(Error) DFM use mode (f_{DFM}) → Low-frequency oscillator → oper
DFM stop → High-frequency oscillator stop

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

(OK) DFM use mode (f_{DFM}) → Set the STOP mode → High-frequency osc
operation mode (f_{OSCH}) → DFM stop → Halt (High-frequency oscilla

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

(Error) DFM use mode (f_{DFM}) → Set the STOP mode → Halt (High-frequ
oscillator stop)

```
LD      (SYSCR2), ––––01––B   ;   Set the 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 imp

(1) Reduced drivability for high-frequency oscillator

(2) Reduced drivability for low-frequency oscillator

(3) Single drive for high-frequency oscillator

(4) Runaway provision with SFR protection register

(5) Runaway provision with ROM protection register

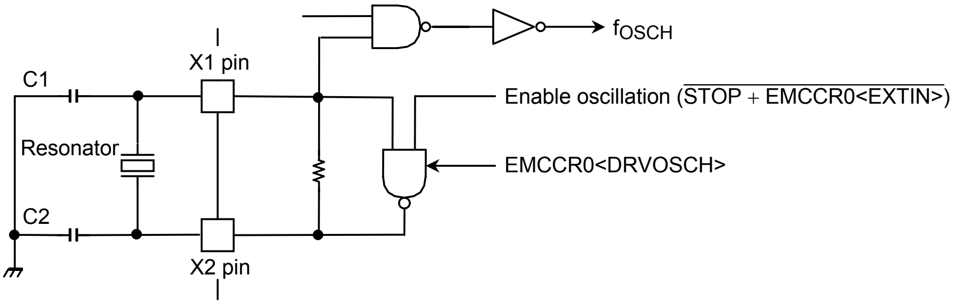
The above functions are performed by making the appr
to EMCCR3 registers.

(1) Reduced drivability for high-frequency oscillator

(Purpose)

Reduces noise and power for oscillator when a reso

(Block diagram)



(Setting method)

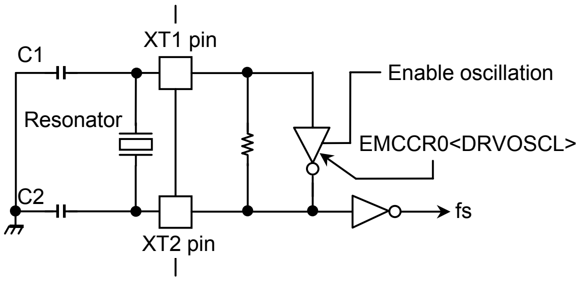
The drivability of the oscillator is reduced by w
register. By reset, <DRVOSCH> is initialized to 1 a
by normal drivability when the power supply is on.

(2) Reduced drivability for low-frequency oscillator

(Purpose)

Reduces noise and power for oscillator when a reso

(Block diagram)



(Setting method)

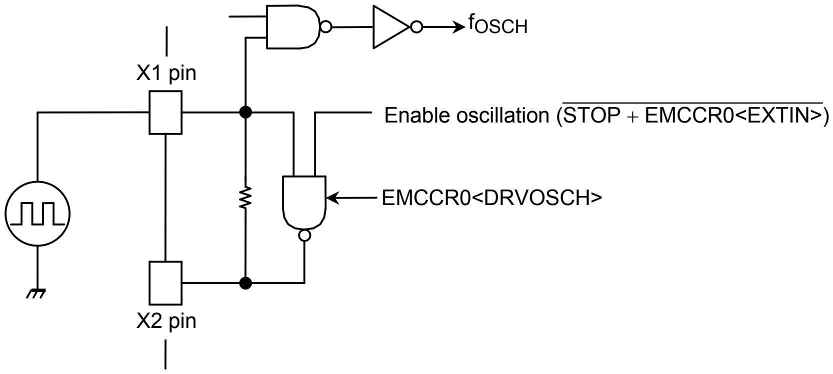
The drivability of the oscillator is reduced by wr register. By reset, <DRVOSCL> is initialized to 1.

(3) Single drive for high-frequency oscillator

(Purpose)

Not need twin-drive and protect mistake operation the external oscillator is used.

(Block diagram)


f_{OSCH}
X1 pin
Enable oscillation ($\overline{\text{STOP + EMCCR0<EXTIN>}}$)
EMCCR0<DRVOSCH>
X2 pin

(Setting method)

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

By reset, <EXTIN> is initialized to 0.

(4) Runaway provision with SFR protection register

(Purpose)

Provision in runaway of program by noise mixing.

Write operation to specified SFR is prohibited so prevents that it is it impossibility fetch which is by sto memory control register clock or (CS/WAIT, 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
DFMCR0/1

(Operation explanation)

Execute and release of protection (Write operatio by setting up a double key to EMCCR1 and EMCCR2 regis

(Double key)

1st-KEY: Succession writes in 5AH at EMCCR1 a

2nd-KEY: Succession writes in A5H at EMCCR1 a

A state of protection can be confirmed by reading E

By reset, protection becomes OFF.

And INTP0 interruption occurs when write operatio with protection on state.

(5) Runaway provision with ROM protection register

(Purpose)

Provision in runaway of program by noise mixing.

(Operation explanation)

When writes operation was executed for external th
program, INTP1 is occurred and detects runaway func

Three kinds of ROM is fixed (as for program ROM), da
program ROM are as follows on the logical addressme

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 dete
with EMCCR3<ENFROM, ENDROM, ENPROM>. And INTP1 in
with which ROM area in the case that occurred
EMCCR3<FFLAG, DFLAG, and PFLAG>. The flag is cleared when write

3.3.7 Standby Controller

(1) HALT modes

When the HALT instruction is executed, the opera
IDLE1 or STOP mode, depending on the contents of
register.

The subsequent actions performed in each mode are

a. IDLE2: Only the CPU halts.
The internal I/O is available to select ope
setting the following register.
Table 3.3.2 shows the registers of set.ting o

Table 3.3.2 SFR Setting Operation during IDLE2 Mode

Internal I/O	SFR
TMRA01	TA01RUN<I2TA01>
TMRA23	TA23RUN<I2TA23>
TMRB0	TB0RUN<I2TB0>
SIO0	SC0MOD1<I2S0>
SIO1	SC1MOD1<I2S1>
AD converter	ADMOD1<I2AD>
WDT	WDMOD<I2WDT>
SBI	SBI0BR0<I2SBI0>

b. IDLE1: Only the oscillator and the RTC (Real ti
operate.

c. STOP: All internal circuits stop operating.

The operation of each of the different HALT modes i

Table 3.3.3 I/O Operation during HALT Modes

HALT Mode		IDLE2	IDLE1	STOP
SYSCR2<HALTM1:0>		11	10	01
Block	CPU	Stop		
	I/O ports	Keep the state when the HALT instruction was executed.	See Table 3.3.6, Table 3.3.7	
	TMRA,TMRB0	Available to select operation block	Stop	
	SIO, SBI			
	AD converter			
	WDT			
	LCDC, SDRAMC Interrupt controller	Operate		
	RTC, MLD		Operational available	

(2) How to release the HALT mode

These halt states can be released by resetting or
release sources are determined by the combination
mask register <IFF2:0> and the HALT mode. The details for releasing t
are shown in Table 3.3.4.

Released by requesting an interrupt

The operating released from the HALT mode depen
status. When the interrupt request level set bef
exceeds the value of interrupt mask register, th
processed after releasing the HALT mode, and
instruction that follows the halt instruction. When the interrupt requ
before executing the halt instruction is less th
register, releasing the HALT mode is not execute
interrupt processing is processed after releasing the HALT mode regardless of t
value of the mask register). However only for IN
INTRTC, INTALM0 to INTALM4, even if interrupt request lev
executing the halt instruction is less than the v
releasing the HALT mode is executed. In this c
CPU starts executing the instruction next to th
interrupt request flag is held at 1.

Note: Usually, interrupts can release all halt status. However, the interrupts ($\overline{\text{NMI}}$, INT0 to INT3, INTKEY, INTRTC, INTALM0 to INTALM4) 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 the 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 resett

When the stop mode is released by reset, it is ne
(See Table 3.3.5) to set the operation of the osci

When releasing the HALT mode by resetting, the i
state before the HALT instruction is executed.
contents are initialized. (Releasing due to int
HALT instruction is executed.)

Table 3.3.4 Source of Halt State Clearance and Halt Clearance Operation

Status of Received Interrupt			Interrupt Enabled (Interrupt level) ≥ (Interrupt mask)			Interrupt Disabled (Interrupt level) < (Interrupt mask)		
HALT mode			IDLE2	IDLE1	STOP	IDLE2	IDLE1	STOP
Source of halt state clearance	Interrupt	NMI	♦	♦	♦*1	–	–	–
		INTWDT	♦	×	×	–	–	–
		INT0 to INT3 (Note 1)	♦	♦	♦*1	○	○	○*1
		INTALM0 to INTALM4	♦	♦	×	○	○	×
		INTTA0 to INTTA3, INTTB00 to INTTB01	♦	×	×	×	×	×
		INTRX0 to INTRX2, TX0 to TX2	♦	×	×	×	×	×
		INTSS0 to INTSS2	♦	×	×	×	×	×
		INTAD	♦	×	×	×	×	×
		INTKEY	♦	♦	♦*1	○	○	○*1
		INTRTC	♦	♦	×	○	○	×
		INTSBI	♦	×	×	×	×	×
		INTLCD	♦	×	×	×	×	×
		RESET	Initialize LSI.					

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

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

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

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

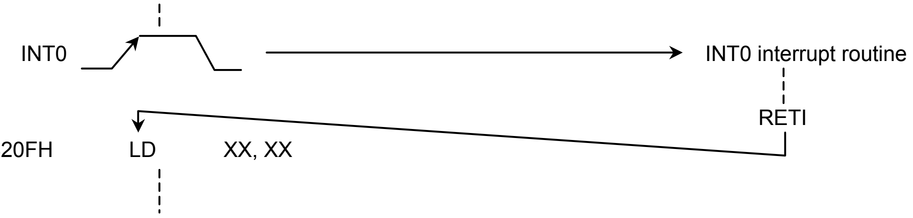
*1: Releasing the HALT mode is executed after passing the warm-up time.

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 releasing IDLE1 mode)

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

```
Address
8203H     LD      (IIMC), 00H           ; Selects INT0 interrupt rising edge.
8206H     LD      (INTE0AD), 06H        ; Sets INT0 interrupt level to 6.
8209H     EI      5                     ; Sets interrupt level to 5 for CPU.
820BH     LD      (SYSCR2), 88H         ; 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 opera
IDLE2 setting register, can take place. Instruct

Figure 3.3.6 illustrates an example of the tim
mode halt state by an interrupt.

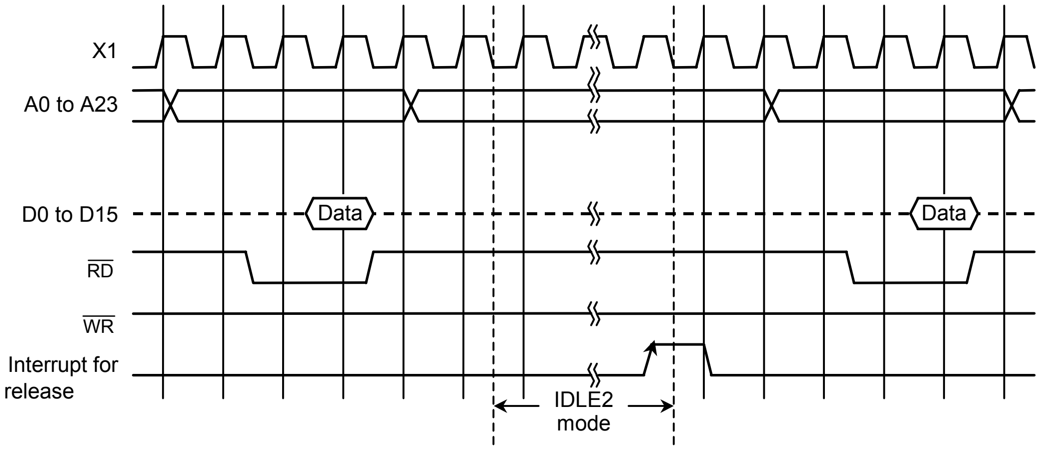


Figure 3.3.6 Timing Chart for IDLE2 Mode Halt State Cleared by Interrupt

b. IDLE1 mode

In IDLE1 mode, only the internal oscillator an
operate. The system clock in the MCU stops. The pi
depended on setting the register SYSCR2<SELDRV
3.3.7 summarizes the state of these pins in the ID

In the halt state, the interrupt request is sam
system clock; however, clearance of the halt st
synchronous with it.

Figure 3.3.7 illustrates the timing for clearan
an interrupt.

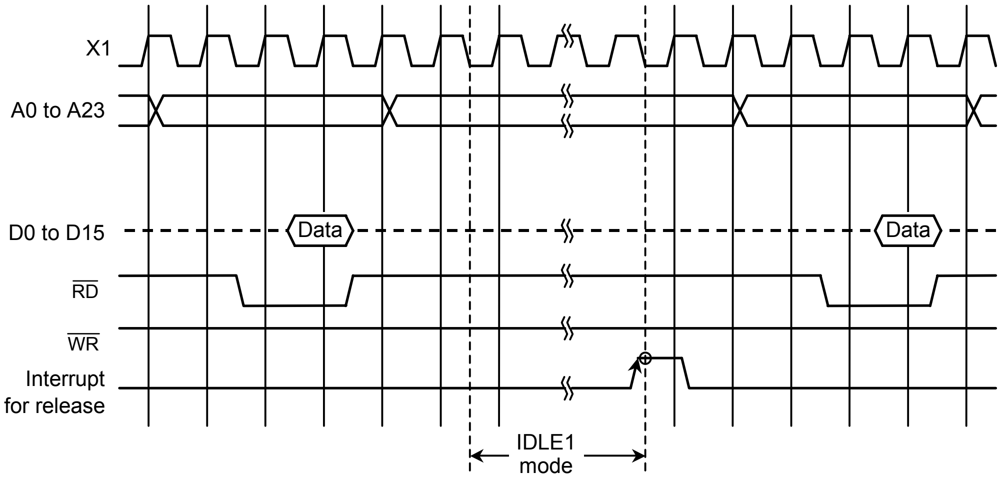


Figure 3.3.7 Timing Chart for IDLE1 Mode Halt State Cleared by Interrupt

c. STOP mode

When STOP mode is selected, all internal circui oscillator pin status in STOP mode depends o SYSCR2<SELDRV, DRVE> register. Table 3.3.6, Table 3.3.7 summariz of these pins in STOP mode.

After STOP mode has been cleared system clock warm-up time has elapsed, in order to allow oscillation to stabili mode has been cleared, either NORMAL mode or SLO using the SYSCR0<RSYSCK> register. Therefore, <RXTEN> must be set swearmth-eupsatmimplees in Table 3.3.5

Figure 3.3.8 illustrates the timing for clearar an interrupt.

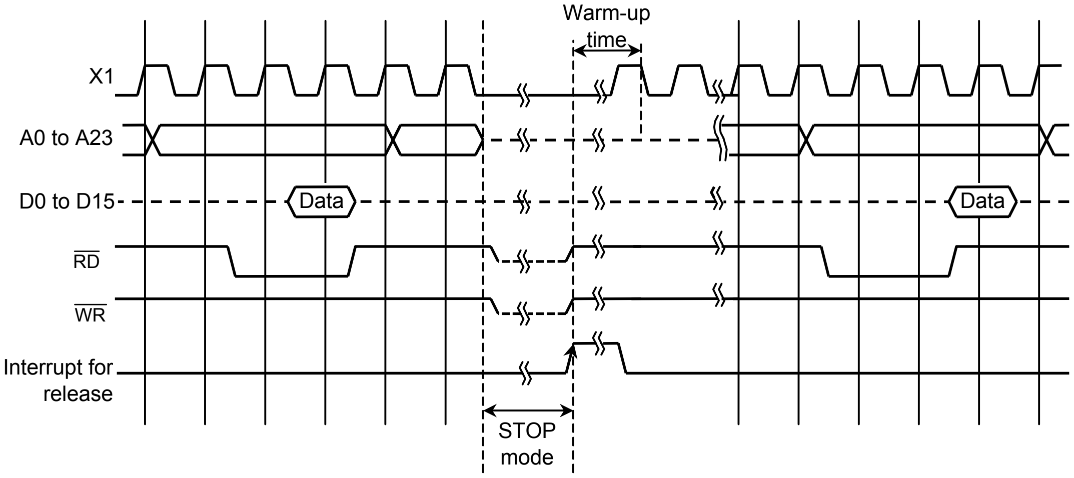


Figure 3.3.8 Timing Chart for STOP Mode Halt State Cleared by Interrupt

Table 3.3.5 Sample Warm-up Times after Clearance of STOP Mode

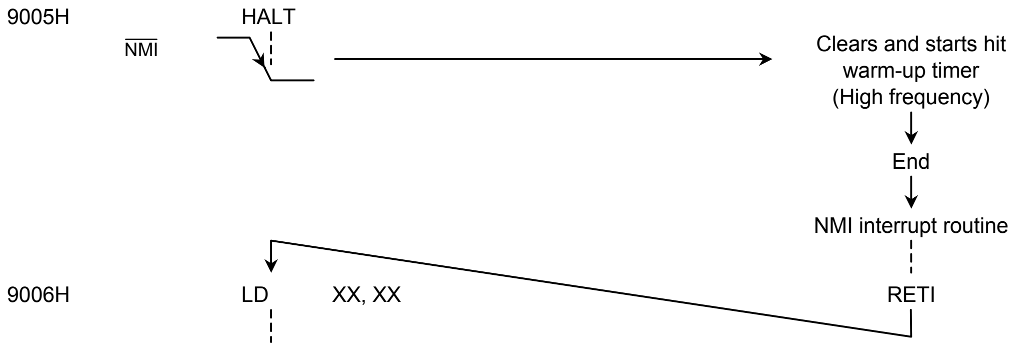
at f_{OSCH} = 16 MHz, fs = 32.768 kHz

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

(Setting example)

The STOP mode is entered when the low frequency operat 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), −X1001X1B      ; Sets warm-up time to 2^14/fOSCH.
9002H                LD    (SYSCR0), 011000 − −B    ; Operates high-frequency after released.
```



−: No change

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

Table 3.3.6 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
P00-07	D0-7	OFF	ON upon external read	ON upon port read	ON upon external read of LCDC	OFF	OFF	OFF	OFF	OFF
P10-17	D8-15	8bit start: OFF 16bit start: ON Built-in ROM start: ON		ON		OFF	OFF	OFF	OFF	OFF
P20-27	—	8bit start: OFF 16bit start: OFF Built-in ROM start: ON	—	ON	—	OFF	—	OFF	—	OFF
P30-37	—									
P40-47	—									
PZ2 (*1)	—	ON	—	ON	—	OFF	—	OFF	—	OFF
PZ3 (*1)	—	ON	—	ON	—	OFF	—	OFF	—	OFF
P56 (*1)	WAIT	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
P70	SCK, OPTRX0	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
P71	SDA	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
P72	SI, SCL	ON	ON	ON	ON	ON	ON	OFF	ON	ON
P73	—	ON	—	ON	—	ON	—	OFF	—	ON
P74	—	ON	—	ON	—	ON	—	OFF	—	ON
P75	—	ON	—	ON	—	ON	—	OFF	—	ON
P76	MSK	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
P77	—	ON	—	ON	—	ON	—	OFF	—	ON
P80 (*2)	—	OFF	—	ON upon port read	—	OFF	—	OFF	—	OFF
P81 (*2)	—	OFF	—		—	OFF	—	OFF	—	OFF
P82 (*2)	—	OFF	—		—	OFF	—	OFF	—	OFF
P83 (*2)	ADTRG	OFF	ON		ON	OFF	ON	OFF	ON	OFF
P84 (*2)	—	OFF	—		—	OFF	—	OFF	—	OFF
P85 (*2)	—	OFF	—		—	OFF	—	OFF	—	OFF
P86 (*2)	—	OFF	—		—	OFF	—	OFF	—	OFF
P87 (*2)	—	OFF	—		—	OFF	—	OFF	—	OFF
P90 (*1)	KI0	ON	ON	ON	ON	ON	ON	ON	ON	ON
P91 (*1)	KI1	ON	ON	ON	ON	ON	ON	ON	ON	ON
P92 (*1)	KI2	ON	ON	ON	ON	ON	ON	ON	ON	ON
P93 (*1)	KI3	ON	ON	ON	ON	ON	ON	ON	ON	ON
P94 (*1)	KI4	ON	ON	ON	ON	ON	ON	ON	ON	ON
P95 (*1)	KI5	ON	ON	ON	ON	ON	ON	ON	ON	ON
P96 (*1)	KI6	ON	ON	ON	ON	ON	ON	ON	ON	ON
P97 (*1)	KI7	ON	ON	ON	ON	ON	ON	ON	ON	ON
PB0	TA0IN	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
PB1	RXD2	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
PB3	INT0	ON	ON	ON	ON	ON	ON	ON	ON	ON
PB4	INT1	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF
PB5	INT2	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF
PB6	INT3	ON	ON	ON	ON	OFF	ON	OFF	ON	OFF
PC0	—	ON	—	ON	—	ON	—	OFF	—	ON
PC1	RXD0	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
PC2	SCLK0, CTS0	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
PC3	—	ON	—	ON	—	ON	—	OFF	—	ON
PC4	RXD1	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
PC5	SCLK1, CTS1	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
PE0-7	—	ON	—	ON	—	OFF	—	OFF	—	OFF
NMI/PS	—	ON	ON	—	ON	—	ON	—	ON	—
RESET (*1)	—	ON	ON	—	ON	—	ON	—	ON	—
AM0, 1	—	ON	ON	—	ON	—	ON	—	ON	—
X1, XT1	—	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:AIN 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 A	Condition A
0	1	Condition B	
1	0		Condition B
1	1		

Table 3.3.7 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
P00-07	D0-7	OFF	ON upon external read	ON	OFF	ON	OFF	OFF	OFF	ON
P10-17	D8-15	OFF		ON	OFF	ON	OFF	OFF	OFF	ON
P20-27	A16-23	8bit start: ON 16bit start: ON Built-in ROM start: OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
P30-37	A8-15									
P40-47	A0-7									
PZ0	RD	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
PZ1	WR									
PZ2	HWR	OFF (*1)								
PZ3	R/W, SRWE									
P56	－	OFF (*1)	－	ON	－	ON	－	OFF	－	ON
P60	CS0	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
P61	CS1, SDCS	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
P62	CS2, CS2A	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
P63	CS3	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
P64	EA24, CS2B	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
P65	EA25, CS2C	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
P66	CS2D, SRLB	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
P67	CS2E, SRUB	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
P70	SCK	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
P71	SO, SDA, OPTTX0	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
P72	SCL	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
P73	CS2F	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
P74	CS2G	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
P75	CSEXA	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
P76	－	OFF	－	ON	－	ON	－	OFF	－	ON
P77	VEECLK	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
PA0	KO0	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
PA1	KO1	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
PA2	KO2	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
PA3	KO3	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
PA4	KO4	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
PA5	KO5	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
PA6	KO6	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
PA7	KO7	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
PB0	TXD2	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
PB1	TA1OUT	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
PB3	－	OFF	－	ON	－	ON	－	OFF	－	ON
PB4	－	OFF	－	ON	－	ON	－	OFF	－	ON
PB5	TA3OUT	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
PB6	TB0OUT0	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
PC0	TXD0	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
PC1	－	OFF	－	ON	－	ON	－	OFF	－	ON
PC2	SCLK0	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
PC3	TXD1	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
PC4	－	OFF	－	ON	－	ON	－	OFF	－	ON
PC5	SCLK1	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
PD0	D1BSCP	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
PD1	D2BLP	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
PD2	D3BFR	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
PD3	DLEBCD	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
PD4	DOFFB	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
PD6	ALARM, MLDALM	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
PD7	MLDALM	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
PE0-7	LD0-7	OFF	ON	ON	ON	ON	OFF	OFF	ON	ON
PF0	SDRAS	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
PF1	SDCAS	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
PF2	SDWE	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
PF3	SDLDQM	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
PF4	SDUDQM	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
PF5	SDCKE	ON	ON	ON	ON	ON	ON in self refresh cycle	OFF	ON	ON
PF6	SDCLK	ON	ON	ON	ON	ON	OFF	OFF	ON	ON
X2, XT2	－	ON	ON	－	ON	－	IDLE1: ON, STOP: output "H" level			

ON: The buffer is always turned on.
OFF: The buffer is always turned off.
－: No applicable

*1:Port having a pull-up/pull-down resistor.
*2:AIN 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 A	Condition A
0	1	Condition B	
1	0		Condition B
1	1		

3.4 Interrupts

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

The TMP91C820A has a total of 43 interrupts divided into the following 5 t

- Interrupts generated by CPU: 9 sources
  (Software interrupts, illegal instruction interru
- Internal interrupts: 28 sources
- Interrupts on external pins ($\overline{\text{NMI}}$ and INT0 to INT3, INTKEY): 6 source

A (Fixed) individual interrupt vector number is assign
One of seven (Variable) priority levels can be assigne
The priority level of non-maskable interrupts is fixed
When an interrupt is generated, the interrupt controller sends the priority of
the CPU. If multiple interrupts are generated simultane
interrupt with the highest priority to the CPU. (The highest level is
non-maskable interrupts.)

The CPU compares the priority level of the interrupt w
mask register <IFF2:0>. If the priority level of the in
interrupt mask register, the CPU accepts the interrupt.

However, software interrupts and illegal instruction
processed without comparison with the <IFF2:0> value.

The interrupt mask register <IFF2:0> value can be up
instruction (Executing EI num sets <IFF2:0> to num). For examp
EI3 enables the acceptance of maskable interrupts whos
controller is 3 or higher, and enables the acceptance of
or EI0 is specified, maskable interrupts with a priorit
interrupts are accepted (Operationally identical to "E

Operationally, the DI instruction (<IFF2:0> is 7) is i
the priority level of maskable interrupts is 0 to 6, the D
interrupt. The EI instruction is vaild immediately aft
EI instruction is vaild after execution of the instruct

In addition to the general-purpose interrupt processi
interrupts have a micro DMA processing mode as well.

Because the CPU transfers (Byte transfer, or 4-byte tr
mode, this mode can be used for speeding up processing, such as tran
I/O. TMP91C820A also has a micro DMA soft start function for requesti
processing by software not by interrupt.

Figure 3.4.1 shows the overall interrupt processing f

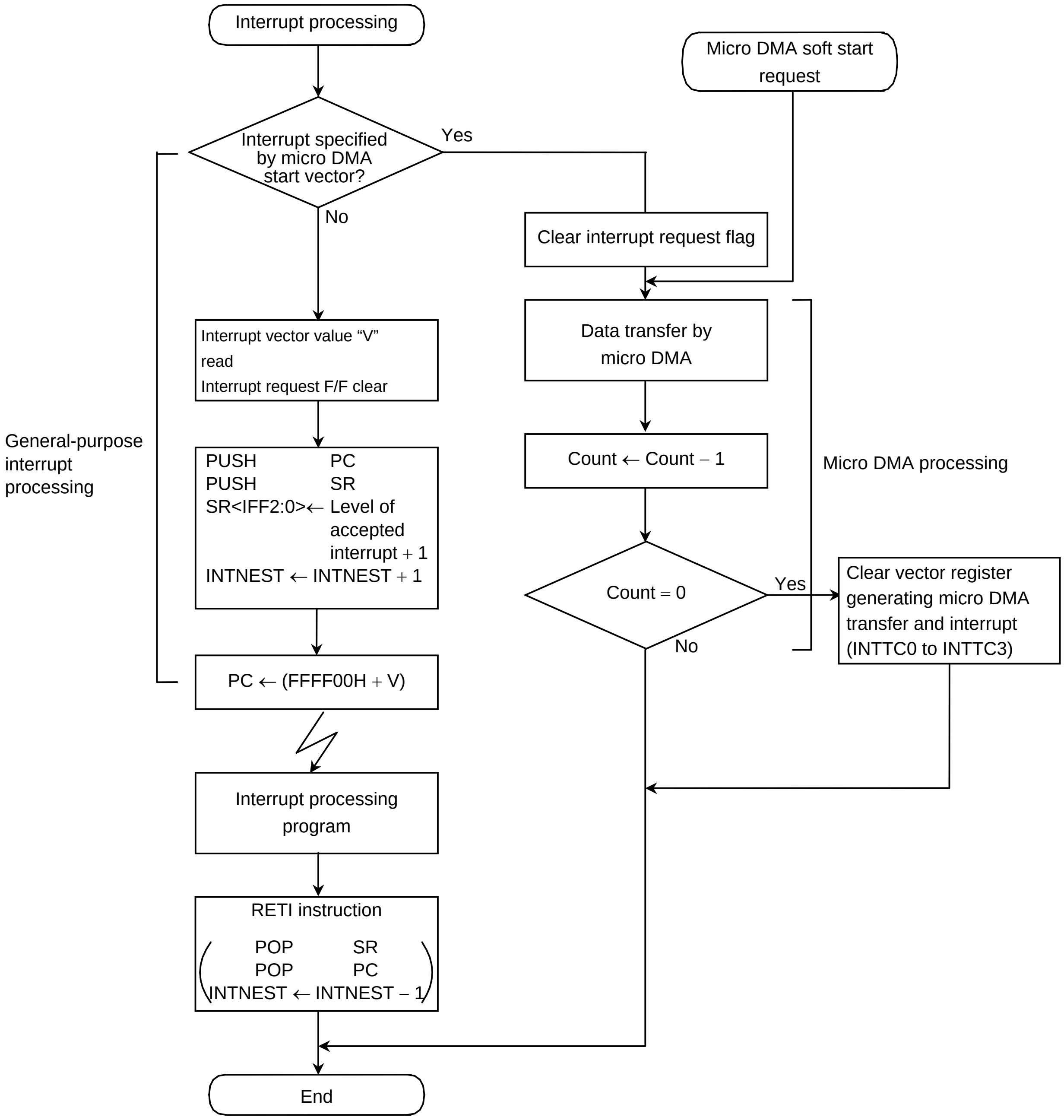

Interrupt processing
Micro DMA soft start request
Interrupt specified by micro DMA start vector?
Yes
No
Clear interrupt request flag
Data transfer by micro DMA
Interrupt vector value "V" read
Interrupt request F/F clear
General-purpose interrupt processing
PUSH PC
PUSH SR
SR<IFF2:0>← Level of accepted interrupt + 1
INTNEST ← INTNEST + 1
Count ← Count − 1
Micro DMA processing
Count = 0
Yes
No
Clear vector register generating micro DMA transfer and interrupt (INTTC0 to INTTC3)
PC ← (FFFF00H + V)
Interrupt processing program
RETI instruction
POP SR
POP PC
INTNEST ← INTNEST − 1
End

Figure 3.4.1 Interrupt and Micro DMA Processing Sequence

3.4.1 General-purpose Interrupt Processing

When the CPU accepts an interrupt, it usually perf
operations. However, in the case of software interru
generated by the CPU, the CPU skips steps a and c and ex

a. The CPU reads the interrupt vector from the inte
If there are simultaneous interrupts set to sam
generates an interrupt vector in accordance wit
the interrupt request.
(The default priority is already fixed: The small
value, the higher the priority level.)

b. The CPU pushes the program counter (PC) and stat
the stack (Pointed to by XSP).

c. The CPU sets the value of the CPU's interrupt m
priority level for the accepted interrupt plus 1. However, if the prior
the accepted interrupt is 7, the register's valu

d. The CPU increments the interrupt nesting counte

e. The CPU jumps to the address indicated by the d
interrupt vector, and starts the interrupt proce

When the CPU completed the interrupt processing, use
the main routine. RETI restores the contents of the
register from the stack and decrements the interrupt

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

If an interrupt request with a priority level eq
than the value set in the CPU interrupt mask register
interrupt. The CPU interrupt mask register <IFF2:0> i
level for the accepted interrupt plus 1.

If, during interrupt processing, an interrupt is generated with a higher l
interrupt begin currently processed, or if, interrup
non-maskable interrupt request is generated from ano
currently processing routine and accepts the later. Then, after
processing the later interrupt, the CPU returns to t
and resumes processing.

If the CPU receives a request for another interrupt w
toe, the second interrupt is sampled immediately after execution of the fir
its interrupt processing routine. Specifying DI as t
interrupt nesting. (Notes: In the 900 and 900/L sampling is performed, befor
the start instruction.)

A reset initializes the interrupt mask register <IF
interrupts.

Table 3.4.1 shows the TMP91C820A interrupt vectors
FFFF00H to FFFFFFH (256 bytes) is designated as the interrupt vector are

Table 3.4.1 TMP91C820A Interrupt Vectors and Micro DMA Start Vectors

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 "SWI0" instruction	0000H	FFFF00H	–
2		"SWI1" instruction	0004H	FFFF04H	–
3		INTUNDEF: Illegal instruction or "SWI2" instruction	0008H	FFFF08H	–
4		"SWI3" instruction	000CH	FFFF0CH	–
5		"SWI4" instruction	0010H	FFFF10H	–
6		"SWI5" instruction	0014H	FFFF14H	–
7		"SWI6" instruction	0018H	FFFF18H	–
8		"SWI7" instruction	001CH	FFFF1CH	–
9		$\overline{\text{NMI}}$ pin	0020H	FFFF20H	–
10		INTWD: Watchdog timer	0024H	FFFF24H	–
–	Maskable	(Micro DMA)	–	–	–
11		INT0 pin	0028H	FFFF28H	0AH
12		INT1 pin	002CH	FFFF2CH	0BH
13		INT2 pin	0030H	FFFF30H	0CH
14		INT3 pin	0034H	FFFF34H	0DH
15		INTALM0: ALM0 (8 kHz)	0038H	FFFF38H	0EH
16		INTALM1: ALM1 (512 Hz)	003CH	FFFF3CH	0FH
17		INTALM2: ALM2 (64 Hz)	0040H	FFFF40H	10H
18		INTALM3: ALM3 (2 Hz)	0044H	FFFF44H	11H
19		INTALM4: ALM4 (1 Hz)	0048H	FFFF48H	12H
20		INTTA0: 8-bit timer 0	004CH	FFFF4CH	13H
21		INTTA1: 8-bit timer 1	0050H	FFFF50H	14H
22		INTTA2: 8-bit timer 2	0054H	FFFF54H	15H
23		INTTA3: 8-bit timer 3	0058H	FFFF58H	16H
24		INTRX0: Serial receives (Channel 0)	005CH	FFFF5CH	17H
25		INTTX0: Serial transmission (Channel 0)	0060H	FFFF60H	18H
26		INTRX1: Serial receives (Channel 1)	0064H	FFFF64H	19H
27		INTTX1: Serial transmission (Channel 1)	0068H	FFFF68H	1AH
28		INTAD: AD conversion end	006CH	FFFF6CH	1BH
29		INTKEY: Key-on wakeup	0070H	FFFF70H	1CH
30		INTRTC: RTC (Alarm interrupt)	0074H	FFFF74H	1DH
31		INTSBI: SBI interrupt	0078H	FFFF78H	1EH
32		INTLCD: LCDC/LP pin	007CH	FFFF7CH	1FH
33		INTP0: Protect 0 (WR to special SFR)	0080H	FFFF80H	20H
34		INTP1: Protect 1 (WR to ROM)	0084H	FFFF84H	21H
35		INTTC0: Micro DMA end (Channel 0)	0088H	FFFF88H	–
36		INTTC1: Micro DMA end (Channel 1)	008CH	FFFF8CH	–
37		INTTC2: Micro DMA end (Channel 2)	0090H	FFFF90H	–
38		INTTC3: Micro DMA end (Channel 3)	0094H	FFFF94H	–
39		Reserved	–	–	–
40		Reserved	–	–	–
41		Reserved	–	–	–
42		INTRX2: Serial receive (Channel 2)	00A4H	FFFFA4H	29H
43		INTTX2: Serial transmission (Channel 2)	00A8H	FFFFA8H	2AH
44		INTTB00: 16-bit timer 0 (TB0RG0)	00ACH	FFFFACH	2BH
45		INTTB01: 16-bit timer 1 (TB0RG1)	00B0H	FFFFB0H	2CH
		(Reserved) to (Reserved)	00B4H to 00FCH	FFFFB4H to FFFFFCH	– to –

3.4.2 Micro DMA Processing

In addition to general-purpose interrupt processin
DMA function. Interrupt requests set by micro DMA per
highest priority level for maskable interrupts (Lev
the particular interrupt source.

Because the micro DMA function has been implemented
of CPU, when CPU is a state of standby by HALT instruc
DMA will be ignored (Pending).

(1) Micro DMA operation

When an interrupt request is generated by an interr
DMA start vector register, the micro DMA request to
interrupt priority level 6 and starts processing the request. The four
channels allow micro DMA processing to be set for up to four types of
one time.

When micro DMA is accepted, the interrupt reques
channel is cleared. The data are automatically tra
address to the transfer address set in the control reg
counter is decremented by 1. If the decremented co
processing ends with no change in the value of the mi
the decremented reading is 0, a micro DMA transfer end interrupt
INTTC3) passes from the CPU to the interrupt contro
start vector register is cleared to 0, the next mic
processing completes.

If a micro DMA request is set for more than one channel at the same time, the pr
based on the interrupt priority level but on the c
channel number the higher the priority (Channel 0 (High) → Channel 3 (Low))

If an interrupt request is triggered for the interr
between the clearing of the micro DMA start vec
general-purpose interrupt processing executes at
only using the interrupt for micro DMA (Not using the inte
general-purpose interrupt: level 6), first set the interrupts level to 0
disabled).

If using micro DMA and general-purpose interrupt
first set the level of the interrupt used for micro DMA processing l
the other interrupt levels. In this case, the cause
edge interrupt.

As with other maskable interrupts, the priority of the micro DMA tr
interrupts is determined by the interrupt level and

While the register for setting the transfer source/destination addresses 32-bit control register, this register can only output 24- Accordingly, micro DMA can access 16 Mbytes (The up valid).

Three micro DMA transfer modes are supported: 1-byte transfer, 2-by transfer, and 4-byte transfer. After a transfe source/destination addresses are incremented, de

This simplifies the transfer of data from I/O to m from I/O to I/O. For details of the transfer modes, s the transfer mode register”. As the transfer count processing can be set for up to 65536 times (The micro DM processing count is maximized when the initial value i

Micro DMA processing can be started by the 31 inter start vectors of Table 3.4.1 and by the micro DMA s interrupts.

Figure 3.4.2 shows the micro DMA cycle in transfer d address INC mode (except for counter mode, the same as for other mode

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

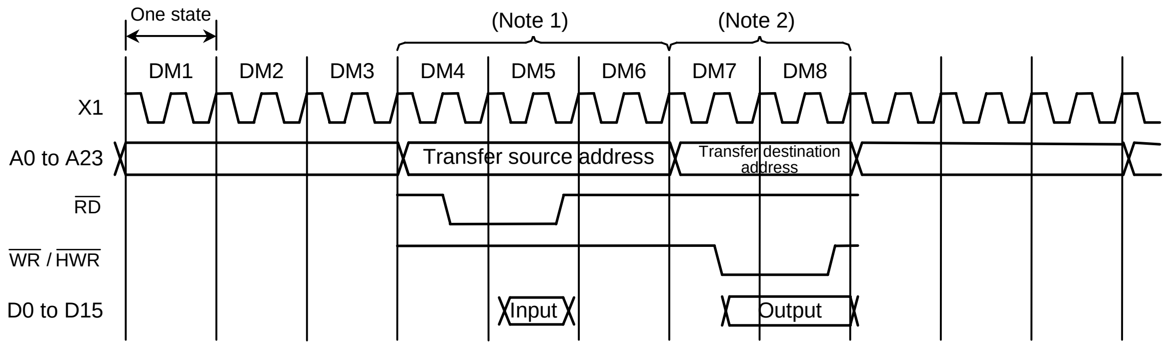


Figure 3.4.2 Timing for Micro DMA Cycle

States 1 to 3: Instruction fetch cycle (Gets next address code).
If three or 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 incremented by two states.
If the source address area is a 16-bit bus and the address starts from an odd number, it is incremented by two states.

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

(2) Soft start function

In addition to starting the micro DMA function by i
a micro DMA software starrtt fsumnicctriooDnMtAhoant tshte genera
write cycle to the DMAR register.

Writing 1 to each bit of DMAR register causes mic
transfer, the bit of the DMAR register which su
automatically cleared to 0.

Only one channel can be set for DMA request at once
one bit.)

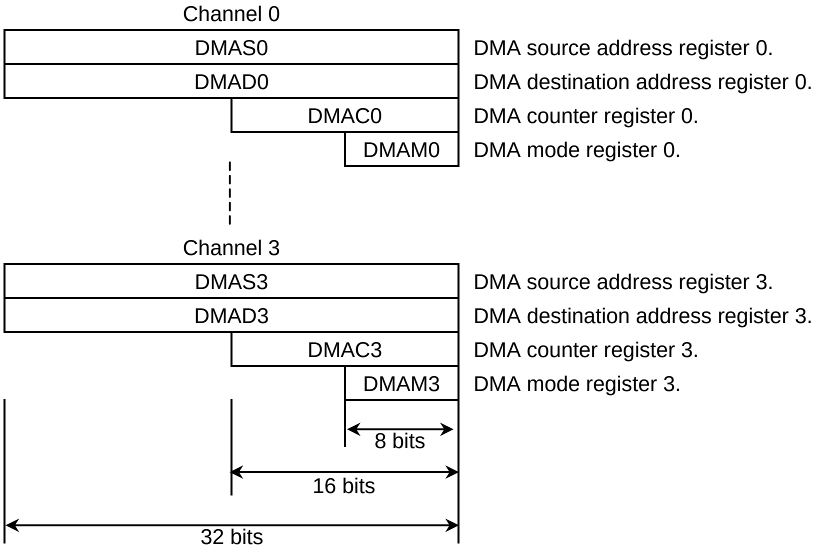
When writing again 1 to the DMAR register, check
writing 1.

When a burst is specified by DMAB register, data is
the value in the micro DMA transfer counter is 0 afte

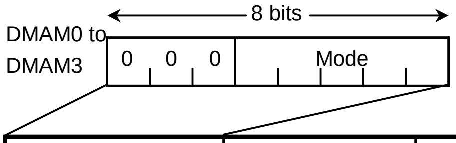
Symbol	Name	Address	7	6	5	4	3	2	1	0
DMAR	DMA request register	89H					DMA request			
							DMAR3	DMAR2	DMAR1	DMAR0
							R/W			
							0	0	0	0

(3) Transfer control registers

The transfer source adsdfreers sdeasntdi tnhaet itoranaddress a
following registers. Arnmi "hLsDtC ucrc t, iro"n coafn tbhee ufsoed to
registers.



(4) Detailed description of the transfer mode regist



Note: When setting a value in this register, write 0 to the upper three bits.

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

(*) For external 16-bit bus, 0 waits, word/4-byte transfer mode, transfer source/transfer destination addresses both have even-numbered values.

Note: n: Corresponding micro DMA channels 0 to 3.
DMADn+/DMASn+: Post increment (Increments register value after transfer).
DMADn–/DMASn–: Post decrement (Decrements register value after transfer).
The I/Os in the table mean fixed address; memory means increment and decrement addresses.
Do not use undefined code, that is, codes other than those listed above for the transfer mode register.

3.4.3 Interrupt Controller Operation

The block diagram in Figure 3.4.3 shows the interrupt circuits. The left- diagram shows the interrupt controller circuit. The right-hand side sh interrupt request signal circuit and the halt release

For each of the 36 interrupt channels there is an int flip-flop), an interrupt priority setting register and a micro DMA start vector regis interrupt request flag latches interrupt requests from the peripherals. The fl zero in the following cases: when reset occurs, when t interrupt it has received, when the CPU receives a micro DMA request (Whe is set), when the micro DMA burst transfer is termina clears the interrupt for that channel is executed (b interrupt priority setting register).

An interrupt priority can be set independently for each interrupt source b priority to the interrupt priority setting register (e.g., INTE0AD or INTE12) priorities levels (1 to 6) are provided. Setting an in disables interrupt requests from that source. The priority of non-maskable i pin interrupts and watchdog timer interrupts) is fixed at 7. If interrupt rec same level are generated at the same time, the default priority (The inte lowest priority or, in other words, the interrupt wi determine which interrupt request is accepted first.

The 3rd and 7th bits of the interrupt priority setti interrupt request flag and thus whether an interrup occurred.

The interrupt controller sends the interrupt request with the highest prio simulateous interrupts and its vector address to the value <IFF2:0> et in the status register by the inter value set; if the latter is higher, the interrupt is ac than the priority value by 1 in the CPU SR<IFF2:0>. In value equals or is higher than the set value are acc previous interrupt routine.

When interrupt processing is completed (after exec CPU restores the priority value saved on the stack before the interrupt wa the CPU SR<IFF2:0>. The interrupt controller also ha micro DMA start vector. Writing the start vector of the interrupt source for t processing (See Table 3.4.1), enables the correspond DMA processing. The values must be set in the micro DMA parameter register (e.g., D and DMAD) prior to the micro DMA processing.

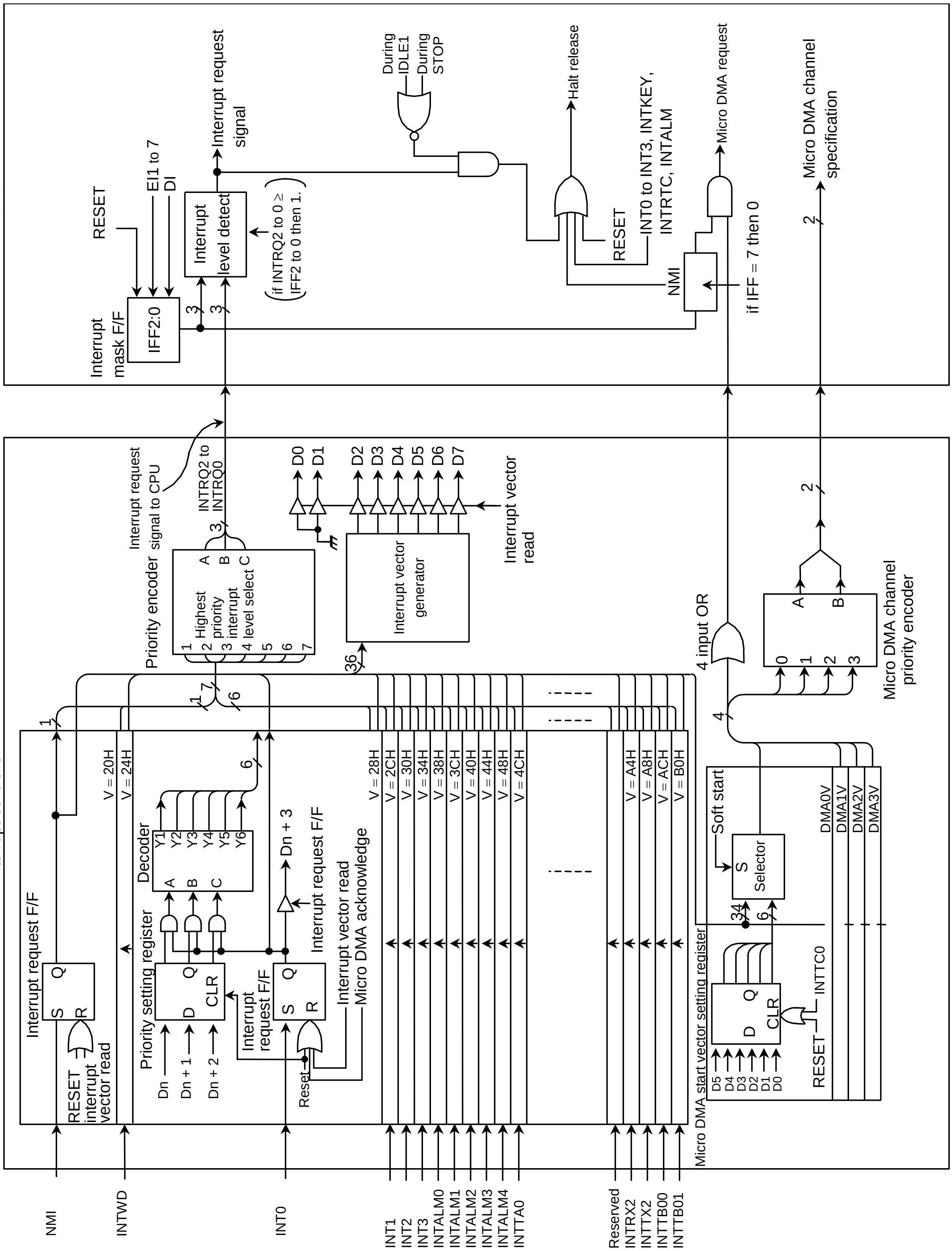
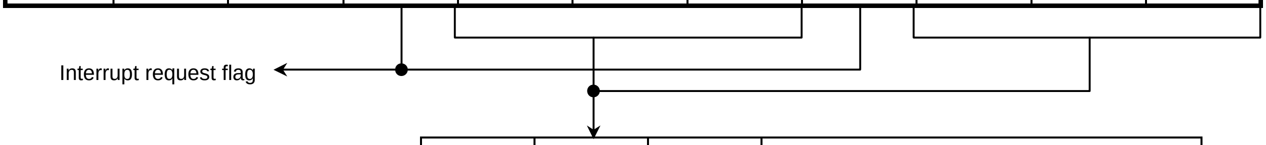


Figure 3.4.3 Block Diagram of Interrupt Controller

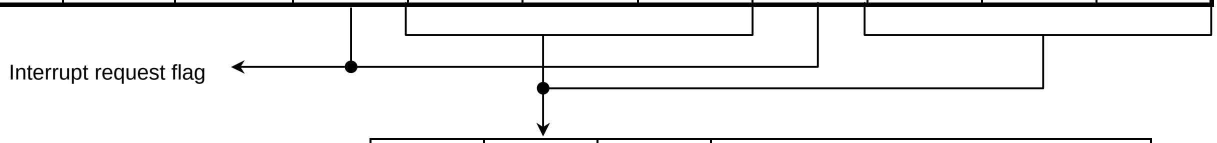
(1) Interrupt level setting registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE0AD	INT0 and INTAD enable	90H	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE12	INT1 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
INTE3 ALM4	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
INTEALM 01	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
INTEALM 23	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
INTETA 01	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
INTETA 23	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
INTERTC KEY	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
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



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
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
INTES2 LCD	INTSBI and INTLCD enable	9AH	INTLCD				INTSBI			
			ILCD1C	ILCDM2	ILCDM1	ILCDM0	ISBIC	ISBIM2	ISBIM1	ISBIM0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTET C01	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
INTET C23	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
INTE P01	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
INTES3	INTRX2 and INTTX2 enable	A0H	INTTX2				INTRX2			
			ITX2C	ITX2M2	ITX2M1	ITX2M0	IRX2C	IRX2M2	IRX2M1	IRX2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETB0	INTTB00 and INTTB01 enable	A1H	INTTB01				INTTB00			
			ITB1C	ITB1M2	ITB1M1	ITB1M0	ITB0C	ITB0M2	ITB0M1	ITB0M0
			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 (No RMW)	–	–	I3EDGE	I2EDGE	I1EDGE	I0EDGE	I0LE	NMIREE
			W							
			0	0	0	0	0	0	0	0
			Write “0”.	Write “0”.	INT3EDGE 0: Rising 1: Falling	INT2EDGE 0: Rising 1: Falling	INT1EDGE 0: Rising 1: Falling	INT0EDGE 0: Rising 1: Falling	0: INT0 edge mode 1: INT0 level mode	1: Operates even on rising/ falling edge of $\overline{\text{NMI}}$

INT0 level enable

0	edge detect INT
1	H level INT

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

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

(3) Interrupt request flag clear register

The interrupt request flag is cleared by writing the appropriate mic vector, as given in Table 3.4.1 to the register INTC

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

INTCLR ← 0AH: Clears interrupt request flag INT0.

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTCLR	Interrupt clear control	88H (No RMW)			CLRV5	CLRV4	CLRV3	CLRV2	CLRV1	CLRV0
					W					
					0	0	0	0	0	0
					Interrupt vector					

(4) Micro DMA start vector registers

This register assigns micro DMA processing to an interrupt sou source with a micro DMA start vector that matches the vector set in thi assigned as the micro DMA start source. When the mic reaches zero, the micro DMA transfer end interrupt sent to the interrupt controller, the micro DMA sta micro DMA start source for the channel is cleared. Therefore, to contin processing, set the micro DMA start vector registe micro DMA transfer end interrupt.

If the same vector is set in the micro DMA start vec channel; the channel with the lowest number has a hi

Accordingly, if the same vector is set in the micro DMA start vector reg channels, the interrupt generated in the channel w until micro DMA transfer is complete. If the vector for th not set again, the next micro DMA channel with the hig (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
counter register reaches zero aSetting a DM which c
the micro DMA channel of the DMAB registers mention

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMAR	DMA software request register	89H					DMAR3	DMAR2	DMAR1	DMAR0
							R/W	R/W	R/W	R/W
							0	0	0	0
							1: DMA software request			
DMAB	DMA burst register	8AH					DMAB3	DMAB2	DMAB1	DMAB0
							R/W			
							0	0	0	0
							1: DMA burst request			

(6) Notes

The instruction execution unit and the bus inter independently. Therefore, immediately before an fetches an instruction that clears the corresponding interrupt reques may execute the instruction that clears the interrupt request flag bet and reading the interrupt vector. In this case, the and reads the interrupt vector address FFFF08H.

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

In the case of changing the value of the interrup execution of POP SR instruction, disable an interrupt by DI instruc execution of POP SR instruction.

In addition, take care as the following 2 circuits 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 an instruction.

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

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

INTRX: Instructions which read the receive buffer.

3.5 Port Functions

The TMP91C820A features 126-bit settings which relate
As well as general-purpose I/O port functionality, the port pins also have I/O functio
relate to the built-in CPU and internal I/Os. Table 3.5.1 lists the func
pin. Tables 3.5.2 and 3.5.3 list I/O registers and their speci

Table 3.5.1 Port Functions (1/2)

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

Port Name	Pin Name	Number of Pins	Direction	R	Direction Setting Unit	Pin Name for Built-in Function
Port 0	P00 to P07	8	I/O	–	Bit	D0 to D7
Port 1	P10 to P17	8	I/O	–	Bit	D8 to D15
Port 2	P20 to P27	8	I/O	–	Bit	A16 to A23
Port 3	P30 to P37	8	I/O	–	Bit	A8 to A15
Port 4	P40 to P47	8	I/O	–	Bit	A0 to A7
Port Z	PZ0	1	Output	–	Bit	$\overline{RD}$
	PZ1	1	Output	–	Bit	$\overline{WR}$
	PZ2	1	I/O	PU	Bit	$\overline{HWR}$
	PZ3	1	I/O	PU	Bit	$R/\overline{W}$, $\overline{SRWE}$
Port 5	P56	1	I/O	PU	Bit	$\overline{WAIT}$
Port 6	P60	1	Output	–	(Fixed)	$\overline{CS0}$
	P61	1	Output	–	(Fixed)	$\overline{CS1}$, $\overline{SDCS}$
	P62	1	Output	–	(Fixed)	$\overline{CS2}$, $\overline{CS2A}$
	P63	1	Output	–	(Fixed)	$\overline{CS3}$
	P64	1	Output	–	(Fixed)	EA24, $\overline{CS2B}$
	P65	1	Output	–	(Fixed)	EA25, $\overline{CS2C}$
	P66	1	Output	–	(Fixed)	$\overline{CS2C}$, $\overline{SRLB}$
	P67	1	Output	–	(Fixed)	$\overline{CS2E}$, $\overline{SRUB}$
Port 7	P70	1	I/O	–	Bit	SCK, OPTRX0
	P71	1	I/O	–	Bit	SO, $\overline{SDA}$, OPTTX0
	P72	1	I/O	–	Bit	SI/SCL
	P73	1	I/O	–	Bit	$\overline{CS2F}$
	P74	1	I/O	–	Bit	$\overline{CS2G}$
	P75	1	I/O	–	Bit	$\overline{CSEXA}$
	P76	1	I/O	–	Bit	MSK
	P77	1	I/O	–	Bit	VEECLK
Port 8	P80 to P87	8	Input	–	(Fixed)	AN0 to AN7, $\overline{ADTRG}$ (P83)
Port 9	P90 to P97	8	Input	U	(Fixed)	KI0 to KI7
Port A	PA0 to PA7	8	Output	–	(Fixed)	KO0 to KO7
Port B	PB0	1	I/O	–	Bit	TA0IN, TXD2
	PB1	1	I/O	–	Bit	TA1OUT, RXD2
	PB3	1	I/O	–	Bit	INT0
	PB4	1	I/O	–	Bit	INT1
	PB5	1	I/O	–	Bit	INT2, TA3OUT
	PB6	1	I/O	–	Bit	INT3, TB0OUT0
Port C	PC0	1	I/O	–	Bit	TXD0
	PC1	1	I/O	–	Bit	RXD0
	PC2	1	I/O	–	Bit	SCLK0, $\overline{CTS0}$
	PC3	1	I/O	–	Bit	TXD1
	PC4	1	I/O	–	Bit	RXD1
	PC5	1	I/O	–	Bit	SCLK1, $\overline{CTS1}$

Table 3.5.2 Port Functions (2/2)

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

Port Name	Pin Name	Number of Pins	Direction	R	Direction Setting Unit	Pin Name for Built-in Function
Port D	PD0	1	Output	–	(Fixed)	D1BSCP
	PD1	1	Output	–	(Fixed)	D2BLP
	PD2	1	Output	–	(Fixed)	D3BFR
	PD3	1	Output	–	(Fixed)	DLEBCD
	PD4	1	Output	–	(Fixed)	DOFFB
	PD6	1	Output	–	(Fixed)	$\overline{\text{ALARM}}$, $\overline{\text{MLDALM}}$
	PD7	1	Output	–	(Fixed)	MLDALM
Port E	PD0 to PD7	8	I/O	–	Bit	LD0 to LD7
Port F	PF0	1	Output	–	(Fixed)	$\overline{\text{SDRAS}}$
	PF1	1	Output	–	(Fixed)	$\overline{\text{SDCAS}}$
	PF2	1	Output	–	(Fixed)	$\overline{\text{SDWE}}$
	PF3	1	Output	–	(Fixed)	SDLDQM
	PF4	1	Output	–	(Fixed)	SDUDQM
	PF5	1	Output	–	(Fixed)	SDCKE
	PF6	1	Output	–	(Fixed)	SDCLK
	PF7	1	Output	–	(Fixed)	

Table 3.5.3 I/O Registers and Specifications (1/3)

Port	Pin Name	Specification	I/O Register			
			Pn	PnCR	PnFC	PnFC2
Port 0	P00 to P07	Input port	X	0	None	None
		Output port	X	1		
		D0 to D7 bus	X	X		
Port 1	P10 to P17	Input port	X	0	0	None
		Output port	X	1	0	
		D8 to D15 bus	X	0	1	
Port 2	P20 to P27	Input port	X	0	X	None
		Output port	X	1	0	
		A16 to A23 output	X	1	1	
Port 3	P30 to P47	Input port	X	0	X	None
		Output port	X	1	0	
		A8 to A15 output	X	1	1	
Port 4	P30 to P47	Input port	X	0	X	None
		Output port	X	1	0	
		A0 to A7 output	X	1	1	
Port Z	PZ0	Output port	X	None	0	None
		$\overline{\text{RD}}$ output	X		1	
	PZ1	Output port	X		0	
		$\overline{\text{WR}}$ output	X		1	
	PZ2, PZ3	Input port (without PU)	0	0	0	
		Input port (with PU)	1	0	0	
		Output port	X	1	0	
	PZ2	$\overline{\text{HWR}}$ output	X	1	1	
	PZ3	R/W output	X	0	1	
		$\overline{\text{SRWE}}$ output	X	1	1	
Port 5	P56	Input port (without PU)	0	0	None	
		Input port (with PU)	1	0		
		Output port	X	1		
		$\overline{\text{WAIT}}$ input (without PU)	0	0		
		$\overline{\text{WAIT}}$ input (with PU)	1	0		
Port 6	P60 to P67	Output port	X	None	0	0
	P60	$\overline{\text{CS0}}$ output	X		1	0
	P61	$\overline{\text{CS1}}$ output	X		1	0
		$\overline{\text{SDCS}}$ output	X		X	1
	P62	$\overline{\text{CS2}}$ output	X		1	0
		$\overline{\text{CS2A}}$ output	X		X	1
	P63	$\overline{\text{CS3}}$ output	X		1	0
	P64	EA24 output	X		1	0
		$\overline{\text{CS2B}}$ output	X		X	1
	P65	EA25 output	X		1	0
		$\overline{\text{CS2C}}$ output	X		X	1
	P66	$\overline{\text{SRLB}}$ output	X		1	0
		$\overline{\text{CS2D}}$ output	X		X	1
	P67	$\overline{\text{SRUB}}$ output	X		1	0
		$\overline{\text{CS2E}}$ output	X		X	1

X: Don't care

Table 3.5.4 I/O Registers and Specifications (2/3)

Port	Pin Name	Specification	I/O Register			
			Pn	PnCR	PnFC	PnFC2
Port 7	P70 to P77	Input port	X	0	0	0
		Output port	X	1	0	0
	P70	SCK input	X	0	0	0
		SCK output	X	1	1	0
		OPTRX0 input (Note 1)	1	0	X	1
	P71	SDA input	X	0	0	0
		SDA output (Note 2)	X	1	1	0
		SO output	X	1	1	0
		OPTTX0 output (Note 1)	1	1	X	1
	P72	SI input	X	0	0	None
		SCL input	X	0	0	
		SCL output (Note 2)	X	1	1	
	P73	$\overline{\text{CS2F}}$ output	X	1	X	1
	P74	$\overline{\text{CS2G}}$ output	X	1	X	1
	P75	$\overline{\text{CSEXA}}$ output	X	1	X	1
	P76	MSK input (Note 3)	X	0	0	0
	P77	VEECLK output	X	1	1	0
Port 8	P80 to P87	Input port	X	None		None
		AN0 to AN7 input (Note 4)	X			
	P83	$\overline{\text{ADTRG}}$ input (Note 5)	X			
Port 9	P90 to P97	Input port	X	None	0	
		KI0 to KI7 input	X		1	
Port A	PA0 to PA7	Output port	X	None	0	
		KO0 to KO7 output (CMOS)	X		0	
		KO0 to KO7 output (Open drain)	X		1	
Port B	PB0 to PB6	Input port	X	0	0	None
		Output port	X	1	0	
	PB0	TA0IN input	X	0	0	
		TXD2 output (Note 1)	X	1	1	
	PB1	TA1OUT output	X	1	1	
		RXD2 input (Note 1)	X	0	0	
	PB3	INT0 input	X	0	1	
	PB4	INT1 input	X	0	1	
	PB5	INT2 input	0	0	1	
		TA3OUT	1	1	1	
	PB6	INT3 input	0	0	1	
		TB0OUT0	1	1	1	

X: Don’t care

Table 3.5.5 I/O Registers and Specifications (3/3)

Port	Pin Name	Specification	I/O Register			
			Pn	PnCR	PnFC	PnFC2
Port C	PC0 to PC5	Input port	X	0	0	None
		Output port	X	1	0	
	PC0	TXD0 output (Note 1)	1	1	1	
	PC1	RXD0 input (Note 1, 6)	1	0	None	
	PC2	SCLK0 input (Note 1)	1	0	0	
		SCLK0 output (Note 1)	1	1	1	
		$\overline{\text{CTS0}}$ input (Note 1)	1	0	0	
	PC3	TXD1 output (Note 1)	1	1	1	
	PC4	RXD1 input (Note 1)	1	0	None	
	PC5	SCLK1 input (Note 1)	1	0	0	
		SCLK1 output (Note 1)	1	1	1	
		$\overline{\text{CTS1}}$ input (Note 1)	1	0	0	
Port D	PD0 to PD7	Output port	X	None	0	
	PD0	D1BSCP output	X		1	
	PD1	D2BLP output	X		1	
	PD2	D3BFR output	X		1	
	PD3	DLEBCD output	X		1	
	PD4	DOFFB output	X		1	
	PD6	$\overline{\text{ALARM}}$ output	1		1	
		$\overline{\text{MLDALM}}$ output	0		1	
	PD7	MLDALM output	X		1	
Port E	PE0 to PE7	Input port	X	0	0	
		Output port	X	1	0	
		LD0 to LD7 output	X	1	1	
Port F	PF0 to PF7	Output port	X	None	0	
	PF0	$\overline{\text{SDRAS}}$ output	X		1	
	PF1	$\overline{\text{SDCAS}}$ output	X		1	
	PF2	$\overline{\text{SDWE}}$ output	X		1	
	PF3	SDLDQM output	X		1	
	PF4	SDUDQM output	X		1	
	PF5	SDCKE output	X		1	
	PF6	SDCLK output	X		1	

X: Don't care

Note 1: As for input ports of SIO1 to SIO3: (OPTTX0, OPTRX0, TXD0, RXD0, SCLK0, $\overline{\text{CTS0}}$, TXD1, RXD1, SCLK1, $\overline{\text{CTS1}}$, TXD2, RXD2), logical selection for output data or input data is determined by the output latch register Pn of each port.

Note 2: When P71/P72 are used as SDA/SCL open-drain outputs, P70DE<ODEP72:71> is used to set the open-drain output mode.

Note 3: In case using P76 for MSK port, set to P7FC<P76F>.

Note 4: When P80 to P87 are used as AD converter input channels, ADMOD1<ADCH2:0> is used to select the channel.

Note 5: When P83 is used as $\overline{\text{ADTRGE}}$ input, ADMOD1<ADTRGE> is used to enable external-trigger input.

Note 6: In case using PC1 for RXD0 port, set "0" to P7FC2<P70F2>.

3.5.1 Port 0 (P00 to P07)

Port 0 is an 8-bit general-purpose I/O port. Each bit can be set individually for input or output using the control register P0CR. Resetting resets all bits of the output latch P0 and the control register P0CR to 0 and sets port 0 to input mode. In addition to functioning as a general-purpose I/O port, port 0 can also function as a data bus (D0 to D7).

When external memory is accessed, port 0 functions as a data bus (D0 to D7) and all bits of P0CR are cleared to 0.

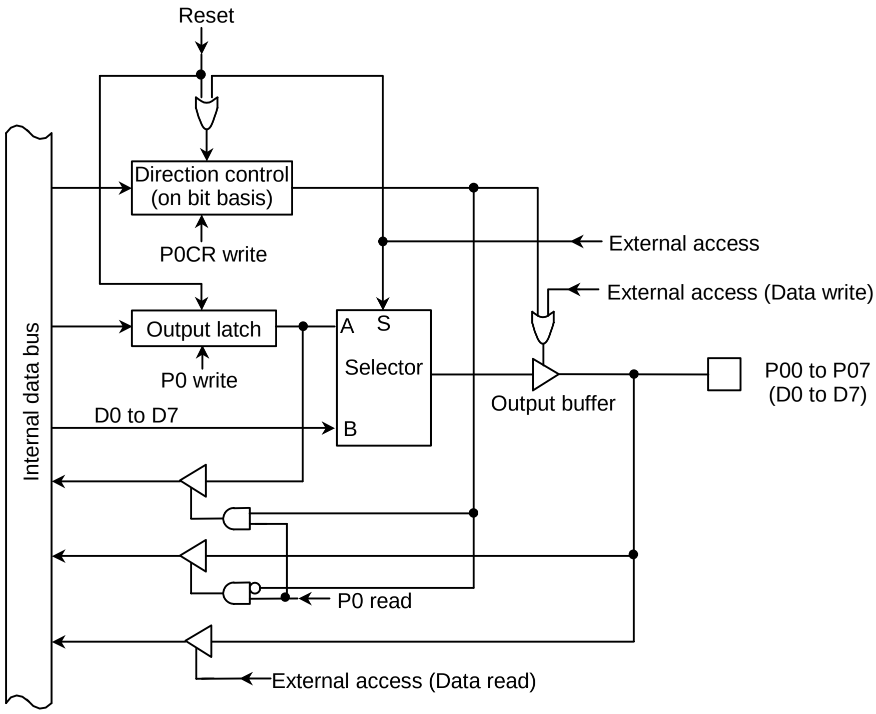


Figure 3.5.1 Port 0

Port 0 Register

		7	6	5	4	3	2	1	0
P0 (0000H)	Bit symbol	P07	P06	P05	P04	P03	P02	P01	P00
	Read/Write	R/W							
	After reset	Data from external port (Output latch register is cleared to 0.)							

Port 0 Control Register

		7	6	5	4	3	2	1	0
P0CR (0002H)	Bit symbol	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	Port 0 input/output settings 0: Input 1:Output							

Note 1: Read-modify-write is prohibited for P0CR.
Note 2: When functioning as a data bus (D0 to D7), P0CR is cleared to 0.

Figure 3.5.2 Register for Port 0

3.5.2 Port 1 (P10 to P17)

Port 1 is an 8-bit general-purpose I/O port. Each bit can be set individual
output using the control register P1CR and the function register P1FC. Rese
bits of the output latch P1, the control register P1C
and sets port 1 to input mode.

In addition to functioning as a general-purpose I/O port, port 1 can also fu
data bus (D8 to D15).

AM1	AM0	P1xF	Function Setting after Reset is Released
0	0	0	Input port
0	1	1	Data bus (D8 to D15)
1	0	–	Don’t use this setting
1	1	0	Input port

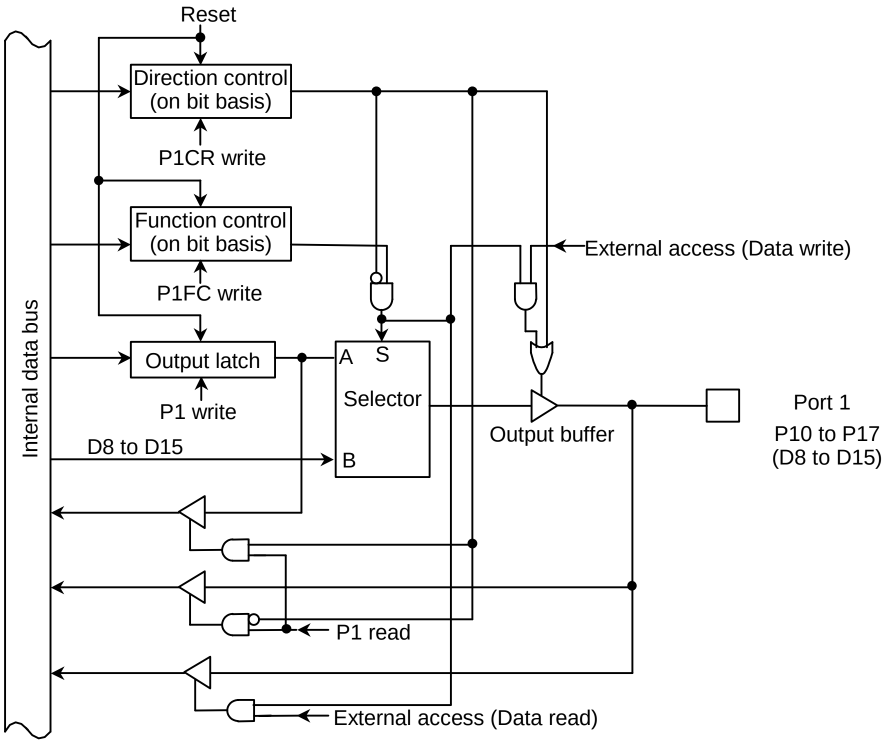


Figure 3.5.3 Port 1

Port 1 Register

		7	6	5	4	3	2	1	0
P1 (0001H)	Bit symbol	P17	P16	P15	P14	P13	P12	P11	P10
	Read/Write	R/W							
	After reset	Data from external port (Output latch register is cleared to 0.)							

Port 1 Control Register

		7	6	5	4	3	2	1	0
P1CR (0004H)	Bit symbol	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	Port 1 function settings							

Port 1 Function Register

		7	6	5	4	3	2	1	0
P1FC (0005H)	Bit symbol	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
	Read/Write	W							
	After reset	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
	Function	Port 1 function settings							

Port 1 function settings

P1FC<P1xF> P1CR<P1xC>	0	1
0	Input port	Data bus (D15 to D8)
1	Output port	Don’t set

Note 1: Read-modify-write is prohibited for P1CR and P1FC.

Note 2: <P1xF> is bit x in register P1FC; <P1xC>, in register P1CR.

Figure 3.5.4 Register for Port 1

3.5.3 Port 2 (P20 to P27)

Port 2 is an 8-bit general-purpose input/output port. Each pin can be set individual output using the control register P2CR and the funct functioning as a general-purpose I/O port, port 2 can also function as an addr to A23).

Setting the AM1 and AM0 pins as shown below initializes the device in to the following function pins.

AM1	AM0	P2xC	P2xF	Function Setting after Reset is Released
0	0	1	1	Address bus (A16 to A23)
0	1	1	1	Address bus (A16 to A23)
1	0	–	–	Don’t use this setting
1	1	0	0	Input port

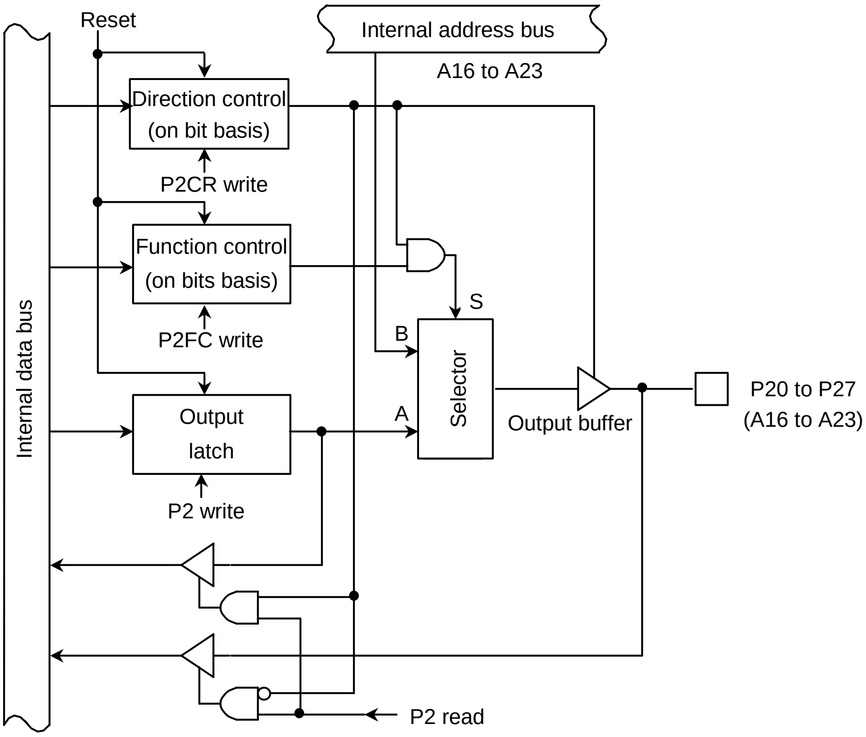


Figure 3.5.5 Port 2

Port 2 Register

		7	6	5	4	3	2	1	0
P2 (0006H)	Bit symbol	P27	P26	P25	P24	P23	P22	P21	P20
	Read/Write	R/W							
	After reset	Data from external port (Output latch register is cleared to 0.)							

Port 2 Control Register

		7	6	5	4	3	2	1	0
P2CR (0008H)	Bit symbol	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
	Read/Write	W							
	After reset	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
	Function	Port 2 function settings							

Port 2 Function Register

		7	6	5	4	3	2	1	0
P2FC (0009H)	Bit symbol	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
	Read/Write	W							
	After reset	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
	Function	Port 2 function settings							

Port 2 function settings

P2FC<P2xF> P2CR<P2xC>	0	1
0	Input port	
1	Output port	Address bus (A16 to A23)

Note 1: Read-modify-write is prohibited for P2CR and P2FC.

Note 2: <P2xF> is bit x in register P2FC; <P2xC>, in register P2CR.

Figure 3.5.6 Register for Port 2

3.5.4 Port 3 (P30 to P37)

Port 3 is an 8-bit general-purpose I/O port. Each bit can be set individual
output using the control register P3CR and the funct
functioning as a general-purpose I/O port, port 3 can
A15).

Setting the AM1 and AM0 pins as shown below allows the device in
to the following function pins.

AM1	AM0	P3xC	P3xF	Function Setting after Reset is Released
0	0	1	1	Address bus (A8 to A15)
0	0	1	1	Address bus (A8 to A15)
1	0	–	–	Don’t use this setting
1	1	0	0	Input port

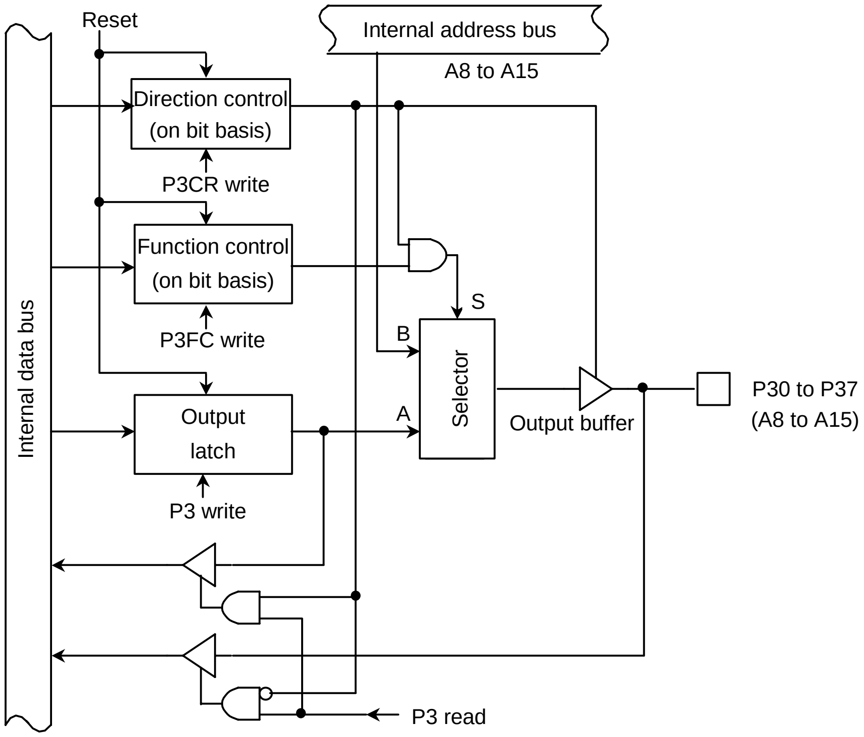


Figure 3.5.7 Port 3

Port 3 Register

P3 (0007H)		7	6	5	4	3	2	1	0
	Bit symbol	P37	P36	P35	P34	P33	P32	P31	P30
	Read/Write	R/W							
	After reset	Data from external port (Output latch register is cleared to 0.)							

Port 3 Control Register

P3CR (000AH)		7	6	5	4	3	2	1	0
	Bit symbol	P37C	P36C	P35C	P34C	P33C	P32C	P31C	P30C
	Read/Write	W							
	After reset	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
	Function	Port 3 function settings							

Port 3 Function Register

P3FC (000BH)		7	6	5	4	3	2	1	0
	Bit symbol	P37F	P36F	P35F	P34F	P33F	P32F	P31F	P30F
	Read/Write	W							
	After reset	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
	Function	Port 3 function settings							

Note 1: Read-modify-write is prohibited for P3CR and P3FC.
Note 2: <P3xF> is bit x in register P3FC; <P3xC>, in register P3CR.

→ Port 3 function settings

P3FC<P3xF> P3CR<P3xC>	0	1
0	Input port	
1	Output port	Address bus (A8 to A15)

Figure 3.5.8 Register for Port 3

3.5.5 Port 4 (P40 to P47)

Port 4 is an 8-bit general-purpose I/O port. Each bit can be set individual
output using the control register P4CR and the funct
functioning as a general-purpose I/O port, port 4 can
A7).

Setting the AM1 and AM0 pins as shown below sets the device in
to the following function pins.

AM1	AM0	P4xC	P4xF	Function Setting after Reset is Released
0	0	1	1	Address bus (A0 to A7)
0	1	1	1	Address bus (A0 to A7)
1	0	–	–	Don’t use this setting
1	1	0	0	Input port

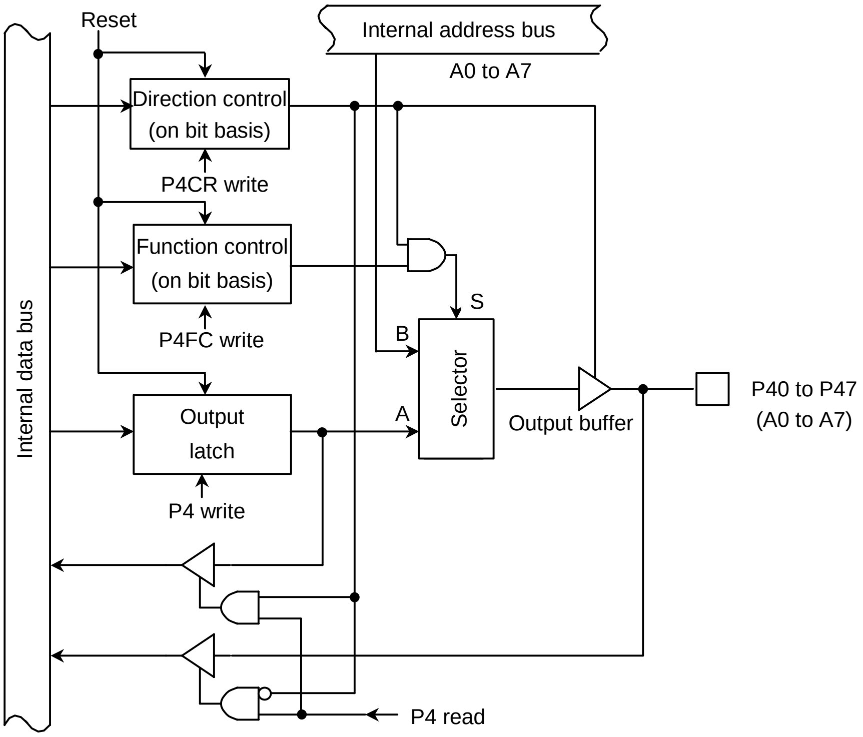


Figure 3.5.9 Port 4

Port 4 Register

		7	6	5	4	3	2	1	0
P4 (000CH)	Bit symbol	P47	P46	P45	P44	P43	P42	P41	P40
	Read/Write	R/W							
	After reset	Data from external port (Output latch register is cleared to 0.)							

Port 4 Control Register

		7	6	5	4	3	2	1	0
P4CR (000EH)	Bit symbol	P47C	P46C	P45C	P44C	P43C	P42C	P41C	P40C
	Read/Write	W							
	After reset	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
	Function	Port 4 function settings							

Port 4 Function Register

		7	6	5	4	3	2	1	0
P4FC (000FH)	Bit symbol	P47F	P46F	P45F	P44F	P43F	P42F	P41F	P40F
	Read/Write	W							
	After reset	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
	Function	Port 4 function settings							

Port 4 function settings

P4FC<P4xF> P4CR<P4xC>	0	1
0	Input port	
1	Output port	Address bus (A0 to A7)

Note 1: Read-modify-write is prohibited for P4CR and P4FC.

Note 2: <P4xF> is bit x in register P4FC; <P4xC>, in register P4CR.

Figure 3.5.10 Register for Port 4

3.5.6 Port Z (PZ0 to PZ3)

Port Z is an 4-bit general-purpose I/O port. Each bit can be set individually for input or output using control register PZCR and PZFC. Resets set the output latch PZ to 1.

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

When PZ0 pin is defined as $\overline{\text{RD}}$ signal output mode (<PZ0F> = 1), clearing the output latch register <PZ0> to 0 outputs the $\overline{\text{RD}}$ strobe (used for the pseudo static RAM) from the PZ0 pin even when the internal address area is accessed. If the output latch register <PZ0> remains 1, the $\overline{\text{RD}}$ strobe signal is output only when the external address area is accessed.

Resetting initializes PZ2 and PZ3 pins to input mode.

Setting the AM1 and AM0 pins as shown below and resetting the device initializes PZ0 and PZ1 pins to the following function pins.

AM1	AM0	PZ0F PZ1F	Function Setting after Reset is Released	
			PZ0 function	PZ1 function
0	0	1	$\overline{\text{RD}}$ pin	$\overline{\text{WR}}$ pin
0	1	1	$\overline{\text{RD}}$ pin	$\overline{\text{WR}}$ pin
1	0	–	Don’t use this setting	Don’t use this setting
1	1	0	Output port	Output port

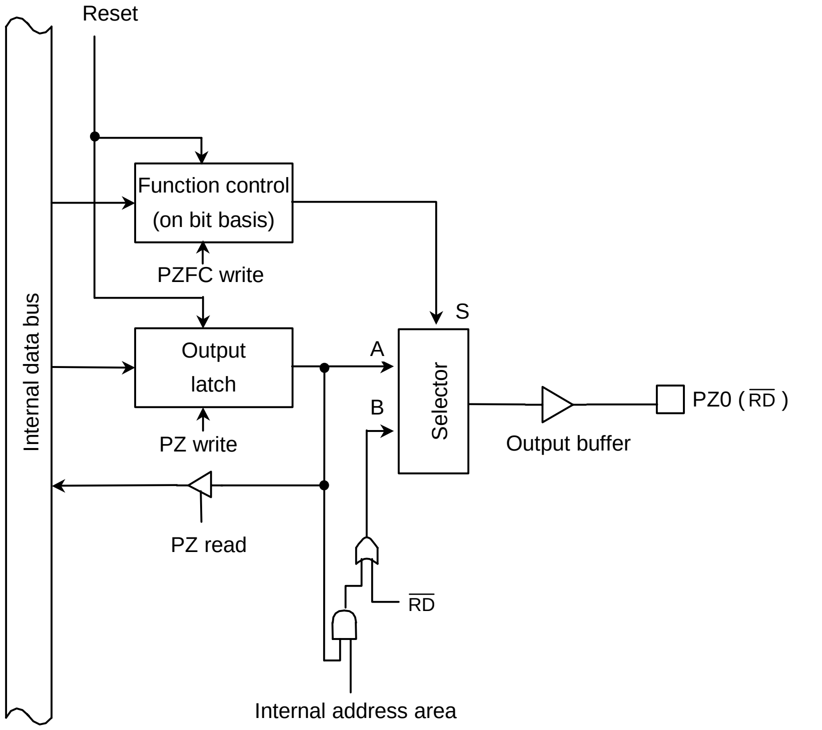


Figure 3.5.11 Port Z (PZ0)

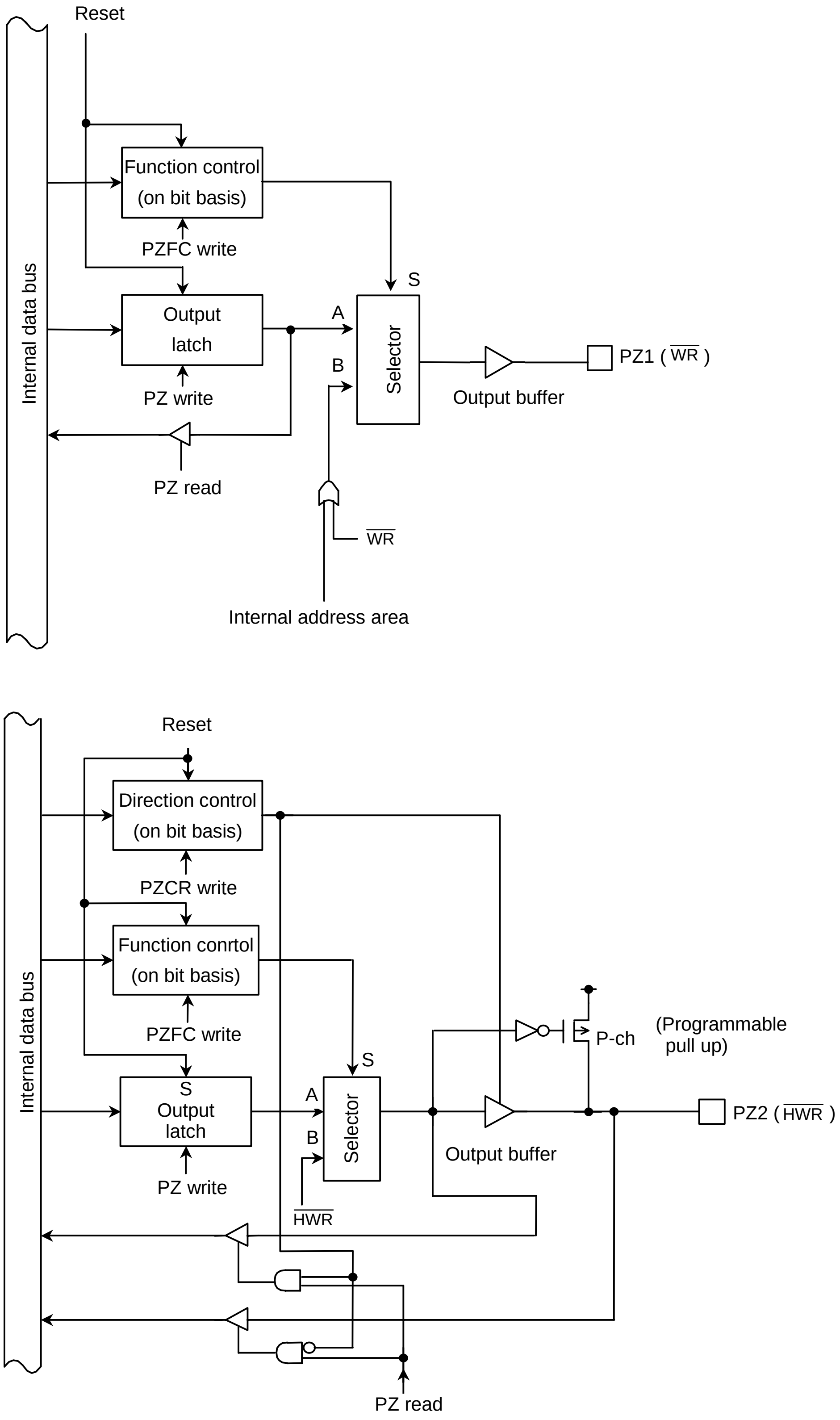


Figure 3.5.12 Port Z (PZ1, PZ2)

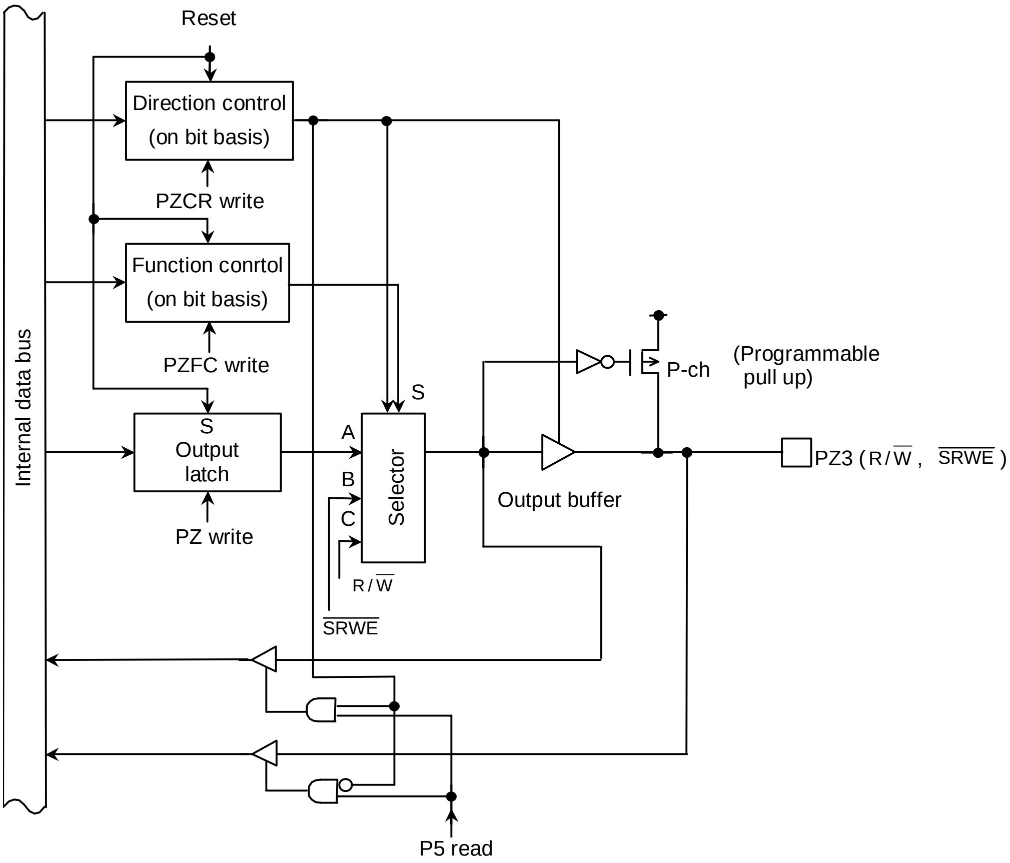


Figure 3.5.13 Port Z (PZ3)

Port Z Register

PZ (007DH)

	7	6	5	4	3	2	1	0
Bit symbol					PZ3	PZ2	PZ1	PZ0
Read/Write					R/W			
After reset					Data form external port (Note 1)		1	1
Function					0: Pull-up resistor OFF 1: Pull-up resistor ON		–	–

Port Z Control Register

PZCR (007EH)

	7	6	5	4	3	2	1	0
Bit symbol					PZ3C	PZ2C		
Read/Write					W			
After reset					0	0		
Function					0: Input 1: Output			

Port Z Function Register

PZFC (007FH)

	7	6	5	4	3	2	1	0
Bit symbol					PZ3F	PZ2F	PZ1F	PZ0F
Read/Write					W			
After reset					0	0	0	0
Function					0: Port 1: R/$\overline{\text{W}}$, $\overline{\text{SRWE}}$	0: Port 1:$\overline{\text{HWR}}$	0: Port 1:$\overline{\text{WR}}$	0: Port 1:$\overline{\text{RD}}$

Note 1: Output latch register is set to 1.
Note 2: Read-modify-write is prohibited for registers PZCR and PZFC.
Note 3: When port Z is used in input mode, the PZ register controls the built-in pull-up resistor. Read-modify-write is prohibited in input mode or I/O mode. Setting the built-in pull-up resistor may be depended on the states of the input pin.

R/$\overline{\text{W}}$, $\overline{\text{SRWE}}$ setting

<PZ3C> <PZ3F>	0	1
0	Input	Output
1	R/W	$\overline{\text{SRWE}}$

PZ0 ($\overline{\text{RD}}$) function setting

<PZ0> <PZ0F>	0	1
0	Output	Output
1	Prohibit this setting	$\overline{\text{RD}}$

PZ1 ($\overline{\text{WR}}$) function setting

<PZ1> <PZ1F>	0	1
0	Output	Output
1	$\overline{\text{WR}}$	

$\overline{\text{HWR}}$ setting

PZFC<PZ2F>	1
PZCR<PZ2C>	1

Figure 3.5.14 Register for Port Z

3.5.7 Port 5 (P56)

Port 5 is an 1-bit general-purpose I/O port. I/O is set using control reg
P5FC. Resetting resets all bits of the output latch P5 to P1.

In addition to functioning as a general-purpose I/O port, P56 also funct
the CPU's control/status signal.

Resetting initializes P56 pins to input mode with pu

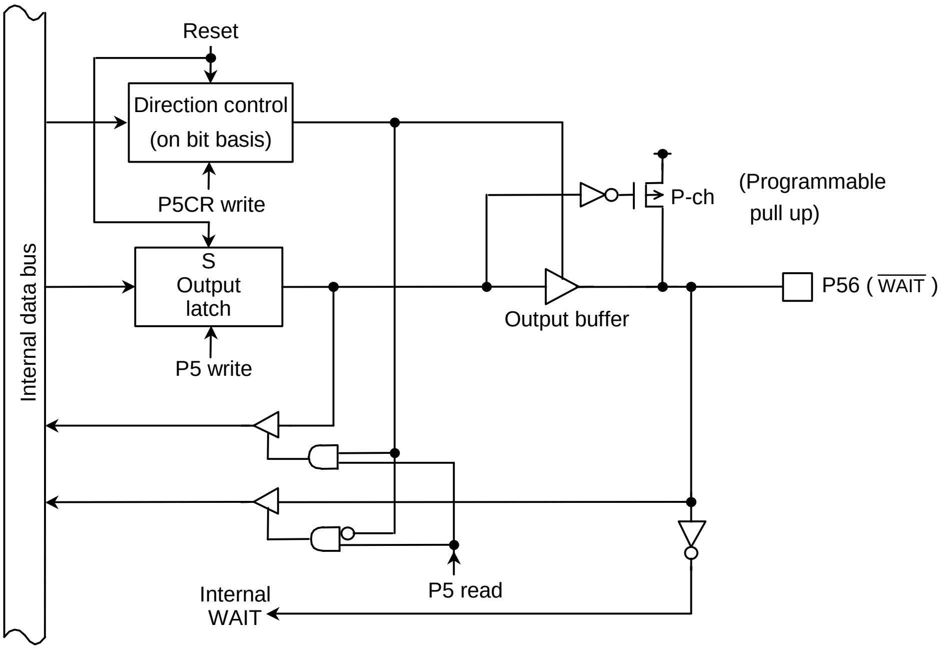


Figure 3.5.15 Port 5 (P56)

Port 5 Register

		7	6	5	4	3	2	1	0
P5 (000DH)	Bit symbol		P56						
	Read/Write		R/W						
	After reset		Data from external port (Output latch register is set to 1.)						
	Function		0: Pull-up resistor OFF 1: Pull-up resistor ON						

Port 5 Control Register

		7	6	5	4	3	2	1	0
P5CR (0010H)	Bit symbol		P56C						
	Read/Write		W						
	After reset		0						
	Function		0: Input 1: Output						

Note: When the P53/WAIT pin is to be use as the WAIT pin, P5CR<P53C> must be set to 0 and <BnW2:0> in the chip select/wait control register must be set 010.

Figure 3.5.16 Register for Port 5

3.5.8 Port 6 (P60 to P67)

Port 60 to 67 are 8-bit output ports. Resetting sets
latches of P60 to P61 and P63 to P67 are set to 1.

Port 6 also function as chip select output ($\overline{\text{CS0}}$ to $\overline{\text{CS3}}$), extend address output (
EA25), extend chip select output ($\overline{\text{CS2A}}$, $\overline{\text{CS2B}}$, $\overline{\text{CS2C}}$, $\overline{\text{CS2D}}$, $\overline{\text{CS2E}}$), SRAM byte contr
output ($\overline{\text{SRUB}}$, $\overline{\text{SRLB}}$), and SDRAM chip-select output ($\overline{\text{SDCS}}$).

Writing 1 in the corresponding bit of P6FC, P6FC2 ena

Resetting reset the P6FC and P6FC2 to 0, and sets all

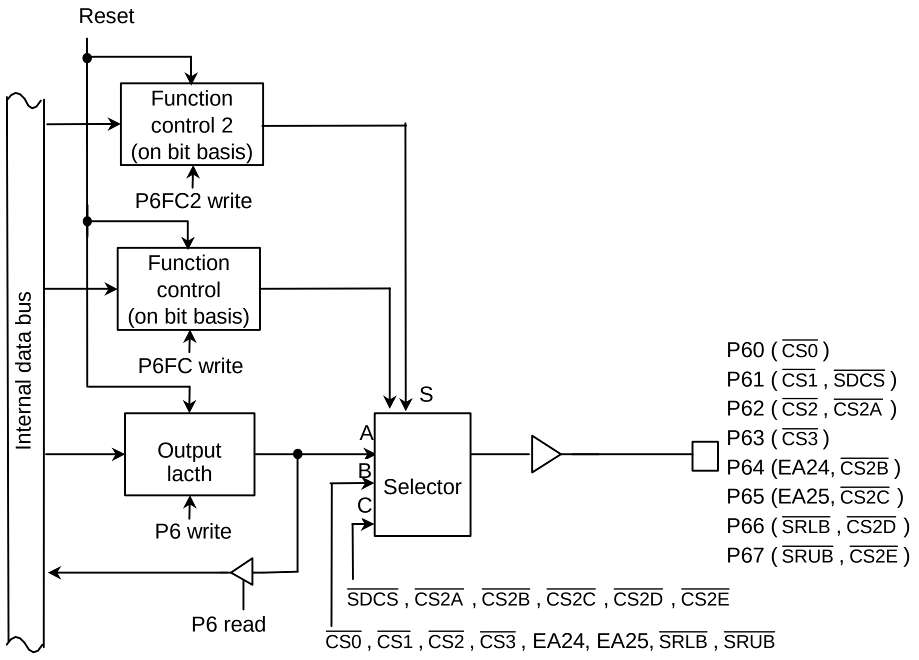

Reset
Function control 2 (on bit basis)
P6FC2 write
Function control (on bit basis)
P6FC write
Internal data bus
Output lacth
P6 write
S
A
B
C
Selector
P6 read
P60 ($\overline{\text{CS0}}$)
P61 ($\overline{\text{CS1}}$, $\overline{\text{SDCS}}$)
P62 ($\overline{\text{CS2}}$, $\overline{\text{CS2A}}$)
P63 ($\overline{\text{CS3}}$)
P64 (EA24, $\overline{\text{CS2B}}$)
P65 (EA25, $\overline{\text{CS2C}}$)
P66 ($\overline{\text{SRLB}}$, $\overline{\text{CS2D}}$)
P67 ($\overline{\text{SRUB}}$, $\overline{\text{CS2E}}$)
$\overline{\text{SDCS}}$, $\overline{\text{CS2A}}$, $\overline{\text{CS2B}}$, $\overline{\text{CS2C}}$, $\overline{\text{CS2D}}$, $\overline{\text{CS2E}}$
$\overline{\text{CS0}}$, $\overline{\text{CS1}}$, $\overline{\text{CS2}}$, $\overline{\text{CS3}}$, EA24, EA25, $\overline{\text{SRLB}}$, $\overline{\text{SRUB}}$

Figure 3.5.17 Port 6

Port 6 Register

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

Port 6 Function Register

		7	6	5	4	3	2	1	0
P6FC (0015H)	Bit symbol	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
	Read/Write	W							
	After reset	0							
	Function	0: Port 1: $\overline{\text{SRUB}}$	0: Port 1: $\overline{\text{SRLB}}$	0: Port 1: EA25	0: Port 1: EA24	0: Port 1: $\overline{\text{CS3}}$	0: Port 1: $\overline{\text{CS2}}$	0: Port 1: $\overline{\text{CS1}}$	0: Port 1: $\overline{\text{CS0}}$

Port 6 Function Register 2

		7	6	5	4	3	2	1	0
P6FC2 (001BH)	Bit symbol	P67F2	P66F2	P65F2	P64F2	–	P62F2	P61F2	–
	Read/Write	W							
	After reset	0							
	Function	0: <P67F> 1: $\overline{\text{CS2E}}$	0: <P66F> 1: $\overline{\text{CS2D}}$	0: <P65F> 1: $\overline{\text{CS2C}}$	0: <P64F> 1: $\overline{\text{CS2B}}$	Always fixed to “0”.	0: <P62F> 1: $\overline{\text{CS2A}}$	0: <P61F> 1: $\overline{\text{SDCS}}$	Always fixed to “0”.

Note: Read-modify-write is prohibited for P6FC and P6FC2.

Figure 3.5.18 Register for Port 6

3.5.9 Port 7 (P70 to P77)

Port 7 is an 8-bit general-purpose I/O port. I/O can be set on bit basis usi
register. Resetting sets port 7 to input port and all b
In addition to functioning as a general-purpose I/O

1. Input/output function for serial bus interface (SCK, SO/SDA, SI/S
2. Input/output function for IrDA (OPTRX0, OPTTX0
3. Extend chip-select output ($\overline{\text{CS1}}$, $\overline{\text{CS2}}$, $\overline{\text{CSEXA}}$)
4. Clock control function for voltage booster of e

Writing 1 in the corresponding bit of P7FC, P7FC2 ena
Resetting resets the P7FC, P7FC2 to 0, and sets all bi

(1) Port 70 (SCK, OPTRX0)

Port 70 is a general-purpose I/O port. It is also used as SCK (clock signal fo
mode) and OPTRX0 (Receive input for IrDA mode of SIO

Used as OPTRX0, it is possible to logical invert by

For port C1, RXD0 or OPTRX0 is used P7FC2<P70F2>.

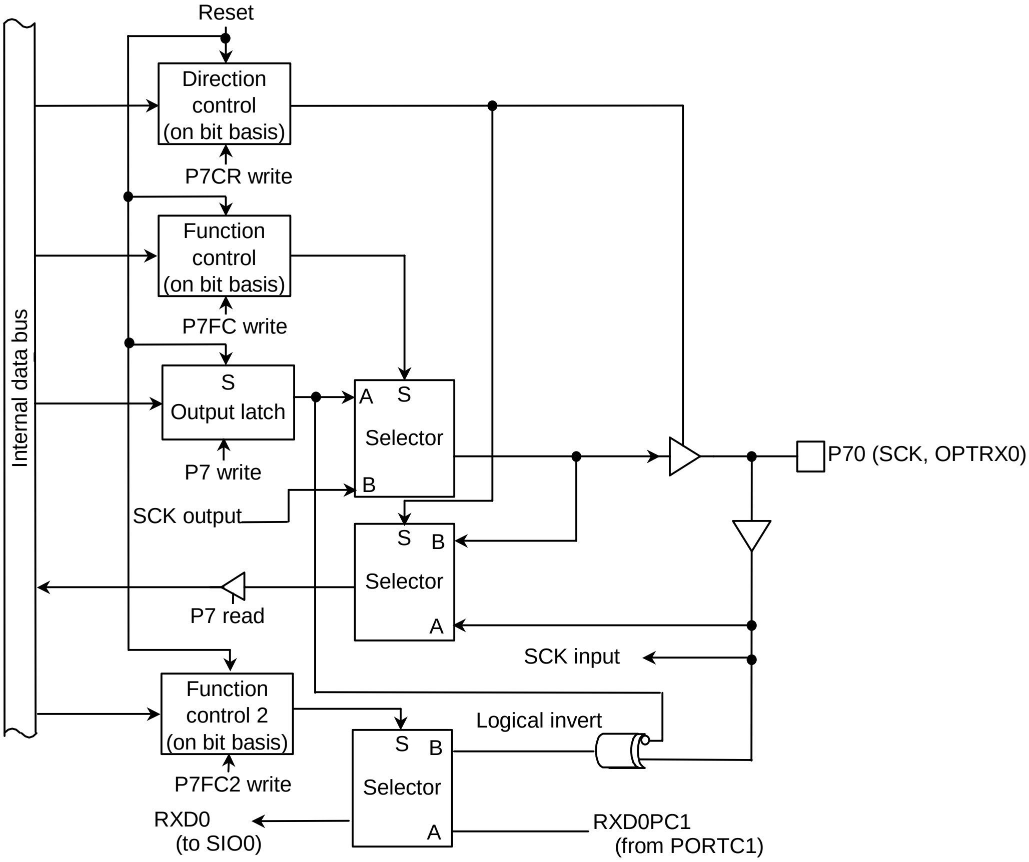


Figure 3.5.19 Port 70

(2) Port 71 (SO/SDA/OPTTX0)

Port 71 is a general-purpose I/O port. It^{2}C bus also us
mode), SO (Data output for SIO mode) for serial bus i
output for IrDA mode of SIO0).

Used as OPTTX0, it is possible to logical invert by

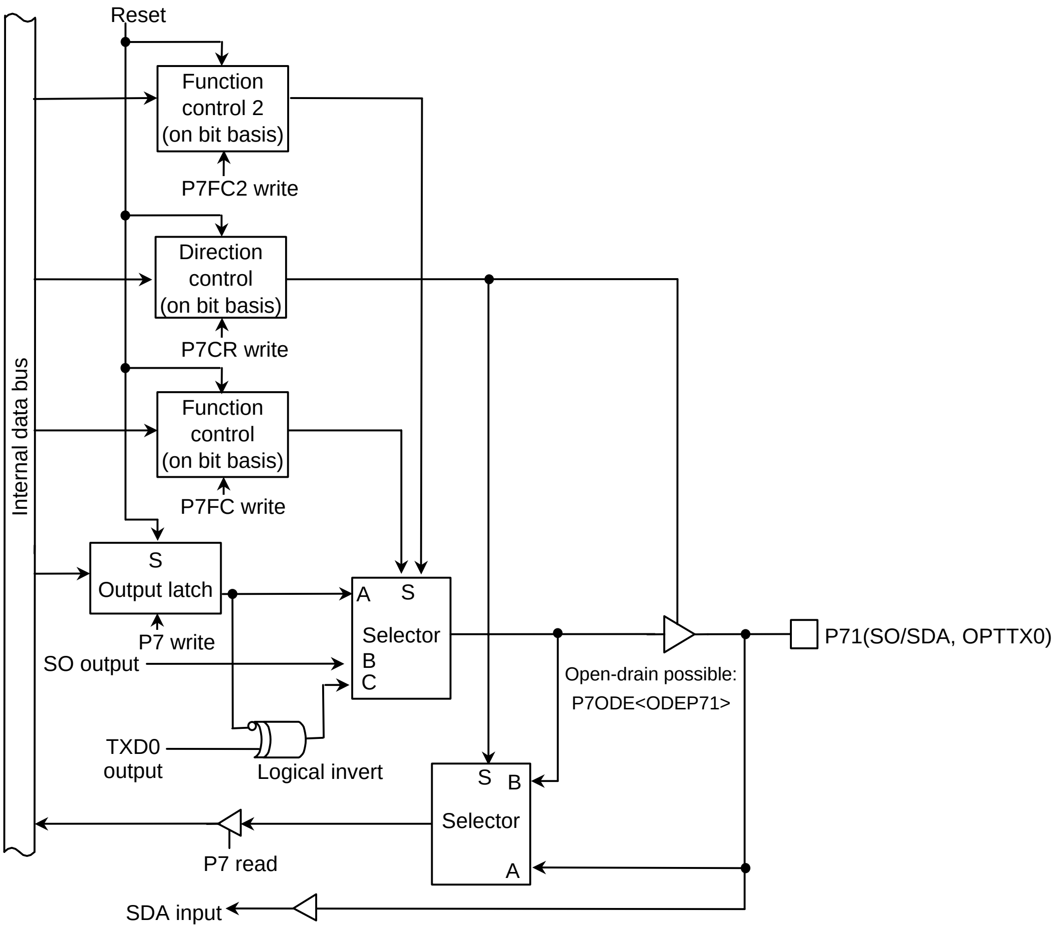


Figure 3.5.20 Port 71

(3) Port 72 (SI/SCL)

Port 72 is a general-purpose I/O port. It is also us
SCL (Clock input/output) for I^2C bus mode of serial bus interface.

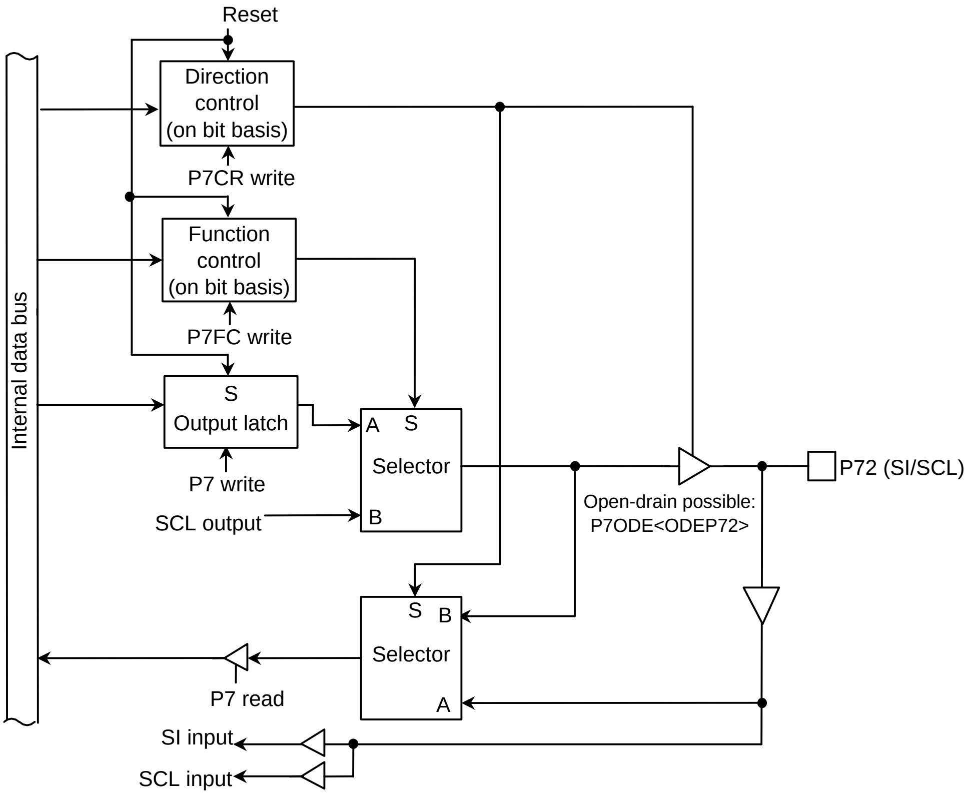


Figure 3.5.21 Port 72

(4) Port 73 ($\overline{CS2F}$), 74 ($\overline{CS2G}$), 75 ($\overline{CSEXA}$)

Port 73 to 75 are general-purpose I/O ports. These
sequential mask ROM and extend chip-select output.

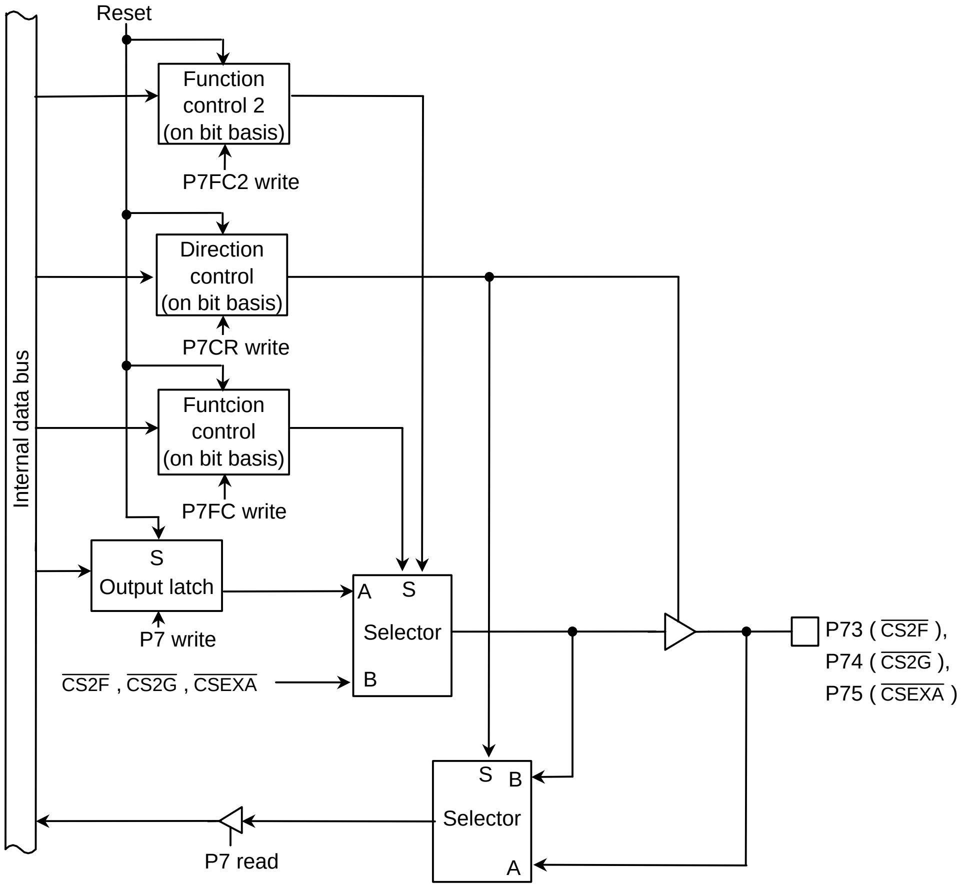

Reset
Function control 2 (on bit basis)
P7FC2 write
Direction control (on bit basis)
P7CR write
Funtcion control (on bit basis)
P7FC write
Internal data bus
S
Output latch
P7 write
$\overline{CS2F}$, $\overline{CS2G}$, $\overline{CSEXA}$
A
S
Selector
B
P73 ($\overline{CS2F}$),
P74 ($\overline{CS2G}$),
P75 ($\overline{CSEXA}$)
S
B
Selector
A
P7 read

Figure 3.5.22 Port 73, 74, 75

(5) Port 76 (MSK), 77 (VEECLK)

Port 76 and 77 are general-purpose I/O ports. These are also used as clo
function for voltage booster of external LCD driver

MSK pin (P76) is an input pin from external LCD driver, clock output f
pin is controlled by state of this pin. It is controlled logic wit

VEECLK pin outputs clock to booster of 32 kHz for 0 level a
request from MSK pin. VEECLK output is controlled w

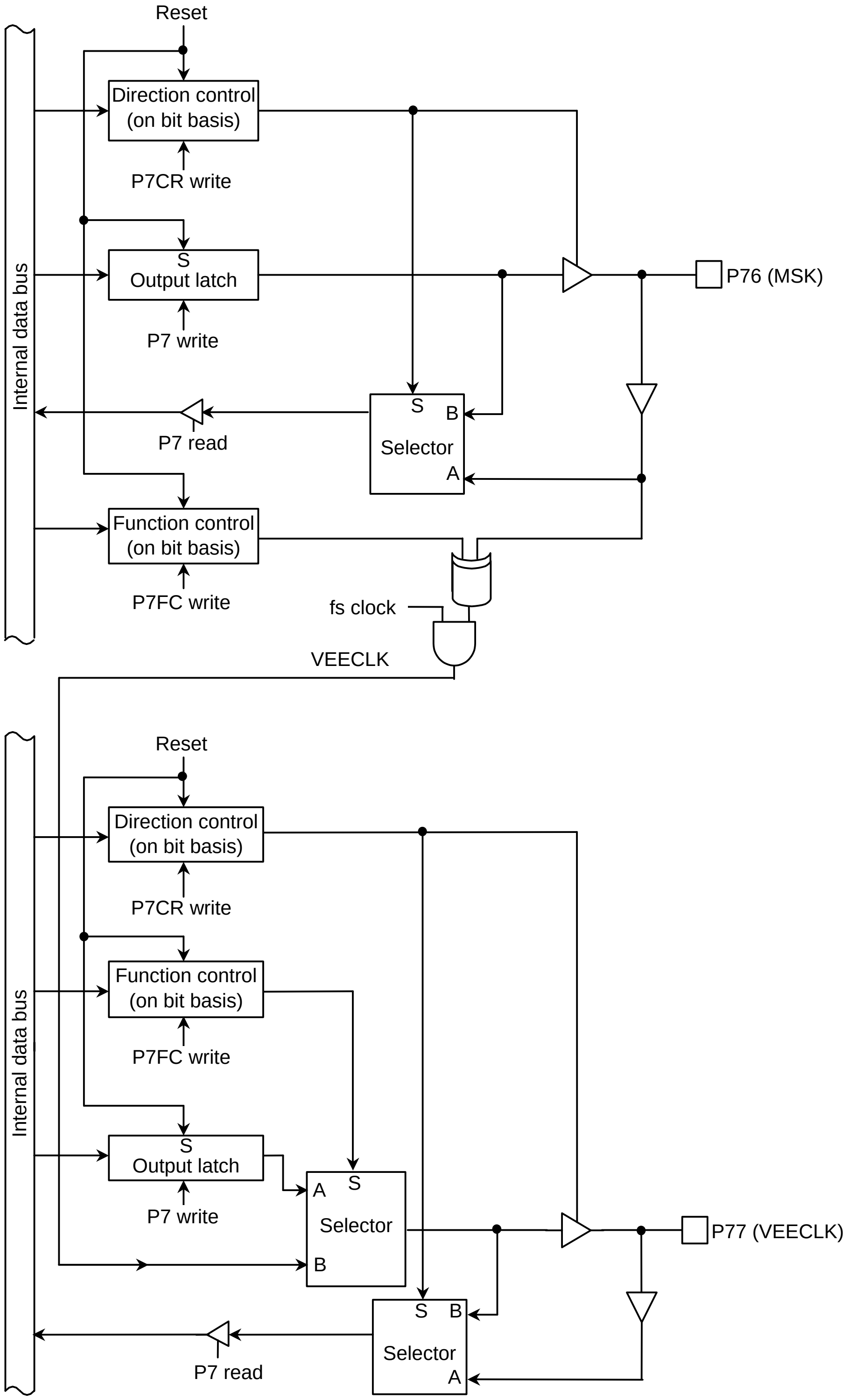


Figure 3.5.23 Port 76, 77

Port 7 Register

		7	6	5	4	3	2	1	0
P7 (0013H)	Bit symbol	P77	P76	P75	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

		7	6	5	4	3	2	1	0
P7CR (0016H)	Bit symbol	P77C	P76C	P75C	P74C	P73C	P72C	P71C	P70C
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Input 1: Output							

Port 7 Function Register

		7	6	5	4	3	2	1	0
P7FC (0017H)	Bit symbol	P77F	P76F	P75F	P74F	P73F	P72F	P71F	P70F
	Read/Write	W							
	After reset	0							
	Function	0:Port 1:VEECLK	MSK select 0: Enable 1: Enable	0: Port	0: Port	0: Port	0: Port 1: SCL/SI	0: Port 1: SDA/SO	0: Port 1: SCK

Port 7 Function Register 2

		7	6	5	4	3	2	1	0
P7FC2 (001CH)	Bit symbol	–	–	P75F2	P74F2	P73F2	–	P71F2	P70F2
	Read/Write	W							
	After reset	0							
	Function	Always fixed to “0”.	Always fixed to “0”.	0: <P75F> 1: $\overline{\text{CSEXA}}$	0: <P74F> 1: $\overline{\text{CS2G}}$	0: <P73F> 1: $\overline{\text{CS2F}}$	Always write “0”.	0: <P71F> 1: OPTTX0	SIO0/RXD0 Pin select 0: RXD0(PC1) 1: OPTRX0 (P70)

Port 7 ODE Register

		7	6	5	4	3	2	1	0
P7ODE (001FH)	Bit symbol	–	–				ODEP72	ODEP71	
	Read/Write	W					W		
	After reset	0	0				0	0	
	Function	Always fixed to “0”.					0: 3 states 1: Open drain		

Note: Read-modify-write is prohibited for P7CR, P7FC, P7FC2 and P7ODE.

Figure 3.5.24 Register for Port 7

3.5.10 Port 8 (P80 to P87)

Port 8 is an 8-bit input port and can also be used as the analog input pins f AD converter. P83 can also be used as ADTRG pin for the AD converter.

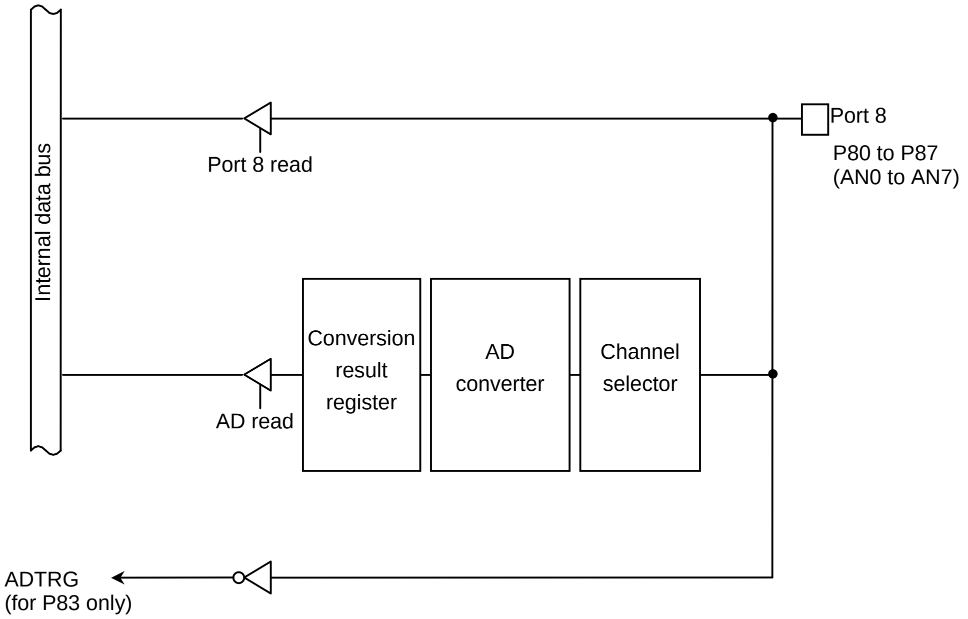


Figure 3.5.25 Port 8

Port 8 Register

		7	6	5	4	3	2	1	0
P8 (0018H)	Bit symbol	P87	P86	P85	P84	P83	P82	P81	P80
	Read/Write	R							
	After reset	Data from external port							

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

Figure 3.5.26 Register for Port 8

3.5.11 Port 9 (P90 to P97)

Port 90 to 97 are 8-bit input ports with pull-up resistor. In addition to general-purpose I/O port, port 90 to 97 can also key interface. The various functions can each be enabled b of the port 9 function register (P9FC).

Resetting resets all bits of the register P9FC to 0 and sets all pins to be inp

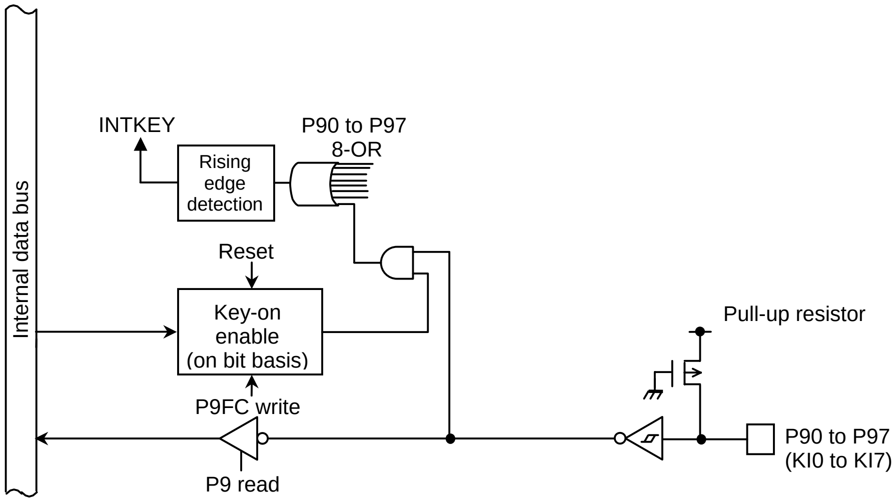


Figure 3.5.27 Port 9

When P9FC=1, if either of input of KI0 to KI7 pins falls generated. INTKEY interrupt can be used to 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

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

Figure 3.5.28 Register for Port 9

3.5.12 Port A (PA0 to PA7)

Port PA0 to PA7 are 8-bit output ports, and also used k
which can set open-drain output buffer.

Writing 1 in the corresponding bit of the port A fun
open-drain output.

Resetting reset bits of the registers PA to 1 and PAFC

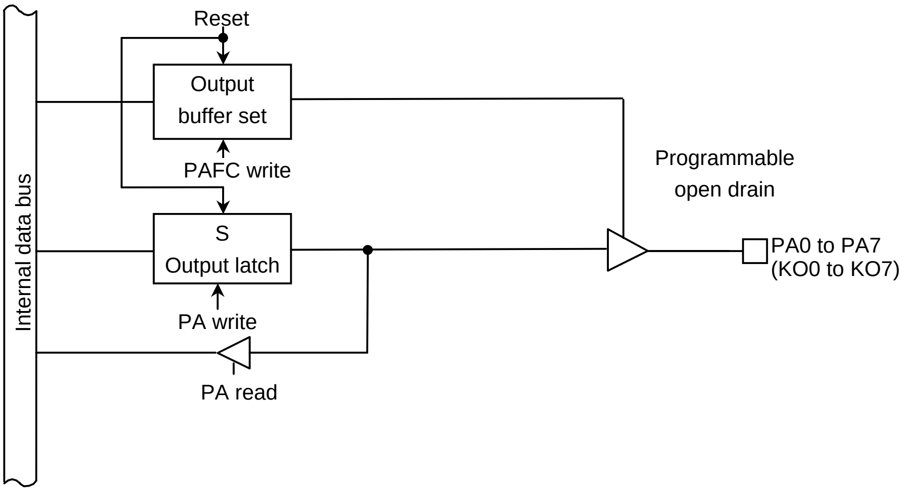


Figure 3.5.29 Port A

Port A Register

		7	6	5	4	3	2	1	0
PA (001EH)	Bit symbol	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
	Read/Write	R/W							
	After reset	1							

Port A Function Register

		7	6	5	4	3	2	1	0
PAFC (0021H)	Bit symbol	PA7F	PA6F	PA5F	PA4F	PA3F	PA2F	PA1F	PA0F
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: CMOS output 1: Open drain							

Note: Read-modify-write is prohibited for PAFC.

Figure 3.5.30 Register for Port A

3.5.13 Port B (PB0 to PB6)

Port B is a 6-bit general-purpose I/O port. Each bit can be individual
output. Resetting sets port B to be an input port.

In addition to functioning as a general-purpose I/O port, port B can also fu
pin for timers (TA0IN, TA1OUT, TA3OUT, TB0OUT), pin for externa
(INT0 to INT3), and I/O for SIO2 (TXD2, RXD2). Above setting
function register PBFC and PBFC2. Edge select of exte
IIMC register, which there is in interruption control

(1) PB0 (TA0IN, TXD2)

As well as functioning as general-purpose I/O port pins, port B0 can also function as se
TXD output pins. In case of use TXD2, it is possible
register PB<PB0>.

And port B0 has a programmable open-drain function which can be c
register PBODE<ODEPB0>.

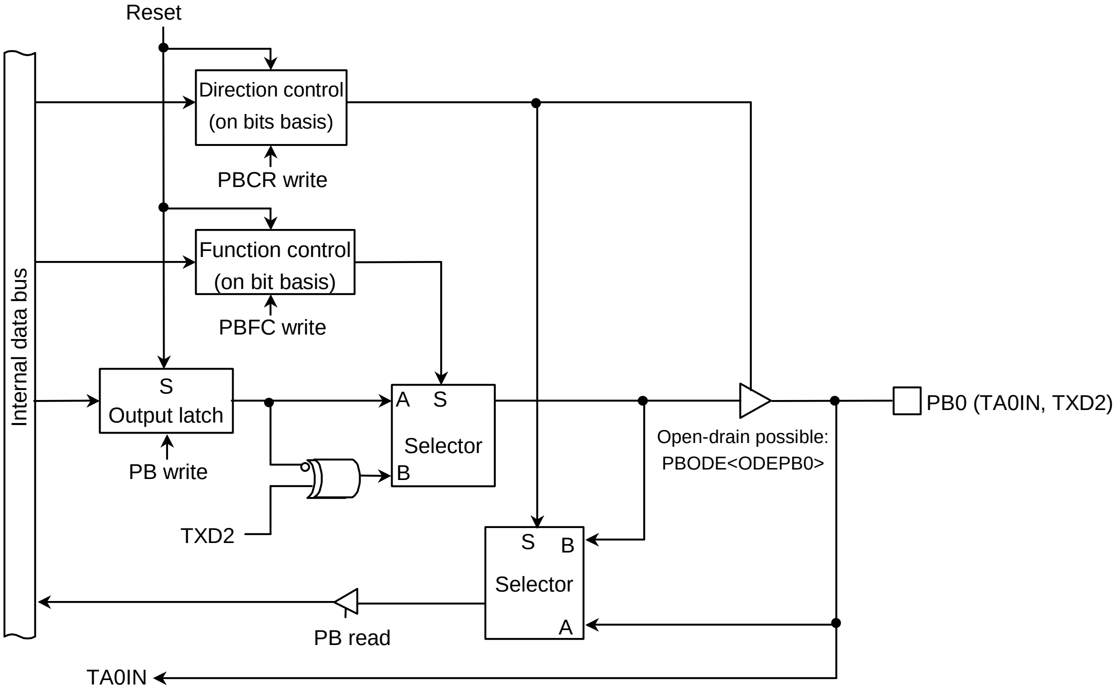


Figure 3.5.31 Port B0

(2) PB1 (TA1OUT, RXD2)

Port B1 is I/O port pin and used as RXD input for the seria
case of use RXD2, it is possible to logical invert by

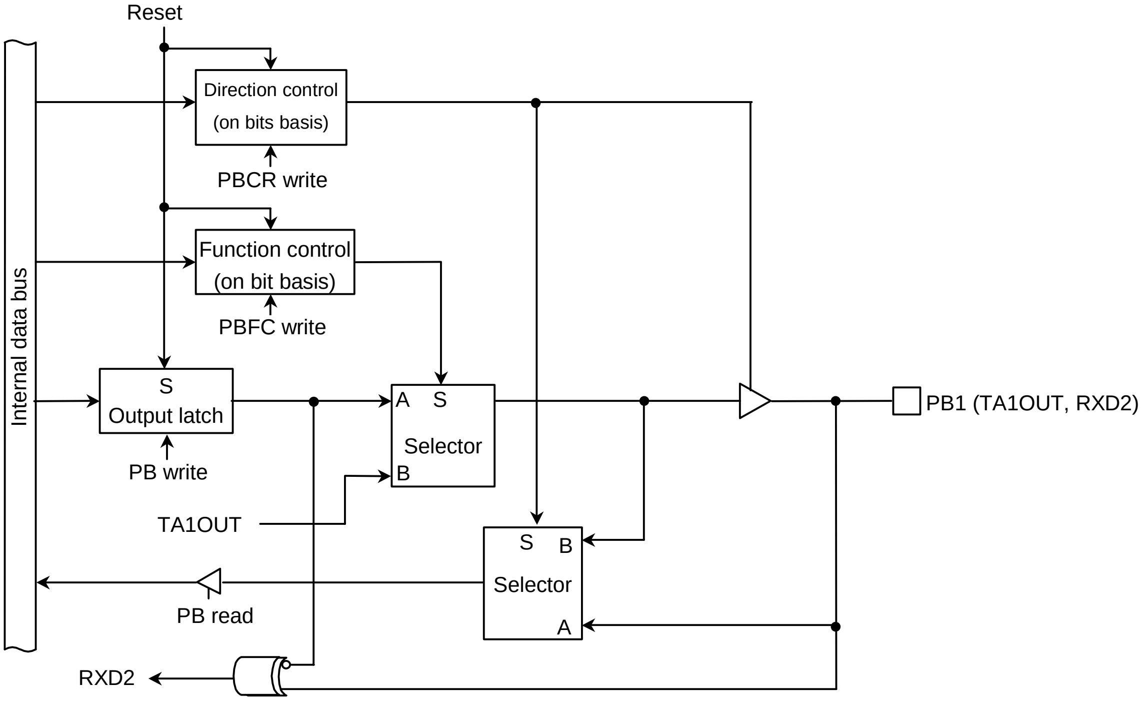


Figure 3.5.32 Port B1

(3) PB3 (INT0)

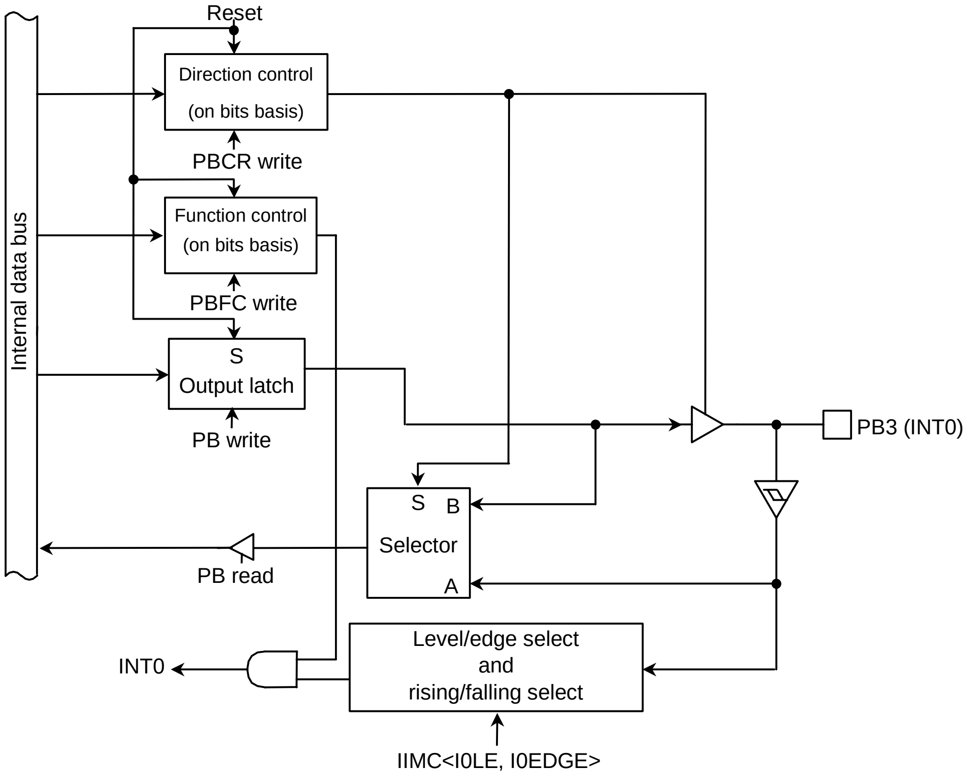


Figure 3.5.33 Port B3

（４） ＰＢ４（ＩＮＴ１），ＰＢ５（ＩＮＴ２，ＴＡ３ＯＵＴ），ＰＢ６（ＩＮＴ３，ＴＢ０ＯＵＴ

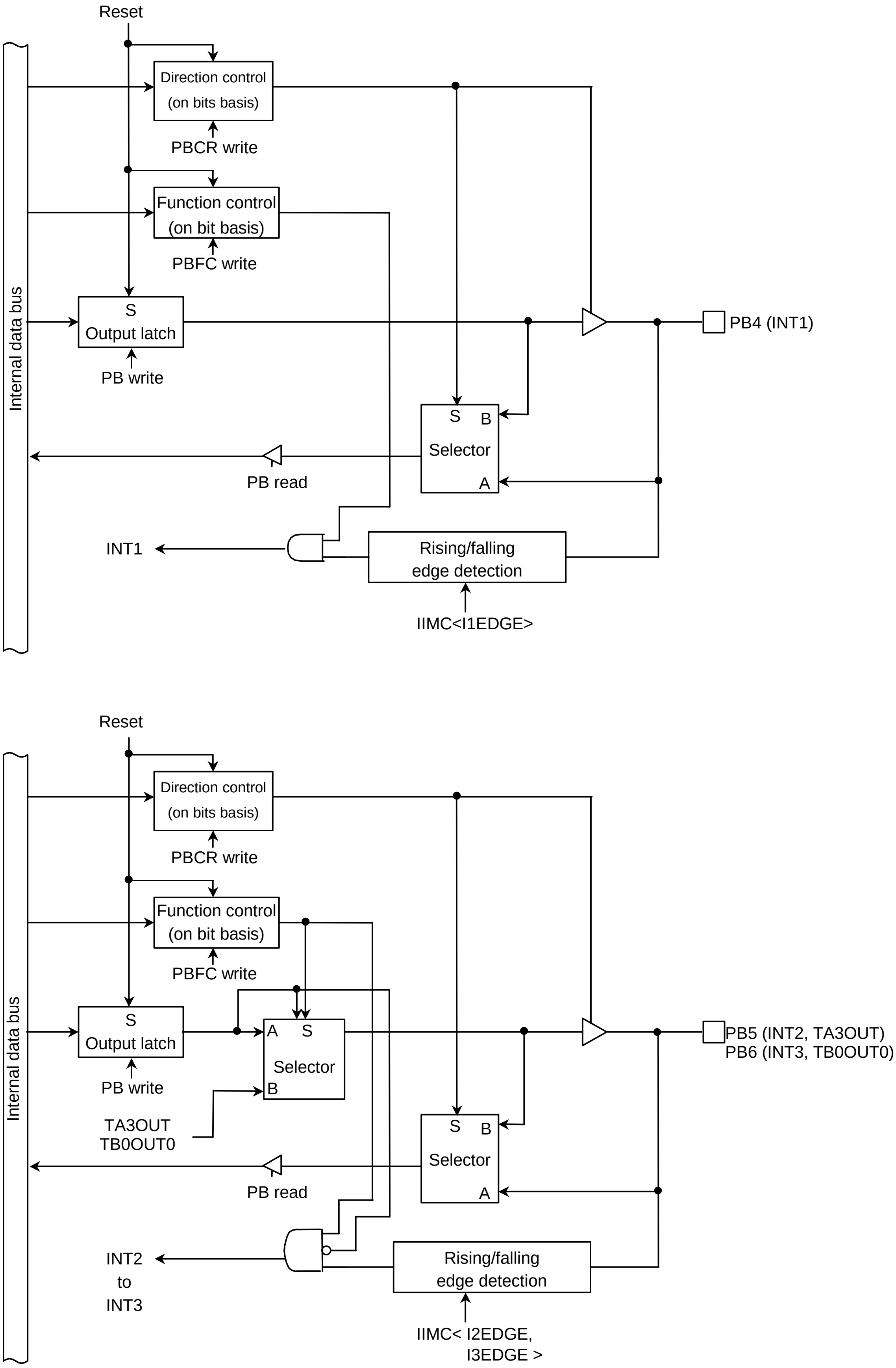


Figure 3.5.34 PB4 to PB6

Port B Register

		7	6	5	4	3	2	1	0
PB (0022H)	Bit symbol		PB6	PB5	PB4	PB3		PB1	PB0
	Read/Write		R/W					R/W	
	After reset		Data from external port (Output latch register is set to 1.)					Data from external port (Output latch register is set to 1.)	
	Function		–	–	–	–		(Note 3)	–

Port B Control Register

		7	6	5	4	3	2	1	0
PBCR (0024H)	Bit symbol		PB6C	PB5C	PB4C	PB3C		PB1C	PB0C
	Read/Write		W					W	
	After reset		0					0	
	Function		0: Input 1: Output					0: Input 1: Output	

Port B Function Register

		7	6	5	4	3	2	1	0
PBFC (0025H)	Bit symbol		PB6F	PB5F	PB4F	PB3F		PB1F	PB0F
	Read/Write		W					W	
	After reset		0					0	
	Function		0: Port 1: INT3 TB0OUT0	0: Port 1: INT2 TA3OUT	0: Port 1: INT1	0: Port 1: INT0		0: Port 1: TA1OUT	0:Port 1:TXD2

INT2, TA3OUT setting

<PB5C> <PB5F>	0	1
0	Input port	Output port
1	INT2	TA3OUT

INT3, TB0OUT0 setting

<PB6C> <PB6F>	0	1
0	Input port	Output port
1	INT3	TB0OUT0

Port B ODE Register

		7	6	5	4	3	2	1	0
PBODE (002BH)	Bit symbol								ODEPB0
	Read/Write								W
	After reset								0
	Function								TXD2 0: CMOS 1: Open drain

Note 1: Read-modify-write is prohibited for the registers PBCR, PBFC and PBODE.

Note 2: PB0/TA0IN pin does not have a register changing PORT/FUNCTION.
For example, when it is used as an input port, the input signal is inputted to 8-bit timer.

Note 3: PB1/RXD1 pin does not have a register changing PORT/FUNCTION.
For example, when it is used as an input port, the input signal is inputted to SIO as the serial receive data.

Figure 3.5.35 Register for Port B

3.5.14 Port C (PC0 to PC5)

Port C0 to C5 are 6-bit general-purpose I/O ports. Each bit can be set in input or output. Resetting sets ports PC0 to PC5 to input ports. It also sets output latch register to 1.

In addition to functioning as general-purpose I/O po as the I/O for serial channels 0 and 1. A pin can be ena corresponding bit of the port C function register (PCFC).

Resetting resets all bits of PCFC and PCCR to 0 and sets all pins ports.

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

As well as functioning as I/O port pins, C0 and C3 pins also functi channel TXD output pins. In case of use TXD0/TXD1, i setting the register PC<PC0, 3>.

And ports C0 to C3 have a programmable open-drain f the register PCODE<ODEPC0, 3>.

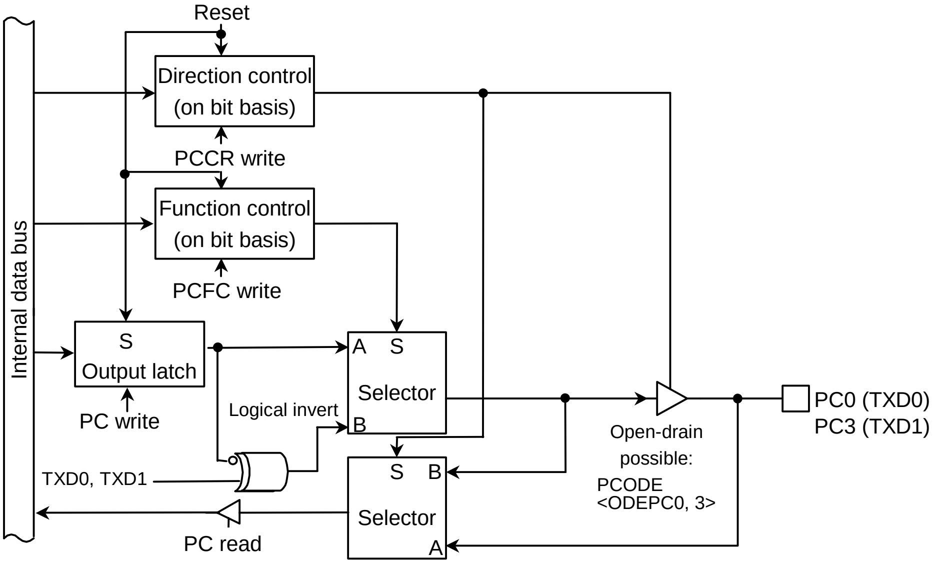


Figure 3.5.36 Port C0 and Port C3

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

Port C1 and C4 are I/O port pins and can also be use channels. In case of use RXD0/RXD1, it is possible register PC<PC1, 4>.

And input data of SIO0 can be selected from RXD0/PC1 or from OPTRX0/P70 b the register PCFC2<P70F2>.

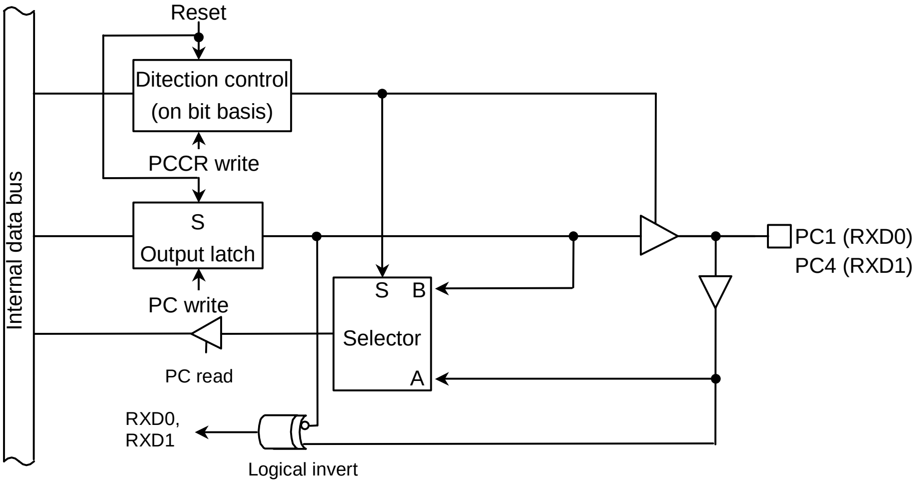


Figure 3.5.37 Port C1 and Port C4

(3) Port C2, C5 (CTS0, SCLK0, CTS1, SCLK1)

Port C2 and C4 are I/O port pins and can also be used as CTS input or SCLK input/output for the serial channels. In case of CTS, SCLK, it is possible to logical invert by setting the register PC<PC2, 5>.

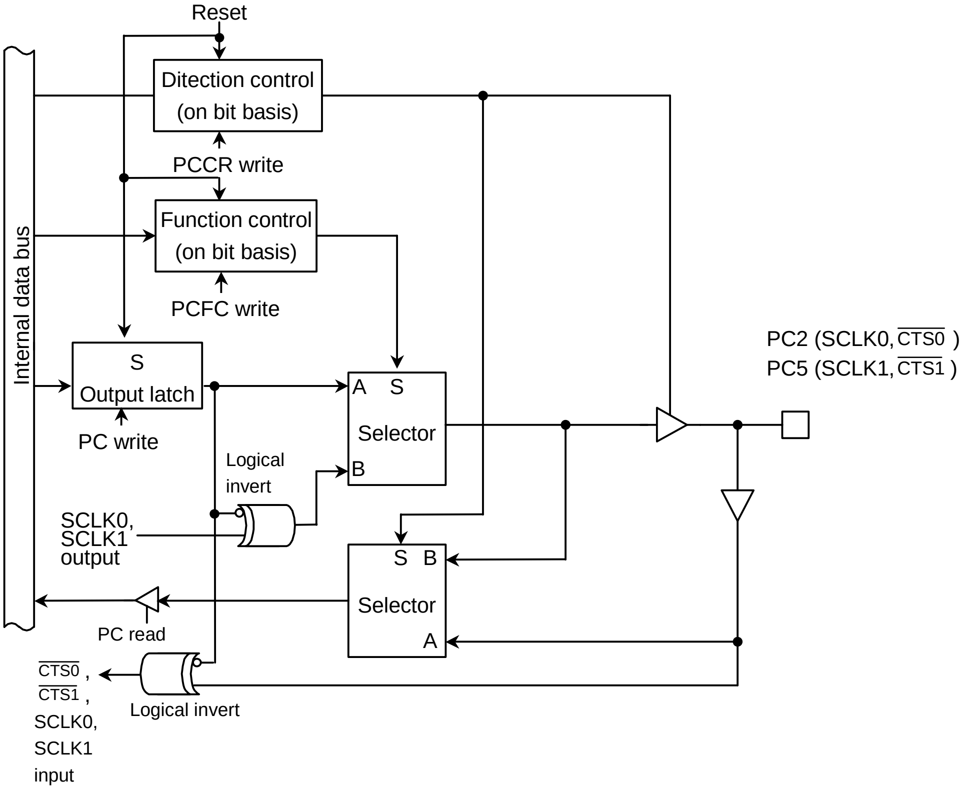


Figure 3.5.38 Port C2 and Port C5

Port C Register

		7	6	5	4	3	2	1	0
PC (0023H)	Bit symbol			PC5	PC4	PC3	PC2	PC1	PC0
	Read/Write			R/W					
	After reset			Data from external port (Output latch register is set to 1.)					

Port C Control Register

		7	6	5	4	3	2	1	0
PCCR (0026H)	Bit symbol			PC5C	PC4C	PC3C	PC2C	PC1C	PC0C
	Read/Write			W					
	After reset			0	0	0	0	0	0
	Function			0: Input 1: Output					

Port C Functon Register

		7	6	5	4	3	2	1	0
PCFC (0027H)	Bit symbol			PC5F		PC3F	PC2F		PC0F
	Read/Write			W		W	W		W
	After reset			0		0	0		0
	Function			0: Port 1: SCLK1 output		0: Port 1: TXD1	0: Port 1: SCLK0 output		0: Port 1: TXD0

Port C ODE Register

		7	6	5	4	3	2	1	0
PCODE (0028H)	Bit symbol					ODEPC3			ODEPC0
	Read/Write					W			W
	After reset					0			0
	Function					TXD1 0: CMOS 1: Open drain			TXD0 0: CMOS 1: Open drain

Note 1: Read-modify-write is prohibited for the registers PCCR, PCFC and PCODE.
Note 2: PC1/RXD0, PC4/RXD1 pins do not have a register changing PORT/FUNCTION. For example, when it is used as an input port, the input signal is inputted to SIO as the serial receive data.

Figure 3.5.39 Register for Port C

3.5.15 Port D (PD0 to PD7)

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

In addition to functioning as output port, port D al controller (DIBSCP, D2BLP, D3BFR, DLEBCD, DOFFB), output pin for ($\overline{\text{ALARM}}$) and output pin for melody/alarm generator ($\overline{\text{MLDALM}}$). Above (M setting is used the function register PDFC.

Only PD6 has two output function $\overline{\text{ALARM}}$ and $\overline{\text{MLDALM}}$. The selectio used PD<PD6>. Resetting resets the function registe output ports.

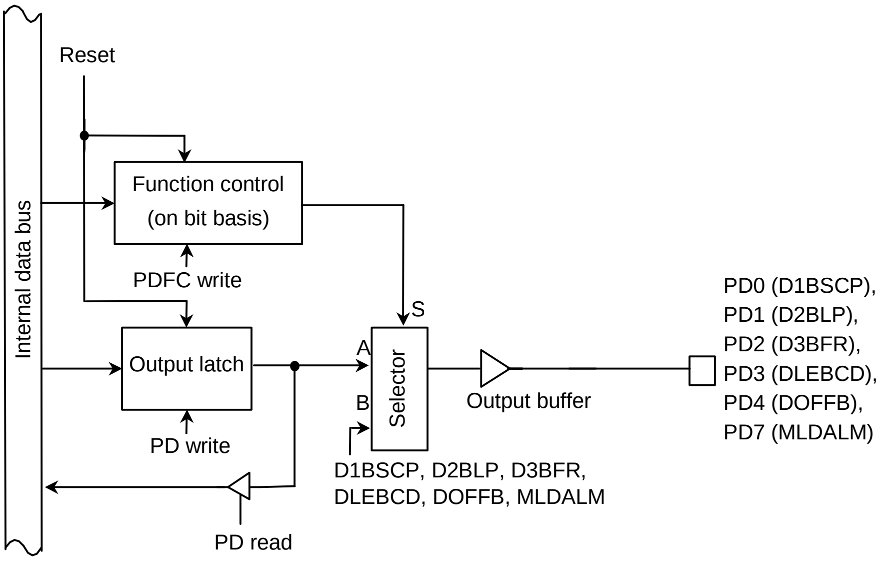


Figure 3.5.40 Port D0 to D4, D7

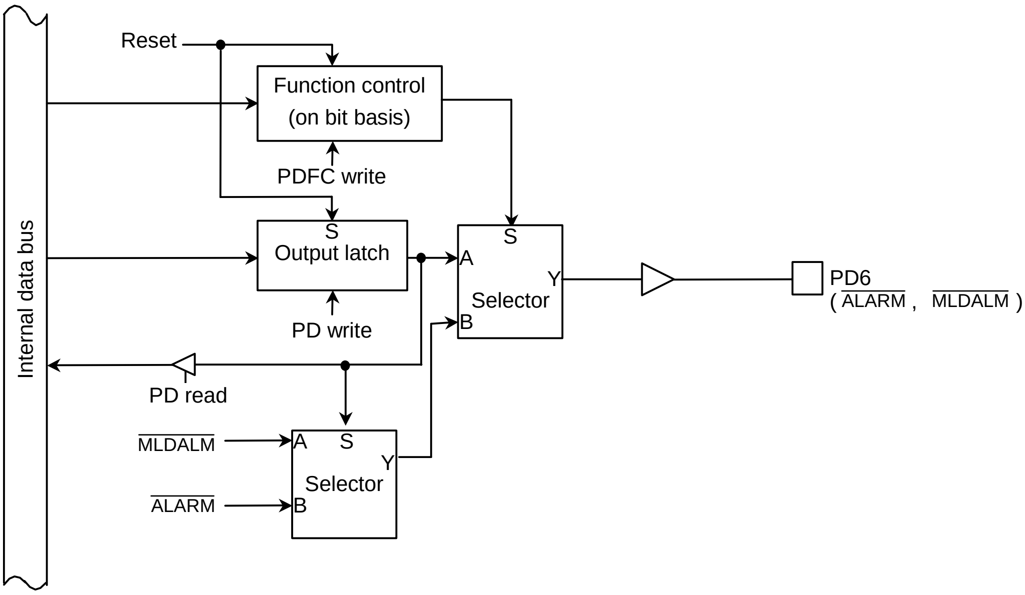


Figure 3.5.41 Port D6

Port D Register

		7	6	5	4	3	2	1	0
PD (0029H)	Bit symbol	PD7	PD6		PD4	PD3	PD2	PD1	PD0
	Read/Write	R/W			R/W				
	After reset	1	1		1	1	1	1	1

Port D Function Register

		7	6	5	4	3	2	1	0
PDFC (002AH)	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

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

Figure 3.5.42 Register for Port D

3.5.16 Port E (PE0 to PE7)

Port E is an 8-bit general-purpose I/O port. Each bit can be set individually for inpu output using the control register PECR. Resetting, th Port E to input port. It also sets all bits of output latch register t

In addition to function as general-purpose I/O port, Port E can also fu data bus for LCD controller (LD0 to LD7). Above sett PEFC.

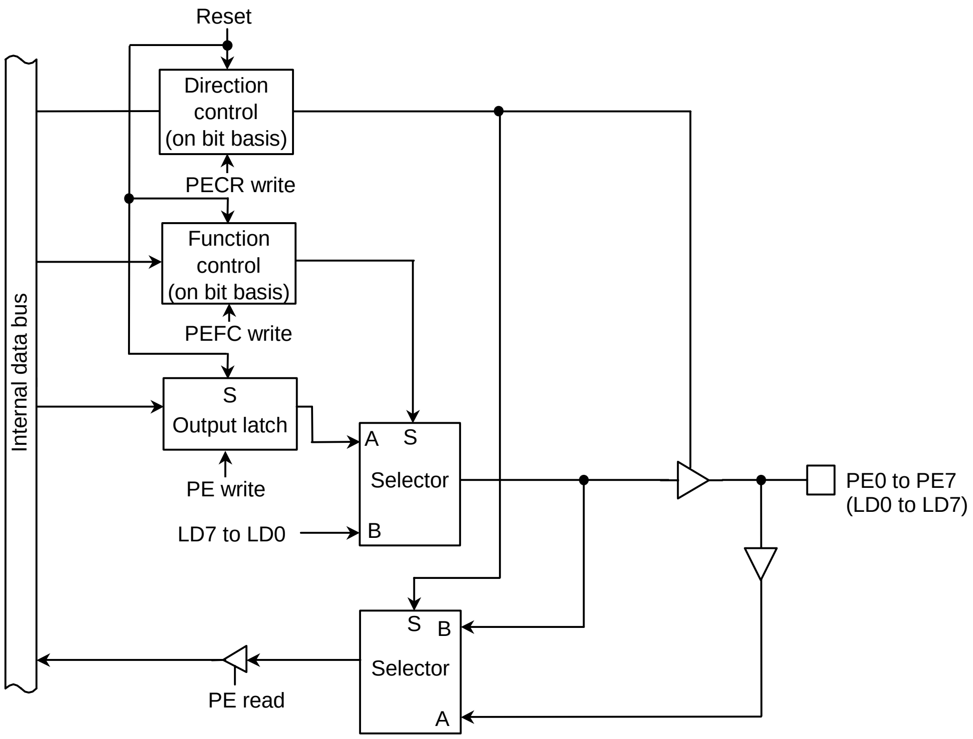


Figure 3.5.43 Port E

Port E Register

		7	6	5	4	3	2	1	0
PE (002CH)	Bit symbol	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
	Read/Write	R/W							
	After reset	Data from external port (Output latch register is set to 1.)							

Port E Control Register

		7	6	5	4	3	2	1	0
PECR (002DH)	Bit symbol	PE7C	PE6C	PE5C	PE4C	PE3C	PE2C	PE1C	PE0C
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Input 1: Output							

Port E Function Register

		7	6	5	4	3	2	1	0
PEFC (002EH)	Bit symbol	PE7F	PE6F	PE5F	PE4F	PE3F	PE2F	PE1F	PE0F
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Port 1: Data bus for LCDC (LD7 to LD0)							

Note: Read-modify-write is prohibited for PECR and PEFC.

Figure 3.5.44 Register for Port E

3.5.17 Port F (PF0 to PF7)

Port F is an 8-bit output port. Resetting sets PF to 1, and P
output 1.

In addition to functioning as output port, port F als
controller (SDCKE, SDLDQM, SDUDQM), and output pin for
(SSCLK). Above setting is used in the function register PFFC.

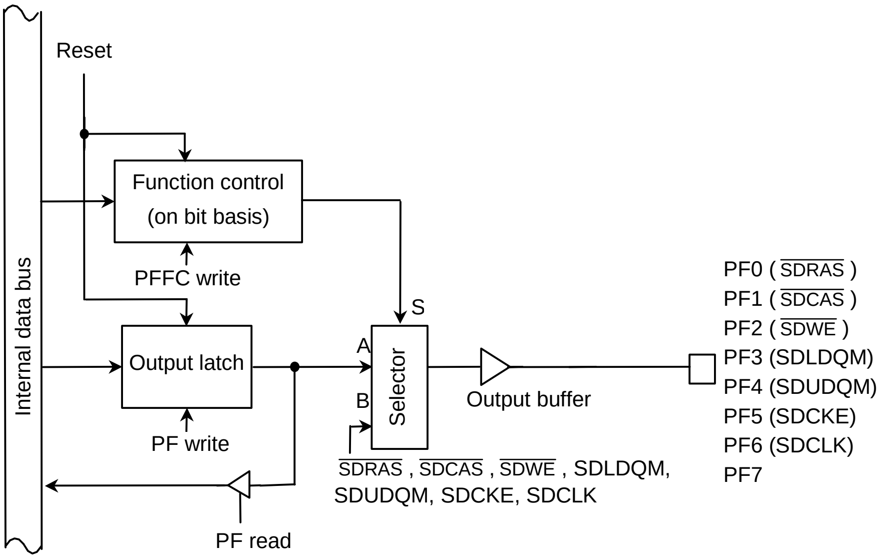

Reset
Internal data bus
Function control
(on bit basis)
PFFC write
Output latch
PF write
S
A
B
Selector
Output buffer
$\overline{\text{SDRAS}}$, $\overline{\text{SDCAS}}$, $\overline{\text{SDWE}}$, SDLDQM,
SDUDQM, SDCKE, SDCLK
PF read
PF0 ($\overline{\text{SDRAS}}$)
PF1 ($\overline{\text{SDCAS}}$)
PF2 ($\overline{\text{SDWE}}$)
PF3 (SDLDQM)
PF4 (SDUDQM)
PF5 (SDCKE)
PF6 (SDCLK)
PF7

Figure 3.5.45 Port F

Port F Register

		7	6	5	4	3	2	1	0
PF (0030H)	Bit symbol	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Port F Function Register

		7	6	5	4	3	2	1	0
PFFC (0032H)	Bit symbol	–	PF6F	PF5F	PF4F	PF3F	PF2F	PF1F	PF0F
	Read/Write	W							
	After reset	0	1	0	0	0	0	0	0
	Function	Always fixed to “0”.	0: Port 1: SDCLK	0: Port 1: SDCKE	0: Port 1: SDUDQM	0: Port 1: SDLDQM	0: Port 1: $\overline{\text{SDWE}}$	0: Port 1: $\overline{\text{SDCAS}}$	0: Port 1: $\overline{\text{SDRAS}}$

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

Figure 3.5.46 Register for Port F

3.6 Chip Select/Wait Controller

On the TMP91C820A, four user-specifiable address areas ($\overline{\text{CS0}}$ to $\overline{\text{CS3}}$) can be set. The d
bus width and the number of waits can be set independentl ($\overline{\text{CS0}}$ to $\overline{\text{CS3}}$
and others).

The pins $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ (which can also function as pins P60 to P63) are the r
output pins for the areas CS0 to CS3. When the CPU specif
the corresponding $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ pin outputs the chip select signal for the specified ad
(in ROM or SRAM). However, in order for the chip select signal to be output, th
register (P6FC) must be set.

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

These pins are CS pin that area and BANK value is fixed withou
CS/WAIT controller.

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

The chip select/wait control registers B0CS to B3CS and
master enable/disable status the data bus width and the

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

3.6.1 Specifying an Address Area

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

At each bus cycle, a compare operation is performed
specified a location in the CS0 to CS3 area. If the result of the comparison i
indicates an access to the corresponding CS area, the $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ pin outputs t
the chip select signal and the bus cycle operates in accordance with the set
select/wait control register B0CS to B3CS. (See 3.6.2

(1) Memory start address registers

Figure 3.6.1 shows the memory start address registers. The memory s registers MSAR0 to MSAR3 set the start addresses for the CS0 to CS3 are upper eight bits (A23 to A16) of the start address in <S23 to S16>. The lower start address (A15 to A0) are permanently set to 0. Accordingly, the star only be set in 64-Kbyte increments, starting from relationship between the start address and the star

Memory Start Address Registers (for areas CS0 to CS3)

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

→ Sets start addresses for areas CS0 to CS3.

Figure 3.6.1 Memory Start Address Register

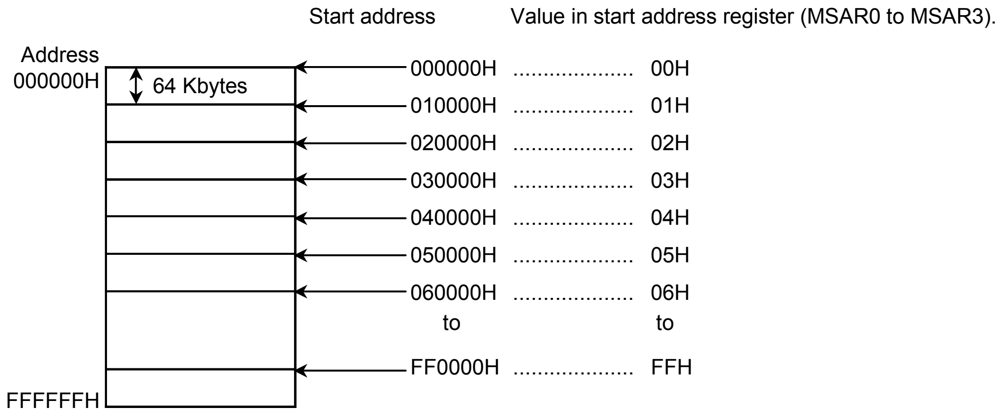

Start address
Value in start address register (MSAR0 to MSAR3).
Address
000000H
64 Kbytes
000000H 00H
010000H 01H
020000H 02H
030000H 03H
040000H 04H
050000H 05H
060000H 06H
to
to
FF0000H .... FFH
FFFFFFH

Figure 3.6.2 Relationship between Start Address and Start Address Register Value

(2) Memory address mask registers

Figure 3.6.3 shows the memory address mask registers. Memory address m registers MAMR0 to MAMR3 are used to set the size of the CS0 to CS3 specifying a mask for each bit of the start address set in memory start addre registers MAMR0 to MAMR3. The compare operation use is in the CS0 to CS3 areas is only performed for bus a set to 0 in these registers. Also, the address bits MAMR3 differ between CS0 to CS3, hence the size that can be different.

Memory Address Mask Register (for CS0 area)

MAMR0 (00C9H)

	7	6	5	4	3	2	1	0
Bit symbol	V20	V19	V18	V17	V16	V15	V14 to V9	V8
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1
Function	Sets size of CS0 area 0: Used for address compare							

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

Memory Address Mask Register (CS1)

MAMR1 (00CBH)

	7	6	5	4	3	2	1	0
Bit symbol	V21	V20	V19	V18	V17	V16	V15 to V9	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)

MAMR2 (00CDH) / MAMR3 (00CFH)

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

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

Figure 3.6.3 Memory Address Mask Registers

(3) Setting memory start addresses and address areas

Figure 3.6.4 shows an example of specifying a 64-Kbyte address area st 010000H using the CS0 areas.

Set 01H in memory start address register MSAR0<S2 upper 8 bits of the start address). Next, calculat address and the anticipated end address (01FFFFH) based on the size of th Bits 20 to 8 of the result correspond to the mask va Setting this value in memory address mask register MSMR0<V20:8> sets t This example sets 07H in MSMR0 to specify a 64K-byte area.

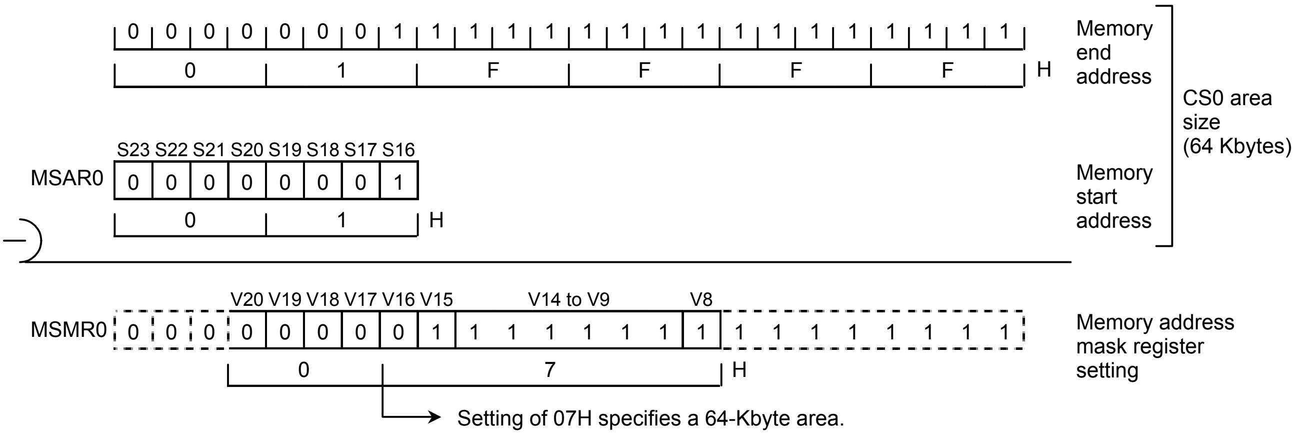


Figure 3.6.4 Example Showing How to Set the CS0 Area

After a reset, MSAR0 to MSAR3 and MAMR0 to MAMR3 are set to B0CS<B0E>, B1CS<B1E> and B3CS<B3E> are reset to 0. T and CS3 areas. However, as B2CS<B2M> is 0 and B2CS<B2E> is set to 1, CS2 is from 000FE0H to 000FFFH and FF0000H to FFFFFFH in TMP91C820A. Also width and number of waits specified in BEXCS are used for accessin outside the specified CS0 to CS3 areas. (See "Control registe

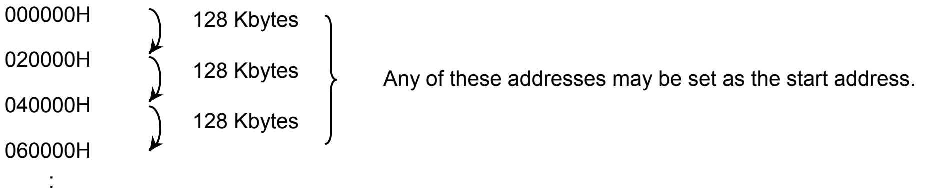
(4) Address area size specification

Table 3.6.1 shows the relationship between CS area and area size. "Δ" indicates areas that cannot be set by memory start address register and address mask regis combinations. When setting an area size using a com start address mask register in the desired steps st

If the CS2 area is set to 16 Mbytes or if two or more area number has the higher priority.

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

a. Valid start addresses



b. Invalid start addresses

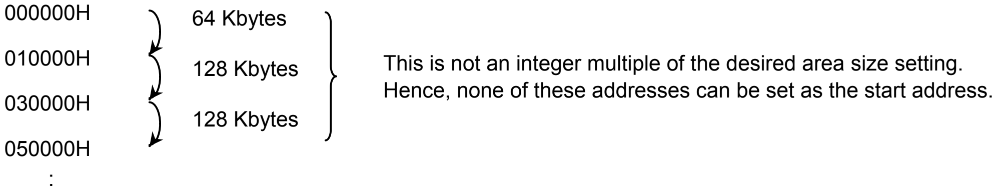

000000H
010000H
030000H
050000H
:
64 Kbytes
128 Kbytes
128 Kbytes
This is not an integer multiple of the desired area size setting.
Hence, none of these addresses can be set as the start address.

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

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

3.6.2 Chip Select/Wait Control Registers

Figure 3.6.5 lists the chip select/wait control reg

The master enable/disable, chip select output wave wait states for each address area (CS0 to CS3 and othe select/wait control registers, B0CS to B3CS and BEXCS

Chip Select/Wait Control Registers

		7	6	5	4	3	2	1	0
B0CS (00C0H)	Bit symbol	B0E		B0OM1	B0OM0	B0BUS	B0W2	B0W1	B0W0
	Read/Write	W		W					
	After reset	0		0	0	0	0	0	0
Read-modify-write instructions are prohibited.	Function	0: Disable 1: Enable		Chip select output waveform selection 00: For ROM/SRAM 01: 10: 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					
	After reset	0		0	0	0	0	0	0
Read-modify-write instructions are prohibited.	Function	0: Disable 1: Enable		Chip select output waveform selection 00: For ROM/SRAM 01: 10: 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							
	After reset	1	0	0	0	0	0	0	0
Read-modify-write instructions are prohibited.	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					
	After reset	0		0	0	0	0	0	0
Read-modify-write instructions are prohibited.	Function	0: Disable 1: Enable		Chip select output waveform selection 00: For ROM/SRAM 01: 10: 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	
BEXCS (00C7H)	Bit symbol					BEXBUS	BEXW2	BEXW1	BEXW0
	Read/Write					0			
	After reset					0	0	0	0
Read-modify-write instructions are prohibited.	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 10 11	Don't care

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

Data bus width selection

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

Figure 3.6.5 Chip Select/Wait Control Registers

(1) Master enable bits

Bit 7 (<B0E>, <B1E>, <B2E> or <B3E>) of a chip sele
master bit, which is used to enable or disable sett
area. Writing 1 to this bit enables the settings.
<B1E> and <B3E>, and enabled (Sets to 1) <B2E>. This

(2) Data bus width selection

Bit 3 (<B0BUS>, <B1BUS>, <B2BUS>, <B3BUS> or <BEXBUS>) of a
select/wait control register is used to specify the width of the data bus. Thi
0 when memory is to be accessed using a 16-bit data b
bus is to be used.

This process of changing the data bus width according to the address b
is known as dynamic bus sizing. For details of this b

Table 3.6.2 Dynamic Bus Sizing

Operand Data Bus Width	Operand Start Address	Memory Data Bus Width	CPU Address	CPU Data		Contorol for READ Cycle							Control for WRITE Cycle						
				D15 to D8	D7 to D0	R/W	$\overline{RD}$	$\overline{WR}$	$\overline{HWR}$	$\overline{SRLB}$	$\overline{SRUB}$	$\overline{SRWR}$	R/W	$\overline{RD}$	$\overline{WR}$	$\overline{HWR}$	$\overline{SRLB}$	$\overline{SRUB}$	$\overline{SRWR}$
8 bits	2n + 0 (Even number)	8 bits	2n + 0	XXXX	b7 to b0	H	L	H	H	L	H	H	L	H	L	H	L	H	L
		16 bits	2n + 0	XXXX	b7 to b0														
	2n + 1 (Odd number)	8 bits	2n + 1	XXXX	b7 to b0					L	H				L	H	L	H	
		16 bits	2n + 1	b7 to b0	XXXX					H	L				H	L	H	L	
16 bits	2n + 0 (Even number)	8 bits	2n + 0 2n + 1	XXXX XXXX	b7 to b0 b15 to b8					L	H				L	H	L	H	
		16 bits	2n + 0	b15 to b8	b7 to b0					L	L				L	L	L	L	
	2n + 1 (Odd number)	8 bits	2n + 1 2n + 2	XXXX XXXX	b7 to b0 b15 to b8					L	H				L	H	L	H	
		16 bits	2n + 1	b7 to b0	XXXX					H	L				H	L	H	L	
			2n + 2	XXXX	b15 to b8					L	H				L	H	L	H	
32 bits	2n + 0 (Even number)	8 bits	2n + 0 2n + 1 2n + 2 2n + 3	XXXX XXXX XXXX XXXX	b7 to b0 b15 to b8 b23 to b16 b31 to b24					L	H				L	H	L	H	
		16 bits	2n + 0 2n + 2	b15 to b8 b31 to b24	b7 to b0 b23 to b16					L	L				L	L	L	L	
	2n + 1 (Odd number)	8 bits	2n + 1 2n + 2 2n + 3 2n + 4	XXXX XXXX XXXX XXXX	b7 to b0 b15 to b8 b23 to b16 b31 to b24					L	H				L	H	L	H	
		16 bits	2n + 1	b7 to b0	XXXX					H	L				H	L	H	L	
			2n + 2	b23 to b16	b15 to b8					L	L				L	L	L	L	
			2n + 4	XXXX	b31 to b24					L	H				L	H	L	H	

xxxx: 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>) o
select/wait control register specify the number of
corresponding memory area is accessed.

The following types of wait operation can be speci
other than those listed in the table should not be ma

Table 3.6.3 Wait Operation Settings

<BxW2:0>	Number of Waits	Wait Operation
000	2 waits	Inserts a wait of 2 states, irrespective of the $\overline{\text{WAIT}}$ pin state.
001	1 wait	Inserts a wait of 1 state, irrespective of the $\overline{\text{WAIT}}$ pin state.
010	(1 + N) waits	Samples the state of the $\overline{\text{WAIT}}$ pin after inserting a wait of one state. If the $\overline{\text{WAIT}}$ pin is low, the waits continue and the bus cycle is extended until the pin goes high.
011	0 waits	Ends the bus cycle without a wait, regardless of the $\overline{\text{WAIT}}$ pin state.
100	Reserved	Invalid setting
101	3 waits	Inserts a wait of 3 states, irrespective of the $\overline{\text{WAIT}}$ pin state.
110	4 waits	Inserts a wait of 4 states, irrespective of the $\overline{\text{WAIT}}$ pin state.
111	8 waits	Inserts a wait of 8 states, irrespective of the $\overline{\text{WAIT}}$ pin state.

A reset sets these bits to 000 (2 waits).

(4) Bus width and wait control for an area other than C

The chip select/wait control register BEXCS width an
waits when memory locations, which are not in one of
areas (CS0 to CS3), are accessed. The BEXCS registe
areas other than CS0 to CS3.

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

Setting B2CS<B2M> (Bit6 of the chip select/wait
designates the 16-Mbyte area (000FE0H to 000FFFH, 003000H to F
CS2 area. Setting B2CS<B2M> to 1 designates the add
address register MSAR2 and the address mask register MAMR2 as CS2
B2CS<B2M> = 1, CS2 is specified in the same manner as CS0

A reset clears this bit to 0, specifying CS2 as 16-M

(6) Procedure for chip selecting/wait control

When using the chip select/wait control function
order:

a. Set the memory start address registers MSAR0 to MSAR3.

Set the start addresses for CS0 to CS3.

b. Set the memory address mask registers MAMR0 to MAMR3.

Set the sizes of CS0 to CS3.

c. Set the chip select/wait control registers B0C

Set the chip select output waveform, data bus w
master enable/disable status for $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$

The CS0 to S3 pins can also function as pins P60 t
signal using one of these pins, set the correspo
register (P6FC) to 1.

If a CS0 to S3 address is specified which is actu
area address, the CPU accesses the internal addre
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
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 e

3.6.3 Connecting External Memory

Figure 3.6.6 shows an example of how to connect external memory to the T
In this example the ROM is connected using a 16-bi
connected using an 8-bit bus.

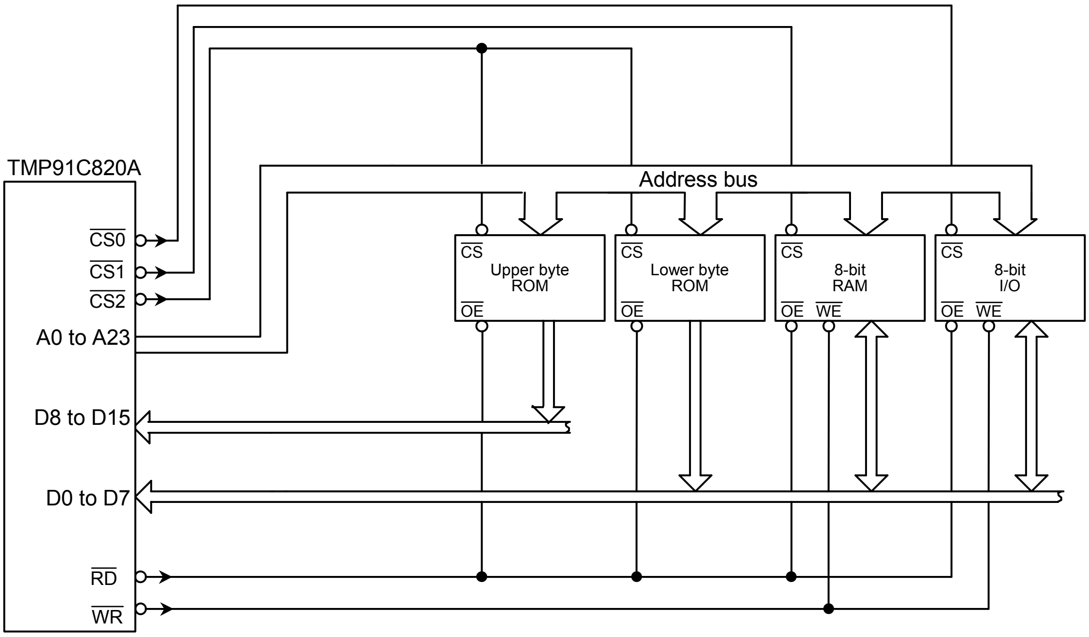


Figure 3.6.6 Example of External Memory Connection
(ROM uses 16-bit bus; RAM and I/O use 8-bit bus.)

A reset clears all bits of the port 6 control register (P6CR) and the port 6 fu
(P6FC) to 0 and disables output of the CS signal. To ou
bit must be set to 1.

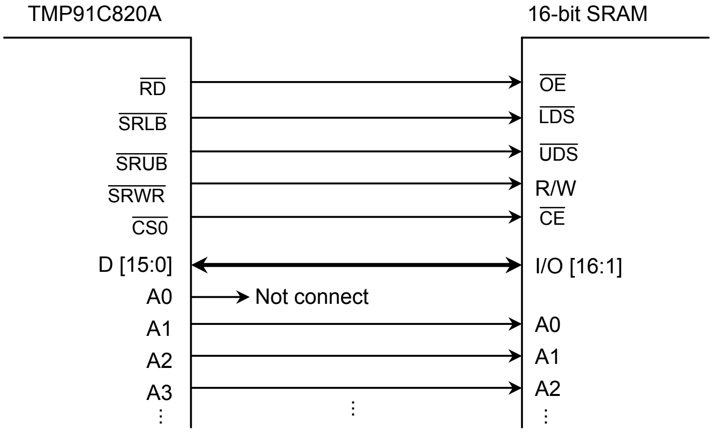


Figure 3.6.7 How to Connect to 16-Bit SRAM for TMP91C820A

3.7 8-Bit Timers (TMRA)

The TMP91C820A features 4 built-in 8-bit timers.

These timers are paired into four modules: TMRA01 and T

two channels and can operate in any of the following four

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

Figure 3.7.1 and 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

In addition, a timer flip-flop and a prescaler are provided for each pair of cha

The operation mode and timer flip-flops are controlled by five control SFRs (S registers).

Each of the two modules (TMRA01 and TMRA23) can be opera operate in the same manner; hence only the operation of T

The contents of this chapter are as follows.

Table 3.7.1 Registers and Pins for Each Module

	Module	TMRA01	TMRA23
External pin	Input pin for external clock	TA0IN (Shared with PB0)	No
	Output pin for timer flip-flop	TA1OUT (Shared with PB1)	TA3OUT (Shared with PB5)
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

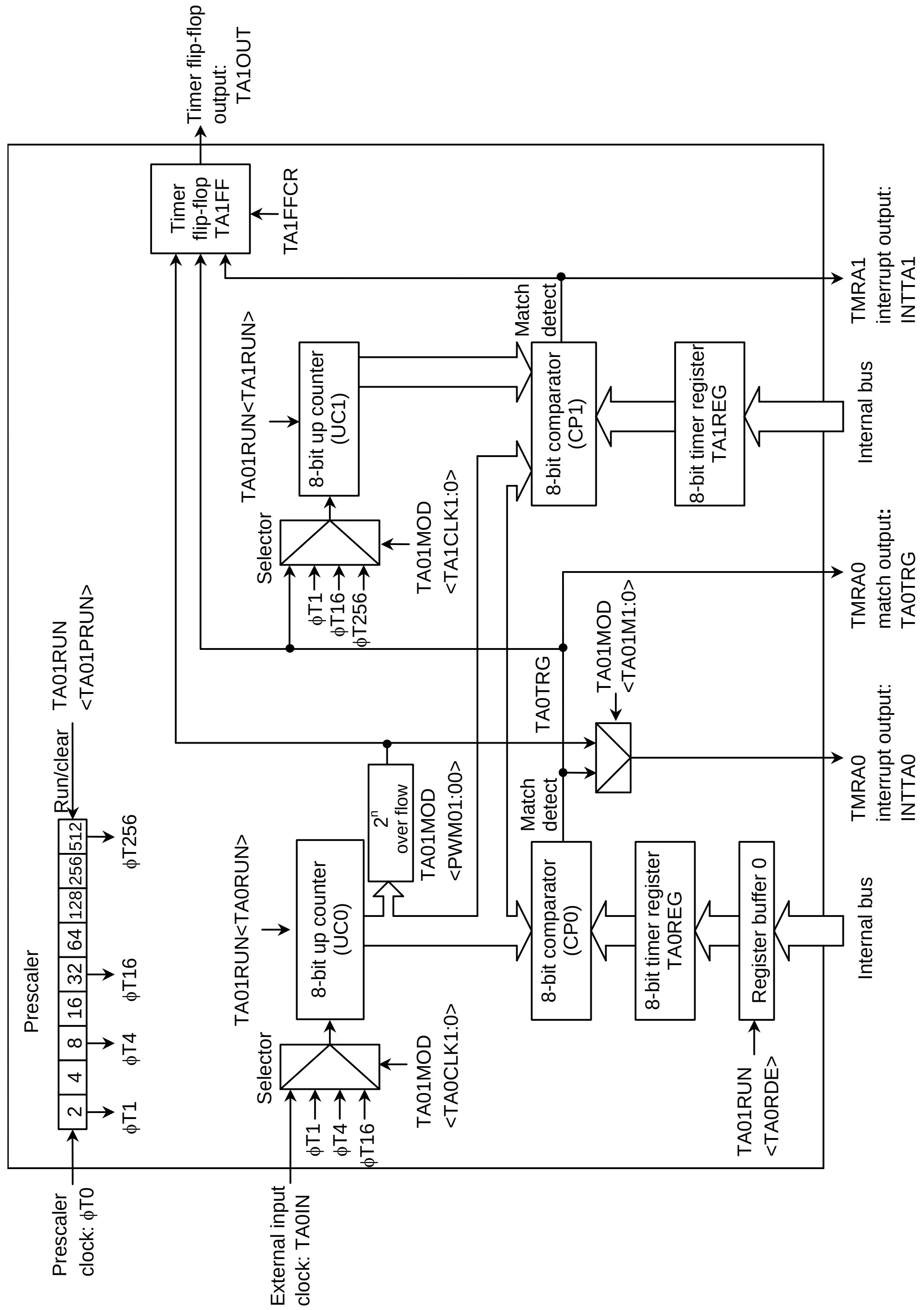


Figure 3.7.1 TMRA01 Block Diagram

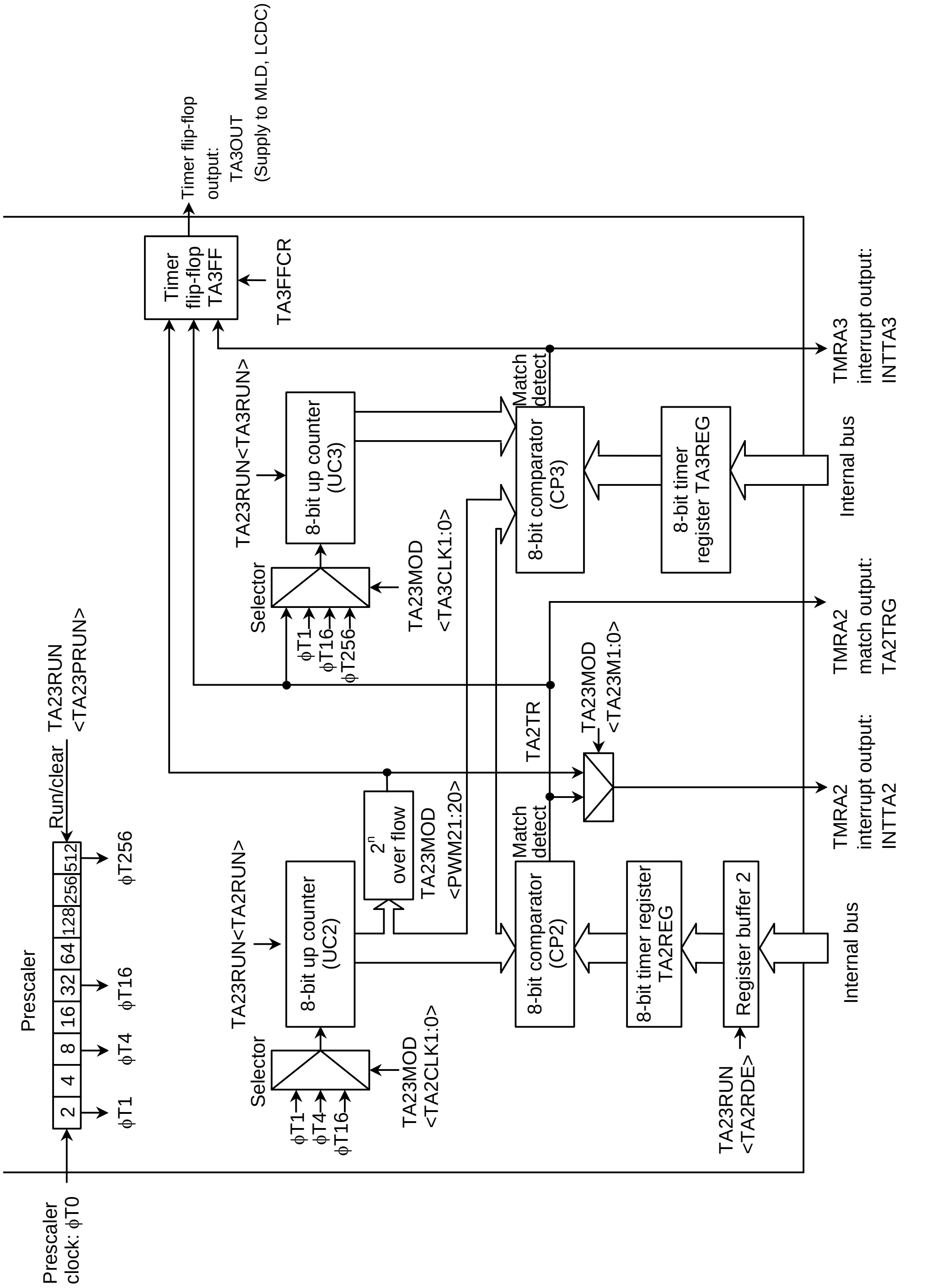


Figure 3.7.2 TMRA23 Block Diagram

3.7.2 Operation of Each Circuit

(1) Prescaler

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

The clock φT0 is divided by 4 and input to this prescaler. φT0 can be either fFPH or fc/16 and is selected using the prescaler clock sel

The prescaler operation can be controlled using TA control register. Setting <TA0PRUN> to 1 starts th clears the prescaler to 0 and stops operation. Tabl output clock resolutions.

Table 3.7.2 Prescaler Output Clock Resolution

at fc = 16 MHz, fs = 32.768 kHz

System Clock Selection <SYSCK>	Prescaler Clock Selection <PRCK1:0>	Gear Value <GEAR2:0>	Prescaler Output Clock Resolution			
			φT1	φT4	φT16	φT256
1 (fs)	00 (f_{FPH})	XXX	fs/2^3(244 μs)	fs/2^5(977 μs)	fs/2^7 (3.9 μs)	fs/2^{11} (62.5 μs)
0 (fc)		000 (fc)	fc/2^3 (0.5 μs)	fc/2^5 (2.0 μs)	fc/2^7 (8.0 μs)	fc/2^{11} (128 μs)
		001 (fc/2)	fc/2^4 (1.0 μs)	fc/2^6 (4.0 μs)	fc/2^8 (16 μs)	fc/2^{12} (256 μs)
		010 (fc/4)	fc/2^5 (2.0 μs)	fc/2^7 (8.0 μs)	fc/2^9 (32 μs)	fc/2^{13} (512 μs)
		011 (fc/8)	fc/2^6 (4.0 μs)	fc/2^8 (16 μs)	fc/2^{10} (64 μs)	fc/2^{14} (1024 μs)
		100 (fc/16)	fc/2^7 (8.0 μs)	fc/2^9 (32 μs)	fc/2^{11} (128 μs)	fc/2^{15} (2048 μs)
	10 (fc/16 CLOCK)	XXX	fc/2^7 (8.0 μs)	fc/2^9 (32 μs)	fc/2^{11} (128 μs)	fc/2^{15} (2048 μs)

xxx: Don't care

(2) Up counters (UC0 and UC1)

These are 8-bit binary counters which count up the specified by TA01MOD.

The input clock for UC0 is selectable and can be either the external cl the TA0IN pin or one of the three internal clocks φT1, φT4, or φT16. The clock sett specified by the value set in TA01MOD<TA01CLK1:0>.

The input clock for UC1 depends on the operation mo overflow output from UC0 is used as the input clock timer mode, the input clock is selectable and can e φT1, φT16, φT256, or the comparator output (the match detection sign TMRA0.

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

(3) Timer registers (TA0REG and TA1REG)

These are 8-bit registers, which can be used to set
set in the timer register TA0REG or TA1REG matches t
up counter, the comparator match detect signal goes active. If the value se
register is 00H, the signal goes active when the up c

The TA0REG are double buffer structure, each of wh
buffer.

The setting of the bit TA01RUN<TA0RDE> determines
buffer structure is enabled or disabled. It is disa = 0 and enabled
<TA0RDE> = 1.

When the double buffer is enabled, data is transfe
timer register when a 2^n overflow occurs in PWM mode, or at the s
in PPG mode. Hence the double buffer cannot be used i

A reset initializes <TA0RDE> to 0, disabling the d
buffer, write data to the timer register, <TA0RDE> to 1, and write th
data to the register buffer. Figure 3.7.3 shows the

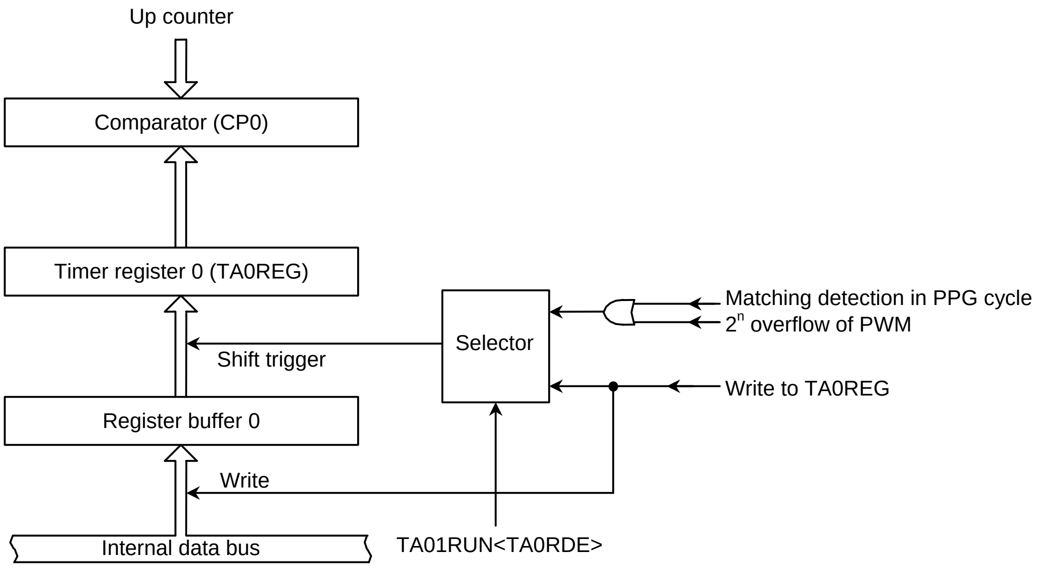


Figure 3.7.3 Configuration of TA0REG

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 re

(4) Comparator (CP0)

The comparator compares the value in an up counter
register. If they match, the up counter is cleared to 0 and an interrupt s
or INTTA1) is generated. If timer flip-flop invers
inverted at the same time.

(5) Timer flip-flop (TA1FF)

The timer flip-flop (TA1FF) is a flip-flop inverte
comparator output) of each interval timer.

Whether inversion is enabled or disabled is dete
TA1FFCR<TA1FFIE> in the timer flip-flops control r
TA1FF to 0. Writing 01 or 10 to TA1FFCR<TA1FFC1:0> sets TA1FF to 0 or 1
00 to these bits inverts the value of TA1FF. (This is

The TA1FF signal is output via the TA1OUT pin. When
output, the timer flip-flop should be set beforehand using the port B func
PBCR, PBFC.

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 to 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 to 6 of TA23RUN are undefined when read.

Figure 3.7.4 Register for TMRA

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 (16-bit timer mode)
01	φT1	
10	φT16	
11	φT256	

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 Register for TMRA

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 (16-bit timer mode)
01	φT1	
10	φT16	
11	φT256	

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 Register for TMRA

TMRA1 Flip-Flop Control Register

TA1FFCR
(0105H)

Read-modify-write instructions are prohibited.

	7	6	5	4	3	2	1	0
Bit symbol					TA1FFC1	TA1FFC0	TA1FFIE	TA1FFIS
Read/Write					R/W		R/W	
After reset					1	1	0	0
Function					00: Invert TA1FF 01: Set TA1FF 10: Clear TA1FF 11: Don’t care		TA1FF control for inversion 0: Disable 1: Enable	TA1FF inversion select 0: TMRA0 1: TMRA1

Inverse signal for timer flip-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 Register for TMRA

TMRA3 Flip-Flop Control Register

TA3FFCR
(010DH)

Read-modify-write instructions are prohibited.

	7	6	5	4	3	2	1	0
Bit symbol					TA3FFC1	TA3FFC0	TA3FFIE	TA3FFIS
Read/Write					R/W		R/W	
After reset					1	1	0	0
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 Register for TMRA

3.7.4 Operation in Each Mode

(1) 8-bit timer mode

Both TMRA0 and TMRA1 can be used independently as 8

a. Generating interrupts at a fixed interval (Usi

To generate interrupts at constant intervals using TMRA1 (INTTA1) TMRA1 then set the operation mode, input clock a TA1REG register respectively. Then, enable the TMRA1 counting.

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

* Clock state

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

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

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 inve
status output via the timer output pin (TA1OUT).

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

* Clock state

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

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

X: Don't care, –: No change

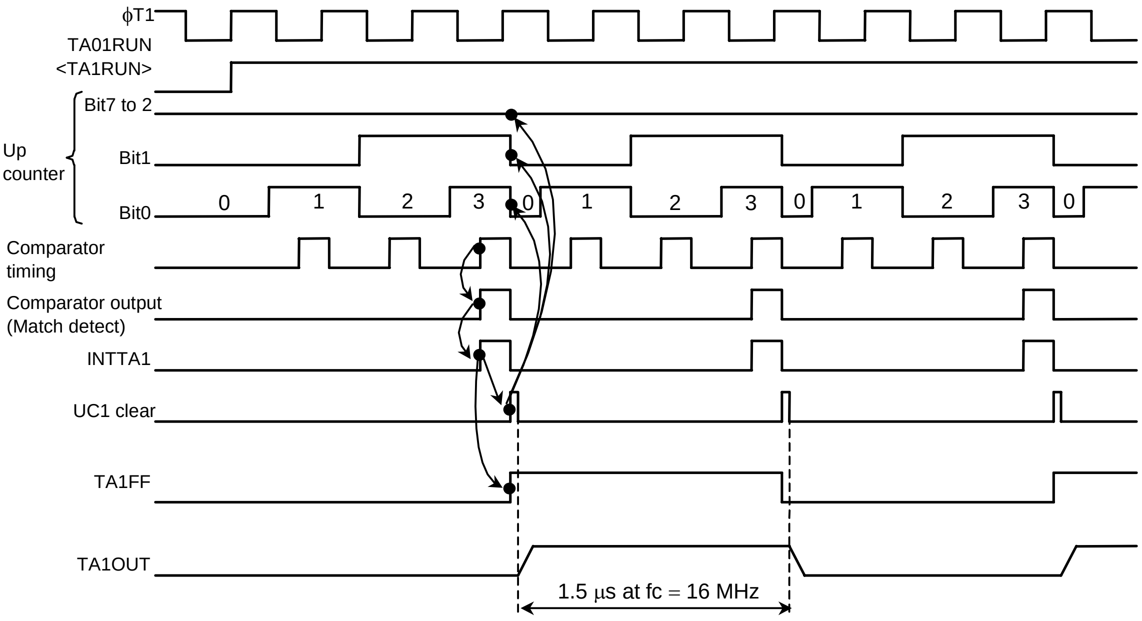


Figure 3.7.9 Square Wave Output Timing Chart (50% duty)

c. Making TMRA1 count up on the match signal from th

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

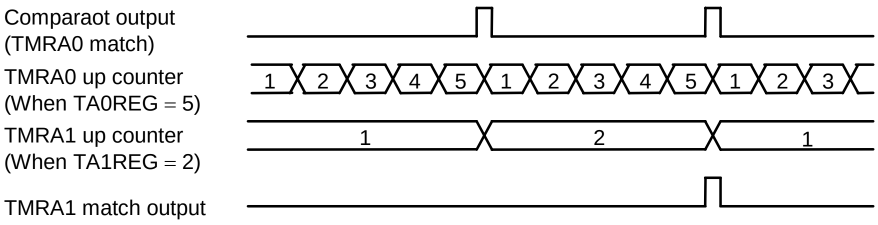


Figure 3.7.10 TMRA1 Count Up on Signal from TMRA0

(2) 16-bit timer mode

Pairing the two 8-bit timers TMRA0 and TMRA1 confi To make a 16-bit interval timer in which TMRA0 and T set TA01MOD<TA01M1:0> to 01.

In 16-bit timer mode, the overflow output from TMRA TMRA1, regardless of the value set in TA01MOD<TA01 the relationship between the timer (Interrupt) cyc

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

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

If φT16 (8.0 μs at 16 MHz) is used as the input clock for co value in the registers: 0.5 s/8.0 μs = 62500 = F424H; e.g. set TA1REG to TA0REG to 24H.

The comparator match signal is output from TMRA0 e matches TA0REG, though the up counter UC0 is not be c

In the case of the TMRA1 comparator, the match det comparator pulse on which the values in the up coun When the match detect signal is output simultaneo TMRA0 and TMRA1, the up counters UC0 and UC1 are clea INTTA1 is generated. Also, if inversion is enable TA1FF is inverted.

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

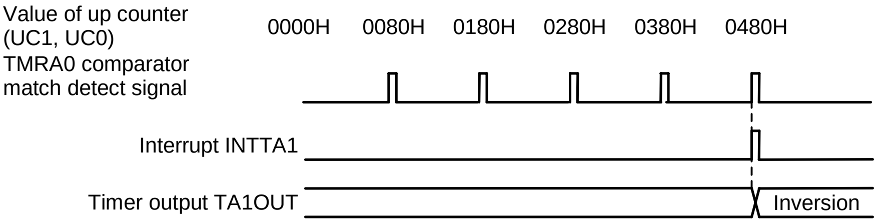


Figure 3.7.11 Timer Output by 16-Bit Timer Mode

(3) 8-bit PPG (Programmable pulse generation) outpu

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

TMRA0 outputs pulses on the TA1OUT pin.

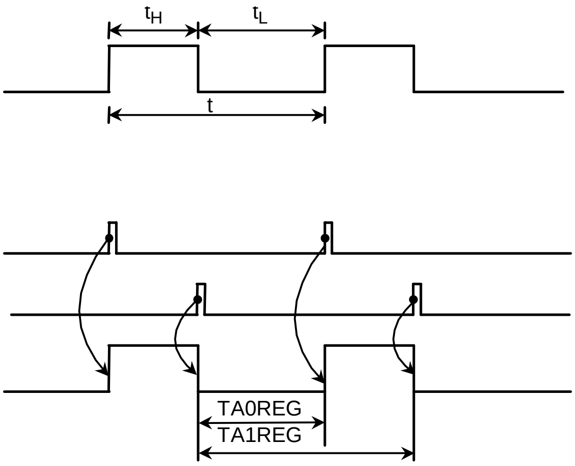


Figure 3.7.12 8-Bit PPG Output Waveforms

In this mode a programmable square wave is genera output each time the 8-bit up counter (UC0) matche registers TA0REG or TA1REG.

The value set in TA0REG must be smaller than the val

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

Figure 3.7.13 shows a block diagram representing t

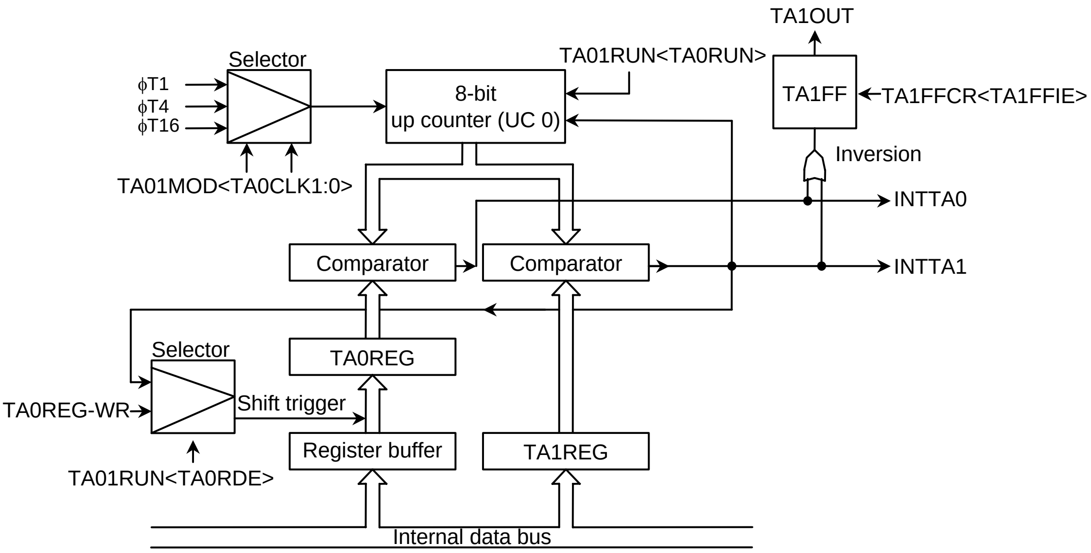


Figure 3.7.13 Block Diagram of 8-Bit PPG Output Mode

If the TA0REG double buffer is enabled in this mode will be shifted into TA0REG each time TA1REG matche

Use of the double buffer facilitates the handling varied).

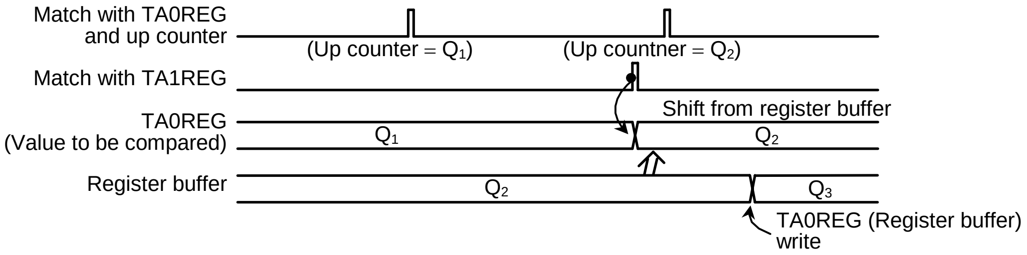
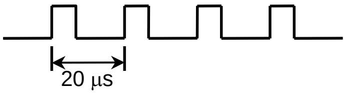


Figure 3.7.14 Operation of Register Buffer

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



* Clock state

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

Calculate the value, which should be set in the time

To obtain a frequency of 50 kHz, the pulse cycle t sho t = 1/50 kHz = 20 μs

φT1 = 0.5 μs (at 16 MHz);

20 μs ÷ 0.5 μs = 40

Therefore set TA1REG to 40 (28H)

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

5 μs ÷ 0.5 μs = 10

Therefore, set TA0REG = 10 = 0AH.

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

When TMRA0 is used the PWM pulse is output on the TA1 be used as an 8-bit timer.

The timer output is inverted when the up counter (U timer register TA0REG match or when 2^n overflow occurs (n = 6, 7 or 8 as specifi TA01MOD<PWM01:00>). The up counter UC0 is cleared when 2^n overflow occurs.

The following conditions must be satisfied before

Value set in TA0REG < Value set for 2^n counter overflow
Value set in TA0REG ≠ 0

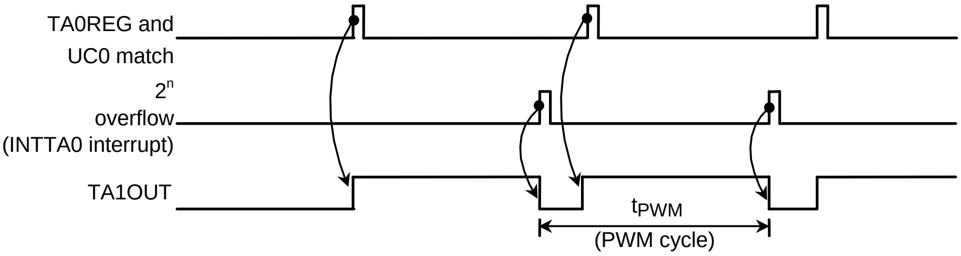


Figure 3.7.15 8-Bit PWM Waveforms

Figure 3.7.16 shows a block diagram representing t

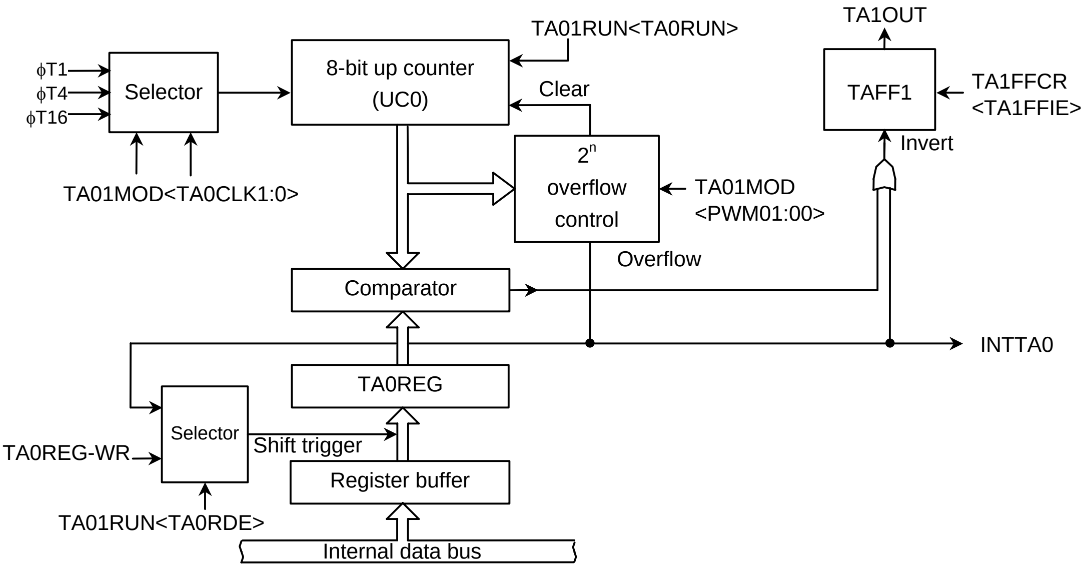


Figure 3.7.16 Block Diagram of 8-Bit PWM Mode

In this mode the value of the register buffer wil
overflow is detected when the TA0REG double buffer

Use of the double buffer facilitates the handling of low duty ratio wav

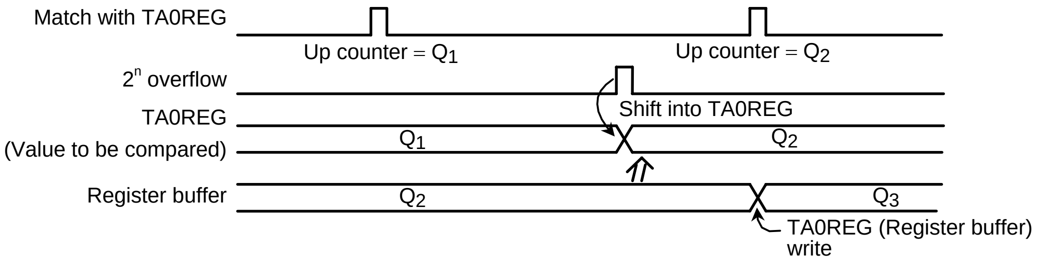
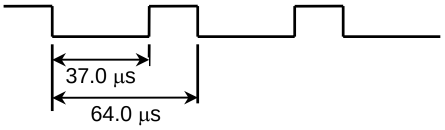


Figure 3.7.17 Register Buffer Operation

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



* Clock state

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

To achieve a 64.0 μs PWM cycle by setting φT1 to 0.5 μs (at fc = 16 MHz):

64 μs ÷ 0.5 μs = 128 = 2^n

Therefore n should be set to 7.

Since the low-level period is 37.0 μs when φT1 = 0.5 μs,

set the following value for TA0REG:

37 μs ÷ 0.5 μs = 74 = 4AH

		MSB							LSB	
		7	6	5	4	3	2	1	0	
TA01RUN	←	–	X	X	X	–	–	–	0	Stop TMRA0 and clear it to 0.
TA01MOD	←	1	1	1	0	–	–	0	1	Select 8-bit PWM mode (cycle: 2^7) and select φT1 as the input clock.
TA0REG	←	0	1	0	0	1	0	1	0	Write 4AH.
TA1FFCR	←	X	X	X	X	1	0	1	X	Clear TA1FF to 0; enable the inversion and double buffer.
PBCR	←	X	–	–	–	–	X	1	–	Set PB1 and the TA1OUT pin.
PBFC	←	X	–	–	–	–	X	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 = 16 MHz, fs = 32.768 kHz

Select System Clock <SYSCK>	Select Prescaler Clock <PRCK1:0>	Gear Value <GEAR2:0>	PWM cycle								
			2^6			2^7			2^8		
			ϕT1	ϕT4	ϕT16	ϕT1	ϕT4	ϕT16	ϕT1	ϕT4	ϕT16
1 (fs)	00 (fFPH)	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)	32.0 μs	128 μs	512 μs	64.0 μs	256 μs	1024 μs	128 μs	512 μs	2048 μs
		001 (fc/2)	64.0 μs	256 μs	1024 μs	128 μs	512 μs	2048 μs	256 μs	1024 μs	4096 μs
		010 (fc/4)	128 μs	512 μs	2048 μs	256 μs	1024 μs	4096 μs	512 μs	2048 μs	8192 μs
		011 (fc/8)	256 μs	1024 μs	4096 μs	512 μs	2048 μs	8192 μs	1024 μs	4096 μs	16.384 ms
		100 (fc/16)	512 μs	2048 μs	8192 μs	1024 μs	4096 μs	16.384 ms	2048 μs	8192 μs	32.768 ms
	10 (fc/16 clock)	XXX	512 μs	2048 μs	8192 μs	1024 μs	4096 μs	16.384 ms	2048 μs	8192 μs	32.768 ms

XXX: Don't care

(5) Settings for each mode

Table 3.7.4 shows he SFR settings for each mode.

Table 3.7.4 Timer Mode Setting Registers

Register Name	TA01MOD				TA1FFCR
<Bit Symbol>	<TA01M1:0>	<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) LCDC and MELODY/ALARM circuit supply mode

This function can operate only TMRA3. It can use
source clock TA3 clock generated by TMRA3. And keep

OPERATE

1. Clock genelate by timer 3
2. Clock supply start (<TA3LCDE>=1) or (<TA3MLDE>=1)
3. Need setup time
4. LCDC or MELODY/ALARM start to operate

STOP

1. LCDC or MELODY/ALARM stop to operate
2. Clock supply cut off (<TA3LCDE>=0) or (<TA3MLDE>=0)

EMCCR0
(00E3H)

	7	6	5	4	3	2	1	0
Bit symbol	PROTECT	TA3LCDE	AHOLD	TA3MLDE	HRESENA	EXTIN	DRVOSCH	DRVOSCL
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	1	1
Function	Protect flag 0: OFF 1: ON	LCDC source clock 0: 32 kHz 1: TA3OUT	Address hold 0: Normal 1: Enable	Melody/alarm source clock 0: 32 kHz 1: TA3OUT	HRESET 0: Disable 1: Enable	1: External clock	fc oscillator driver ability 1: Normal 0: Weak	fs oscillator driver ability 1: Normal 0: Weak

3.8 External Memory Extension Function (MMU)

This is MMU function which can expand program/data are
area.

Address pins to external memory are 2 extended address pins (EA24, E
extended chip select pins ($\overline{CS2A}$ to $\overline{CS2G}$ and $\overline{CSEXA}$) in addition to 24 address b
A23) which are common specification of TLCS900 family ($\overline{CS0}$ to $\overline{CS3}$)
output from CS/WAIT controller.

The feature and the recommendation setting method of t
In addition, "AH" in the table is the value of address bus 23 to 16 disp

Purpose	Item	For Many Kinds Class Extended Memory
Program ROM	Maximum memory size	2 Mbytes: COMMON2 + 14 Mbytes: BANK (16 Mbytes × 1 pcs)
	Used local area, BANK number	LOCAL2 (AH = C0 − DF: 2 Mbytes × 7 BANK)
	Setting CS/WAIT	Setup AH = 80 − FF to CS2
	Used $\overline{CS}$ pin	$\overline{CS2A}$
Data ROM	Maximum memory size	96 Mbytes (16 Mbytes × 6 pcs)
	Used local area, BANK number	LOCAL3 (AH = 80 − BF: 4 Mbytes × 24 BANK)
	Setting CS/WAIT	Setup AH = 80 − FF to CS2
	Used $\overline{CS}$ pins	$\overline{CS2B}$, $\overline{CS2C}$, $\overline{CS2D}$, $\overline{CS2E}$, $\overline{CS2F}$, $\overline{CS2G}$
Data SDRAM*	Maximum memory size	2 Mbytes: COMMON1 + 14 Mbytes: BANK (16 Mbytes × 1 pcs)
	Used local area, BANK number	LOCAL1 (AH = 40 − 5F: 2 Mbytes × 7 BANK)
	Setting CS/WAIT	Setup AH = 40 − 7F to CS1
	Used $\overline{CS}$ pin	$\overline{CS1}$
Data RAM	Maximum memory size	1 Mbyte: COMMON0 + 7 Mbytes: BANK (8 Mbytes × 1pcs)
	Used local area, BANK number	LOCAL0 (AH = 10 − 1F: 1 Mbyte × 7 BANK)
	Setting CS/WAIT	Setup AH = 00 − 1F to CS3
	Used $\overline{CS}$ pin	$\overline{CS3}$
Extended memory 1	Maximum memory size	1 Mbyte (1 Mbyte × 1 pcs)
	Used local area, BANK number	None
	Setting CS/WAIT	Setup AH = 20 − 2F to CS0
	Used $\overline{CS}$ pin	$\overline{CS0}$
Extended memory 2	Maximum memory size	256 Kbytes (256 Kbytes × 1 pcs)
	Used local area, BANK number	None
	Setting CS/WAIT	Setup AH = 30 − 3F to $\overline{CSEX}$
	Used $\overline{CS}$ pin	$\overline{CSEXA}$
Extended memory 3 (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	Setup AH = 30 − 3F to $\overline{CSEX}$
	Used $\overline{CS}$ pin	D1BSCP, D2BLP, D3BFR, DLEBCD
Extended memory 4	Maximum memory size	512 Kbytes
	Used local area, BANK number	None
	Setting CS/WAIT	Setup AH = 30 − 3F to $\overline{CSEX}$
	Used $\overline{CS}$ pin	None

Note: SDRAM must be mapped in LOCAL1 area. It can't use other area.

3.8.1 Recommendable Memory Map

The recommendation logic address memory map at the t correspondence is shown in Figure 3.8.1. And physical address map is sh 3.8.2.

However, when memory area is less than 16 Mbytes and i section of CS/WAIT controller. Setting MMU is not necessar Since it is being fixed, the local area address cannot be changed. When SDRAM is used, must locate to LOCAL1 area.

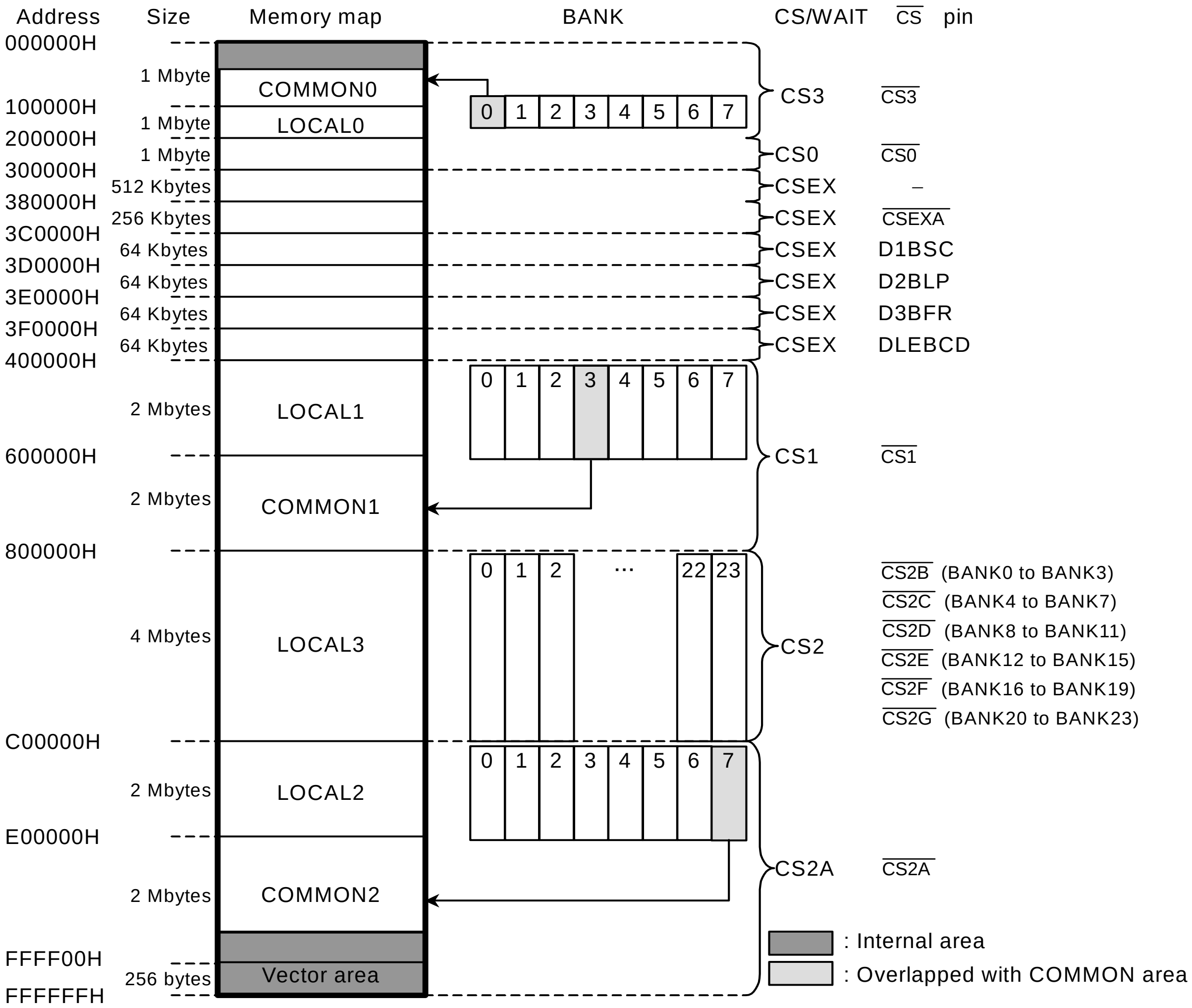


Figure 3.8.1 Logical Address Map

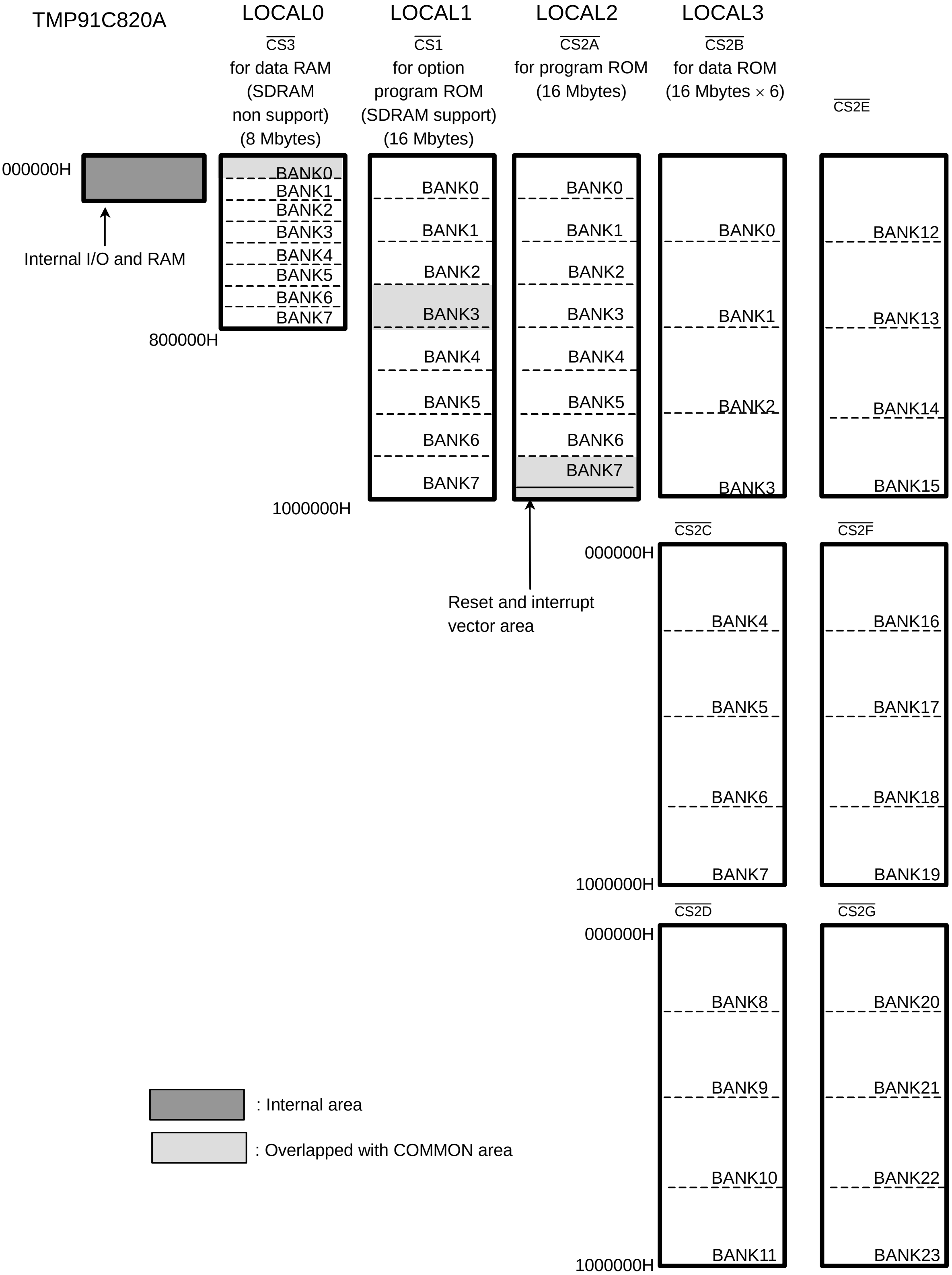
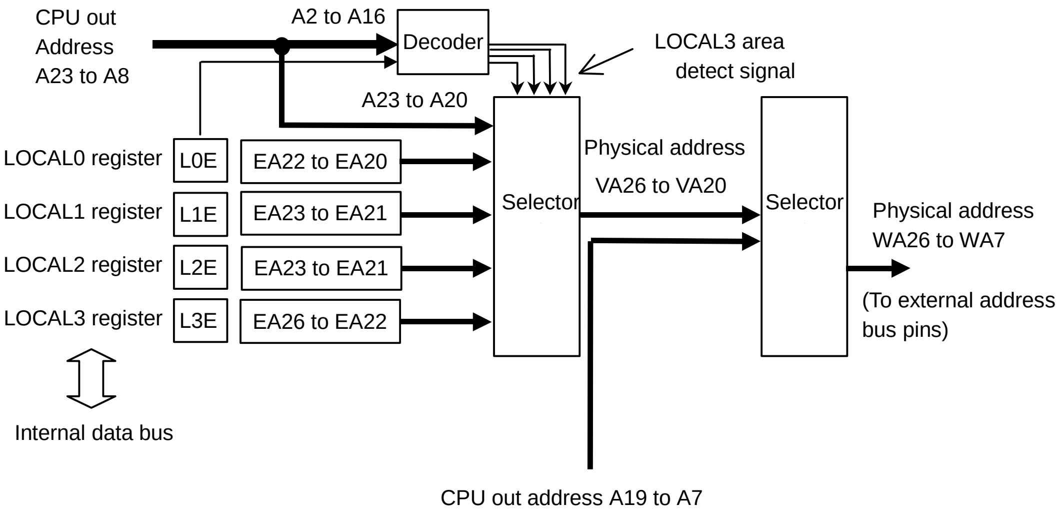


Figure 3.8.2 Physical Address Map

3.8.2 Block Diagram



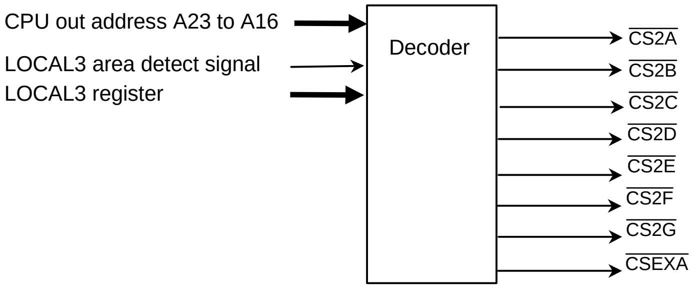


Figure 3.8.3 Block Diagram of MMU

3.8.3 Control Registers

LOCAL0 Register

		7	6	5	4	3	2	1	0
LOCAL0 (0350H)	Bit symbol	L0E					L0EA22	L0EA21	L0EA20
	Read/Write	R/W					R/W		
	After reset	0					0	0	0
	Function	Use BANK for LOCAL0 0: Not use 1: Use					Setting BANK number for LOCAL0		

LOCAL1 Register

		7	6	5	4	3	2	1	0
LOCAL1 (0351H)	Bit symbol	L1E					L1EA23	L1EA22	L1EA21
	Read/Write	R/W					R/W		
	After reset	0					0	0	0
	Function	Use BANK for LOCAL1 0: Not use 1: Use					Setting BANK number for LOCAL1		

LOCAL2 Register

		7	6	5	4	3	2	1	0
LOCAL2 (0352H)	Bit symbol	L2E					L2EA23	L2EA22	L2EA21
	Read/Write	R/W					R/W		
	After reset	0					0	0	0
	Function	Use BANK for LOCAL2 0: Disable 1: Enable					Setting BANK number for LOCAL2		

LOCAL3 Register

		7	6	5	4	3	2	1	0
LOCAL3 (0353H)	Bit symbol	L3E			L3EA26	L3EA25	L3EA24	L3EA23	L3EA22
	Read/Write	R/W			R/W	R/W	R/W	R/W	R/W
	After reset	0			0	0	0	0	0
	Function	Use BANK for LOCAL3 0: Disable 1: Enable			01000 to 01011: $\overline{\text{CS2D}}$ 00000 to 00011: $\overline{\text{CS2B}}$ 00100 to 00111: $\overline{\text{CS2C}}$ 01100 to 01111: $\overline{\text{CS2E}}$ 10000 to 10011: $\overline{\text{CS2F}}$ 10100 to 10111: $\overline{\text{CS2G}}$ 11000 to 11111: Set prohibition				

Figure 3.8.4 Register for LOCAL0 to LOCAL3

3.8.4 Operational Description

Setup bank value and bank use in bank setting registe register in COMMON area. Moreover, in that case, a co CS/WAIT controller simultaneously sets up mapping. W the LOCAL area, MMU outputs physical address to the o bank setting register. Access of external memory becomes possible t

Please do not use as bank that other banks since this CO overlaps with either of 8 BANKs of LOCAL area on the phy

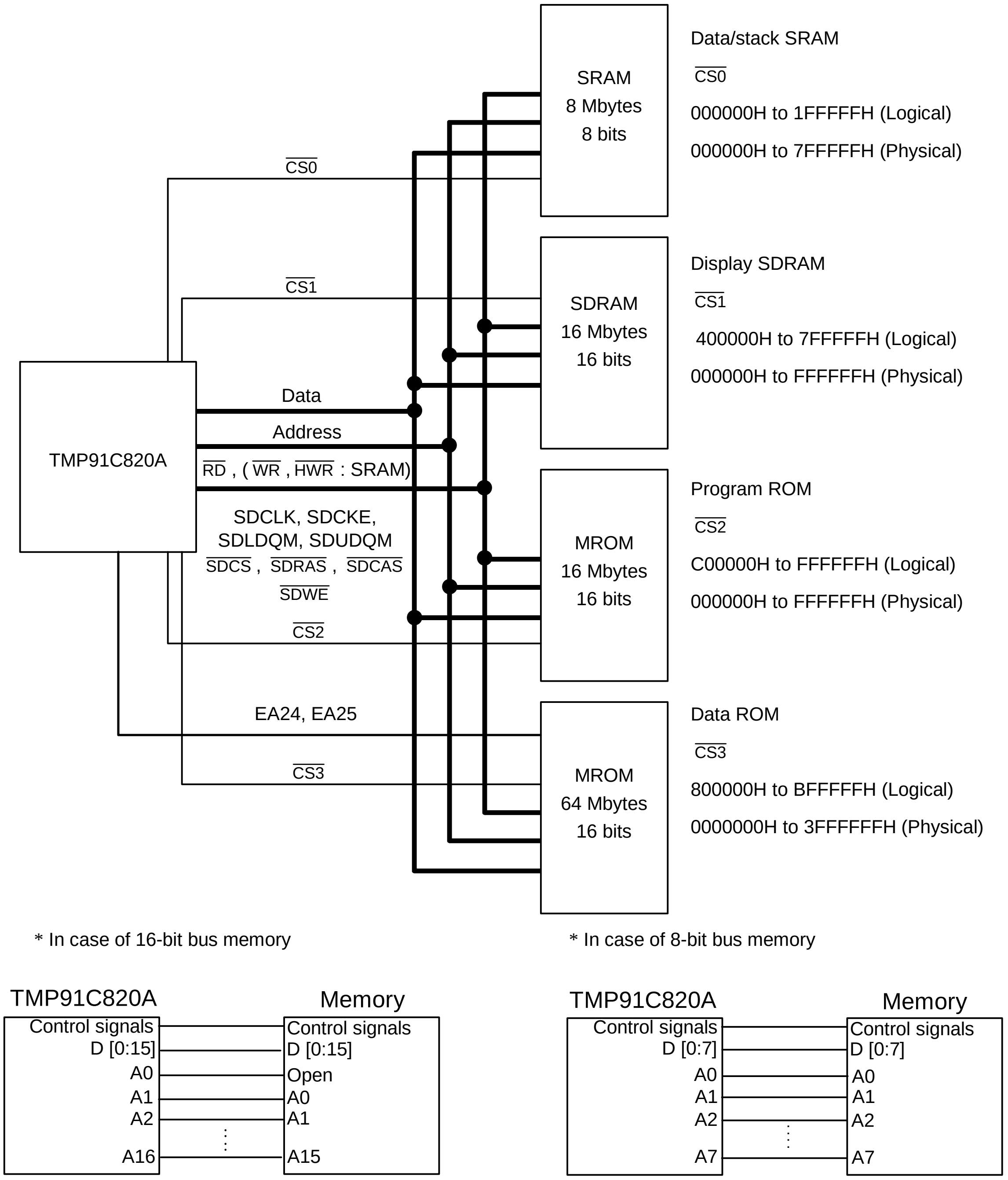


Figure 3.8.5 (a) H/W Setting Example

At Figure 3.8.5 (a), it shows example to connect TMP91C820A and some program ROM: MROM, 16 Mbyte, data ROM: MROM, 64 Mbyte, d 8-bit bus, display RAM: SDRAM, 16 Mbytes.

In case of 16-bit bus memory, connect to 1-bit shifted address bus f and 8-bit bus case, connect address bus of TMP91C820A.

In that figure, logical address are shown. And each memo chip select signal is $\overline{\text{CS0}}$, SRAM; $\overline{\text{CS1}}$, SDRAM; $\overline{\text{CS2}}$, program MROM; $\overline{\text{CS3}}$, data MROM. In case of this example, as data ROM is 64 Mbyte, MROM connect to EA24 and E

Initial condition after reset, because TMP91C820A acc program ROM. It can set free setting except program ROM.

```
;Initial Setting
;CS0
        LD      (MSAR0),00H      ; Logical address area: 000000H to 1FFFFFH
        LD      (MAMR0),FFH      ; Logical address size: 2 Mbytes
        LD      (B0CS),89H       ; Condition: 8 bits, 1 wait (8 Mbytes, SRAM)
;CS1
        LD      (MSAR1),40H      ; Logical address area: 400000H to 5FFFFFH
        LD      (MAMR1),FFH      ; Logical address size: 4 Mbytes
        LD      (B1CS),83H       ; Condition: 16 bits, 0 waits (16 Mbytes, SDRAM)
;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),85H       ; 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),02H      ; CS1 → SDCS setting
to
        LDW     (PZCR),0707H     ; HWR, WR, RD
        LD      (PFFC),7FH       ; PF [6:0] = SDRAM control
        LD      (SDACR),0ADH     ; Add-MUX enable, 128-M select

to                               ; SDRAM setup time

        LD      (SDACR),06DH     ; Add MUX enable, 128-M select
        LD      (SDRCR),01H      ; Interval reflesh
```

Figure 3.8.6 (b) Bank Operation S/W Example 1

Secondly, it shows example of initial setting at Figure

Because $\overline{CS0}$ connect to RAM: 8-bit bus, 8 Mbyte, 8-bit bus, it set bus. At thi set 1-wait setting. $\overline{CS1}$ is set the same 16-bit bus way. $\overline{CS2}$ is set 0 wait 16-bit bus and wait. $\overline{CS3}$ is set 16-bit bus and 3 waits.

By CS/WAIT controller, each channel's memory size, don't set memory size, need to set local address size: fitting to each lo address is set by BANK register setting.

CSEX setting of CS/WAIT controller is except above CS example isn't used CSEX setting.

Finally pin condition is set $\overline{CS0}$, $\overline{CS1}$, $\overline{CS2}$, $\overline{CS3}$ to port 6, EA24, EA25 and SDRAM condition.

```
;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      (LOCAL3),85H         ; LOCAL3 BANK5 set 14xxxxH
       LDW     HL,(800000H)         ; Load data (5555H) form  BANK5 (140000H: Physical address)
                                                                  of LOCAL3 ( CS3 )
       LD      (LOCAL3),88H         ; LOCAL3 BANK8 set 20xxxxH
       LDW     BC,(800000H)         ; Load data (AAAAH) form  BANK8 (200000H: Physical address)
                                                                  of LOCAL3 ( CS3 )
to
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
to
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
to
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 3.8.7 (c) BANK Operation S/W Example 2

Here shows examples of data access between one BANK and other BANK. F
one software example. A dot line square area shows one me
$\overline{CS2}$'s program ROM and $\overline{CS3}$'s data ROM. Program start from E00000H a
to BANK register of LOCAL3 area upper 5-bit address of ac

In case of this example, because most upper address bi
upper address bit of BANK register is 4-bit, 5-bit of upper address mear
BANKs. After setting BANK5, accessing 800000H to BFFFFFH address
address, actually access to physical 1400000H to 17000

```
;BANK Operation
;***** CS2 *****
ORG   000000H                    ; Program ROM: Start address at BANK0 of LOCAL2
ORG   200000H                    ; Program ROM: Start address at BANK1 of LOCAL2
      NOP                        ; Operation at BANK1of LOCAL2
to
      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
to
      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       (LOCAL2),81H      ; LOCAL2 BANK1 set 20xxxxH
      JP       C00000H           ; Jump to BANK1 (200000H: Physical address) of LOCAL2
to
ORG   E00100H
      LD       (LOCAL2),83H      ; LOCAL2 BANK3 set 60xxxxH
      JP       C00000H           ; Jump to BANK3 (600000H: Physical address) of LOCAL2
to
ORG   E00200H
      LD       (LOCAL1),00H      ; Disable BANK
to                               ; LCD display set
      LD       (LSARCH),60H      ; C_area start address
      LD       (LSARCM),00H      ; C_area start address
      LD       (LSARCL),00H      ; C_area start address
      SET      0,(LCTCTL)        ; LCD display start
ORG   FFFFFFH                    ; Program ROM: End address at BANK7( = COMMON2) of LOCAL2
```

```
;***** CS1 *****
ORG   000000H                    ; SDRAM: Start address at BANK0 of LOCAL1
ORG   200000H                    ; SDRAM: Start address at BANK1 of LOCAL1
ORG   400000H                    ; SDRAM: Start address at BANK2 of LOCAL1
ORG   600000H                    ; SDRAM: Start address at BANK3( = COMMON1) of LOCAL1
      dl       01234567H         ; Display data
to

ORG   800000H                    ; SDRAM: Start address at BANK4 of LOCAL1
ORG   A00000H                    ; SDRAM: Start address at BANK5 of LOCAL1
ORG   C00000H                    ; SDRAM: Start address at BANK6 of LOCAL1
ORG   E00000H                    ; SDRAM: Start address at BANK7 of LOCAL1
ORG   FFFFFFH                    ; SDRAM: End address at BANK7 of LOCAL1
```

Figure 3.8.8 (d) Bank Operation S/W Example 3

At Figure 3.8.8 (d), it shows example of program jump.

In the same way with before example, to CS2 is program ROM and CS1 is (SDCS) SDRAM. Program is COMMON address E00000H, firstly BANK register of LOCAL2 area upper 3-bit address of jump

After setting BANK1, jump FFC000H to logical LOCAL actually jump to physical address 200000H. When return to COMMO can only jump to E00000H to FFFFFFH without setting of LOCAL2 area.

By a way of setting of BANK register, the setting that BA conflict with is possible here. When two kinds addresses COMMON area exist, management of BANK is confused. We recommends no address and COMMON address conflict with.

When using LCD display data from SDRAM, it arrange display COMMON area in SDRAM. Because of, LCD displays DMA occeur at sy changed, you can use only COMMON area.

It is a mark paid attention to here, it needs to go by way moves from a BANK to a BANK. In other words, it must wri COMMON area and it prohibits writing the BANK registers bank register's data in BANK area, program runaway.

Please do not set Bank function of display RAM MMU. This is because r display data is not controlled by the CPU. Therefore if B LCD displaying, it cannot display collect. It is recommended display data area.

3.9 Serial Channels

TMP91C820A includes 3 serial I/O channels. For each channels
(Asynchronous transmission) or I/O interface mode (Synchronous transmission) can be sele
(Channel 2 can be selected only UART mode.)

- I/O interface mode — Mode 0: For transmitting and receivi
  synchronizing signal SCLK for ex

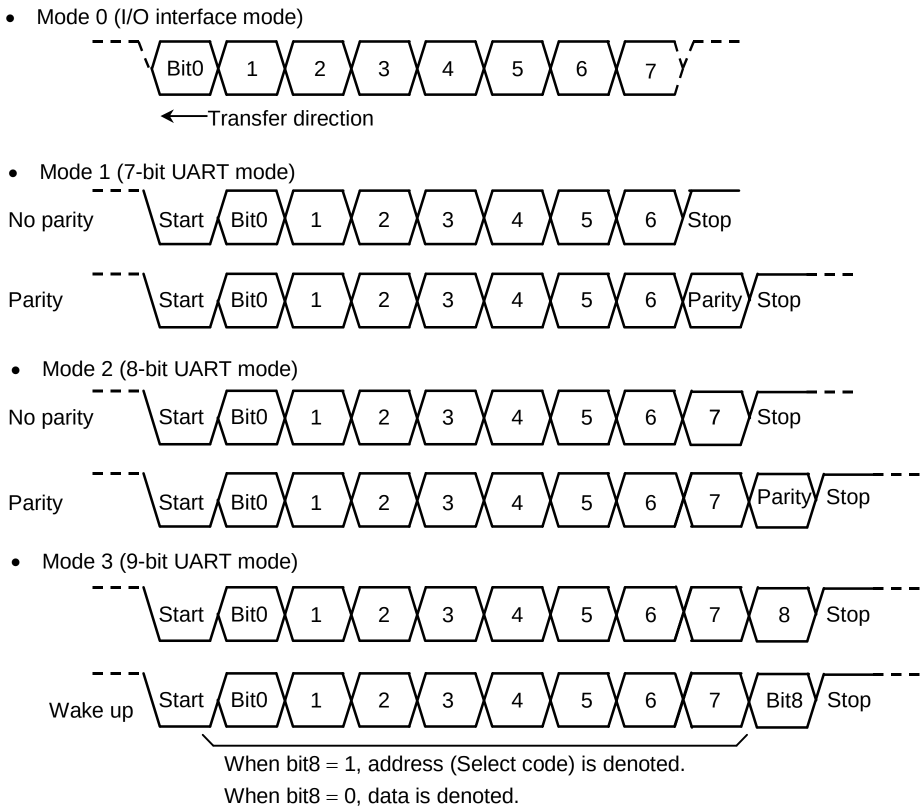
- UART mode
  - Mode 1: 7-bit data
  - Mode 2: 8-bit data
  - Mode 3: 9-bit data

In mode 1 and mode 2 a parity bit can be added. Mode 3 has a wakeup function fo
master controller to start slaves via a serial link (A multi-co
Figure 3.9.2, 3.9.3, 3.9.4 are block diagrams for each channel.
Each channel can be used independently.
Each channel operates in the same fashion except for t
operation of channel 0 is explained below.

Table 3.9.1 Differences between Channels 0 to 2

	Channel 0	Channel 1	Channel 2
Pin name	TXD0 (PC0) RXD0 (PC1) $\overline{\text{CTS0}}$ /SCLK0 (PC2)	TXD1 (PC3) RXD1 (PC4) $\overline{\text{CTS1}}$ /SCLK1 (PC5)	TXD2 (PB0) RXD2 (PB1)
IrDA mode	Yes	No	No

This chapter contains the following sections:



When bit8 = 1, address (Select code) is denoted.
When bit8 = 0, data is denoted.

Figure 3.9.1 Data Formats

3.9.1 Block Diagrams

Figure 3.9.2 is a block diagram representing serial

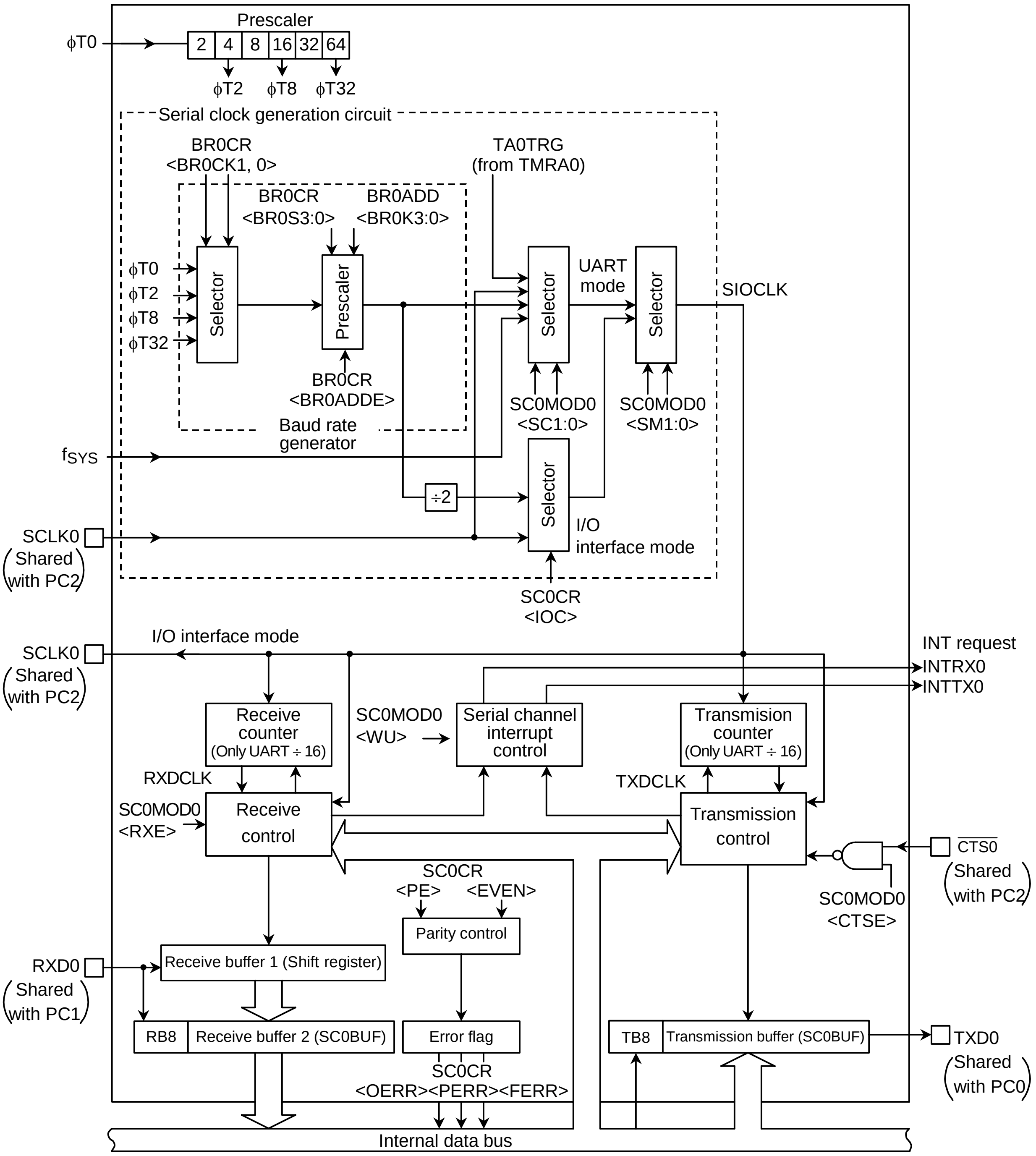


Figure 3.9.2 Block Diagram of the Serial Channel 0

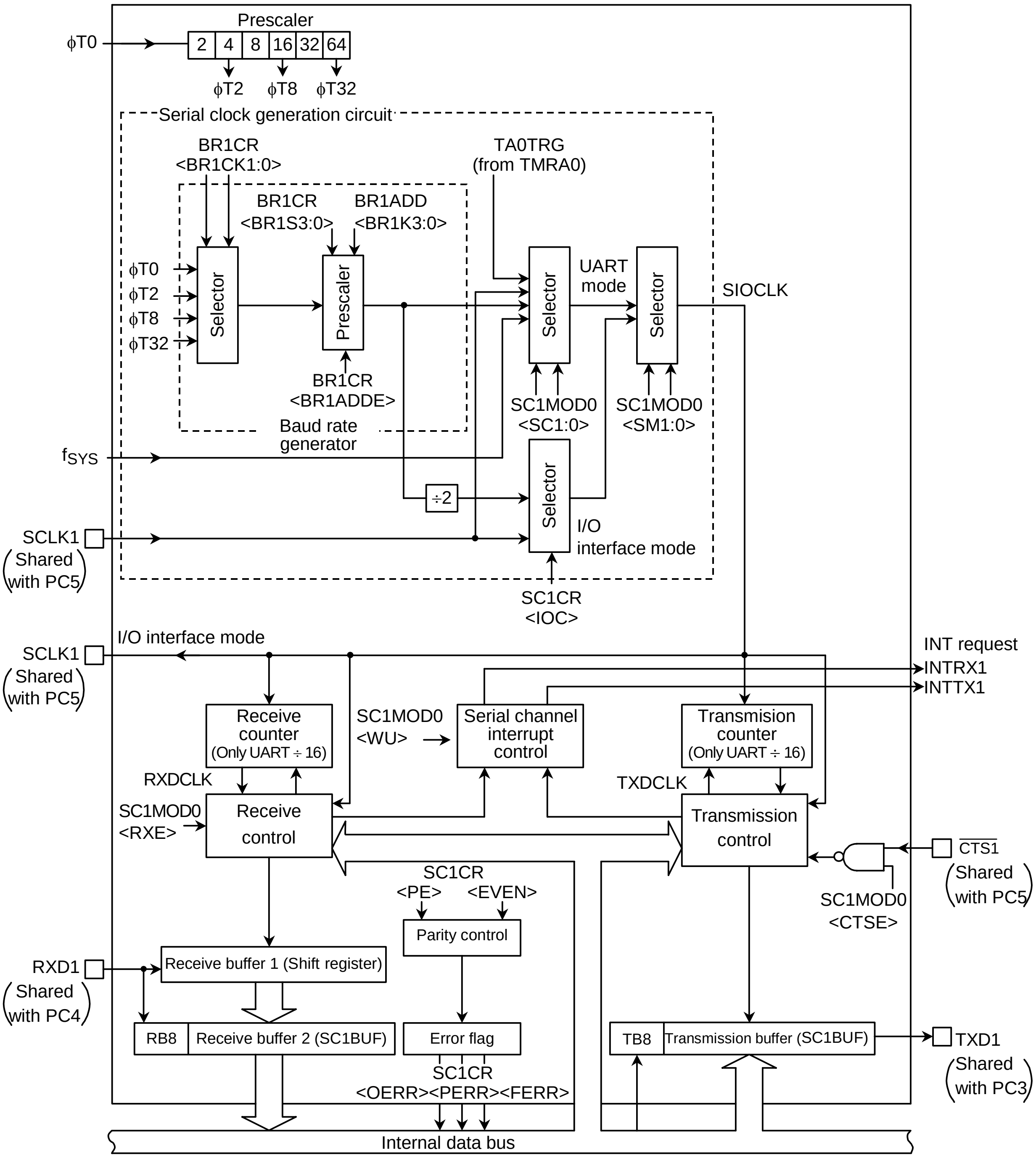


Figure 3.9.3 Block Diagram of the Serial Channel 1

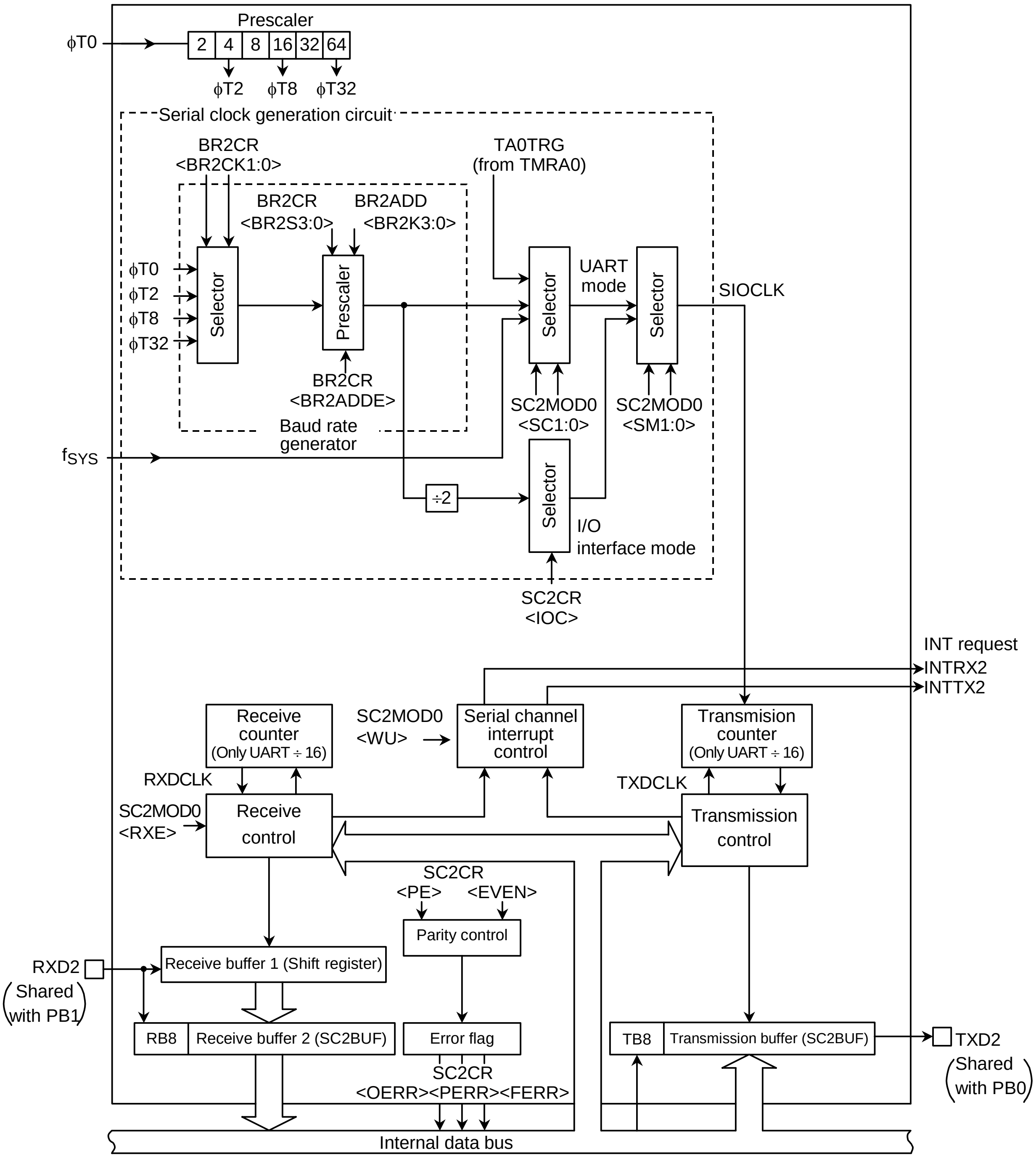


Figure 3.9.4 Block Diagram of the Serial Channel 2

3.9.2 Operation of Each Circuit

(1) Prescaler, prescaler clock selects

There is a 6-bit prescaler for serial clock. The clock sel
SYSCR<PRCK1:0> is divided by 4 and input to the ϕT0. The prescaler
be run by selecting the baud rate as the waking serial cloc

Table 3.9.2 shows prescaler clock resolution to baud rate genera

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	$fs/2^2$	$fs/2^4$	$fs/2^6$	$fs/2^8$
0 (fc)		000 (fc)	$fc/2^2$	$fc/2^4$	$fc/2^6$	$fc/2^8$
		001 (fc/2)	$fc/2^3$	$fc/2^5$	$fc/2^7$	$fc/2^9$
		010 (fc/4)	$fc/2^4$	$fc/2^6$	$fc/2^8$	$fc/2^{10}$
		011 (fc/8)	$fc/2^5$	$fc/2^7$	$fc/2^9$	$fc/2^{11}$
		100 (fc/16)	$fc/2^6$	$fc/2^8$	$fc/2^{10}$	$fc/2^{12}$
	10 (fc/16 clock)	XXX	–	$fc/2^8$	$fc/2^{10}$	$fc/2^{12}$

X: Don’t care, –: Cannot be used

The baud rate generator selects between 4-clock ϕT0, ϕT2, ϕT8, and ϕT32 clock i
among the prescaler outputs.

(2) Baud rate generator

The baud rate generator is a circuit which generates transmission a
clocks that determine the transfer rate of the seri

The input clock to the baud rate generator, φT0, φT2, φT8 or φT32, is generatec
the 6-bit prescaler which is shared by the timers. One of these input cloc
using the BR0CR<BR0CK1:0> field in the baud rate ge

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

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

- In UART mode

(1) When BR0CR<BR0ADDE> = 0

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

(2) When BR0CR<BR0ADDE> = 1

The N + (16 − K)/16 division function is enabled. T
divides the selected prescaler clock by N + (16 − K)/16 using the value of
BR0CR<BR0S3:0> (N = 2, 3 ... 15) and the value of K set in B
(K = 1, 2, 3 ... 15)

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

- In I/O interface mode

The N + (16 − K)/16 division function is not availabl
BR0CR<BR0ADDE> to 0 before dividing by N.

The method for calculating the transfer rate wh
used is explained below.

- In UART mode

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

- In I/O interface mode

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

- Integer divider (N divider)

For example, when the source clock frequency (fc) = 12.288 MHz, the input c
frequency = φT2 (fc/16), the frequency divider N = 5, and (BR0
BR0CR<BR0ADDE> = 0, the baud rate in UART mode is as fol

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

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

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

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

- N + (16 – K)/16 divider (Only UART mode)

Accordingly, when the source clock frequency (fc) = 4.8 MHz, the input c
frequency = φT0, the frequency divider N = 7 (BR0CR<BR
(BR0ADD<BR0K3:0> = 3), and BR0CR<BR0ADDE> = 1, the baud rate in U
mode is as follows:

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

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

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

Table 3.9.3, 3.9.4 show UART mode transfer rates.

Additionally, the external clock input is avai
channels 0, 1). The method for calculating the ba

- 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 BR0CR<BR0ADDE> = 0)

Unit (kbps)

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

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 and the system clock is the prescaler clock input.

Table 3.9.4 Selection of Transfer Rate
(When TMRA0 with input clock ϕT1 is used)

Unit (kbps)

fc TA0REG0	12.288 MHz	12 MHz	9.8304 MHz	8 MHz	6.144 MHz
1H	96		76.8	62.5	48
2H	48		38.4	31.25	24
3H	32	31.25			16
4H	24		19.2		12
5H	19.2				9.6
8H	12		9.6		6
AH	9.6				4.8
10H	6		4.8		3
14H	4.8				2.4

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

$$\text{Transfer rate} = \frac{\text{Clock frequency determined by SYSCR0<PRCK1:0>}}{\text{TA0REG} \times 2^3 \times 16}$$

(when TMRA0 (Input clock ϕT1)is used)

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

Note 2: The values in this table are calculated for when fc is selected as the system clock, the clock gear is set for fc and the system clock is the prescaler clock input.

(3) Serial clock generation circuit

This circuit generates the basic clock for transmi

- In I/O interface mode

In SCLK output mode with the setting SC0CR<IOC> = 0, the basic cloc generated by dividing the output of the baud rate generator by 2, a previously.

In SCLK input mode with the setting SC0CR<IOC> = 1, the rising edg falling edge will be detected according to the register to generate the basic clock.

- In UART mode

The SC0MOD0<SC1:0> setting determines whether the baud rate g clock, the internal clock f_{SYS}, the match detect signal from or the external clock (SCLK0) is used to generate

(4) Receiving counter

The receiving counter is a 4-bit binary counter used in UART mode counts up the pulses of the SIOCLK clock. It takes bit of data; each data bit is sampled three times cycles.

The value of the data bit is determined from th majority rule.

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

(5) Receiving control

- In I/O interface mode

In SCLK output mode with the setting SC0CR<IOC> = 0, the RXD0 signal sampled on the rising or falling edge of the shi SCLK0 pin, according to the SC0CR<SCLKS> setting.

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

- In UART mode

The receiving control block has a circuit, whi majority rule. Received bits are sampled three three samples are 0, the bit is recognized as start bit and the operation commences.

The values of the data bits that are received a majority rule.

(6) The receiving buffers

To prevent overrun errors, the receiving buffers
structure.

Received data is stored one bit at a time in recei
register). When 7 bits or 8 bits of data have been sto
data is transferred to receiving buffer 2 (SC0BUF);
be generated. The CPU only reads receiving buffer 2
reads receiving buffer 2 (SC0BUF), the received dat
However, unless receiving buffer 2 (SC0BUF) is read before all bits of the
received by receiving buffer 1, an overrun error occurs. If an
contents of receiving buffer 1 will be lost, althou
and SC0CR<RB8> bit will be preserved.

SC0CR<RB8> is used to store either the parity bit –
the most significant bit (MSB) – in 9-bit UART mode.

In 9-bit UART mode the wakeup function for the sl
setting SC0MOD0<WU> to 1; in this mode INTRX0 inte
value of SC0CR<RB8> is 1.

(7) Transmission counter

The transmission counter is a 4-bit binary counter
which, like the receiving counter, counts the SIOCL
generated every 16 SIOCLK clock pulses.

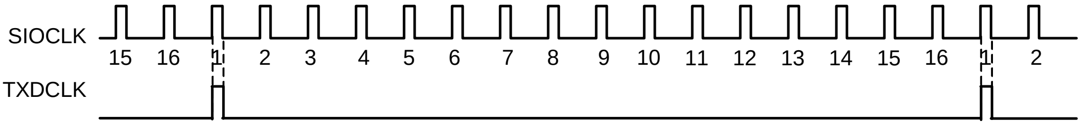


Figure 3.9.5 Generation of the Transmission Clock

(8) Transmission control

- In I/O interface mode

In SCLK output mode with the setting SC0CR<IOC> = 0,
transmission buffer is output one bit at a time t
falling edge of the shift clock which is output or
SC0CR<SCLKS> setting.

In SCLK input mode with the setting SC0CR<IOC> = 1,
transmission buffer is output one bit at a time o
falling edge of the SCLK0 input, according to the

- In UART mode

When transmission data sent from the CPU is wr
buffer, transmission starts on the rising edge o
transmission shift clock TXDSFT.

Handshake function

Serial channels 0 has a $\overline{\text{CTS}}$ pin. Use of this pin allows da
in units of one frame; thus, overrun errors can
functions is enabled or disabled by the SC0MOD<C

When the $\overline{\text{CTS}}$ pin goes high on completion of the cur
transmission is halted until the $\overline{\text{CTS}}$ pin goes low again. However,
interrupt is generated, it requests the next data to the CPU. The ne
written in the transmission buffer and data sending is halted.

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

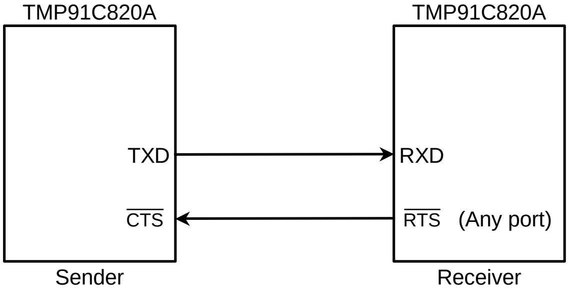
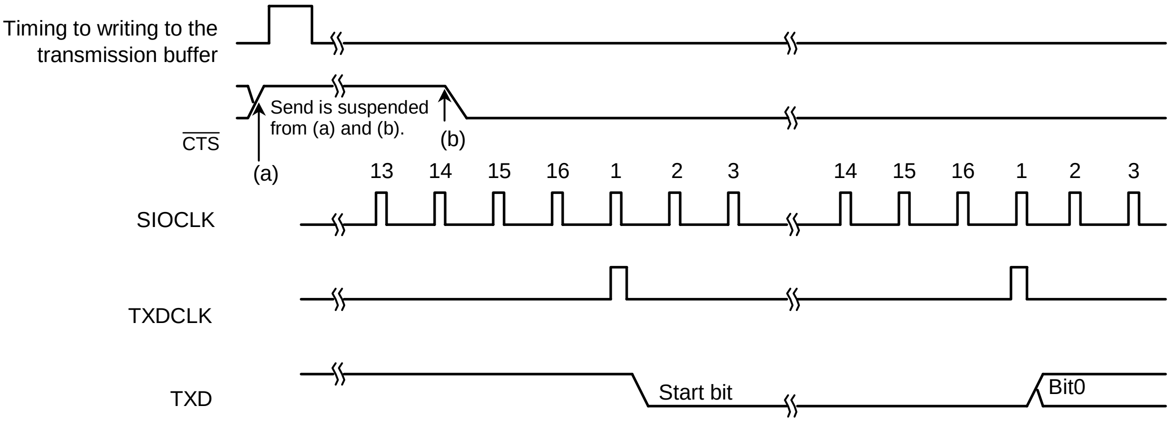


Figure 3.9.6 Handshake Function


Timing to writing to the transmission buffer
Send is suspended from (a) and (b).
CTS
(a)
(b)
13
14
15
16
1
2
3
14
15
16
1
2
3
SIOCLK
TXDCLK
TXD
Start bit
Bit0

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

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

Figure 3.9.7 $\overline{\text{CTS}}$ (Clear to send) Timing

(9) Transmission buffer

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

(10) Parity control circuit

When SC0CR<PE> in the serial channel control regi
transmit and receive data with parity. However, parity can be added on
UART mode or 8-bit UART mode. The SC0CR<EVEN> fiel
control register allows either even or odd parity to be selected.

In the case of transmission, parity is automatically generated when d
to the transmission buffer SC0BUF. The data is tran
been stored in SC0BUF<TB7> in 7-bit UART mode or in
UART mode. SC0CR<PE> and SC0CR<EVEN> must be set before the transmissi
data is written to the transmission buffer.

In the case of receiving, data is shifted into rece
after the data has been transferred to receiving buffer 2 (SC0BUF), and t
with SC0BUF<RB7> in 7-bit UART mode or with SC0CR<R
If they are not equal, a parity error is generated an

(11) Error flags

Three error flags are provided to increase the rel

1. Overrun error <OERR>

If all the bits of the next data item have been rec
valid data still remains stored in receiving buf
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 SC0MOD

b) Wait to terminate current frame

c) Read receiving buffer

d) Read error flag

e) Set to enable receiving (Write 1 to SC0MOD0<RXE>)

f) Request to transmit again

4) Other

2. Parity error <PERR>

The parity generated for the data shifted into receiving buffer 2 (
compared with the parity bit received via the RX
parity error is generated.

3. Framing error <FERR>

The stop bit for the received data is sampled th
the majority of the samples are 0, a framing error

(12) Timing generation

a. In UART mode

Receiving

Mode	9 Bits (Note)	8 Bits + Parity (Note)	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 bits and 8 bits + parity modes, interrupts coincide with the ninth bit pulse. Thus, when servicing the interrupt, it is necessary to wait for a 1-bit period (to allow the stop bit to be transferred) to allow checking for a framing error.

Transmitting

Mode	9 Bits	8 Bits + Parity	8 Bits, 7 Bits + Parity, 7 Bits
Interrupt timing	Just before stop bit is 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 last bit data. (See figure 3.9.25)
	SCLK input mode	Immediately after rise of last SCLK signal rising mode, or immediately after fall in falling mode. (See figure 3.9.26)
Receiving interrupt timing	SCLK output mode	Timing used to transfer received to data receive buffer 2 (SC0BUF) (e.g., immediately after last SCLK). (See figure 3.9.27)
	SCLK input mode	Timing used to transfer received data to receive buffer 2 (SC0BUF) (e.g., immediately after last SCLK). (See figure 3.9.28)

3.9.3 SFRs

SC0MOD0
(0202H)

	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	Wakeup function 0: Disable 1: Enable	Serial transmission mode 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial transmission clock (UART) 00: TMRA0 trigger 01: Baud rate generator 10: Internal clock f_{SYS} 11: External clcok (SCLK0 input)	

Serial transmission clock source (UART)

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

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

Serial transmission mode

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

Wakeup function

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

Receiving function

0	Receive disabled
1	Receive enabled

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

0	Disabled (Always transferable)
1	Enabled

Transmission data bit8

Figure 3.9.8 Serial Mode Control Register (Channel 0, 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	Wakeup function 0: Disable 1: Enable	Serial transmission mode 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial transmission clock (UART) 00: TMRA0 trigger 01: Baud rate generator 10: Internal clock f_{SYS} 11: External clcok (SCLK0 input)	

Serial transmission clock source (for UART)

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

Serial transmission mode

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

Wakeup function

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

Receiving function

0	Receive disabled
1	Receive enabled

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

0	Disabled (Always transferable)
1	Enabled

Transmission data bit8

Figure 3.9.9 Serial Mode Control Register (Channel 1, SC1MOD0)

SC2MOD0
(0212H)

	7	6	5	4	3	2	1	0
Bit symbol	TB8	–	RXE	WU	SM1	SM0	SC1	SC0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Transfer data bit8	Always fixed to "0".	Receive function 0: Receive disable 1: Receive enable	Wakeup function 0: Disable 1: Enable	Serial transmission mode 00: Reserved 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: Reserved	

Serial transmission clock source (for UART)

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

Serial transmission mode

00	Reserved	
01	UART mode	7-bit mode
10		8-bit mode
11		9-bit mode

Wakeup function

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

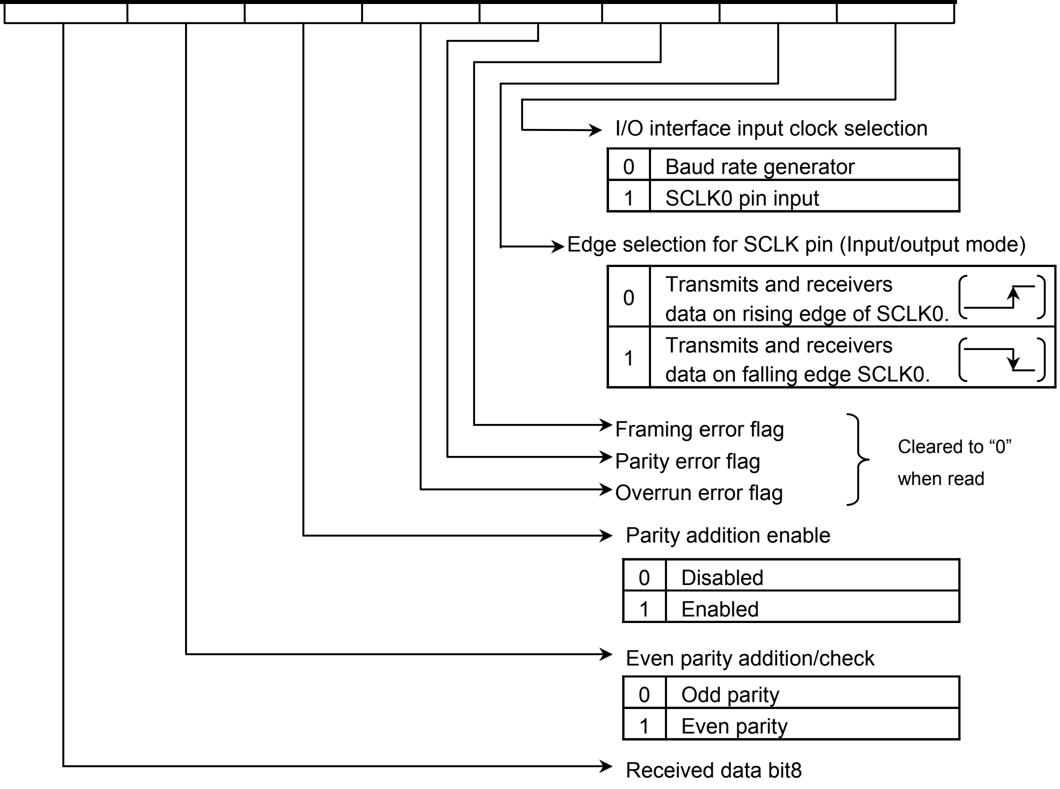
Receiving function

0	Receive disabled
1	Receive enabled

Transmission data bit8

Figure 3.9.10 Serial Mode Control Register (Channel 2, SC2MOD0)

SC0CR (0201H)		7	6	5	4	3	2	1	0
	Bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
	Read/Write	R	R/W		R (Cleared to 0 when read)			R/W	
	After reset	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 Overrun	Parity	Framing	0: SCLK0 ↑ 1: SCLK0 ↓	0: Baud rate generator 1: SCLK0 pin input



I/O interface input clock selection

0	Baud rate generator
1	SCLK0 pin input

Edge selection for SCLK pin (Input/output mode)

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

Framing error flag
Parity error flag
Overrun error flag
Cleared to “0” when read

Parity addition enable

0	Disabled
1	Enabled

Even parity addition/check

0	Odd parity
1	Even parity

Received data bit8

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

Figure 3.9.11 Serial Control Register (Channel 0, 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)			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 SCLK pin (Input/output mode)

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

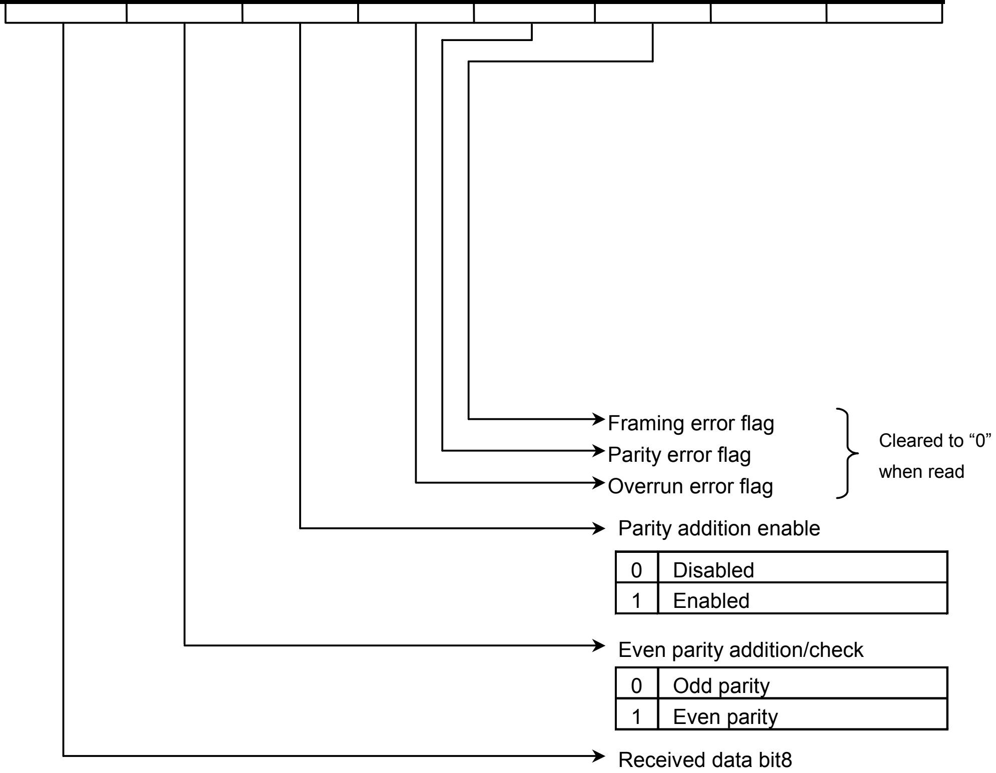
Received data bit8

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

Figure 3.9.12 Serial Control Register (Channel 1, SC1CR)

SC2CR
(0211H)

	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)			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 Overrun	Parity	Framing	Always fixed to “0”.	Always fixed to “0”.



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 3.9.13 Serial Control Register (Channel 2, SC2CR)

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 fixed to "0".	+ (16 − K)/16 division 0: Disable 1: Enable	00: φT0 01: φT2 10: φT8 11: φT32		Setting of the divided frequency			

+ (16 − K)/16 division enable

0	Disable
1	Enable

Setting the input clock of baud rate generator

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

BR0ADD
(0204H)

	7	6	5	4	3	2	1	0
Bit symbol					BR0K3	BR0K2	BR0K1	BR0K0
Read/Write					R/W			
After reset					0	0	0	0
Function					Sets frequency divisor K (Divided by N = (16 − K)/16)			

Sets baud rate generator frequency divisor

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

Note1: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.

Figure 3.9.14 Baud Rate Generator Control (Channel 0, 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 fixed to "0".	+ (16 − K)/16 division 0: Disable 1: Enable	00: φT0 01: φT2 10: φT8 11: φT32		Divided frequency setting			

+ (16 − K)/16 division enable

0	Disabled
1	Enabled

Input clock selection for baud rate generator

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

BR1ADD (020CH)

	7	6	5	4	3	2	1	0
Bit symbol					BR1K3	BR1K2	BR1K1	BR1K0
Read/Write					R/W			
After reset					0	0	0	0
Function					Set frequency divisor K (Divided by N = (16 − K)/16)			

Baud rate generator frequency divisor setting

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

BR1CR <BR1S3:0>
BR1ADD <BR1K3:0>
	0000(N = 16)
	or
	0001(N = 1)
		0010 (N = 2)
		to
		1111 (N = 15)
			0001(N = 1) (Only UART)
			to
			1111 (N = 15)
			0000 (N = 16)

0000	Disable	Disable	Divided by N
0001 (K = 1) to 1111 (K = 15)	Disable	Disabled by N + (16 −K)/16	

Note1: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.

Figure 3.9.15 Baud Rate Generator Control (Channel 1, BR1CR, BR1ADD)

		7	6	5	4	3	2	1	0
BR2CR (0213H)	Bit symbol	−	BR2ADDE	BR2CK1	BR2CK0	BR2S3	BR2S2	BR2S1	BR2S0
	Read/Write	R/W							
	After reset	0	0	0	0	0	0	0	0
	Function	Always fixed to “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

		7	6	5	4	3	2	1	0
BR2ADD (0214H)	Bit symbol					BR2K3	BR2K2	BR2K1	BR2K0
	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

	BR2CR<BR2ADDE> = 1		BR2CR<BR2ADDE> = 0
BR2CR <BR2S3:0> BR2ADD <BR2K3:0>	0000 (N = 16) or 0001 (N = 1)	0010 (N = 2) to 1111 (N = 15)	0001 (N = 1) (Only UART) to 1111 (N = 15) 0000 (N = 16)
0000	Disable	Disable	Divided by N
0001 (K = 1) to 1111 (K = 15)	Disable	Disabled by N + (16 − K)/16	

Note1: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 BR2CR <BR2ADDE> to 1 after setting K (K = 1 to 15) to BR2ADD<BR2K3:0> when +(16-K)/16 division function is used.

Figure 3.9.16 Baud Rate Generator Control (Channel 2, BR2CR, BR2ADD)

SC0BUF
(0200H)

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

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

Note: Prohibit read-modify-write for SC0BUF.

Figure 3.9.17 Serial Transmission/Receiving Buffer Registers (Channel 0, 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 3.9.18 Serial Mode Control Register 1 (Channel 0, SC0MOD1)

SC1BUF
(0208H)

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

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

Note: Prohibit read-modify-write for SC1BUF.

Figure 3.9.19 Serial Transmission/Receiving Buffer Registers (Channel 1, 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 3.9.20 Serial Mode Control Register 1 (Channel 1, SC1MOD1)

SC2BUF
(0210H)

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

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

Note: Prohibit read-modify-write for SC2BUF.

Figure 3.9.21 Serial Transmission/Receiving Buffer Registers (Channel 2, SC2BUF)

SC2MOD1
(0215H)

	7	6	5	4	3	2	1	0
Bit symbol	I2S2	FDPX2						
Read/Write	R/W	R/W						
After reset	0	0						
Function	IDLE2 0: Stop 1: Run	Duplex 0: Half 1: Full						

Figure 3.9.22 Serial Mode Control Register 1 (Channel 2, SC2MOD1)

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 p

data to or receive data from an external shift register.

This mode includes the SCLK output mode to output s

SCLK input mode to input external synchronous clock

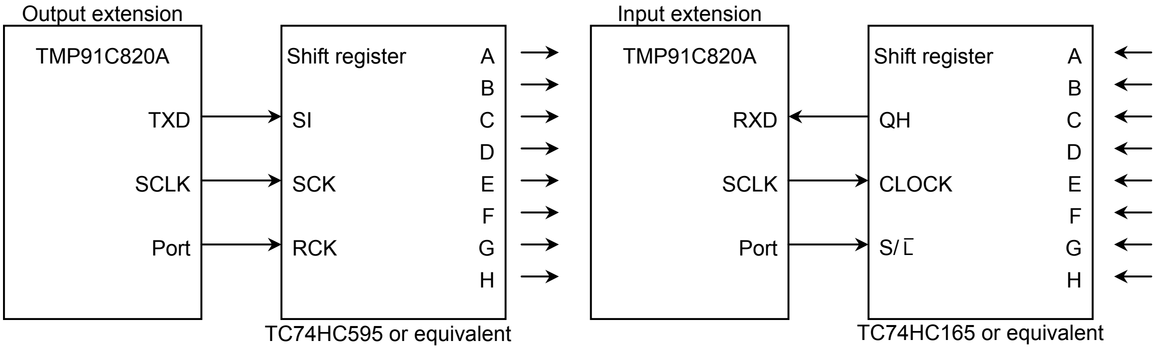


Figure 3.9.23 SCLK Output Mode Connection Example

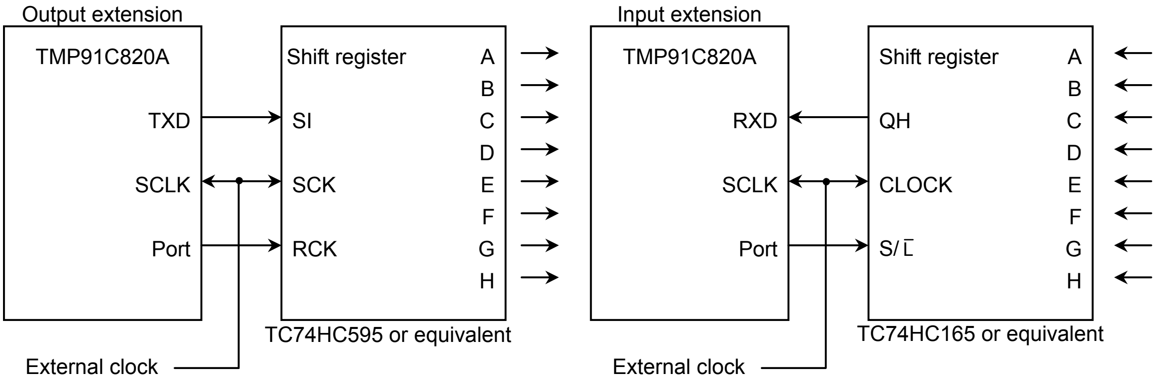


Figure 3.9.24 Example of SCLK Input Mode Connection

a. Transmission

In SCLK output mode 8-bit data and a synchronou TXD0 and SCLK0 pins respectively each time the C transmission buffer. When all data is output, I generate the INTTX0 interrupt.

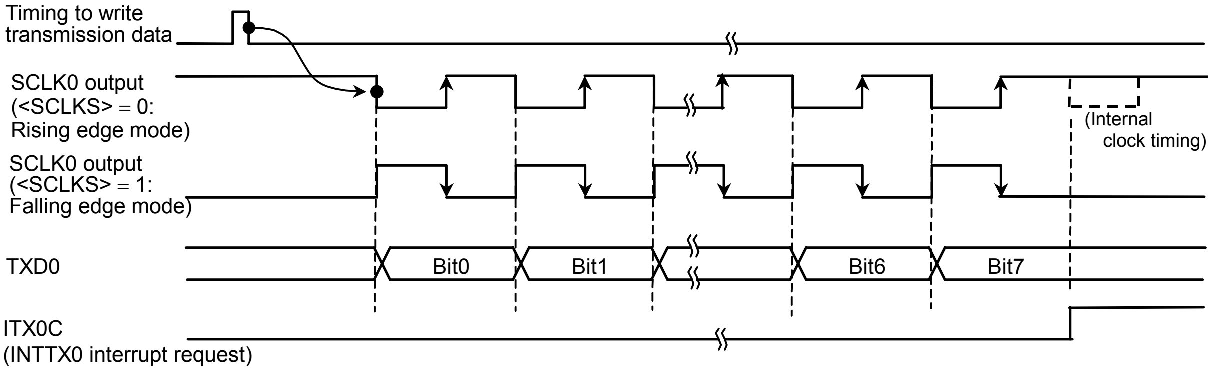


Figure 3.9.25 Transmitting Operation in I/O Interface Mode (SCLK0 output mode) (Channel 0)

In SCLK input mode, 8-bit data is output on the input becomes active after writing the data to the transmissi the CPU.

When all data is output, INTES0<ITX0C> will be interrupt.

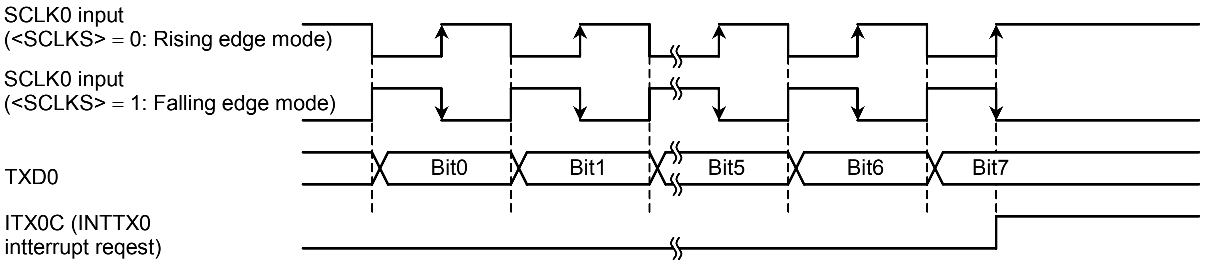


Figure 3.9.26 Transmitting Operation in I/O Interface Mode (SCLK0 input mode) (Channel 0)

b. Receiving

In SCLK output mode, the synchronous clock is ou
the data is shifted to receiving buffer 1. This starts when the receive
INTES0<IRX0C> is cleared by reading received data. When 8-bi
received, the data will be transferred to receiving buffer 2 (SC0BUF) according
the timing shown below) and INTES0<IRX0C> will be set to generat
interrupt.

The outputting for the first SCLK0 starts by set

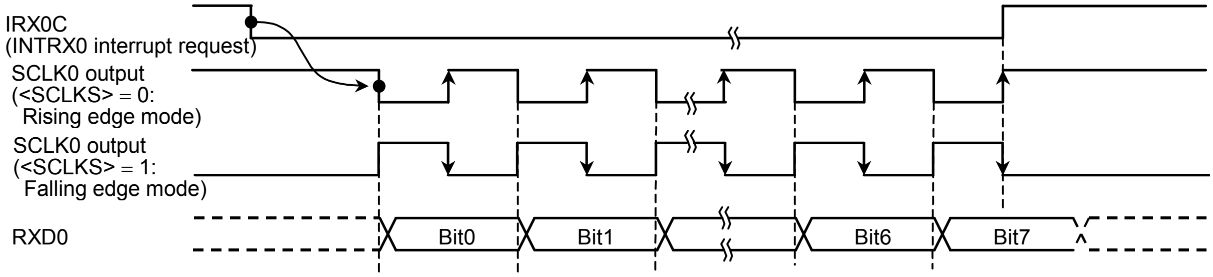


Figure 3.9.27 Receiving Operation in I/O Interface Mode (SCLK0 output mode)
(Channel 0)

In SCLK input mode, the data is shifted to recei
input becomes active after the receive interrup
by reading the received data. When 8-bit data is r
to receiving buffer 2 (SC0BUF according to the ti
<IRX0C> will be set to generate INTRX0 interrupt.

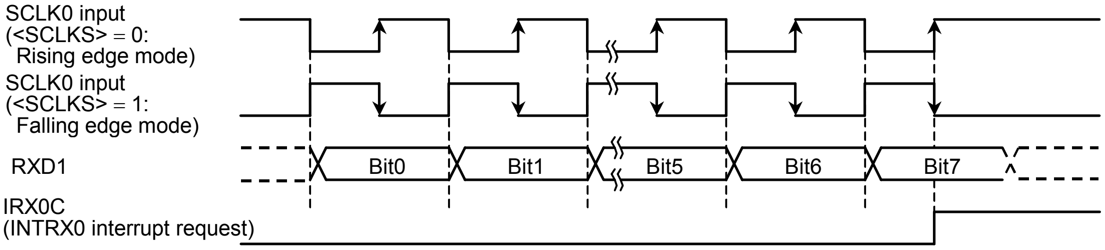


Figure 3.9.28 Receiving Operation in I/O Interface Mode (SCLK0 input mode)
(Channel 0)

Note: The system must be put in the receive enable state (SCMOD0<RXE> = 1) before data can be received.

c. Transmission and receiving (Full duplex mode)

When the full duplex mode is used, set the receive interru
enable the level of transmit interrupt. In the tra
receiving buffer before setting the next transmi

The example is following.

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

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

Main routine

	7	6	5	4	3	2	1	0	
INTES0	0	0	0	1	0	0	0	0	Set the INTTX0 level to 1. Set the INTRX0 level to 0.
PCCR	–	–	–	–	–	1	0	1	Set PC0, PC1 and PC2 to function as the TXD0, RXD0 and SCLK0 pins respectively.
PCFC	–	–	–	–	–	1	–	1	
SC0MOD0	0	0	0	0	0	0	0	0	Select I/O interface mode.
SC0MOD1	1	1	0	0	0	0	0	0	Select full duplex mode.
SC0CR	0	0	0	0	0	0	0	0	SCLK_out, transmit on negative edge, receive on positive edge.
BR0CR	0	0	1	1	0	0	1	1	Baud rate = 9600 bps.
SC0MOD0	0	0	1	0	0	0	0	0	Enable receiving.
SC0BUF	*	*	*	*	*	*	*	*	Set the transmit data and start.
INTTX0 interrupt routine									
Acc SC0BUF									Read the receiving buffer.
SC0BUF	*	*	*	*	*	*	*	*	Set the next transmit data.

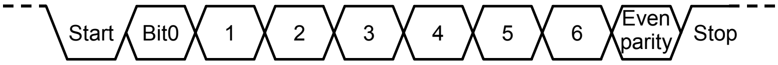
X: Don't care, −: No change

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

7-bit UART mode is selected by setting the ser
SC0MOD0<SM1:0> field to 01.

In this mode a parity bit can be added. Use of a parity bit is enabled o
the setting of the serial channel control register SC0CR<PE> bit; whether e
or odd parity will be used is determined by the S
SC0CR<PE> is set to 1 (Enabled).

Example: When transmitting data of the following fo
be set as described below. This example applies to chann



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

* Clock state

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

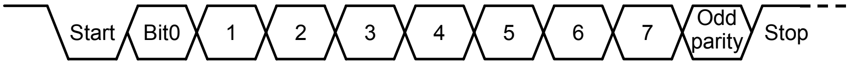
		7	6	5	4	3	2	1	0	
PCCR	←	–	–	–	–	–	–	–	1	Set PC0 to function as the TXD0 pin.
PCFC	←	–	–	–	–	–	–	–	1	
SC0MOD	←	X	0	–	X	0	1	0	1	Select 7-bit UART mode.
SC0CR	←	X	1	1	X	X	X	0	0	Add even parity.
BR0CR	←	0	0	1	0	0	1	0	1	Set the transfer rate to 2400 bps.
INTES0	←	1	1	0	0	–	–	–	–	Enable the INTTX0 interrupt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

X: Don’t care, –: No change

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

8-bit UART mode is selected by setting SC0MOD0<SM1:0> to 10. In thi
parity bit can be added (Use of a parity bit is enabled or disabled by t
SC0CR<PE>); whether even parity or odd parity will be used is determi
SC0CR<EVEN> setting when SC0CR<PE> is set to 1 (Ena

Example: When receiving data of the following form
set as described below.



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

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

Main settings
                7 6 5 4 3 2 1 0
PCCR       ←   – – – – – – 0 –     Set PC1 to function as the RXD0 pin.
SC0MOD     ←   – 0 1 X 1 0 0 1     Enable receiving in 8-bit UART mode.
SC0CR      ←   X 0 1 X X X 0 0     Add even parity.
BR0CR      ←   0 0 0 1 0 1 0 1     Set the transfer rate to 9600 bps.
INTES0     ←   – – – – 1 1 0 0     Enable the INTRX0 interrupt and set it to interrupt level 4.
Interrupt processing
Acc        ←   SC0CR AND 00011100  }
if Acc     ≠   0 then ERROR        }  Check for errors.
Acc        ←   SC0BUF                 Read the received data.

X: Don't care, –: No change
```

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

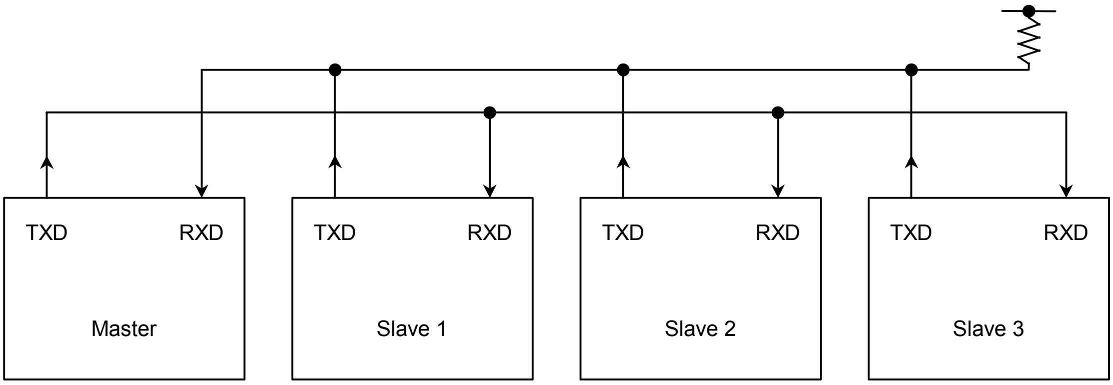
9-bit UART mode is selected by setting SC0MOD0<SM1

bit cannot be added.

In the case of transmission the MSB (9th bit) is wr
case of receiving it is stored in SC0CR<RB8>. When t
MSB is read or written first, before the rest of the

Wakeup function

In 9-bit UART mode, the wakeup function for slave c
SC0MOD0<WU> to 1. The interrupt INTRX0 occurs only



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

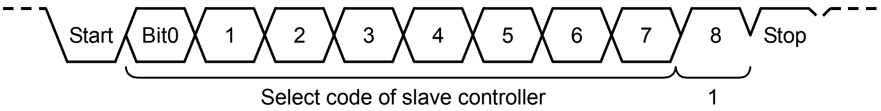
Figure 3.9.29 Serial Link Using Wakeup Function

Protocol

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

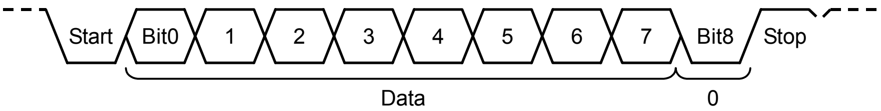
b. Set the SC0MOD0<WU> bit on each slave controller t

c. The master controller transmits one-frame data i
slave controllers. The MSB (Bit8) <TB8> is set to 1.



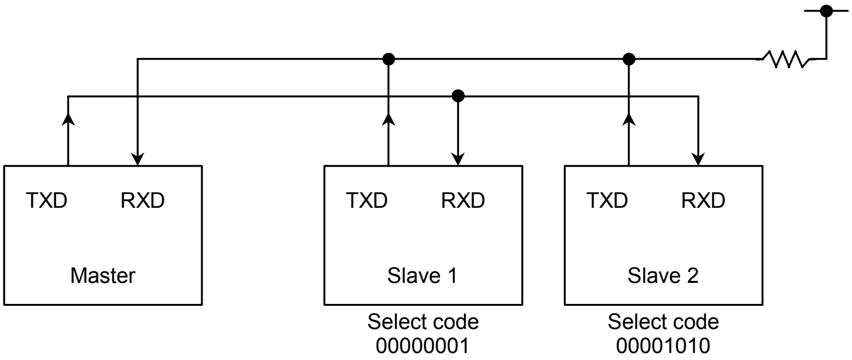
d. Each slave controller receives the above frame. Ea
code against its own select code. The controller whose code matches clears
0.

e. The master controller transmits data to the sp
SC0MOD<WU> bit is cleared to 0. The MSB (Bit8) <TB8>



f. The other slave controllers (whose <WU> bits remain at 1) ignore the r
because their MSBs (Bit8 or <RB8>) are set to 0, disabling IN
The slave controller (WU bit = 0) can transmit data to the master c
possible to indicate the end of data receiving t
transmission.

Example: To link two slave controllers serially wi
internal clock f_{SYS} as the transfer clock.



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

- Setting the master controller

Main		
PCCR	← – – – – – – 0 1	Set PC0 and PC1 to function as the TXD0 and RXD0 pins respectively.
PCFC	← – – – – – – X 1	
INTES0	← 1 1 0 0 1 1 0 1	Enable the INTTX0 interrupt and set it to interrupt level 4. Enable the INTRX0 interrupt and set it to interrupt level 5.
SC0MOD0	← 1 0 1 0 1 1 1 0	Set f_{SYS} as the transmission clock for 9-bit UART mode.
SC0BUF	← 0 0 0 0 0 0 0 1	Set the select code for slave controller 1.
INTTX0 interrupt		
SC0MOD0	← 0 – – – – – – –	Set TB8 to 0.
SC0BUF	← * * * * * * * *	Set data for transmission.

- Setting the slave controller

Main		
PCCR	← – – – – – – 0 1	Select PC1 and PC0 to function as the RXD0 and TXD0 pins respectively (Open-drain output).
PCFC	← – – – – – – X 1	
PCODE	← X X X X – X X 1	
INTES0	← 1 1 0 1 1 1 1 0	Enable INTRX0 and INTTX0.
SC0MOD0	← 0 0 1 1 1 1 1 0	Set <WU> to 1 in 9-bit UART transmission mode using f_{SYS} as the transfer clock.

INTRX0 interrupt
Acc ← SC0BUF
if Acc = select code
Then SC0MOD0 ← – – – 0 – – – – Clear <WU> to 0.

3.9.5 Support for IrDA

SIO0 includes support for the IrDA 1.0 infrared data communication spec
Figure 3.9.30 shows the block diagram.

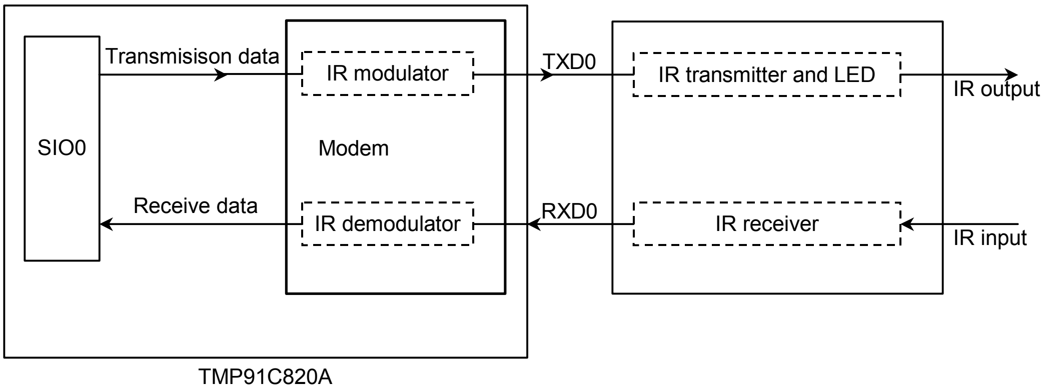


Figure 3.9.30 Block Diagram

(1) Modulation of the transmission data

When the transmit data is 0, the modem outputs 1 to TXD0 pin with eit
1/16 times for width of baud-rate. The pulse width is selected by the SIR
When the transmit data is 1, the modem outputs 0.

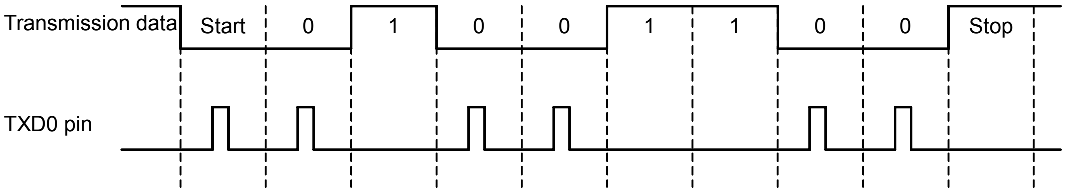


Figure 3.9.31 Transmission Example

(2) Demodulation of the receive data

When the receive data is the effective width of pul
Otherwise the modem outputs 1 to SIO0. The effecti
SIRCR<SIRWD3:0>.

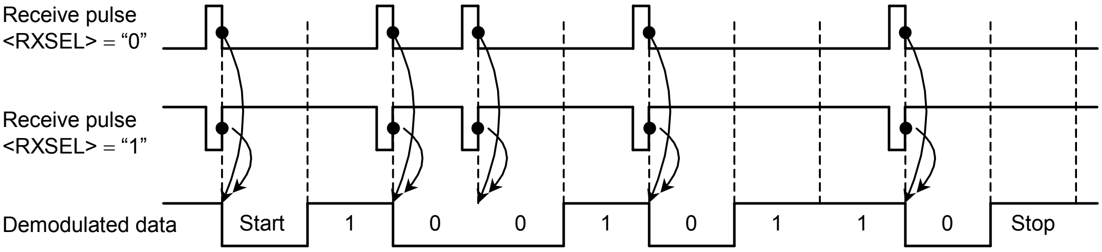


Figure 3.9.32 Receiving Example

(3) Data format

The data format is fixed as follows:

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

(4) SFR

Figure 3.9.33 shows the control register SIRCR. Se stopping. 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 trans

↓

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 s to SC0MOD0<SC1:0>. To use another source (TA0TR and SCLK0 input) are not allowed.

2) As the IrDA 1.0 physical layer specification, the data transf infra-red pulse width is specified.

The IrDA 1.0 specification is defined in Table 3.9

Table 3.9.5 Baud Rate and Pulse Width Specifications

Baud Rate	Modulation	Rate Tolerance (% of rate)	Pulse Width (Minimum)	Pulse Width (Typical)	Pulse width (Maximum)
2.4 kbps	RZI	±0.87	1.41 μs	78.13 μs	88.55 μs
9.6 kbps	RZI	±0.87	1.41 μs	19.53 μs	22.13 μs
19.2 kbps	RZI	±0.87	1.41 μs	9.77 μs	11.07 μs
38.4 kbps	RZI	±0.87	1.41 μs	4.88 μs	5.96 μs
57.6 kbps	RZI	±0.87	1.41 μs	3.26 μs	4.34 μs
115.2 kbps	RZI	±0.87	1.41 μs	1.63 μs	2.23 μs

The pulse width is defined either 1.6 μs or T × 3/16 (1.6 μs is equal to T × 3/16 pulse width when baud rate is 115.2 kbps).

The TMP91C820A has the function selects the pulse 3/16 or 1/16. But 1/16 pulse width can be used when the baud rate is e than 38.4 kbps.

As the same reason, +(16 – K)/16 division function in the bau 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 functi be used.

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.

		7	6	5	4	3	2	1	0
SIRCR (0207H)	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 × (Value + 1) + 100 ns Can be set: 1 to 4 Can not be set: 0, 15			

Select receive pulse width

Formula: Effective pulse width ≥ 2x × (Value + 1) + 100 ns

$x = 1/f_{FPH}$

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

Receive operation

0	Disabled
1	Enabled

Transmit operation

0	Disabled
1	Enabled

Select transmit pulse width

0	3/16
1	1/16

Note: If pulse width complying with the IrDA 1.0 standard (1.6 μs 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 3.9.33 IrDA Control Register

3.10 Serial Bus Interface (SBI)

The TMP91C820A has a one-channel serial bus interfa

synchronous 8-bit SIO mode and an I^{2}C bus mode

The serial bus interface is connected to an external device through P71 (SDA)

in the I^{2}C bus mode; and through P70 (SCK), P71 (SO), P72 (SI) in the clocked

8-bit SIO mode.

Each pin is specified as follows.

	P7ODE<ODE72, ODE71>	P7CR<P72C, P71C, P70C>	P7FC<P72F, P71F, P70F>
I^{2}C bus mode	11	11X	11X
Clocked synchronous 8-bit SIO mode	XX	011 010	111

X: Don't care

3.10.1 Configuration

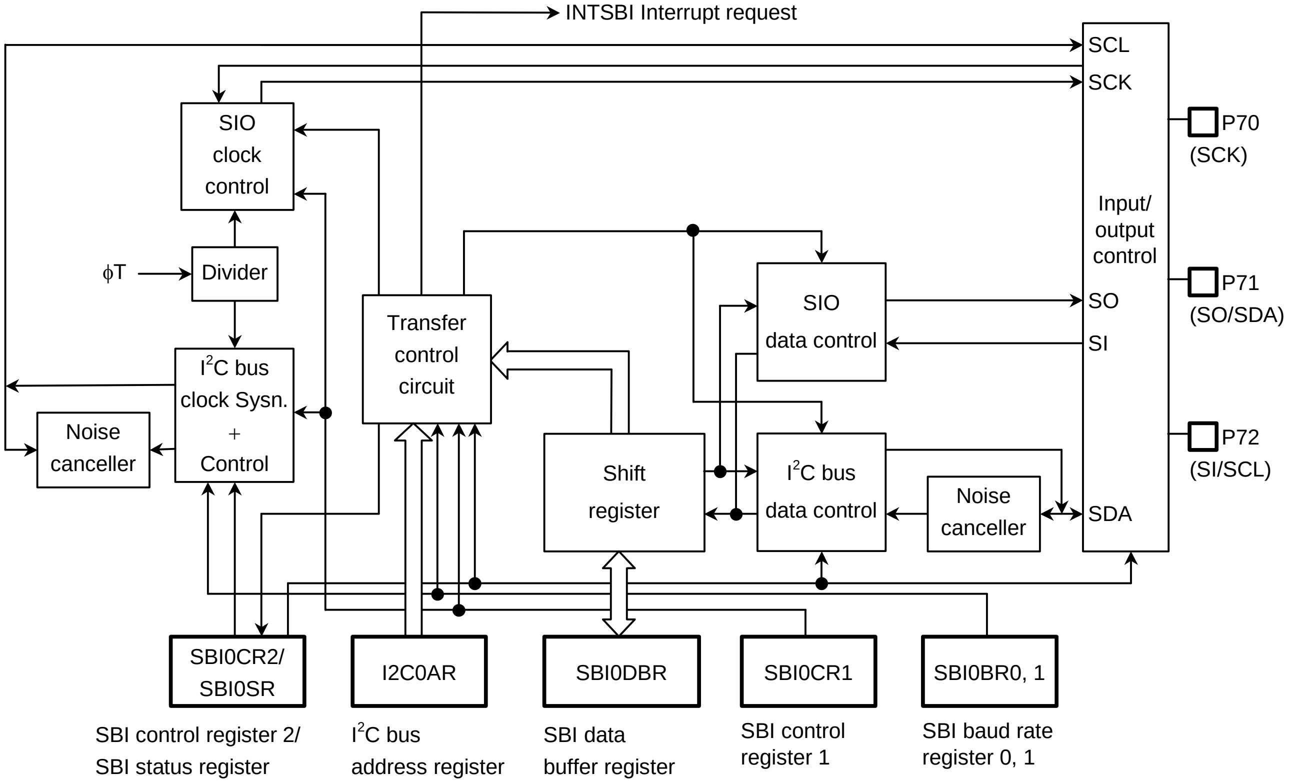


Figure 3.10.1 Serial Bus Interface (SBI)

3.10.2 Serial Bus Interface (SBI) Control

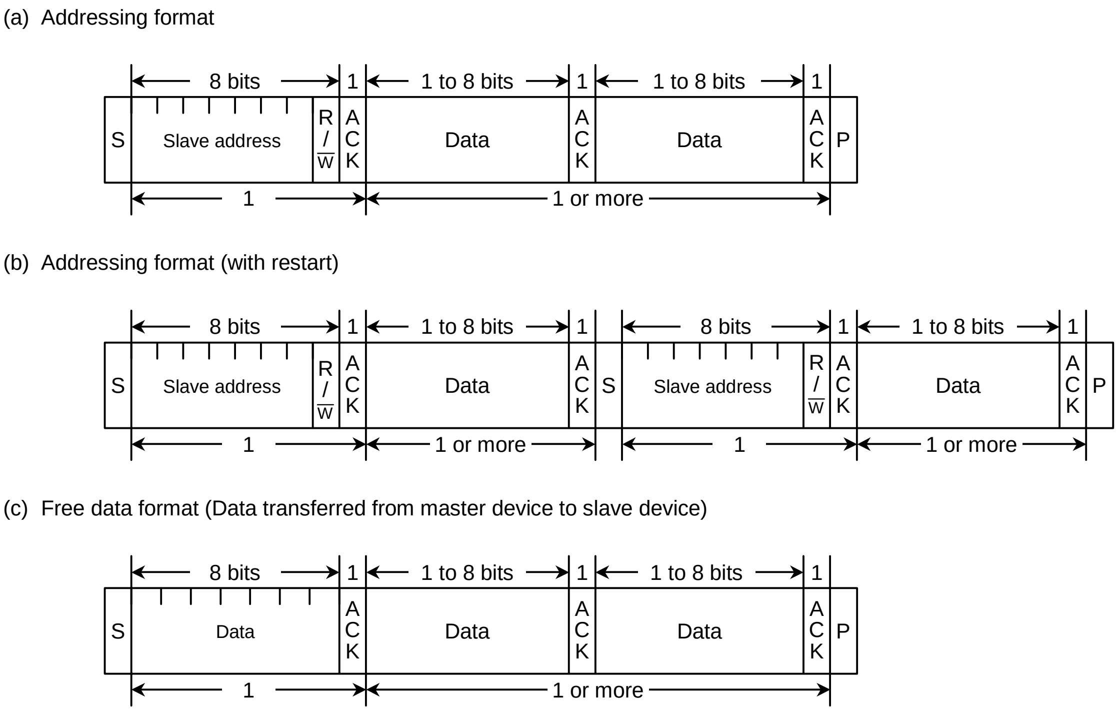
The following registers are used to control the ser operation status.

- Serial bus interface control register 1 (SBI0CR1
- Serial bus interface control register 2 (SBI0CR2
- Serial bus interface data buffer register (SBI0D
- I2C bus address register (I2C0AR)
- Serial bus interface status register (SBI0SR)
- Serial bus interface baud rate register 0 (SBI0B
- Serial bus interface baud rate register 1 (SBI0B

The above registers differ depending on a mode to be u
Refer to section 3.10.4 "I^{2}C Bus Mode Control" and 3.10.7 "Clocke
SIO Mode Control".

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

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


(a) Addressing format

(b) Addressing format (with restart)

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

S: Start condition

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

ACK: Acknowledge bit

P: Stop condition

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

3.10.4 I^{2}C Bus Mode Control

The following registers are used to control and moni the serial bus interface (SBI) in the I^{2}C bus mode.

Seirial Bus Interface Conrol Register 1

		7	6	5	4	3	2	1	0
SBI0CR1 (0240H)	Bit symbol	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0/ SWRMON
	Read/Write	W			R/W		W		R/W
	After Reset	0	0	0	0		0	0	0/1 (Note 3)
Prohibit read-modify-write	Function	Number of transferred bits (Note 1)			Acknowledge mode specification 0: Not generate 1: Generate		Internal serial clock selection and software reset monitor (Note 2)		

Internal serial clock selection <SCK2:0> at write

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

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

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

Software reset state monitor <SWRMON> at read

0	During software reset
1	Initial data

Acknowledge mode specification

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

Number of bits transferred

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

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

Note 2: For the frequency of the SCL line clock, see 3.10.5 (3) “Serial clock”.

Note 3: Initial data of SCK0 is “0”, SWRMON is “1”.

Note 4:This I^{2}C bus circuit dose not support high-speed mode, it supports standard mode only.

Figure 3.10.3 Registers for the I^{2}C Bus Mode

Serial Bus Interface Control Register 2

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

Serial bus interface operating mode selection (Note 2)

00	Port mode (Serial bus interface output disabled)
01	Clocked synchronous 8-bit SIO mode
10	I^{2}C bus mode
11	(Reserved)

INTSBI interrupt request

0	–
1	Cancel interrupt request

Start/stop condition generation

0	Generates the stop condition
1	Generates the start condition

Transmitter/receiver selection

0	Receiver
1	Transmitter

Master/slave selection

0	Slave
1	Master

Note 1: Reading this register function as SBI0SR register.

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

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

Serial Bus Interface Status Register

SBI0SR
(0243H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
Read/Write	R							
After reset	0	0	0	1	0	0	0	0
Function	Master/ slave status monitor	Transmitter/ receiver status monitor	I^{2}C bus status monitor	INTSBI interrupt request monitor	Arbitration lost detection monitor 0: Undetected 1: Detected	Slave address match detection monitor 0: Undetected 1: Detected	GENERAL CALL detection monitor 0: Undetected 1: Detected	Last received bit monitor 0: 0 1: 1

Last received bit monitor

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

GENERAL CALL detection monitor

0	Undetected
1	GENERAL CALL detected

Slave address match detection monitor

0	Undetected
1	Slave address match or GENERAL CALL detected

Arbitration lost detection monitor

0	–
1	Arbitration lost

INTSBI interrupt request monitor

0	Interrupt requested
1	Interrupt canceled

I^{2}C bus status monitor

0	Free
1	Busy

Transmitter/receiver status monitor

0	Receiver
1	Transmitter

Master/slave status monitor

0	Slave
1	Master

Note: Writing in this register functions as SBI0CR2.

Figure 3.10.5 Registers for the I^{2}C Bus Mode

Serial Bus Interface Baud Rate Regster 0

SBI0BR0
(0244H)

Prohibit read-modify-write

		7	6	5	4	3	2	1	0
SBI0BR0 (0244H)	Bit symbol	–	I2SBI0						
	Read/Write	W	R/W						
	After reset	0	0						
Prohibit read-modify-write	Function	Write "0".	IDLE2 0: Stop 1: Run						

Operation during IDLE 2 mode

0	Stop
1	Operation

Serial Bus Interface Baud Rate Register 1

		7	6	5	4	3	2	1	0
SBI0BR1 (0245H)	Bit symbol	P4EN	–						
	Read/Write	W	W						
	After reset	0	0						
Prohibit read-modify-write	Function	Internal clock 0: Stop 1: Operate	Write "0".						

Baud rate clock control

0	Stop
1	Operate

Sirial Bus Interface Data Buffer Register

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

Note 1: When writing transmitted data, start from the MSB (Bit7). Receiving data is placed from LSB (Bit0).

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

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

I^{2}C Bus Address Register

		7	6	5	4	3	2	1	0
I2C0AR (0242H)	Bit symbol	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
Prohibit read-modify-write	Function	Slave address selection for when device is operating as slave device							Address recognition mode specification

Address recognition mode specification

0	Slave address recognition
1	Non slave address recognition

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

3.10.5 Control in I^{2}C Bus Mode

(1) Acknowledge mode specification

Set the SBI0CR1<ACK> to 1 for operation in the
TMP91C820A generates an additional clock pulse fo
operating in master mode. In the transmitter mode c
SDA pin is released in order to receive the acknowle
receiver mode during the clock pulse cycle, the SDA
generate the acknowledge signal.

Clear the <ACK> to 0 for operation in the non-acknow
does not generate a clock pulse for the acknowledg
master mode.

(2) Number of transfer bits

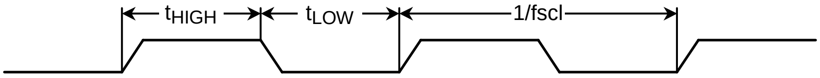
The SBI0CR1<BC2:0> is used to select the number of bits for next trans
receiving data.

Since the <BC2:0> is cleared to 000 as a start co
direction bit transmission are executed in 8 bits
retains a specified value.

(3) Serial clock

a. Clock source

The SBI0CR1<SCK2:0> is used to select a maxim
outputted on the SCL pin in master mode. Set the b
calculated according to the formula below, to meet I^{2}C bus
such as the smallest pulse width of t_{LOW}.



$t_{LOW} = 2^{n-1}/f_{SBI}$

$t_{HIGH} = 2^{n-1}/f_{SBI} + 8/f_{SBI}$

$$fscl = 1/(t_{LOW} + t_{HIGH}) = \frac{f_{SBI}}{2^n + 8}$$

SBI0CR1<SCK2:0>	n
000	5
001	6
010	7
011	8
100	9
101	10
110	11

Note 1: f_{SBI} shows f_{FPH}.
Note 2: It's prohibit to use fc/16 prescaler clock when using SBI block.
(I^{2}C bus and clock synchronous.)

Figure 3.10.7 Clock Source

b. Clock synchronization

In the I2C bus mode, in order to wired-AND a bus, a ma
down a clock line to low level, in the first place,
master device which generates a high-level clock
high-level clock pulse needs to detect the situa
procedure.

The TMP91C820A has a clock synchronization f
transfer even when more than one master exists on

The example explains the clock synchronizatio
simultaneously exist on a bus.

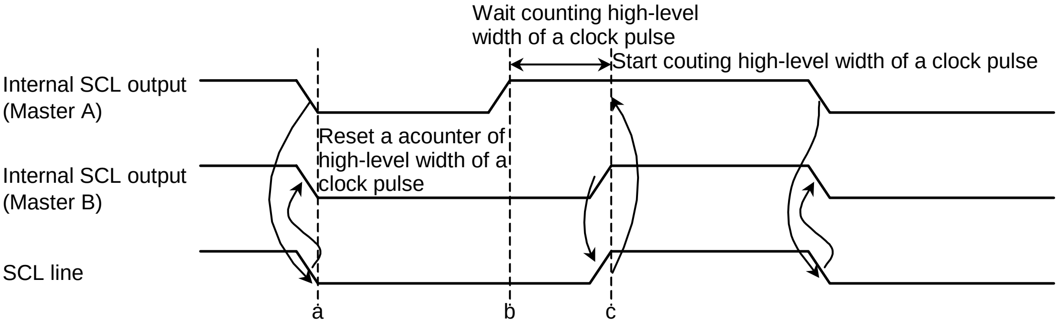

Wait counting high-level width of a clock pulse
Start couting high-level width of a clock pulse
Internal SCL output (Master A)
Reset a acounter of high-level width of a clock pulse
Internal SCL output (Master B)
SCL line
a
b
c

Figure 3.10.8 Clock Synchronization

As master A pulls down the internal SCL output t
SCL line of the bus becomes the low level. After de
resets a counter of high-level width of an own clock pulse and
SCL output to the low level.

Master A finishes counting low-level width of a
sets the internal SCL output to the high level. Si
of the bus at the low level, master A wait for coun
clock pulse. After master B finishes counting low
at point c and master A detects the SCL line of th
starts counting high level of an own clock pulse
determined by the master device with the shorte
master device with the longest low-level width f
connected to the bus.

(4) Slave address and address recognition mode specification

When the TMP91C820A is used as a slave device, set t
<ALS> to the I2C0AR<ALS> to 0 for the address recogn

(5) Master/slave selection

Set the SBI0CR2<MST> to 1 for operating the TMP91
Clear the SBI0CR2<MST> to 0 for operation as a slave device. The <MST> is
0 by the hardware after a stop condition on the bus is

(6) Transmitter/receiver selection

Set the SBI0CR2<TRX> to 1 for operating the TMP91C820A as a transmit
the <TRX> to 0 for operation as a receiver. When dat
transferred in slave mode, when a slave address wit
or a GENERAL CALL is received (All 8-bit data are 0 after a start con
<TRX> is set to 1 by the hardware if the direction ($\overline{W}$) sent from the mas
device is 1, and is cleared to 0 by the hardware if th
an acknowledge signal is returned from the slave de
the hardware if a transmitted direction bit is 1, and is set to 1 by the hardwa
When an acknowledge signal is not returned, the current condition is m

The <TRX> is cleared to 0 by the hardware after a st ^{2}C bus is
detected or arbitration is lost.

(7) Start/stop condition generation

When the SBI0SR<BB> is 0, slave address and direction bit whic
SBI0DBR are output on a bus after generating a star
SBI0CR2 <MST, TRX, BB, PIN> to 1. It is necessary to transmitted data a
buffer register (SBI0DBR) and set 1 to <ACK> before

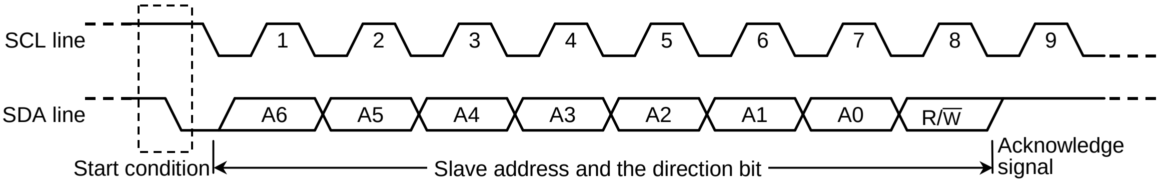


Figure 3.10.9 Start Condition Generation and Slave Address Generation

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

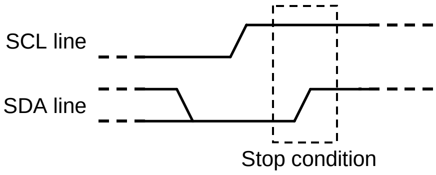


Figure 3.10.10 Stop Condition Generation

The state of the bus can be ascertained by reading
SBI0SR<BB> will be set to 1 if a start condition has b
be cleared to 0 if a stop condition has been detected

And about generation of stop condition in master m
point. Please refer 3.10.6 (4) “Stop condition generation”.

(8) Interrupt service requests and interrupt cancel

When a serial bus interface interrupt request (INT is cleared to 0. During the time that the SBI0CR2<P down to the low level.

The <PIN> is cleared to 0 when a 1 word data is transmitted or rec writing/reading data to/from SBI0DBR sets the <PIN

The time from the <PIN> being set to 1 until the SCL Low.
In the address recognition mode (<ALS> = 0), <PIN> is cleared to 0 when slave address is the same as the value set at the I2C0 received (All 8-bit data are 0) after a start condition. Although SBI0CR set to 1 by the program, the <PIN> is not clear it to 0

(9) Serial bus interface operation mode selection

SBI0CR2<SBIM1:0> is used to specify the serial bu SBI0CR2<SBIM1:0> to 10 when the device is used in I^2C bus mode after confirming pin condition of serial bus interface to

Switch a mode to port after confirming a bus is free

(10) Arbitration lost detection monitor

Since more than one master device can exist I^2C bus simult mode, a bus arbitration procedure has been impleme integrity of transferred data.

Data on the SDA line is used for I^2C bus arbitration.

The following shows an example of a bus arbitrati devices exist simultaneously on the bus. Master A a until point a. After master A outputs L and master B wire-AND and the SDA line is pulled down to the low-l line of the bus is pulled up at point b, the slave dev that is, data in master A. A data transmitted from ma in master B is called "ARBITRATION LOST". Master B d releases the internal SDA output in order not to af masters with arbitration. When more than one master sends the same data at the first word, arbitration occurs continuously after the second word.

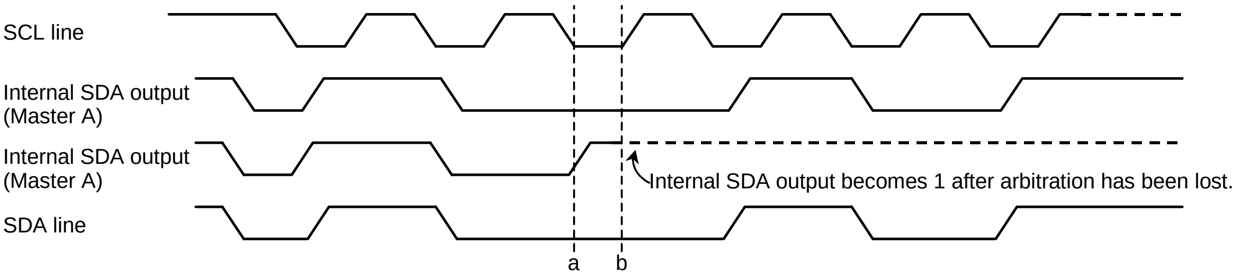

SCL line
Internal SDA output (Master A)
Internal SDA output (Master A)
SDA line
Internal SDA output becomes 1 after arbitration has been lost.
a
b

Figure 3.10.11 Arbitration Lost

The TMP91C820A compares the levels on the bus's S internal SDA output on the rising edge of the SCL l arbitration is lost and SBI0SR<AL> is set to 1.

When SBI0SR<AL> is set to 1, SBI0SR<MST, TRX> are cleared to 00 an is switched to slave receiver mode. Thus, clock output is stopped in data t setting <AL> = "1".

SBI0SR<AL> is cleared to 0 when data is written to or read from SBI0D data is written to SBI0CR2.

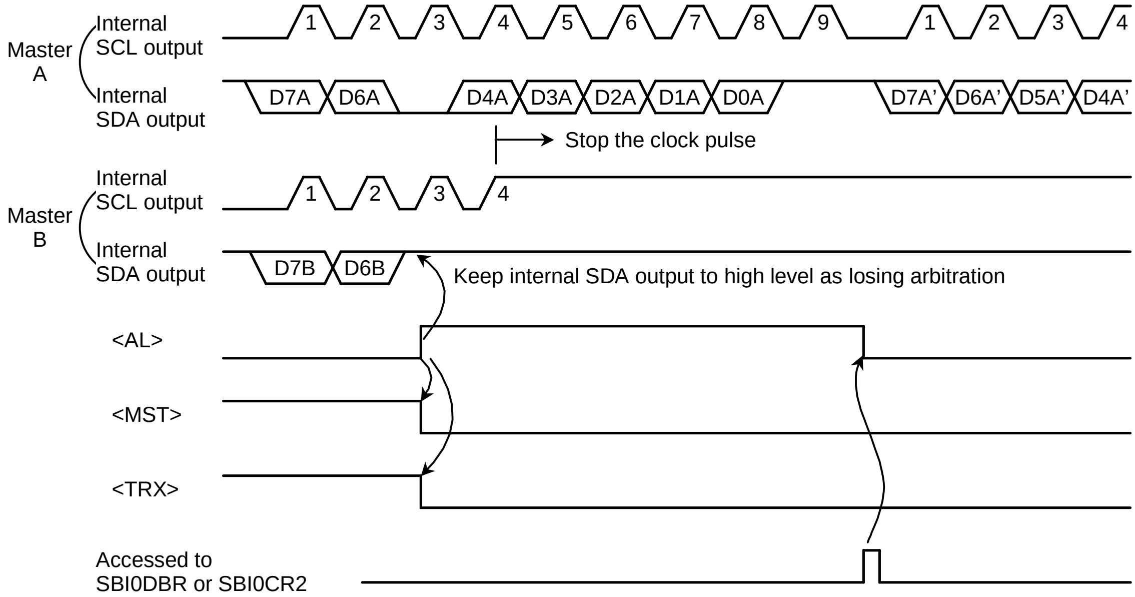


Figure 3.10.12 Example of when TMP91C820A is a Master Device B (D7A = D7B, D6A = D6B)

(11) Slave address match detection monitor

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

(12) GENERAL CALL detection monitor

SBI0SR<AD0> is set to 1 in slave mode, when a GENER 8-bit received data is 0, after a start condition). SBI0SR<AD0> is clear start condition or stop condition is detected on th

(13) Last received bit monitor

The SDA line value stored at the rising edge of SBI0SR<LRB>. In the acknowledge mode, immediately after an INTSB request is generated, an acknowledge signal is read by reading the cont SBI0SR<LRB>.

(14) Software reset function

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

An internal reset signal pulse can be generated by
10 and 01. This initializes the SBI circuit int
SBI0CR2<SBIM1:0>) registers and status registers are initialized as

SBI0CR1<SWRMON> is automatically set to 1 after the SBI circui
initialized.

(15) Serial bus interface data buffer register (SBI0D

The received data can be read and transfer data can be written by
writing the SBI0DBR.

In the master mode, after the start condition is ge
direction bit are set in this register.

(16) I2CBUS address register (I2C0AR)

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

The slave address output from the master device
I2C0AR<ALS> to 0. The data format is the addressing
is not recognized at the <ALS> = 1, the data format is the free data

(17) Baud rate register (SBI0BR1)

Write 1 to SBI0BR1<P4EN> before operation commenc

(18) Setting register for IDLE2 mode operation (SBI0E

SBI0BR0<I2SBI0> is the register setting operat
Therefor, setting <I2SBI0> is necessary before the

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

(1) Device initialization

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

Set a slave address <SA6:0> and the <ALS> (<ALS> = 0 when an addressing f
to the I2C0AR.

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

(2) Start condition and slave address generation

a. Master mode

In the master mode, the start condition and the s
follows.

Check a bus free status (when <BB> = 0).

Set the SBI0CR1<ACK> to 1 (Acknowledge mode) and specify a sla
and a direction bit to be transmitted to the SBI0D

When SBI0CR2<BB> = 0, the start condition are generate
SBI0CR2<MST, TRX, BB, PIN>. Subsequently to the
are output from the SCL pin. While eight clocks ar
the direction bit which are set to the SBI0DBR. A
released and the acknowledge signal is received

An INTS2 interrupt request occurs at the falli
<PIN> is cleared to 0. In the master mode, the SC
low-level while <PIN> is 0. When an interrupt request occurs,
changed according to the direction bit only wh
returned from the slave device.

b. Slave mode

In the slave mode, the start condition and the sl

After the start condition is received from the m
are output from the SCL pin, the slave address an
output from the master device are received.

When a GENERAL CALL or the same address as the s
I2C0AR is received, the SDA line is pulled down to the low level at the
and the acknowledge signal is output.

An INTSBI interrupt request occurs on the falli
<PIN> is cleared to 0. In slave mode the SCL line i
while the <PIN> = 0.

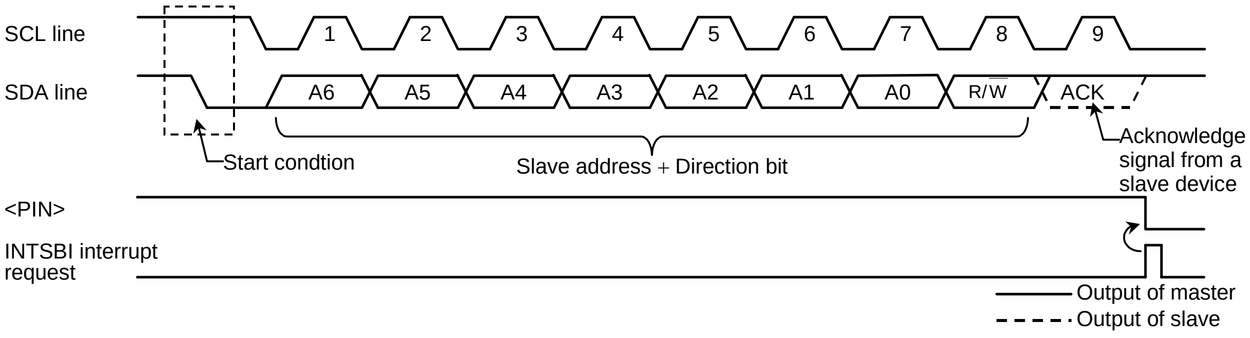


Figure 3.10.13 Start Condition Generation and Slave Address Transfer

(3) 1-word data transfer

Check the <MST> by the INTSBI interrupt process af
completed, and determine whether the mode is a mast

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

Check the <TRX> and determine whether the mode i

When the <TRX> = 1 (Transmitter mode)

Check the <LRB>. When <LRB> is 1, a receiver d
Implement the process to generate a stop condit
terminate data transfer.

When the <LRB> is 0, the receiver requests new data. When t
transmitted data is 8 bits, write the transmitted data to SBI0DBR. Whe
transmitted data is other than 8 bits, set the <B
write the transmitted data to SBI0DBR. After written the data, <PIN
a serial clock pulse is generated for transferring a new 1 word of dat
SCL pin, and then the one-word data is transmitte
an INTSBI interrupt request occurs. The <PIN> becomes 0 and the SC
pulled down to the low level. If the data to be transferred is more than
length, repeat the procedure from the <LRB> chec

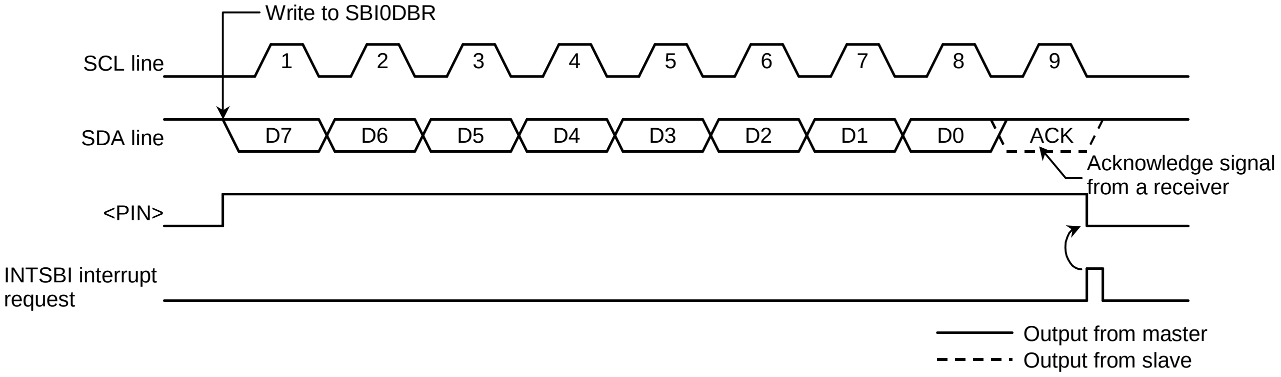


Figure 3.10.14 Example in which <BC2:0> = 000 and <ACK> = 1 in Transmitter Mode

When the <TRX> = 0 (Receiver mode)

When the next transmitted data is other than 8 b
<ACK> to 1 and read the received data from SBI0DBR to release the
(Data which is read immediately after a slave add
the data is read, <PIN> becomes 1.

Serial clock pulse for transferring new 1 word of data is defined on SCL and ou
level from SDA pin with acknowledge timing.

An INTSBI interrupt request then occurs and the
TMP91C820A pulls SCL pin to the low level. The TMP9
a clock pulse for 1 word of data transfer and the
that received data is read from the SBI0DBR.

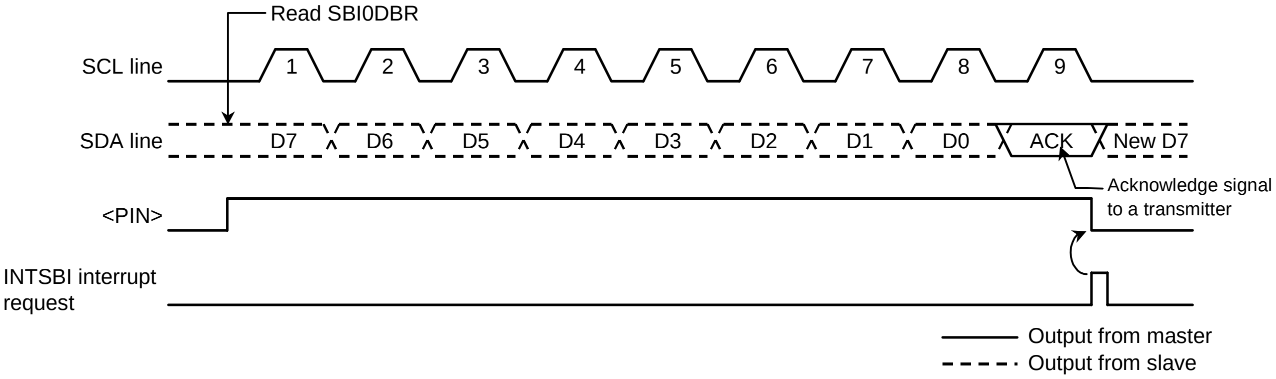


Figure 3.10.15 Example of when <BC2:0> = 000, <ACK> = 1 in Receiver Mode

In order to terminate the transmission of data t
0 before reading data which is 1 word before the la
data word does not generate a clock pulse as the a
data has been transmitted and an interrupt requ
<BC2:0> to 001 and read the data. The TMP91C820A g
1-bit data transfer. Since the master device is a
remains high. The transmitter interprets the hi
receiver indicates to the transmitter that data

After the one data bit has been received and a
generated, the TMP91C820A generates a stop condition (See Sectio
and terminates data transfer.

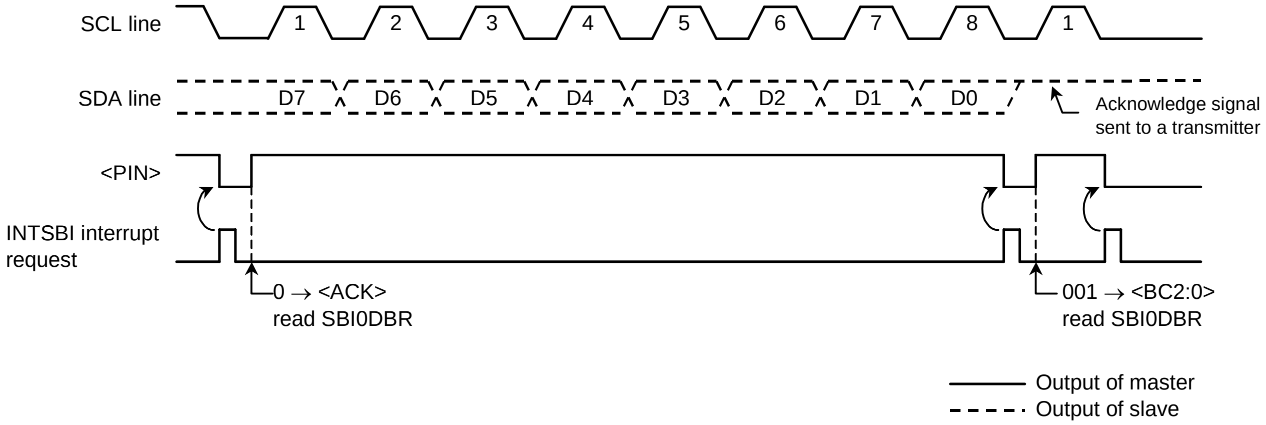


Figure 3.10.16 Termination of Data Transfer in Master Receiver Mode

b. If <MST> = 0 (Slave mode)

In the slave mode the TMP91C820A operates either slave mode after losing arbitration.

In the slave mode, an INTSBI interrupt request o receives a slave address or a GENERAL CALL from th GENERAL CALL is received and data transfer is co received address. In the master mode, the TMP91C if it losing arbitration. An INTSBI interrupt request occurs when transfer terminates after losing arbitration. When an INTSBI interru occurs the <PIN> is cleared to 0 and the SCL pin is pulled down to the Either reading/writing from/to the SBI0DBR or se the SCL pin after taking t_{Low} time.

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

Table 3.10.1 Operation in the Slave Mode

<TRX>	<AL>	<AAS>	<AD0>	Conditions	Process
1	1	1	0	The TMP91C820A loses arbitration when transmitting a slave address and receives a slave address for which the value of the direction bit sent from another master is 1.	Set the number of bits a word in <BC2:0> and write the transmitted data to SBI0DBR.
	0	1	0	In salve receiver mode the TMP91C820A receives a slave address for which the value of the direction bit sent from the master is 1.	
		0	0	In salve transmitter mode a single word of is transmitted. Set <BC2:0> to the number of bits in a word.	Check the <LRB> setting. If <LRB> is set to 1, set <PIN> to 1 since the receiver win no request the data which follows. Then, clear <TRX> to 0 to release the bus. If <LRB> is cleared to 0 of and write the transmitted data to SBI0DBR since the receiver requests next data.
0	1	1	1/0	The TMP91C820A loses arbitration when transmitting a slave address and receives a slave address or GENERAL CALL for which the value of the direction bit sent from another master is 0.	Read the SBI0DBR for setting the <PIN> to 1 (Reading dummy data) or set the <PIN> to 1.
		0	0	The TMP91C820A loses arbitration when transmitting a slave address or data and terminates word data transfer.	
	0	1	1/0	In slave receiver mode the TMP91C820A receives a slave address or GENERAL CALL for which the value of the direction bit sent from the master is 0.	
		0	1/0	In slave receiver mode the TMP91C820A terminates receiving word data.	Set <BC2:0> to the number of bits in a word and read the received data from SBI0DBR.

(4) Stop condition generation

When the SBI0SR<BB> is "1", the sequence of generating a stop condit
by setting "111" to the SBI0CR2<MST, TRX, PIN> and "
not modify the contents of the SBI0CR2<MST, TRX, PI
generated on a bus. When a SCL line of bus is pulled down by other device
generates a stop condition after SCL line is released and the SDA becom

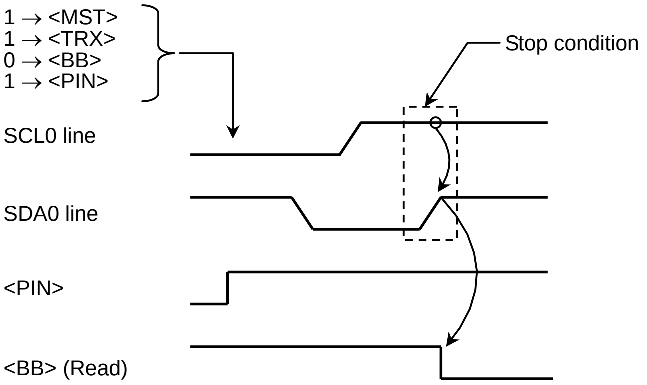


Figure 3.10.17 Stop Condition Generation (Single master)

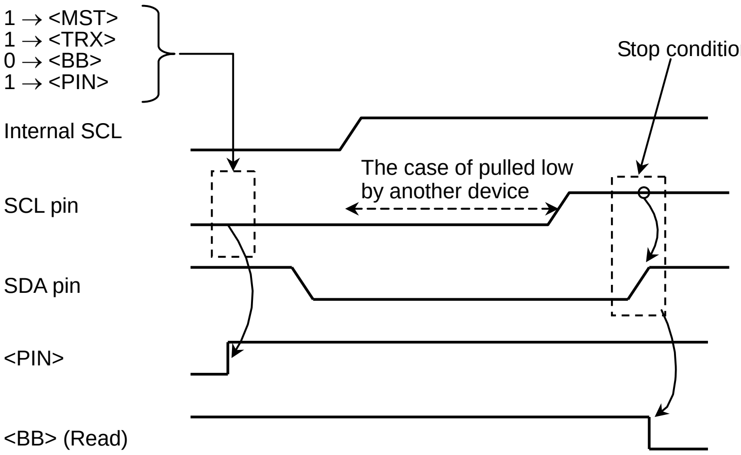


Figure 3.10.18 Stop Condition Generation (Multi master)

(5) Restart

Restart is used to change the direction of data transfer between a mast a slave device during transferring data. The follow TMP91C820A is in the master mode.

Clear 0 to the SBI0CR2<MST, TRX, BB> and set 1 to <PIN> and rele The SDA line remains the high level and the SCL pin is is not generated on a bus, a bus is assumed to be in a busy state from ot Check the SBI0SR<BB> until it becomes 0 to check tha Check the <LRB> until it becomes 1 to check that the down to the low level by other devices. After confirming that a bus stays free generate a start condition with procedure 3.10.6 (2

In order to meet setup time when restarting, take 4.7 μs of waiting time software from the time of restarting to confirm that a bus is free unti generate the start condition.

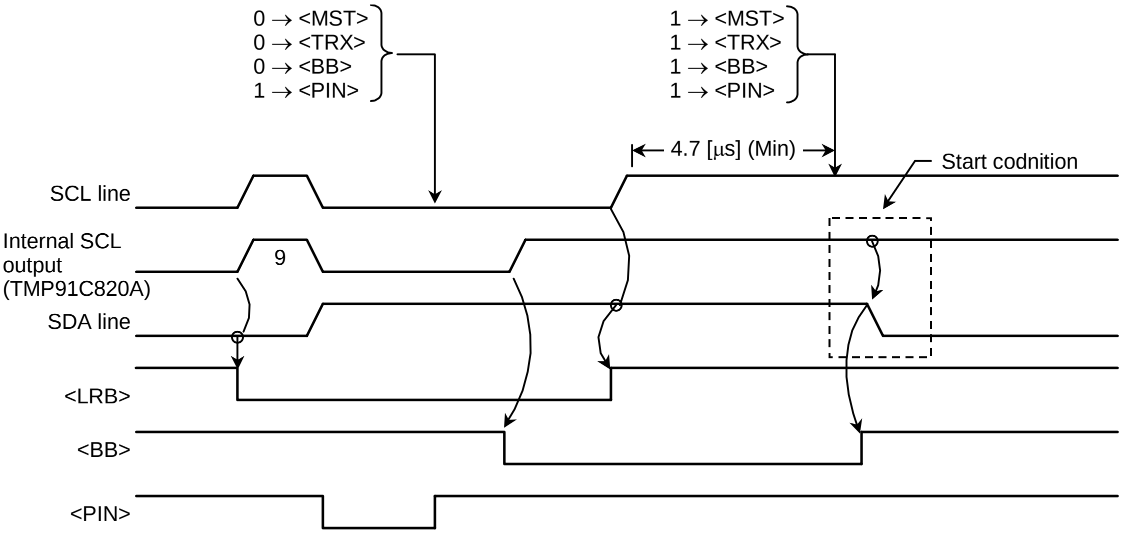


Figure 3.10.19 Timing Diagram for TMP91C820A Restart

3.10.7 Clocked Synchronous 8-Bit SIO Mode control

The following registers are used to control and mon

serial bus interface (SBI) is being operated in clock

Serial Bus Interface Control Register 1

		7	6	5	4	3	2	1	0
SBI0CR1 (0240H)	Bit symbol	SIOS	SIOINH	SIOM1	SIOM0		SCK2	SCK1	SCK0
	Read/Write	W					W		W
	After reset	0	0	0	0		0	0	0
Prohibit read-modify-write	Function	Transfer start 0: Stop 1: Start	Continue/ abort transfer 0: Continue transfer 1: Abort transfer	Transfer mode select 00: Transmit mode 01: (Reserved) 10: Transmit/receive mode 11: Receive mode			Serial clock selection and reset monitor		

Serial clock selection <SCK2:0> at write

000	n = 4	1 MHz
001	n = 5	500 kHz
010	n = 6	250 kHz
011	n = 7	125 kHz
100	n = 8	62.5 kHz
101	n = 9	31.25 kHz
110	n = 10	1.625 kHz
111	–	External clock (Inputted from sck pin)

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

Transfer mode selection

00	8-bit transmit mode
01	(Reserved)
10	8-bit transmit/received mode
11	8-bit received mode

Continue/abort transfer

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

Transfer start/stop

0	Stopped
1	Started

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

Serial Bus Interface Data Buffer Register

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

Figure 3.10.20 Register for the SIO Mode (1/3)

Serial Bus Interface Control Register 2

		7	6	5	4	3	2	1	0
SBI0CR2 (0242H)	Bit symbol					SBIM1	SBIM0	–	–
	Read/Write					W		W	W
	After reset					0	0	0	0
Prohibit read-modify-write	Function					Serial bus interface operation mode selection 00: Port mode 01: SIO mode 10: I^2C bus mode 11: (Reserved)		(Note 2)	(Note 2)

Serial bus interface operation mode selection

00	Port mode (Serial bus interface output disabled)
01	Clocked synchronous 8-bit SIO mode
10	I^2C bus mode
11	(Reserved)

Note 1: Set the SBI0CR1<BC2:0> 000 before switching to a clocked synchronous 8-bit SIO mode.

Note 2: Please always write “00“ to SBICR2<1:0>.

Serial Bus Interface Status Register

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

Shift operation status monitor

0	Shift operation terminated
1	Shift operation in progress

Serial transfer operating status monitor

0	Transfer terminated
1	Transfer in progress

Figure 3.10.21 Registers for the SIO Mode (2/3)

Serial Bus Interface Baud Rate Register 0

		7	6	5	4	3	2	1	0
SBI0BR0 (0244H)	Bit symbol	–	I2SBI0						
	Read/Write	W	R/W						
Prohibit read-modify-write	After reset	0	0						
	Function	Always fixed to “0”.	IDLE2 0: STOP 1: RUN						

Operation in IDLE 2 mode

0	Stop
1	Operate

Serial Bus Interface Baud Rate Register 1

		7	6	5	4	3	2	1	0
SBI0BR1 (0245H)	Bit symbol	P4EN	–						
	Read/Write	W	W						
Prohibit read-modify-write	After reset	0	0						
	Function	Internal clock 0: Stop 1: Operate	Always fixed to “0”.						

Baud rate clock control

0	Stop
1	Operate

Figure 3.10.22 Registers for the SIO Mode (3/3)

(1) Serial Clock

a. Clock source

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

Internal clock

In internal clock mode one of seven frequencies
signal is output to the outside on the SCK pin. Th
transfer starts. When the device is writing (in
receive mode), data cannot follow the serial clock rate, so an aut
function is executed which automatically stops t
shift operation until reading or writing has bee

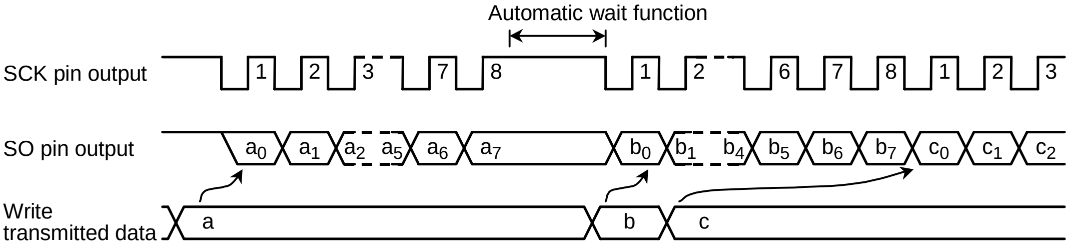


Figure 3.10.23 Automatic Wait Function

External clock (<SCK2:0> = 111)

An external clock input via the SCK pin is used a
ensure the integrity of shift operations, both t
pulse widths shown below must be maintained. The maximum data
frequency is 1 MHz (when fc = 16 MHz).

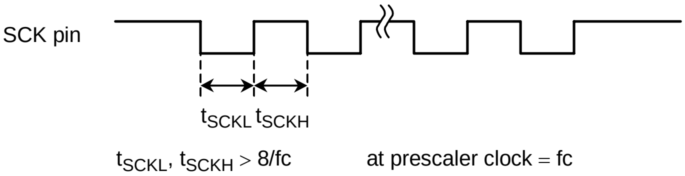


Figure 3.10.24 Maximum Data Transfer Frequency when External Clock Input Used

b. Shift edge

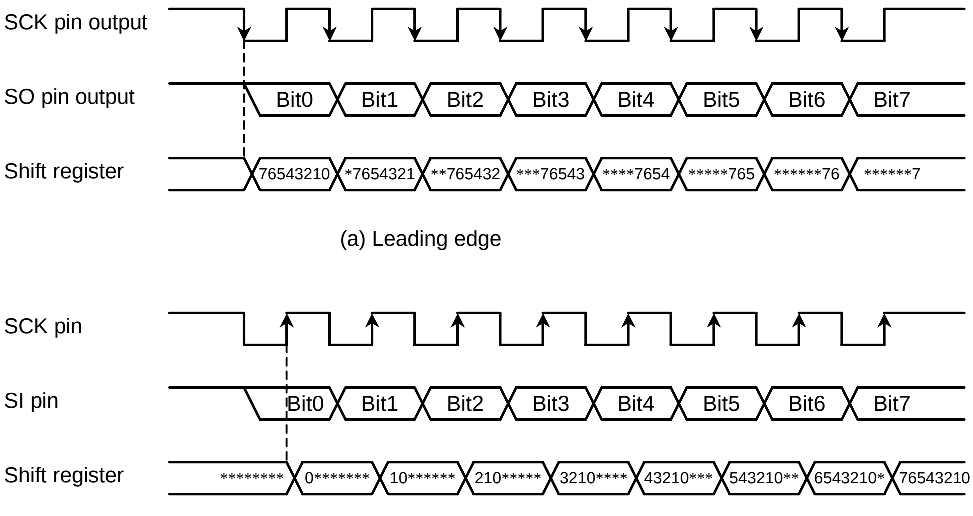
Data is transmitted on the leading edge of the cl
edge.

Leading edge shift

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

Trailing edge shift

Data is shifted on the trailing edge of the seria
SCK pin input/output).



(a) Leading edge

*: Don't care

(b) Trailing edge

Figure 3.10.25 Shift Edge

(2) Transfer modes

The SBI0CR1<SIOM1:0> is used to select a transmi mode.

a. 8-bit transmit mode

Set a control register to a transmit mode and w SBI0DBR.

After the transmit data is written, set the SBI0 transfer. The transmitted data is transferred from SBI0DBR to the shi and output to the SO pin in synchronized with the least significant bit (LSB). When the transmit data is transferre register, the SBI0DBR becomes empty. An INTSBI request is generated to request new data.

When the internal clock is used, the serial clo function will be initiated if new data is not load the specified 8-bit data is transmitted. When new transmit data is automatic-wait function is canceled.

When the external clock is used, data should be w data is shifted. The transfer speed is determined by the maximum d between the time when an interrupt request is gen is written to SBI0DBR by the interrupt service pr

When the transmit is started, after the SBI0SR<SIOF> goes 1 outpu SO pin holds final bit of the last data until fall

Transmitting data is ended by clearing the <SIOS> to 0 by the buff interrupt service program or setting the <SIOIN cleared, the transmitted mode ends when all data data is surely transmitted by the program, set the <SIOF> (Bit3 of SBI sensed. The SBI0SR<SIOF> is cleared to 0 when tra the <SIOINH> is set to 1, transmitting data stops. SBI0SR<SIOF> t

When an external clock is used, it is also necessary to clear SBI0SF before new data is shifted; otherwise, dummy data is transmitted an ends.

Example: Program to stop data transmission (when an exte

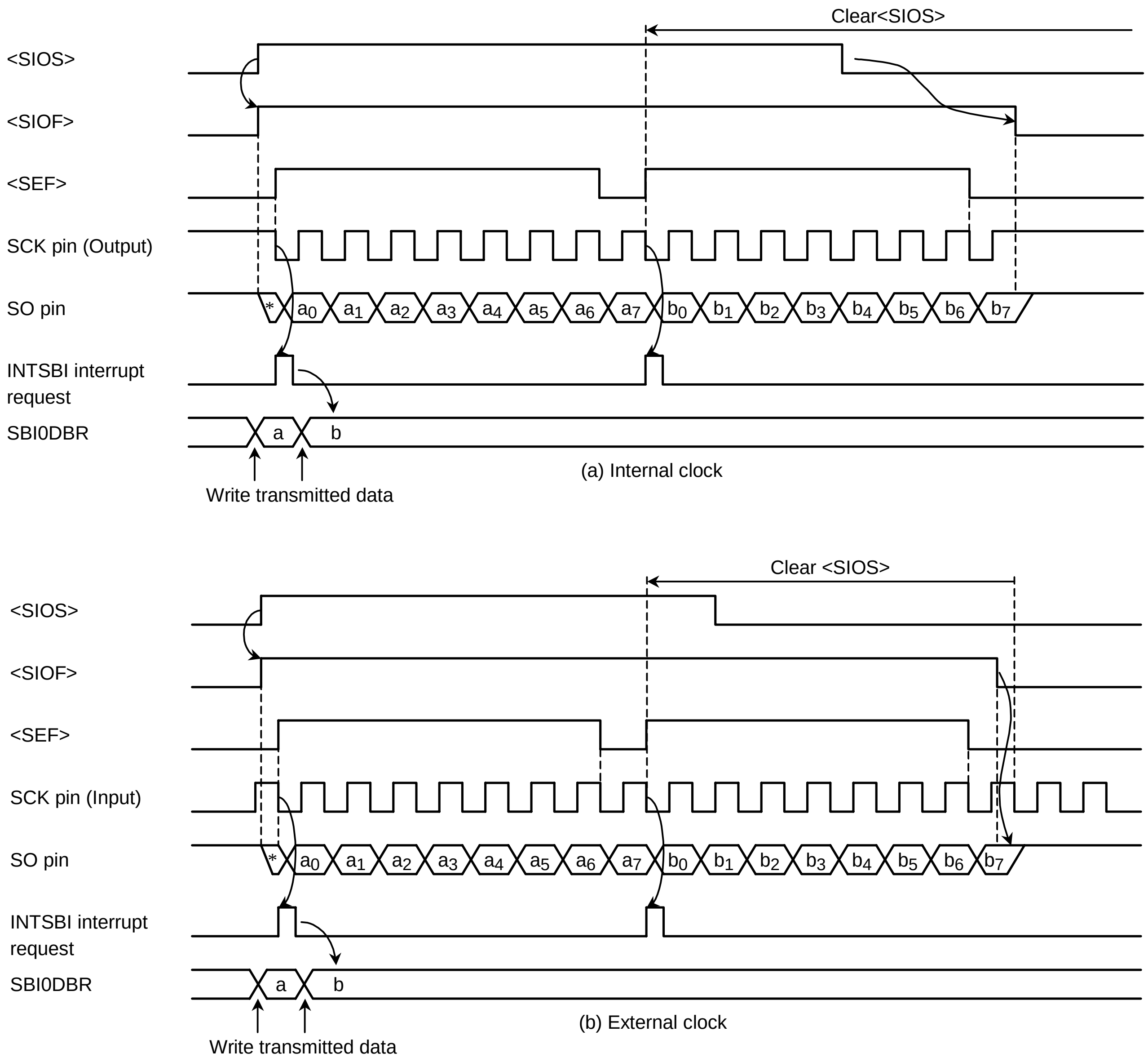


Figure 3.10.26 Transfer Mode

```
STEST1: BIT     SEF, (SBI0SR)   ; If <SEF> = 1 then loop
        JRNZ, STEST1
STEST2: BIT     0, (P7)         ; If SCK = 0 then loop
        JRZ, STEST2
        LD      (SBI0CR1), 00000101B ; <SIOS> ← 0
```

b. 8-bit receive mode

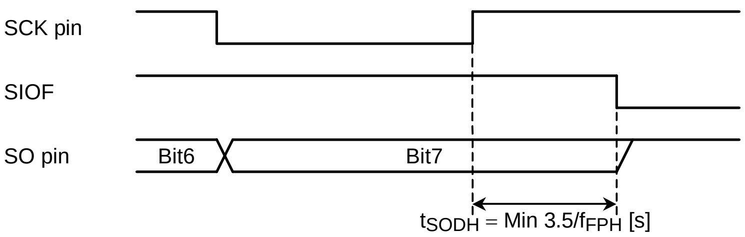


Figure 3.10.27 Transmitted Data Hold Time at End of Transmission

Set the control register to receive mode and set SBI0CR1<SIOS> switching to receive mode. Data is received into and synchronized with the serial clock, starting from the least si (LSB). When 8-bit data is received, the data is transferred from the to SBI0DBR. An INTSBI (Buffer full) interrupt re that the received data be read. The data is then interrupt service program.

When an internal clock is used, the serial clock function will be in effect until the received dat

When an external clock is used, since shift ope external clock pulse, the received data should b next serial clock pulse is input. If the received data is not read, any which is to be received, is canceled. The maxim external clock is used is determined by the delay interrupt request is generated and the time when

Receiving of data ends when <SIOS> is cleared to service program or when <SIOINH> is set to 1. When <SIOS> is cleared to data is transferred to SBI0DBR in complete blocks the transfer is completed. To confirm whether data is bei properly by the program, set SBI0SR<SIOF> to be s when receiving has been completed. When it is confirmed that recei completed, the last data is read. When <SIOINH> i <SIOF> is cleared to 0. (The received data become read it.)

Note: When the transfer mode is changed, the contents of SBI0DBR will be lost. If the mode must be changed, conclude data receiving by clearing <SIOS> to 0, read the last data, then change the mode.

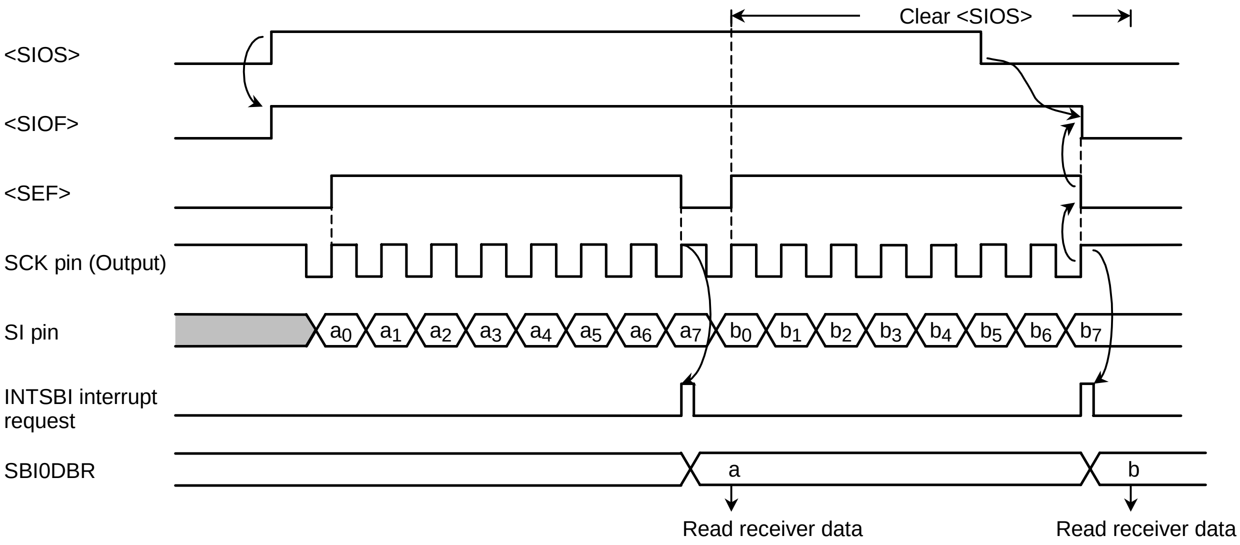


Figure 3.10.28 Receiver Mode (Example: Internal clock)

c. 8-bit transmit/receive mode

Set a control register to a transmit/receive mo After the data has been written, set SBI0CR transmitting/receiving. When data is transmitt pin, starting from the least significant bit (LSB), and synchronize leading edge of the serial clock signal. When dat the SI pin on the trailing edge of the serial clock signal. 8-bit data from the shift register to SBI0DBR and an INTSBI interrupt request i The interrupt service program reads the receiv register and writes the data which is transmitted. SBI0DBR is u transmitting and receiving. Transmitted data should always be wri received data has been read.

When an internal clock is used, the automatic w until the received data has been read and the next

When an external clock is used, since the shift the external clock, received data is read and transmitted data is wri new shift operation is executed. The maximum tra clock is used is determined by the delay time betw request is generated and the received data is read and data is written.

When the transmit is started, after SBI0SR<SIOF> goes to 1 outpu SO pin holds final bit of the last data until fall

Transmitting/receiving data ends when <SIOS> i interrupt service program or when SBI0CR1<SIOINH> is set to 1. When is cleared to 0, received data is transferred to transmit/receive mode ends when the transfer is whether data is being transmitted/received prop to be sensed. <SIOF> is set to 0 when transmitting When <SIOINH> is set to 1, data transmitting/rec cleared to 0.

Note: When the transfer mode is changed, the contents of SBI0DBR will be lost. If the mode must be changed, conclude data transmitting/receiving by clearing <SIOS> to 0, read the last data, then change the transfer mode.

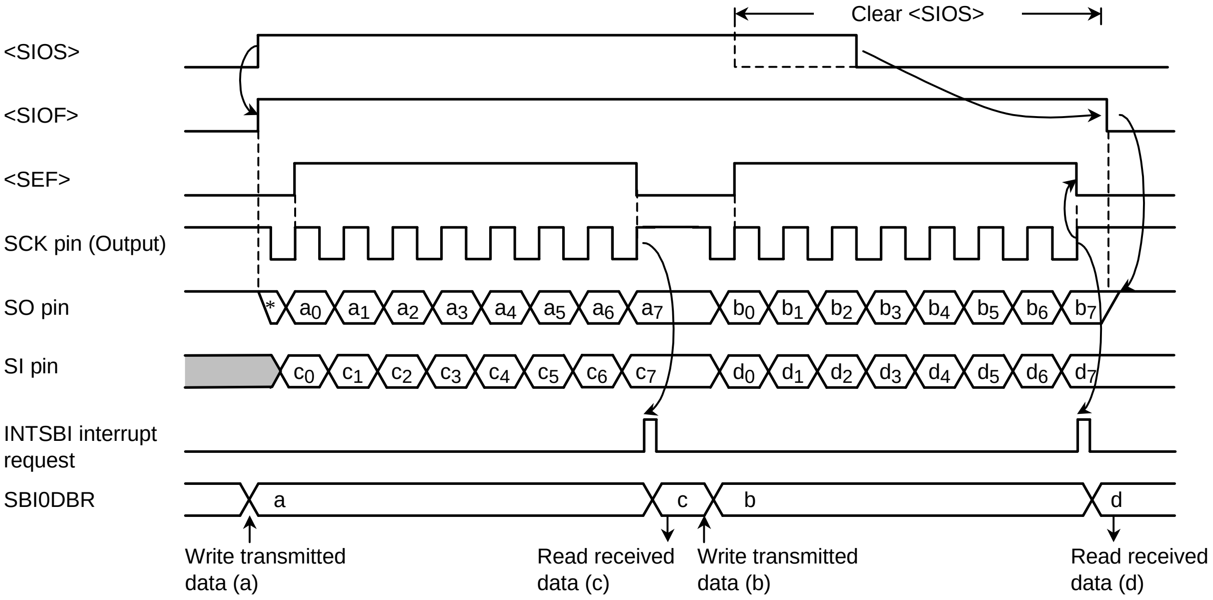


Figure 3.10.29 Transmit/Received Mode (Example using internal clock)

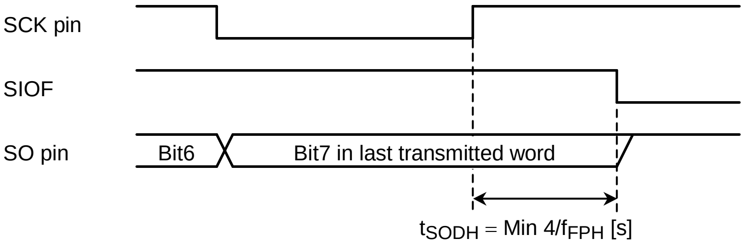

SCK pin
SIOF
SO pin
Bit6
Bit7 in last transmitted word
t_{SODH} = Min 4/f_{FPH} [s]

Figure 3.10.30 Transmitted Data Hold Time at End of Transmit/Receive

3.11 Analog/Digital Converter

The TMP91C820A incorporates a 10-bit successive ap
converter (AD converter) with 8-channel analog input.
Figure 3.11.1 is a block diagram of the AD converter. Th
AN7) are shared with the input-only port 8 and can thus b

Note: When IDLE2, IDLE1 or STOP mode is selected, so as to reduce the power, with some timings the system may enter a standby mode even though the internal comparator is still enabled. Therefore be sure to check that AD converter operations are halted before a HALT instruction is executed.

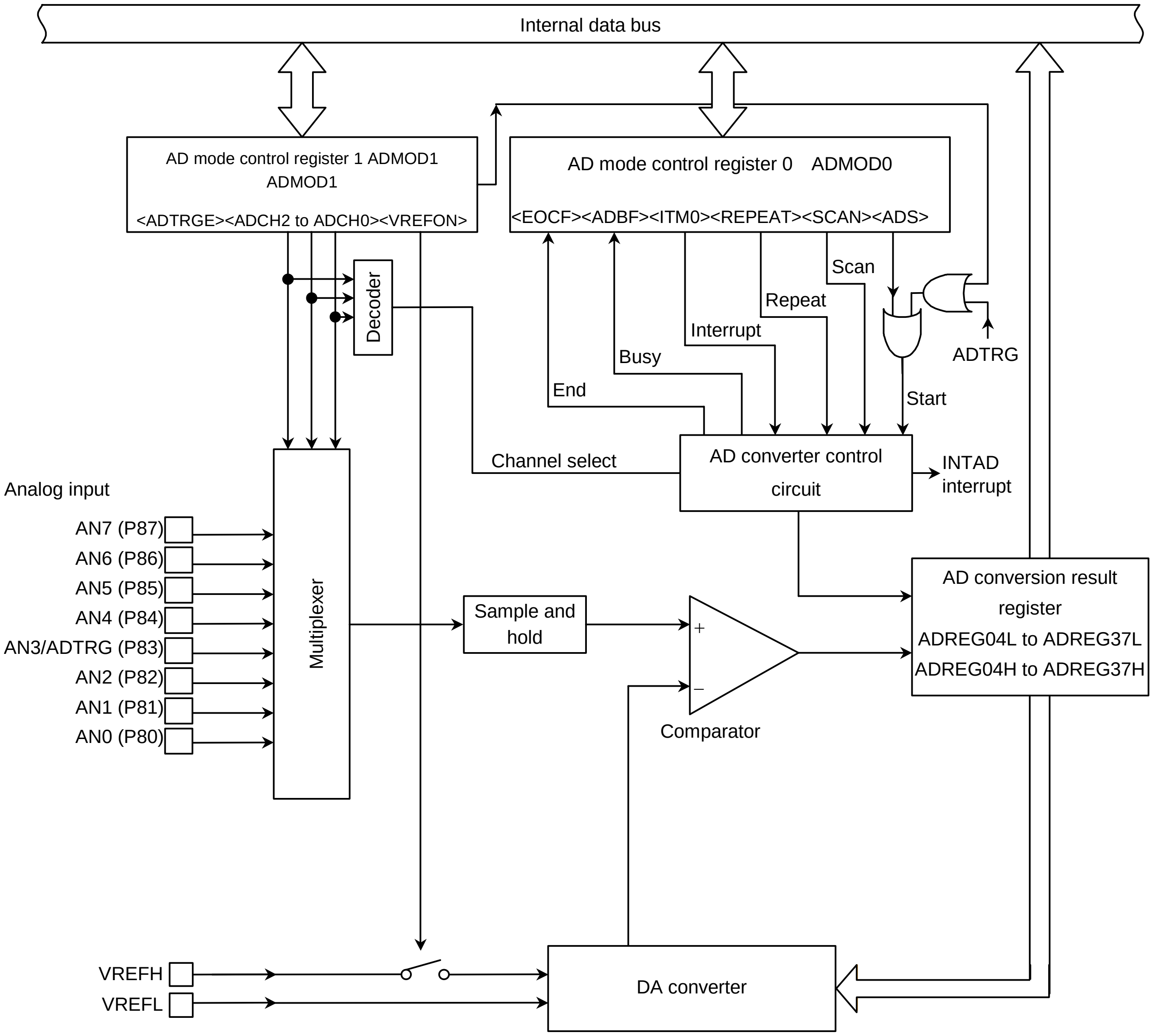


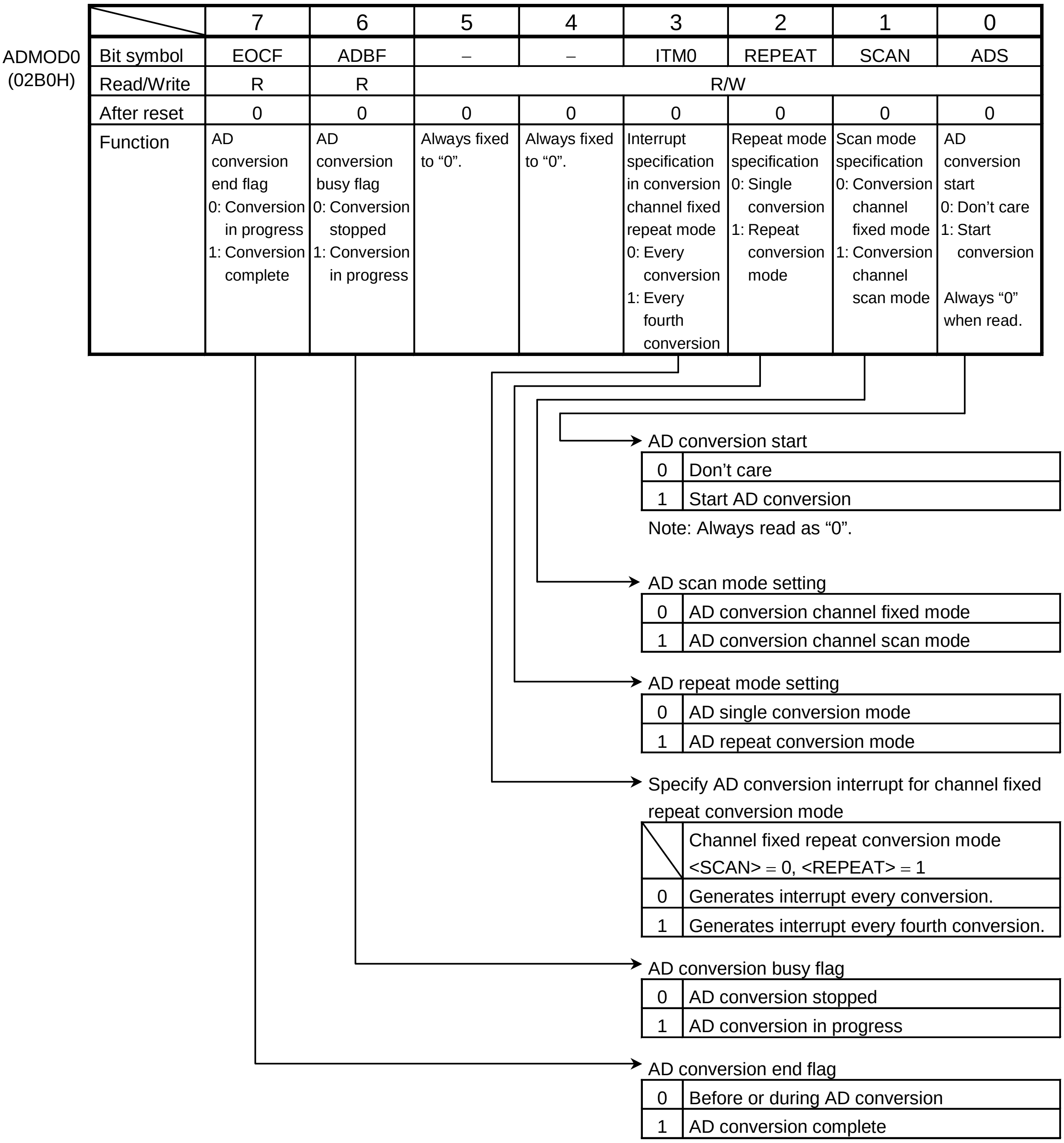
Figure 3.11.1 Block Diagram of AD Converter

3.11.1 Analog/Digital Converter Registers

The two AD mode control registers control the AD converter: ADMOD0 and

The eight AD conversion data upper and lower registers (ADREG04H/L,
ADREG26H/L and ADREG37H/L) store the results of AD co

Figure 3.11.2 shows the registers related to the AD c



AD Mode Control Register 0

ADMOD0 (02B0H)		7	6	5	4	3	2	1	0
	Bit symbol	EOCF	ADBF	–	–	ITM0	REPEAT	SCAN	ADS
	Read/Write	R	R	R/W					
	After reset	0	0	0	0	0	0	0	0
	Function	AD conversion end flag 0: Conversion in progress 1: Conversion complete	AD conversion busy flag 0: Conversion stopped 1: Conversion in progress	Always fixed to “0”.	Always fixed to “0”.	Interrupt specification in conversion channel fixed repeat mode 0: Every conversion 1: Every fourth conversion	Repeat mode specification 0: Single conversion 1: Repeat conversion mode	Scan mode specification 0: Conversion channel fixed mode 1: Conversion channel scan mode	AD conversion start 0: Don’t care 1: Start conversion Always “0” when read.

AD conversion start

0	Don’t care
1	Start AD conversion

Note: Always read as “0”.

AD scan mode setting

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

AD repeat mode setting

0	AD single conversion mode
1	AD repeat conversion mode

Specify AD conversion interrupt for channel fixed repeat conversion mode

	Channel fixed repeat conversion mode <SCAN> = 0, <REPEAT> = 1
0	Generates interrupt every conversion.
1	Generates interrupt every fourth conversion.

AD conversion busy flag

0	AD conversion stopped
1	AD conversion in progress

AD conversion end flag

0	Before or during AD conversion
1	AD conversion complete

Figure 3.11.2 AD Converter Related Register

AD Mode Control Register 1

ADMOD1 (02B1H)

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

Analog input channel selection

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

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

0	Disabled
1	Enabled

IDLE2 control

0	Stopped
1	In operation

Control of application of reference voltage to AD converter

0	OFF
1	ON

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

Note: As pin AN3 also functions as the $\overline{\text{ADTRG}}$ input pin, do not set <ADCH2:0> = 011 when using $\overline{\text{ADTRG}}$ with <ADTRGE> set to 1.

Figure 3.11.3 AD Converter Related Register

AD Conversion Data Low Register 0/4

ADREG04L
(02A0H)

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

AD Conversion Data Upper Register 0/4

ADREG04H
(02A1H)

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

AD Conversion Data Lower Register 1/5

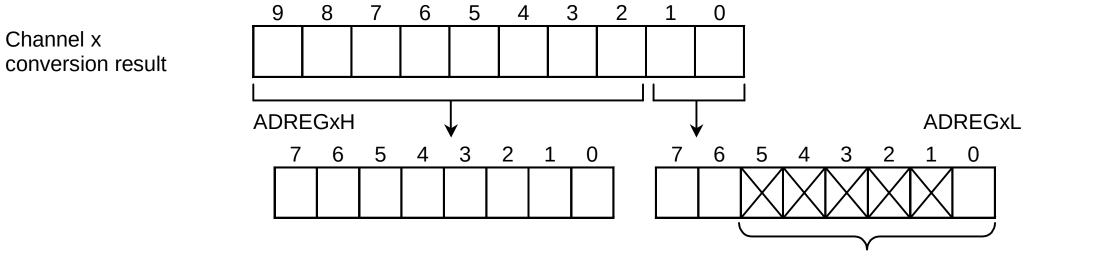
ADREG15L
(02A2H)

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

AD Conversion Data Upper Register 1/5

ADREG15H
(02A3H)

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



- Bits 5 to 1 are always read as 1
- Bit0 is the AD conversion data storage flag <ADRxRF>. When the AD conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.11.4 AD Converter Related Registers

AD Conversion Result Lower Register 2/6

ADREG26L (02A4H)

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

AD Conversion Data Upper Register 2/6

ADREG26H (02A5H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
Read/Write	R							
After reset	Undefined							
Function	Stores upper 8 bits of AD conversion result.							

AD Conversion Data Lower Register 3/7

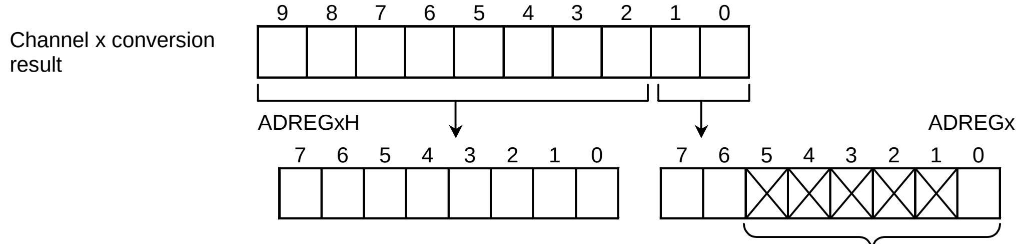
ADREG37L (02A6H)

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

AD Conversion Result Upper Register 3/7

ADREG37H (02A7H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
Read/Write	R							
After reset	Undefined							
Function	Stores upper 8 bits of AD conversion result.							



- Bits 5 to1 are always read as 1.
- Bit0 is the AD conversion data storage flag <ADRxRF>. When the AD conversion result is stored, the flag is set to 1. When either of the registers (ADREGxH, ADREGxL) is read, the flag is cleared to 0.

Figure 3.11.5 AD Converter Related Registers

3.11.2 Description of Operation

(1) Analog reference voltage

A high-level analog reference voltage is applied t reference voltage is applied to the VREFL pin. To pe voltage, the difference between VREFH and VREFL is divided by 1024 us resistance. The result of the division is then comp

To turn off the switch between VREFH and VRE ADMOD1<VREFON> in AD mode control register 1. To s OFF state, first write a 1 to ADMOD1<VREFON>, wait μs until the in reference voltage stabilizes ([illegible]), then set ADMOD0<

(2) Analog input channel selection

The analog input channel selection varies depends converter.

- In analog input channel fixed mode (ADMOD0<SCAN> = 0)
  Setting ADMOD1<ADCH2:0> selects one of the inpu input channel.
- In analog input channel scan mode (ADMOD0<SCAN> = 1)
  Setting ADMOD1<ADCH2:0> selects one of the eight scan modes.

Table 3.11.1 illustrates analog input channel sel

On a reset, ADMOD0<SCAN> is set to 0 and ADMOD1<ADC 000. Thus pin AN0 is selected as the fixed input channel. Pins not **used** as analog channels can be used as standard input port pins.

Table 3.11.1 Analog Input Channel Selection

<ADCH2:0>	Channel Fixed <SCAN> = 0	Channel Scan <SCAN> = 1
000	AN0	AN0
001	AN1	AN0 → AN1
010	AN2	AN0 → AN1 → AN2
011	AN3	AN0 → AN1 → AN2 → AN3
100	AN4	AN4
101	AN5	AN4 → AN5
110	AN6	AN4 → AN5 → AN6
111	AN7	AN4 → AN5 → AN6 → AN7

(3) Starting AD conversion

To start AD conversion, write a 1 to ADMOD0<ADS> in or ADMOD1<ADTRGE> in AD mode control register 1, a $\overline{\text{ADTRG}}$ pin. When AD conversion starts, the AD co ADMOD0<ADBF> will be set to 1, indicating that AD co

Writing a 1 to ADMOD0<ADS> during AD conversion re time, to determine whether the AD conversion resul value of the conversion data storage flag ADREGxL<

During AD conversion, a falling edge on the $\overline{\text{ADTRG}}$ pin will be ignored.

(4) AD conversion modes and the AD conversion end inte

The four AD conversion modes are:

- Channel fixed single conversion mode
- Channel scan single conversion mode
- Channel fixed repeat conversion mode
- Channel scan repeat conversion mode

The ADMOD0<REPEAT> and ADMOD0<SCAN> settings in register 0 determine the AD mode setting.

Completion of AD conversion triggers an INTAD AD request. Also, ADMOD0<EOCF> will be set to 1 to ind been completed.

a. Channel fixed single conversion mode

Setting ADMOD0<REPEAT> and ADMOD0<SCAN> to 00 channel fixed single conversion mode.

In this mode data on one specified channel is co conversion has been completed, the ADMOD0<EO ADMOD0<ADBF> is cleared to 0, and an INTAD interrupt request is

b. Channel scan single conversion mode

Setting ADMOD0<REPEAT> and ADMOD0<SCAN> to 01 channel scan single conversion mode.

In this mode data on the specified scan channels scan conversion has been completed, ADMOD0<EOCF> is ADMOD0<ADBF> is cleared to 0, and an INTAD interrupt request is

c. Channel fixed repeat conversion mode

Setting ADMOD0<REPEAT> and ADMOD0<SCAN> to 10
channel fixed repeat conversion mode.

In this mode data on one specified channel is c
conversion has been completed, ADMOD0<EOCF>
ADMOD0<ADBF> is not cleared to 0 but held at 1. I
generation timing is determined by the setting of ADMOD0<ITM0>.

Setting <ITM0> to 0 generates an interrupt re
conversion is completed.

Setting <ITM0> to 1 generates an interrupt request on completi
fourth conversion.

d. Channel scan repeat conversion mode

Setting ADMOD0<REPEAT> and ADMOD0<SCAN> to 11
channel scan repeat conversion mode.

In this mode data on the specified scan channels
each scan conversion has been completed, ADMOD0<EOC
INTAD interrupt request is generated. ADMOD0<AD
held at 1.

To stop conversion in a repeat conversion mode (
to ADMOD0<REPEAT>. After the current conversio
repeat conversion mode ends and ADMOD0<ADBF> is cleared to

Switching to a halt state (IDLE2 mode with ADMO
IDLE1 mode or STOP mode) immediately stops oper
even when AD conversion is still in progress. In r
cases c and d), when the halt is released, conversion restarts from t
In single conversion mode (cases a and b), conversi
when the halt is released (The converter remains

Table 3.11.2 shows the relationship between t
interrupt requests.

Table 3.11.2 Relationship between AD Conversion Modes and Interrupt Requests

Mode	Interrupt Request Generation	ADMOD0		
		<ITM0>	<REPEAT>	<SCAN>
Channel fixed single conversion mode	After completion of conversion	X	0	0
Channel scan single conversion mode	After completion of scan conversion	X	0	1
Channel fixed repeat conversion mode	Every conversion	0	1	0
	Every forth conversion	1		
Channel scan repeat conversion mode	After completion of every scan conversion	X	1	1

X: Don’t care

(5) AD conversion time

84 states (10.5 μs at f_{PH} = 16 MHz) are required for the AD co channel.

(6) Storing and reading the results of AD conversion

The AD conversion data upper and lower registers (ADREG04H/L to ADREG37H/L store the results of AD conversion. (ADREG04H/L t registers.)

In channel fixed repeat conversion mode, the conversion results successively in registers ADREG04H/L to ADREG37H/ AN4, AN1 and AN5, AN2 and AN6, and AN3 and AN7 conversion results a ADREG04H/L, ADREG15H/L, ADREG26H/L and ADREG37H/L respectivel

Table 3.11.3 shows the correspondence between the registers, which are used to hold the results of AD c

Table 3.11.3 Correspondence between Analog Input Channels and AD Conversion Result Registers

Analog Input Channel (Port 8)	AD Conversion Result Register	
	Conversion Modes Other than at Right	Channel Fixed Repeat Conversion Mode (Every 4th conversion)
AN0	ADREG04H/L	ADREG04H/L ↓ ADREG15H/L ↓ ADREG26H/L ↓ ADREG37H/L
AN1	ADREG15H/L	
AN2	ADREG26H/L	
AN3	ADREG37H/L	
AN4	ADREG04H/L	
AN5	ADREG15H/L	
AN6	ADREG26H/L	
AN7	ADREG37H/L	

<ADRxRF> bit0 of the AD conversion data lower re conversion data storage flag. The storage flag inc result register has been read or not. When a conver conversion result register, the flag is set to 1. When either of the A registers (ADREGxH or ADREGxL) is read, the flag is

Reading the AD conversion result also clears t ADMOD0<EOCF> to 0.

Setting example:

a. Convert the analog input voltage on the AN3 pin a
address 1000H using the AD interrupt (INTAD) pro

Main routine:

		7 6 5 4 3 2 1 0	
INTE0AD	←	X 1 0 0 – – – –	Enable INTAD and set it to interrupt level 4.
ADMOD1	←	1 1 X X 0 0 1 1	Set pin AN3 to be the analog input channel.
ADMOD0	←	X X 0 0 0 0 0 1	Start conversion in channel fixed single conversion mode.

Interrupt routine processing example:

WA	←	ADREG37	Read value of ADREG37L and ADREG37H into 16-bit general-purpose register WA.
WA	> > 6		Shift contents read into WA six times to right and zero-fill upper bits.
(1000H)	←	WA	Write contents of WA to memory address 1000H.

b. This example repeatedly converts the analog input voltages on the thr
AN1 and AN2, using channel scan repeat conversio

INTE0AD	←	X 0 0 0 – – – –	Disable INTAD.
ADMOD1	←	1 X X X 0 0 1 0	Set pins AN0 to AN2 to be the analog input channels.
ADMOD0	←	X X 0 0 0 1 1 1	Start conversion in channel scan repeat conversion mode.

X: Don’t care, –: No change

3.12 Watchdog Timer (Runaway detection timer)

The TMP91C820A features a watchdog timer for detecting

The watchdog timer (WDT) is used to return the CPU to nor
CPU has started to malfunction (Runaway) due to causes
timer detects a malfunction, it generates a non-maskabl
the malfunction.

Connecting the watchdog timer output pin internally forces a r

3.12.1 Configuration

Figure 3.12.1 is a block diagram of he watchdog timer

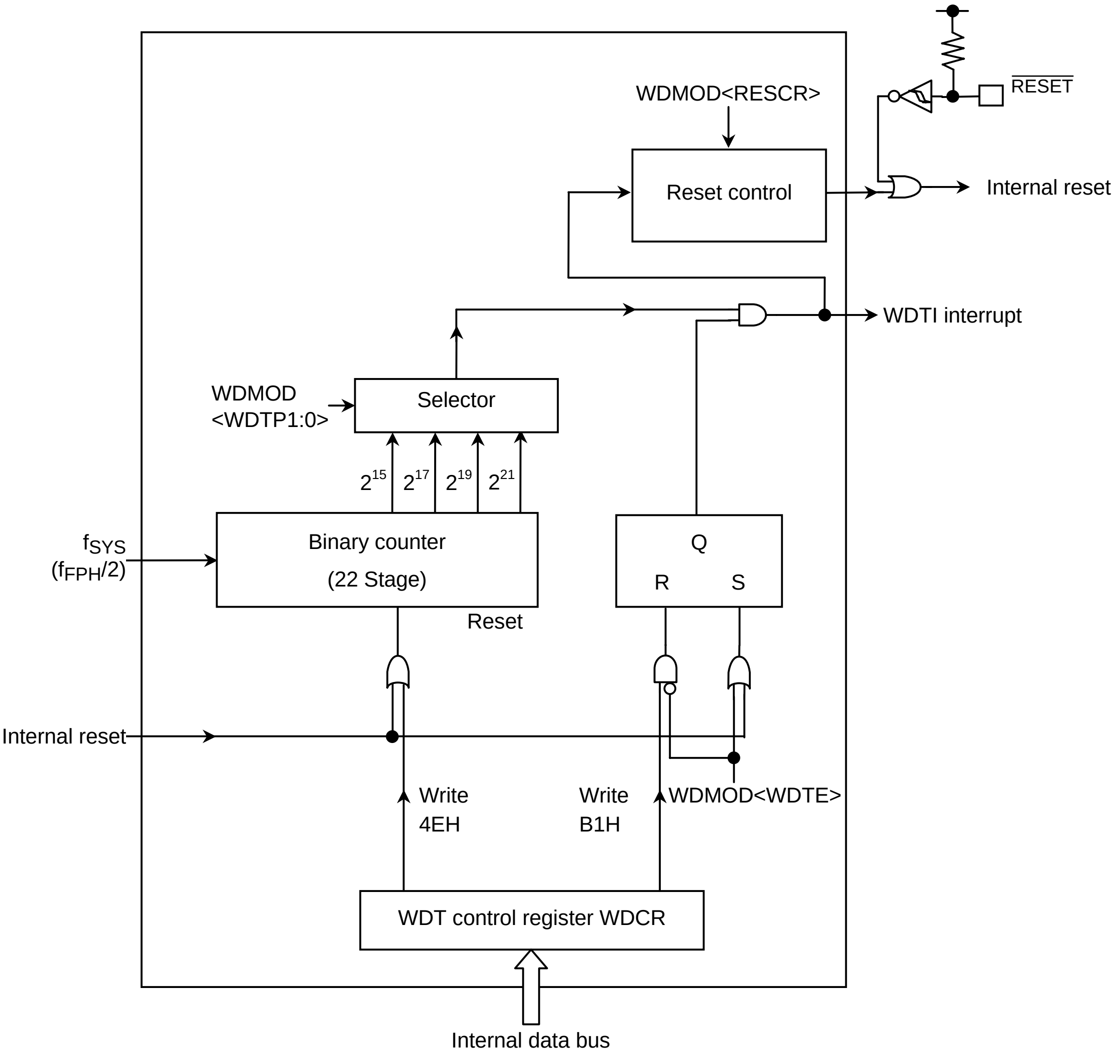


Figure 3.12.1 Block Diagram of Watchdog Timer

The watchdog timer consists of a 22-stage binary cou
(f_{SYS}) as the input clock. The binary counter can output $f_{SYS}/2^{15}$, $f_{SYS}/2^{17}$, $f_{SYS}/2^{19}$ and outpu
$f_{SYS}/2^{21}$. Selecting one of the outputs using WDMOD<WDTP1
interrupt and outputs watchdog timer out when an over

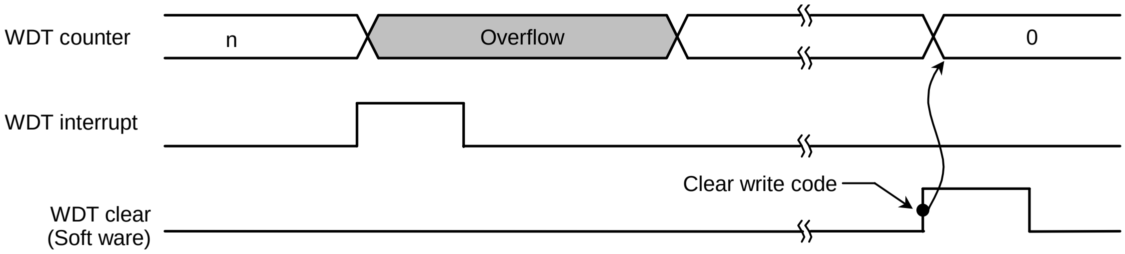


Figure 3.12.2 Normal Mode

The runaway detection result can also be connected to the reset pin inte
In this case, the reset time will be between 22 and 29

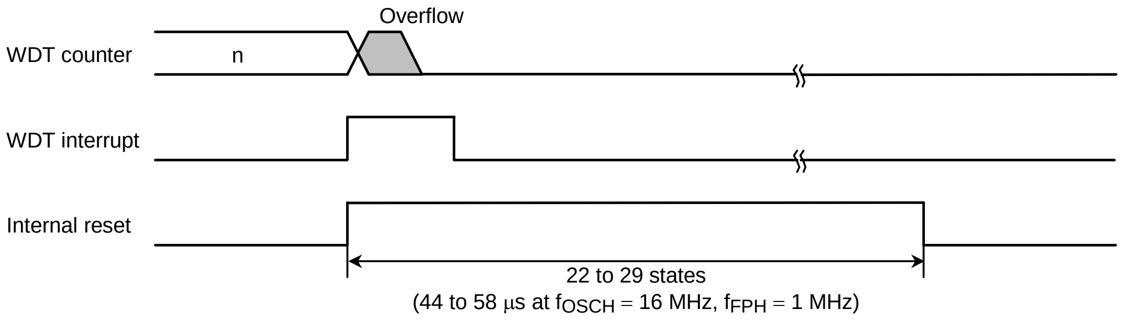


Figure 3.12.3 Reset Mode

3.12.2 Control Registers

The watchdog timer WDT is controlled by two controls

(1) Watchdog timer mode register (WDMOD)

a. Setting the detection time for the watchdog tim

This 2-bit register is used for setting the wat when detecting runaway. On a reset this reg WDMOD<WDTP1:0> = 00.

The detection times for WDT are shown in Figure 3

b. Watchdog timer enable/disable control registe

On a reset WDMOD<WDTE> is initialized to 1, enab

To disable the watchdog timer, set this bit to 0 and disable code (B1H) to the watchdog timer control register (WDCR). T difficult for the watchdog timer to be disabled by runaway.

However, it is possible to return the watchdog t 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 RESET terminal internally. Since WDMOD<RESCR> i 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 bina

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

```
WDMOD    ← 0 – – – – – – 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 t (4EH) to the WDCR register.

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

WDMOD
(0300H)

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

Watchdog timer out control

0	–
1	Connects WDT out to a reset

IDLE2 Control

0	Stop
1	Operation

Watchdog timer detection time

at fc = 16 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)	4.096 ms	16.384 ms	65.536 ms	262.144 ms
	001 (fc/2)	8.192 ms	32.768 ms	131.072 ms	524.288 ms
	010 (fc/4)	16.384 ms	65.536 ms	262.144 ms	1.049 s
	011 (fc/8)	32.768 ms	131.072 ms	524.288 ms	2.097 s
	100 (fc/16)	65.536 ms	262.144 ms	1.049 s	4.194 s

Watchdog timer enable/disable control

0	Disabled
1	Enabled

Figure 3.12.4 Watchdog Timer Mode Register

WDCR
(0301H)

Prohibit read-modify-write

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

Disable/clear WDT

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

Figure 3.12.5 Watchdog Timer Control Register

3.12.3 Operation

The watchdog timer generates an INTWD interrupt whe WDMOD<WDTP1:0> has elapsed. The watchdog timer mus before an INTWD interrupt will be generated. If the C occurs) due to causes such as noise, but does not execute the instruction u binary counter, the binary counter will overflow and an INTWD interrupt will b The CPU will detect malfunction (runaway) due to the INTWD interrupt and in is possible to return to the CPU to normal operation by means of an anti- program.

The watchdog timer does not operate in IDLE1 or STOP continues counting during bus release (when $\overline{\text{BUSAK}}$ goes low)

When the device is in IDLE2 mode, the operation WDMOD<I2WDT> setting. Ensure that WDMOD<I2WDT> is se 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 – – – – 0
```

c. Disable the watchdog timer.

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

3.13 Real Time Clock (RTC)

3.13.1 Function Description for RTC

1) Clock function (Hour, minute, second)

2) Calendar function (Month and day, day of the week,

3) 24- or 12-hour (AM/PM) clock function

4) +/- 30 second adjustment function (by software)

5) Alarm function (Alarm output)

6) Alarm interrupt generate

3.13.2 Block Diagram

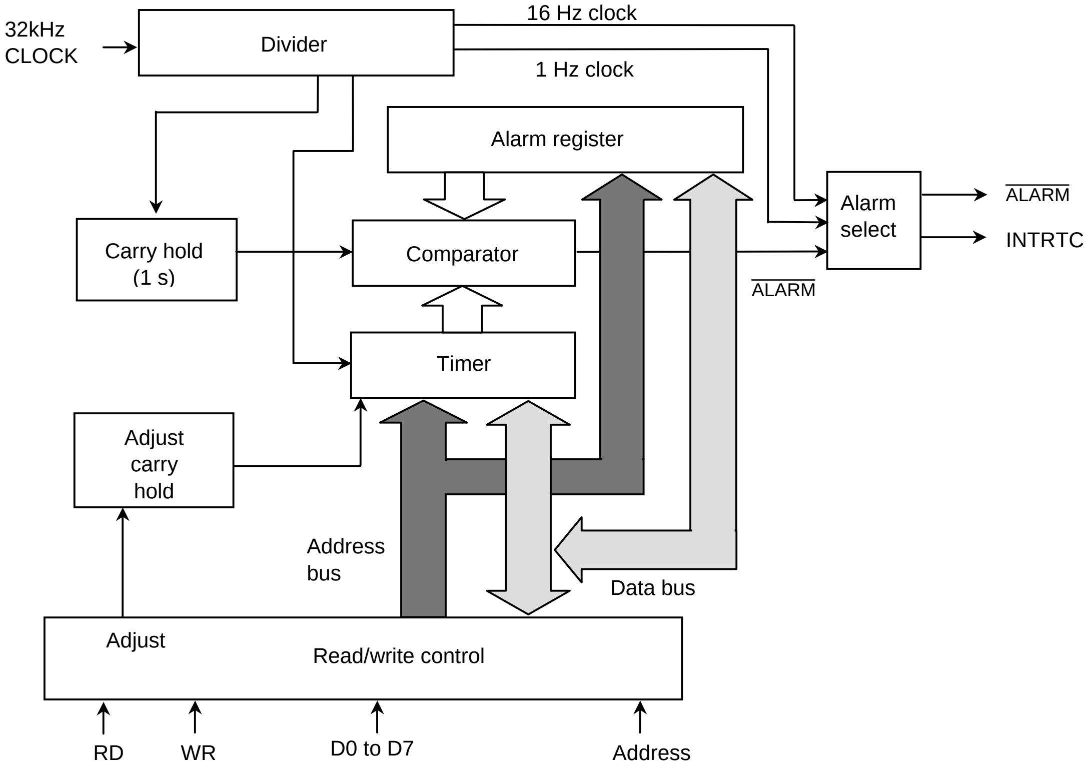


Figure 3.13.1 RTC Block Diagram

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 is divisible with 400, is a leap year. But there is not this product for the correspondence to the above exception. Because there are only with the year which is divisible with 4 as a leap year, please cope with the system side if this function is problem.

3.13.3 Control Registers

Table 3.13.1 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	Second column	R/W
MINR	0321H		40 min.	20 min.	10 min.	8 min.	4 min.	2 min.	1 min.	Minute column	R/W
HOURR	0322H			20 hours	10 hours	8 hours	4 hours	2 hours	1 hour	Hour column	R/W
DAYR	0323H						W2	W1	W0	Day of the week column	R/W
DATER	0324H			Day 20	Day 10	Day 8	Day 4	Day 2	Day 1	Day column	R/W
MONTHR	0325H				Oct.	Aug.	Apr.	Feb.	Jan.	Month column	R/W
YEARR	0326H	Year 80	Year 40	Year 20	Year 10	Year 8	Year 4	Year 2	Year 1	Year column (Lower two columns)	R/W
PAGER	0327H	INTENA			ADJUST	ENATMR	ENAALM		PAGE	PAGE register	W, R/W
RESTR	0328H	DIS1HZ	DIS16HZ	RSTTMR	RSTALM	0	0	0	0	Reset register	Write only

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

Table 3.13.2 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 min.	Minute column for alarm	R/W
HOURR	0322H			20 hours	10 hours	8 hours	4 hours	2 hours	1 hour	Hour column for alarm	R/W
DAYR	0323H						W2	W1	W0	Day of the week column for alarm	R/W
DATER	0324H			Day 20	Day 10	Day 8	Day 4	Day 2	Day 1	Day column for alarm	R/W
MONTHR	0325H								24/12	24-hour clock mode	R/W
YEARR	0326H							LEAP1	LEAP0	Leap-year mode	R/W
PAGER	0327H	INTENA				ENATMR	ENAALM		PAGE	PAGE register	W, R/W
RESTR	0328H	DIS1HZ	DIS16HZ	RSTTMR	RSTALM	0	0	0	0	Reset register	Write only

3.13.4 Detailed Explanation of Control Register

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

(1) Second column register (for PAGE0 only)

		7	6	5	4	3	2	1	0
SECR (0320H)	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	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

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

		7	6	5	4	3	2	1	0
MINR (0321H)	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.

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

a. In case of 24-hour clock mode (MONTHR<MO0> = 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 hour column	10 hour column	8 hour column	4 hour column	2 hour column	1 hour column

0	0	0	0	0	0	0 o’clock
0	0	0	0	0	1	1 o’clock
0	0	0	0	1	0	2 o’clock

0	0	1	0	0	0	8 o’clock
0	0	1	0	0	1	9 o’clock
0	1	0	0	0	0	10 o’clock

0	1	1	0	0	1	19 o’clock
1	0	0	0	0	0	20 o’clock

1	0	0	0	1	1	23 o’clock

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

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/ $\overline{\text{AM}}$	10 hour column	8 hour column	4 hour column	2 hour column	1 hour column

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

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

		7	6	5	4	3	2	1	0
DAYR (0323H)	Bit symbol						WE2	WE1	WE0
	Read/Write						R/W		
	After reset						Undefined		
	Function	"0" is read.					W2	W1	W0

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

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

		7	6	5	4	3	2	1	0
DATER (0324H)	Bit symbol			DA5	DA4	DA3	DA2	DA1	DA0
	Read/Write			R/W					
	After reset			Undefined					
	Function	"0" is read.		Day 20	Day 10	Day 8	Day 4	Day 2	Day 1

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

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

0	1	1	0	0	1	19th day
1	0	0	0	0	0	20th day

1	0	1	0	0	1	29th day
1	1	0	0	0	0	30th day
1	1	0	0	0	1	31st day

(6) Month column register (for PAGE0 only)

		7	6	5	4	3	2	1	0
MONTHR (0325H)	Bit symbol				MO4	MO3	MO2	MO1	MO0
	Read/Write				R/W				
	After reset				Undefined				
	Function	"0" is read.			10 months	8 months	4 months	2 months	1 month

↓

0	0	0	0	1	January
0	0	0	1	0	February
0	0	0	1	1	March
0	0	1	0	0	April
0	0	1	0	1	May
0	0	1	1	0	June
0	0	1	1	1	July
0	1	0	0	0	August
0	1	0	0	1	September
1	0	0	0	0	October
1	0	0	0	1	November
1	0	0	1	0	December

(7) Select 24-hour clock or 12-hour clock (for PAGE1 only)

		7	6	5	4	3	2	1	0
MONTHR (0325H)	Bit symbol								MO0
	Read/Write								R/W
	After reset								Undefined
	Function	"0" is read.							1: 24 hours 0: 12 hours

(8) Year column register (for PAGE0 only)

		7	6	5	4	3	2	1	0
YEARR (0326H)	Bit symbol	YE7	YE6	YE5	YE4	YE3	YE2	YE1	YE0
	Read/Write	R/W							
	After reset	Undefined							
	Function	80 years	40 years	20 years	10 years	8 years	4 years	2 years	1 year

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

(9) Leap-year register (for PAGE1 only)

		7	6	5	4	3	2	1	0
YEARR (0326H)	Bit symbol							LEAP1	LEAP0
	Read/Write							R/W	
	After reset							Undefined	
	Function	"0" is read.						00: Leap year 01: One year after leap year 10: Two years after leap year 11: Three years after leap year	

0	0	Current year is leap year
0	1	Present is next year of a leap year
1	0	Present is two years after a leap year
1	1	Present is three years after leap year

(10) Setting PAGE register (for PAGE0/1)

PAGER
(0327H)

	7	6	5	4	3	2	1	0
Bit symbol	INTENA			ADJUST	ENATMR	ENAALM		PAGE
Read/Write	R/W			W	R/W			R/W
After reset	0			Undefined	Undefined			Undefined
Function	1: INT enable (Note)	"0" is read.		1:Adjust	Timer 1:Enable 0:Disable	Alarm 1:Enable 0:Disable	"0" is read.	PAGE select

Prohibit read-modify-write.

Note: When INTRTC is used, set following,
ld (pager), 0 ch
ld (pager), 8 ch

0	Select Page 0
1	Select Page 1

0	–
1	Adjust sec. counter. When set this bit to 1 the sec. counter become to 0 when the value of sec. counter is 0 to 29. And in case that value of sec. counter is 30 to 59, min. counter is carried and become sec. counter to 0.

(11) Setting reset register (for PAGE0/1)

RESTR
(0328H)

	7	6	5	4	3	2	1	0
Bit symbol	DIS1HZ	DIS16HZ	RSTTMR	RSTALM	RE3	RE2	RE1	RE0
Read/Write	W							
After reset	Undefined							
Function	0: 1 Hz	0: 16 Hz	1: Timer reset	1:Alarm reset	Write "0".			

Prohibit read-modify-write.

0	–
1	Reset alarm register

0	–
1	Timer reset

0	Enable 16 Hz clock ($\overline{\text{ALARM}}$ output, INTRTC)
1	Disable 16 Hz clock ($\overline{\text{ALARM}}$ output, INTRTC)

0	Enable 1 Hz clock ($\overline{\text{ALARM}}$ output, INTRTC)
1	Disable 1 Hz clock ($\overline{\text{ALARM}}$ output, INTRTC)

3.13.5 Operational Description

(1) Reading timer data

a. There is the case, which reads wrong data when
happens during the operation which timer data re
times with the following correct data.

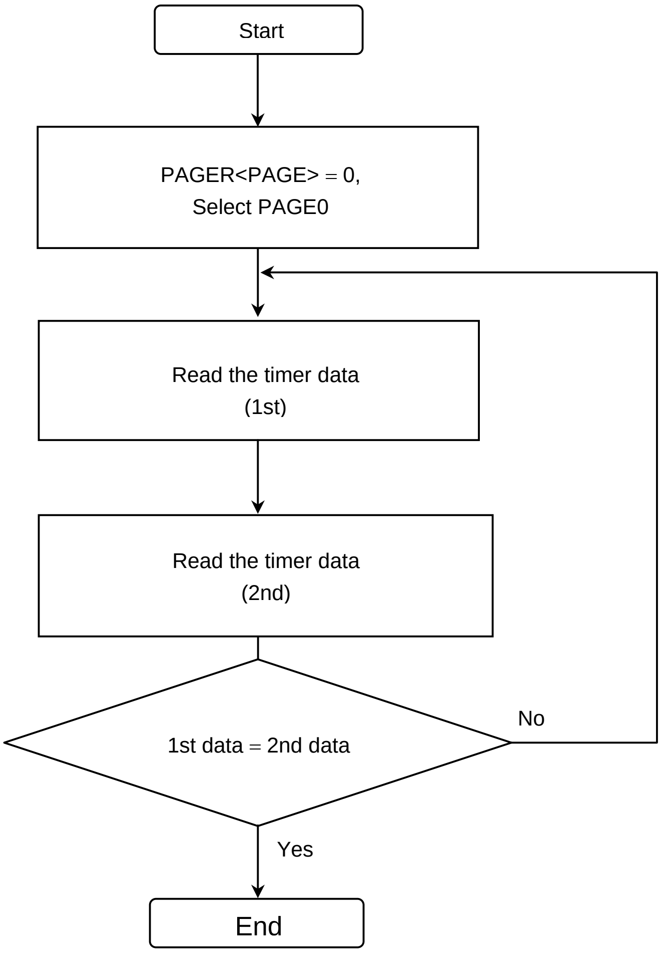
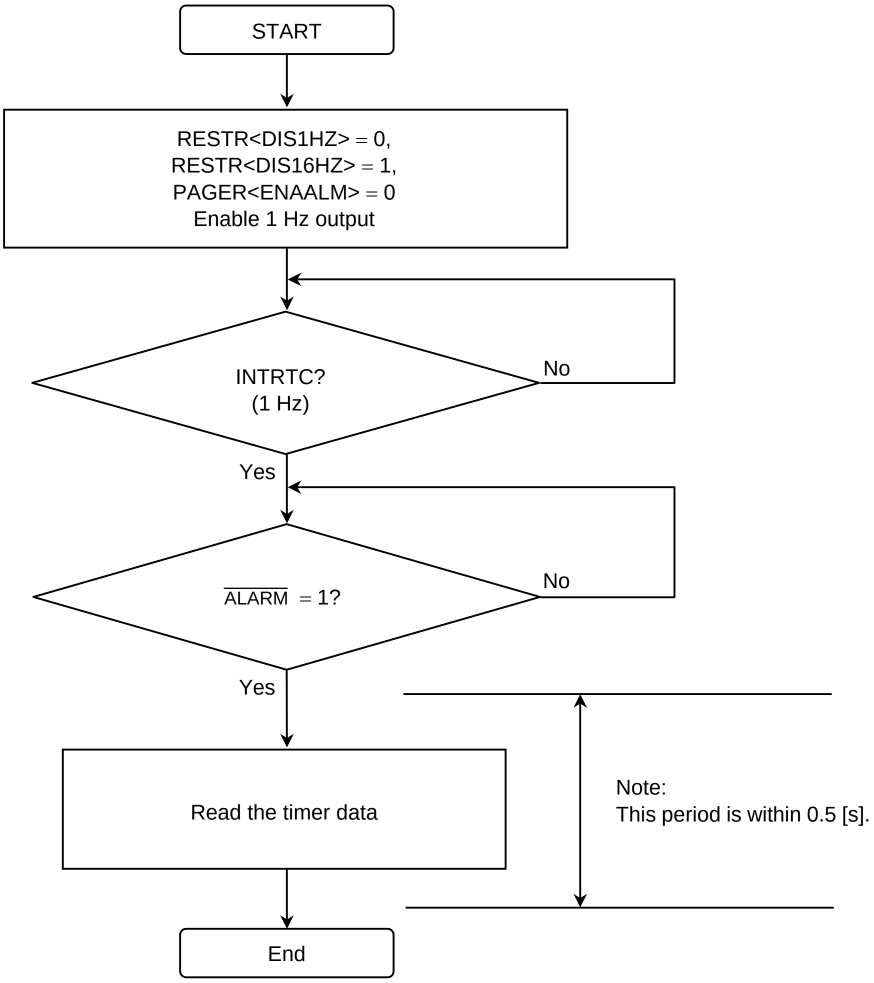
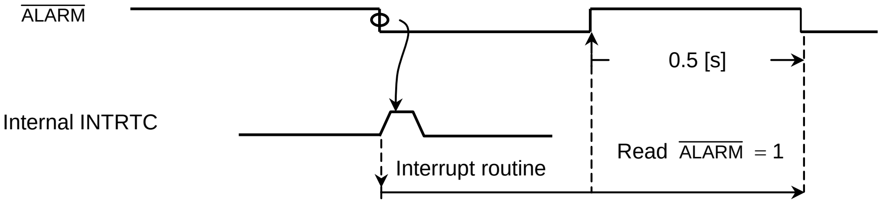


Figure 3.13.2 Flowchart of Timer Data Read

b. Readout of timer ALARM output used

Timer data can be read with using ALARM output by detecting ALARM = 1 with interrupt routine of INTRTC of 1 Hz.





The reason why read a timer of RTC after reading PORT in interrupt routine of $\overline{\text{ALARM}}$ = 1 is that carry of RTC timer occurs with rising edge of pulse period of 1 Hz. By reading timer during 0.5 second after carry happening, right data (A timer value) can be read.

Figure 3.13.3 Readout of the Timer Table Used $\overline{\text{ALARM}}$ Output

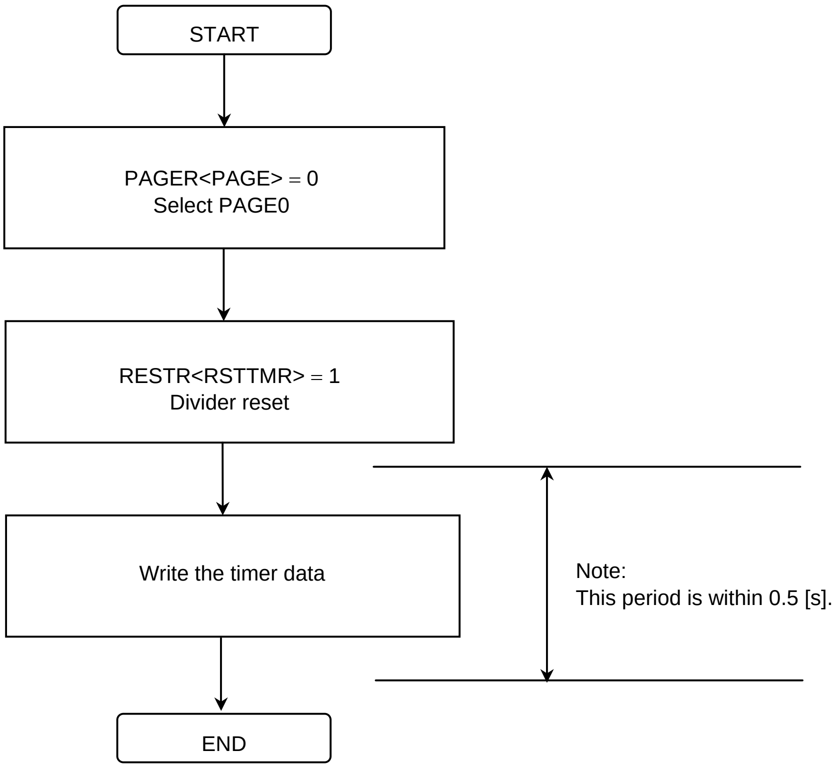
(2) Writing timer data

When there is carry on the way of write operation,
exactly.

Therefore, in order to write in data exactly pleas

a. Resetting a divider

In RTC inside, there are 15-stage dividers, wh
32.768 kHz. Carry of a timer is not done for one se
write in data at this interval.



Note:
This period is within 0.5 [s].

Figure 3.13.4 Flowchart of Data Write

b. Disabling the timer

Carry of a timer is prohibited when write in 0 at prevent malfunction by CLOCK HOLD. During a timer p CLOCK HOLD circuits hold a one-sec. carry signal, which is gene divider. After becoming a timer enable, carry signal is output to time revise time and continue operation. However, when the timer-disab state continues for one second or more. During ti system power is down. In this case, timer is stopped and time i

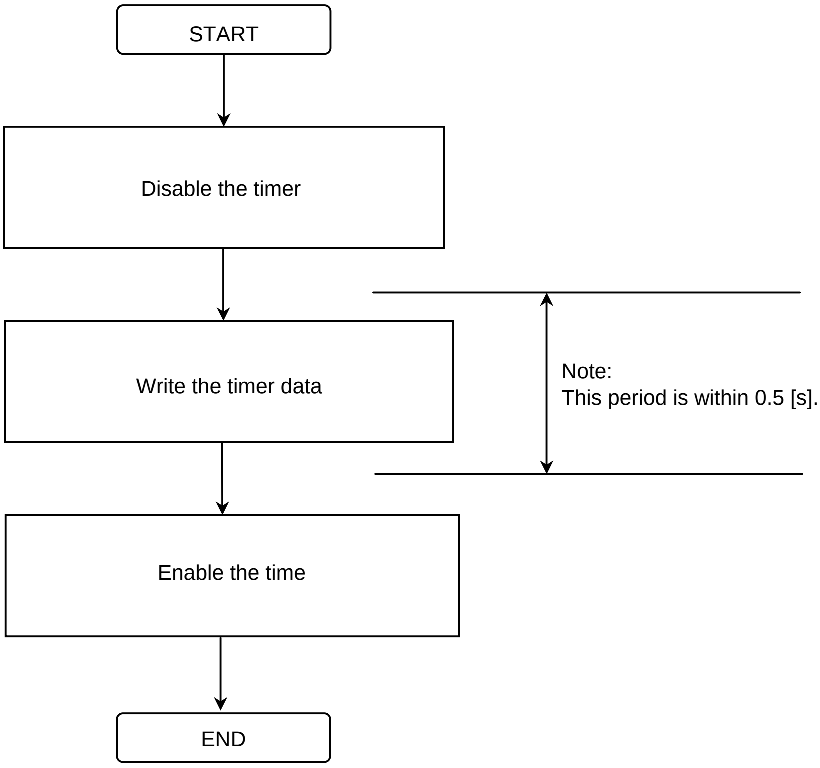


Figure 3.13.5 Flowchart of Timer Disable

3.13.6 Explanation of the Alarm Function

Can use alarm function by setting of register of PAGE
to $\overline{\text{ALARM}}$ pin as follows.

(1) In accordance of alarm register and the timer,

(2) Output clock of 1Hz.

(3) Output clock of 16Hz.

Resetting does not initialize RTC. So the case of usi
flag INTERTCKEY<IRC> after reset.

(1) In accordance of alarm register and a timer, outpu

When value of a timer of PAGE0 accorded with alarm register of PAGE1 w
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 R
all alarm becomes don't care. In this case, always accord with value o
occur INTRTC interrupt if PAGER<ENAALM> is 1.

Setting alarm min., alarm hour, alarm day and ala
writing in data at each register of PAGE1.

When all setting contents accorded, RTC genera
PAGER<ENAALM> is 1. However, contents set (Don't care) which does
is considered to always accord.

The contents, which set it up once, cannot be re
independence. Initialization of alarm and resetti

Follows are example program at outputting $\overline{\text{ALARM}}$ at noon a
(PM12:00) every day

```
        LD      (PAGER), 09H    ; Alarm disable, sett
        LD      (RESTR), D0H    ; Alarm initialize
        LD      (MONTHR), 01H   ; 24-hour clock mode
        LD      (HOURR), 12H    ; Setting 12 o'clock
        LD      (MINR), 00H     ; Setting 00 min.
                                ; Setup time 31 μs   Note)
        LD      (PAGER), 0CH    ; Alarm enable
       ( LD     (PAGER), 8CH    ; Interrupt enable )
```

When CPU is operated by high-frequency oscillatio
clock at 32 kHz, about 30 μs () for the time register setting becomes valid. In th
example, it is necessary to set up time 31 μs between setting th
enabling the alarm register.

Note: This setup time is unnecessary under SLOW mode when you use only internal interruption.

(2) When output clock of 1Hz

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

(3) When output clock of 16Hz

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

3.14 LCD Controller (LCDC)

The TMP91C820A incorporates two types liquid crystal c
LCD Driver LSI. One circuit handles a RAM build-in type
data in the LCD driver itself, and the other handles shift-register type LCD driver th
must serially transfer the display data to LCD driver for each display pictu

Shift-register type LCD driver control mode (SR type)
Set the mode of operation, display data address in memory and LCD
before start SR type.
After started SR type LCDC, LCDC requests bus to CPU and read data
data memory. After that LCDC transmits data to external
through exclusive data bus (LD7:0). At this control signal is connected L
specified waveform synchronizes with data transmissio
After finish display data reading, LCDC cancels the bus
As the display RAM, SRAM or DRAM can be used in TMP91C820A.

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 SFR, when moves inst
LCDC outputs chip select signal to LCD driver connect
(D1BSCP etc.). Therefore control of data transmission r
is controlled by software. At this time, LCD controlle
transmission data output from data bus (D7:0).

This section is constituted as follows.

3.14.1 Feature of LCDC of Each Mode

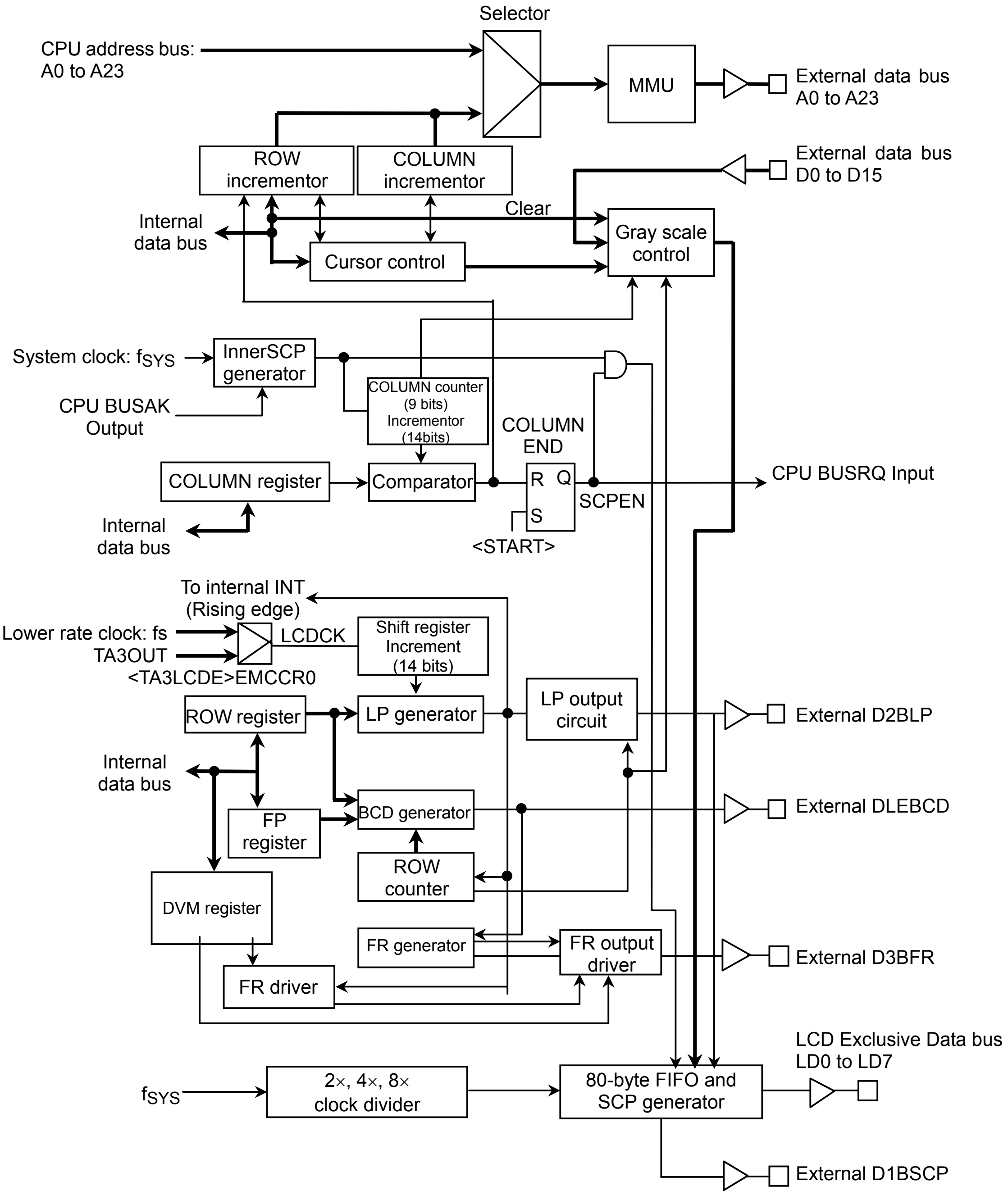
Each feature and operation of pin is as follows.

Table 3.14.1 Feature of LCDC of Each Mode

(Example: Toshibamade LCD driver T6C13B, T6B66

		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): 128, 160, 200, 240, 320, 400, 480 Segment (Column): 128, 160, 240, 320, 400, 480, 560, 640	There is not a limitation
Display memory data bus width		16-bit fixed	Depend on the setting of CS/WAIT controller.
LCD driver data bus width		8-bit fixed	
Transfer rate (at f_{FPH} = 36 [MHz])		Min 55 ns/1 word at SDRAM/BURST Min 111 ns/1 word at SRAM	
External pins	Data bus (D7 to D0)	Not used	Data bus; connect to data bus of LCD driver.
	LCD data bus: (LD7 to LD0)	Data bus; connect to data bus of LCD driver.	Not used
	Bus state	Not used	Bus state; connect with write enable pin of segment/common driver.
	Address bus: (A0)	Not used	Address 0; connect with D/I pin of segment driver. When A0 = 1 data bus value means display data, when A0 = 0 data bus means instruction data.
	Shift clock pulse: (D1BSCP)	Shift clock pulses; connect with SCP pin of segment driver. Driver latches data bus value by falling edge of this pin.	Chip enable for segment driver 1; connect with $\overline{CE}$ pin of segment driver 1.
	Latch pulse: (D2BLP)	Latch pulses output; connect with LP pin of segment/common driver. Display data is renewed in output register in LCD driver by rising edge of this pin.	Chip enable for segment driver 2; connect with $\overline{CE}$ pin of segment driver 2.
	Frame: (D3BFR)	LCD frame output; connect with FR pin of segment/common driver.	Chip enable for segment driver 3; connect with $\overline{CE}$ pin of segment driver 3.
	Cascade pulse: (DLEBCD)	Cascade pulses output; connect with DIO1 pin of row driver. These pin outputs 1 shot pulse by every D3BFR pin changes.	Chip enable for common driver; connect with $\overline{LE}$ pin of common driver.
	Display off: ($\overline{DOFF}$)	Display off output; connect with $\overline{DSPOF}$ terminal of segment/common driver. L means display off and H means display on.	

3.14.2 Block Diagram



Note: Row means common, and column means segment.

Figure 3.14.1 LCDC Block Diagram

3.14.3 SFR

LCD Mode Register

		7	6	5	4	3	2	1	0
LCDMODE (04B0H)	Bit symbol	BAE	AAE	SCPW1	SCPW0	−	BULK	RAMTYPE	MODE
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
	After reset	0	0	1	0	0	0	0	0
	Function	Used by B AREA 0: Disable 1: Enable	Used by A AREA 0: Disable 1: Enable	00: Base SCP 01: 2 clocks 10: 4 clocks 11: 8 clocks		Always write “0”.	SDRAM BULK 0: 64 Mbits 1: 128 Mbits	Display RAM 0: SRAM 1: SDRAM	LCD driver type selection 0: RAM 1: SR

Note 1: <BULK> is effective only if 1 is set to <RAMTYPE>.

Note 2: SCPW [1:0] is introduced in section 3.14.4.6.

Divide FrameRegister

		7	6	5	4	3	2	1	0
LCDDVM (04B1H)	Bit symbol	FMN7	FMN6	FMN5	FMN4	FMN3	FMN2	FMN1	FMN0
	Read/Write	R/W							
	After reset	0							
	Function	Setting Frame invert adjustment function bit7 to bit0							

LCD Size Setting Register

		7	6	5	4	3	2	1	0
LCDSIZE (04B2H)	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	Setting the LCD common number for SR mode 0000: 128 0001: 160 0010: 200 0011: 240 0100: 320 0101: 400 0110: 480 Other: Reserved				Setting the LCD segment number for SR type 0000: 128 0001: 160 0010: 240 0011: 320 0100: 400 0101: 480 0110: 560 0111: 640 Other: Reserved			

LCD Control Register

		7	6	5	4	3	2	1	0
LCDCTL (04B3H)	Bit symbol	LCDON	ALL0	FRMON	–	FP9	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	$\overline{\text{DOFF}}$ pin 0: Display OFF 1: Display ON	Transfer data of exclusive bus for LCD 0: Normal 1: All display data 0	Divided FR mode 0: Disable 1: Enable	Always write “0”.	Setting bit9 for f_{FP} [9:0]	Specify address of LCD driver with built-in RAM 0: Sequential 1: Random	Setting bit8 for f_{FP} [9:0]	Start control in SR type 0: STOP 1: START

LCDC start/stop bit

0	LCDC START
1	LCDC STOP

Type select for internal RAM LCD driver

0	Sequential access type (No address pin in LCD driver)
1	Random access type (Address pin in LCD driver)

Frame invert adjustment function

0	Disable
1	Enable

All data of exclusive bus for LCD (LD7:0)

0	Normal
1	All 0

Note: This bit forces sending data to LCD driver to 0 (Data off) by writing 1. Usually this bit is 0.

LCD Driver pin $\overline{\text{DOFF}}$

0	Driver OFF
1	Driver ON

Note: This bit determines the status of $\overline{\text{DOFF}}$ pin

0: $\overline{\text{DOFF}}$ pin outputs 0

1: $\overline{\text{DOFF}}$ pin outputs 1

LCD f_{FP} Register

		7	6	5	4	3	2	1	0
LCDFFP (04B4H)	Bit symbol	FP7	FP6	FP5	FP4	FP3	FP2	FP1	FP0
	Read/Write	R/W							
	After reset	0							
	Function	Setting bit7 to bit0 for f_{FP}							

LCD Gray Level Setting Register

		7	6	5	4	3	2	1	0
LCDGL (04B5H)	Bit symbol							GRAY1	GRAY0
	Read/Write							R/W	
	After reset							0	0
	Function							00: Monochrome 01: 4 levels 10: 8 levels 11: 16 levels	

Table 3.14.2 LCD Start/End Address Register

	Start Address Register			End Address Register		
	H (bit23 to bit16)	M (bit15 to bit8)	L (bit7 to bit0)	H (bit23 to bit16)	M (bit15 to bit8)	L (bit7 to bit0)
A-area	LSARAH (04C1H)	LSARAM (04C0H)	–	LEARAH (04C3H)	LEARAM (04C2H)	–
After reset	40H	00H		40H	00H	
B-area	LSARBH (04C5H)	LSARBM (04C4H)	–	LEARBH (04C7H)	LEARBM (04C6H)	–
After reset	40H	00H		40H	00H	
C-area	LSARCH (04CAH)	LSARCM (04C9H)	LSARCL (04C8H)	–	–	–
After reset	40H	00H	00H			

Note: All registers are available for R (Read)/W (Write).

LCD Cursor Setting Register

		7	6	5	4	3	2	1	0
LCDCM (04B6H)	Bit symbol	CDE	CCS					CBE1	CBE0
	Read/Write	R/W	R/W					R/W	R/W
	After reset	0	0					0	0
	Function	Cursor 0: OFF 1: ON	Cursor color 0: White 1: Black					Cursor blink interval (fs:32 kHz) 00: Don’t blink 01: 2 Hz 10: 1 Hz 11: 0.5 Hz	

Note 1: The function of cursor blink is effective only when low-frequency oscillator is input.

Note 2: The function of cursor blink depends on the low-frequency oscillator (fs) even if you use timer out (TA3OUT) as LCDCK.

LCD Cursor Width Setting Register

		7	6	5	4	3	2	1	0
LCDCW (04B7H)	Bit symbol				CW4	CW3	CW2	CW1	CW0
	Read/Write				R/W	R/W	R/W	R/W	R/W
	After reset				0	0	0	0	0
	Function				Cursor width 00000: 1 dot (Min) 11111: 32 dots (Max)				

LCD Cursor Height Setting Register

		7	6	5	4	3	2	1	0
LCDCH (04B8H)	Bit symbol				CH4	CH3	CH2	CH1	CH0
	Read/Write				R/W	R/W	R/W	R/W	R/W
	After reset				0	0	0	0	0
	Function				Cursor height 00000: 1 dot (Min) 11111: 32 dots (Max)				

LCD Cursor Start Address Setting Register

LCDCPL
(04BAH)

	7	6	5	4	3	2	1	0
Bit symbol	CAP7	CAP6	CAP5	CAP4	CAP3	CAP2	CAP1	CAP0
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	Setting bit7 to bit0 for cursor start address							

LCD Cursor Start Address Setting Register

LCDCPM
(04BBH)

	7	6	5	4	3	2	1	0
Bit symbol	CAP15	CAP14	CAP13	CAP12	CAP11	CAP10	CAP9	CAP8
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	Setting bit15 to bit8 for cursor start address							

LCD Cursor Start Address Setting Register

LCDCPH
(04BCH)

	7	6	5	4	3	2	1	0
Bit symbol	CAP23	CAP22	CAP21	CAP20	CAP19	CAP18	CAP17	CAP16
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	1	0	0	0	0	0	0
Function	Setting bit23 to bit16 for cursor start address							

LCD Cursor Hot Point Pixel Number (Bit correction) Setting Register

LCDCP
(04B9H)

	7	6	5	4	3	2	1	0
Bit symbol					APB3	APB2	APB1	APB0
Read/Write					R/W			
After reset					0			
Function					Setting bit 3 to bit0 of pixel for correction of hot point (for 1-dot correction)			

LCDC1L, LCDC1H, LCDC2L, LCDC2H, LCDC3L, LCDC3H, LCDR1L, LCDR1H 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 TMP91C820A. These ar and display registers of external built-in RAM sequential-type LCD driver.

Address as Table 3.14.3 is assigned to these registers, and the chip enable becomes active when accessing corresponding address.

And, the area of these address, $\overline{RD}$, $\overline{WR}$ terminal become active as act external access.

Table 3.14.4 shows the address map in the case of co access type LCD driver.

The explanation part of MMU also explains this.

This setup is performed by LCDCTL <MMULCD>.

Table 3.14.3 Memory Mapping for Built-in RAM Sequential Access Type

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

Table 3.14.4 Memory Mapping for Built-in RAM Random Access Type

Address	Purpose Random Access Type	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

Note 1: We call built-in RAM sequential access type LCD driver that use register to access to display RAM without address. (e.g., T6B65A, T6C84 etc: mar/2000)

Note 2: We call built-in RAM random access type LCD driver that is same method to access to SRAM. (e.g., T6C23, T6K01 etc: mar/2000)

3.14.4 Shift-Register Type LCD Driver Control Mode (SR type)

3.14.4.1 Operation

Set the mode of operation, start address of displ
level and LCD size to control registers before star
After start it LCDC outputs bus release request to
data memory. After that [illegible] to ext
driver through LCD perso[illegible]trol sig
etc.) connected LCD driver output specified wav
transmission. [illegible]n, LCDC cancels the b
and CPU will restart. [illegible]on wavefor
DLEBCD and D2BLP pins. LCDCK select TAOUT that be
frequency oscillator [illegible] setting
<TA3LCDE> register. [illegible] external reset
frequency oscillator (fs) is set.

Note: When set LCDC to SR type, during data reading (during DMA operation), CPU is stopped by internal BUSREQ signal. When using SR type LCDC, programmer need to care the CPU stop time. For detail, see the Table 3.14.8.

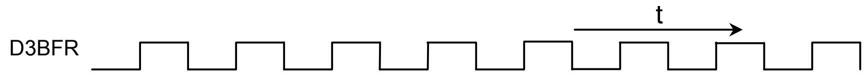
3.14.4.2 Gray Scale Mode Indication

Monochrome, 4-, 8- and 16-gray scale mode can be selected by sett
<GRAY1:0>.

And when SDRAM mode, you can select the size of SDRA
<BULK>.

TMP91C820A realize gray scale display by thinni
control palette is defined by 16-bit register (LGn
selected according to the gray scale level (Monoch
Table 3.14.6). On/off for data (e.g., each channel) by 16-
register (LGnL/H) respectively. However each palette has a initial
adjust finely which matches to LCD driver you use
panel.

Table 3.14.5 Gray Scale Control Palette Default Setting



Level Code	Density	Data Setting Register (Address/after reset)	bit0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
F	16/16	LGFH/L (04EF-E/FFFFH)	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
E	14/16	LGEH/L (04ED-C/FDFDH)	●	○	●	●	●	●	●	●	●	○	●	●	●	●	●	●
D	1316	LGDH/L (04EB-A/FDDDH)	●	○	●	●	●	○	●	●	●	○	●	●	●	●	●	●
C	12/16	LGCH/L (04E9-8/DDDDH)	●	○	●	●	●	○	●	●	●	○	●	●	●	○	●	●
B	11/16	LGBH/L (04E7-6/DDD5H)	●	○	●	○	●	○	●	●	●	○	●	●	●	○	●	●
A	10/16	LGAH/L (04E5-4/D5D5H)	●	○	●	○	●	○	●	●	●	○	●	○	●	○	●	●
9	9/16	LG9H/L (04E3-2/D555H)	●	○	●	○	●	○	●	○	●	○	●	○	●	○	●	●
8	8/16	LG8H/L (04E1-0/AAAAH)	○	●	○	●	○	●	○	●	○	●	○	●	○	●	○	●
7	7/16	LG7H/L (04DF-E/8AAAH)	○	●	○	●	○	●	○	●	○	●	○	●	○	○	○	●
6	6/16	LG6H/L (04DD-C/8A8AH)	○	●	○	●	○	○	○	●	○	●	○	●	○	○	○	●
5	5/16	LG5H/L (04DB-A/888AH)	○	●	○	●	○	○	○	●	○	○	○	●	○	○	○	●
4	4/16	LG4H/L (04D9-8/8888H)	○	○	○	●	○	○	○	●	○	○	○	●	○	○	○	●
3	3/16	LG3H/L (04D7-6/8880H)	○	○	○	○	○	○	○	●	○	○	○	●	○	○	○	●
2	2/16	LG2H/L (04D5-4/8080H)	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	●
1	1/16	LG1H/L (04D3-2/8000H)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●
0	0/16	LG0H/L (04D1-0/0000H)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

●: Display ON, ○: Display OFF

Table 3.14.6 Gray Scale Control Palette Effective Registers for Each Gray Level

	LG0 L/H	LG1 L/H	LG2 L/H	LG3 L/H	LG4 L/H	LG5 L/H	LG6 L/H	LG7 L/H	LG8 L/H	LG9 L/H	LGA L/H	LGB L/H	LGC L/H	LGD L/H	LGE L/H	LGE L/H
16-gray level	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
8-gray level	●	×	●	×	●	×	●	×	●	×	●	×	●	×	×	●
4-gray level	●	×	×	×	●	×	×	×	●	×	×	×	×	×	×	●
Monochrome	●															●

×: Don't care, ●: Effective

3.14.4.3 Memory Mapping

The LCDC can display the LCD panel divided horizon
parts; upper, middle and lower. Each area calls A
characteristics showing below.

Start/end address of each area in the physical mem
LCD start/end address registers (See Table 3.14.2
address.)

A and B areas are selectable enable or not in LCDMC
area are disable, the C area take over all panel spac

The displaying priority is A > B > C. If A area is set to enable while th
is defined as all C area (That is A and B area are dis
the LCD panel and A area is inserted from the top of t
area set to enable while the panel area is defined as
the bottom of the C area overlapping.

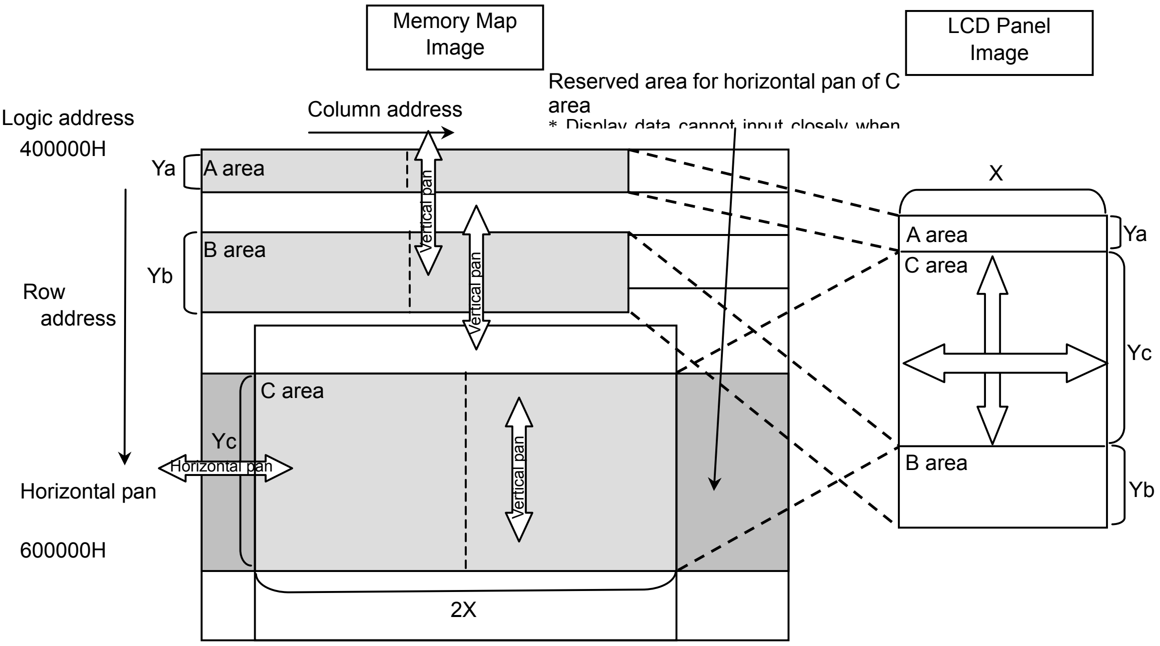


Figure 3.14.2 Memory Mapping from Physical Memory to LCD Panel

- Display memory mapping and panning function

LCDC can change the panel window if only you ch
A and B area can be vertical panned by changing
can be vertical and horizontal panned by changing row and column

An important thing is that display data from
cannot be input continuously even if you don't use the panning fu
row address of display RAM corresponds to 1st
display data of 2nd line cannot be set within t
RAM even if the necessary data for display does not fill
capacity of 1st row address of display RAM. Ad
panel is equal to assigned to row address of display R

And another important thing is, this limitati
RAM without address multiplex. When you use SDI
can select the size for display RAM capacity of
SRAM, display RAM capacity is fixed to 512 bytes.

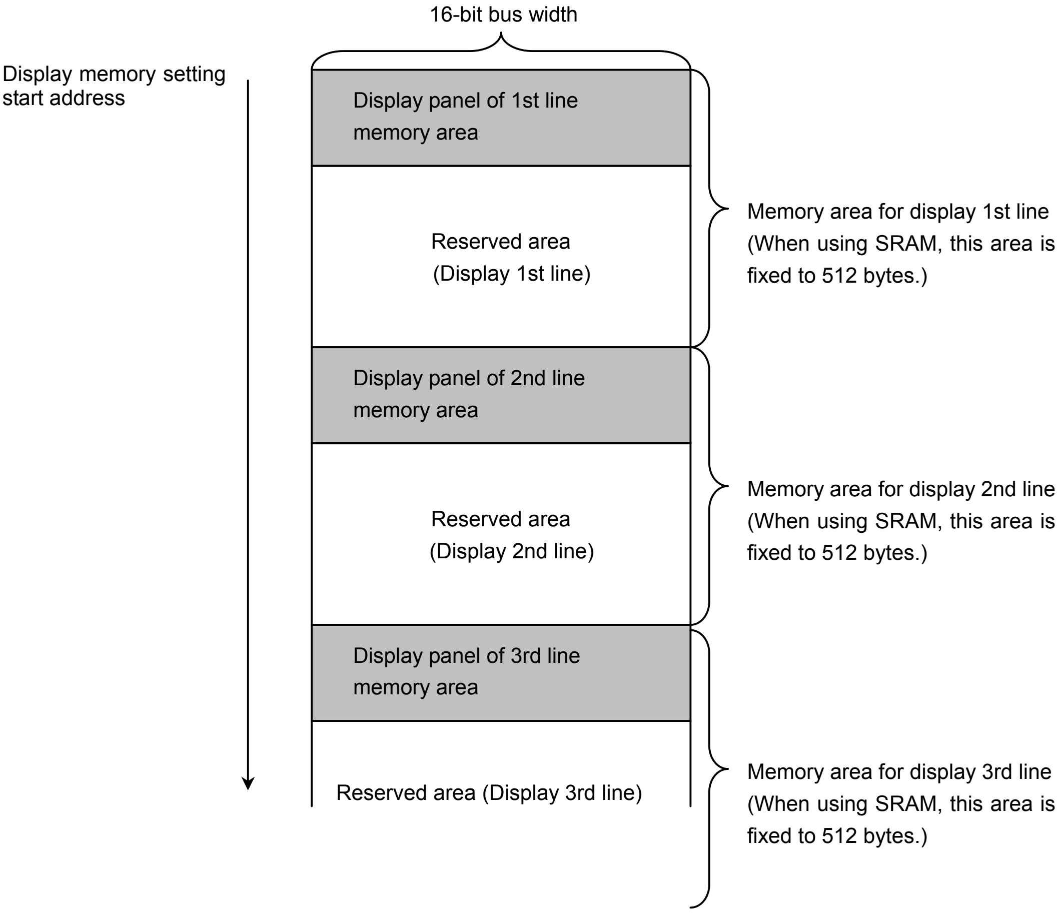

16-bit bus width
Display memory setting start address
Display panel of 1st line memory area
Reserved area (Display 1st line)
Memory area for display 1st line (When using SRAM, this area is fixed to 512 bytes.)
Display panel of 2nd line memory area
Reserved area (Display 2nd line)
Memory area for display 2nd line (When using SRAM, this area is fixed to 512 bytes.)
Display panel of 3rd line memory area
Reserved area (Display 3rd line)
Memory area for display 3rd line (When using SRAM, this area is fixed to 512 bytes.)

Figure 3.14.3 Memory Mapping Image for SRAM as Display RAM (Only A and B area)

TMP91C820A can select four display mode; monochrome, 4 gray, 8 gray a With the intrinsic property of gray levels, a pixe different memory size.

A pixel use a bit in memory for monochrome, 2 bits in m gray, 3 bits for 8 gray and 4 bits for 16 gray. So display memory has som differences for each gray scale of memory.

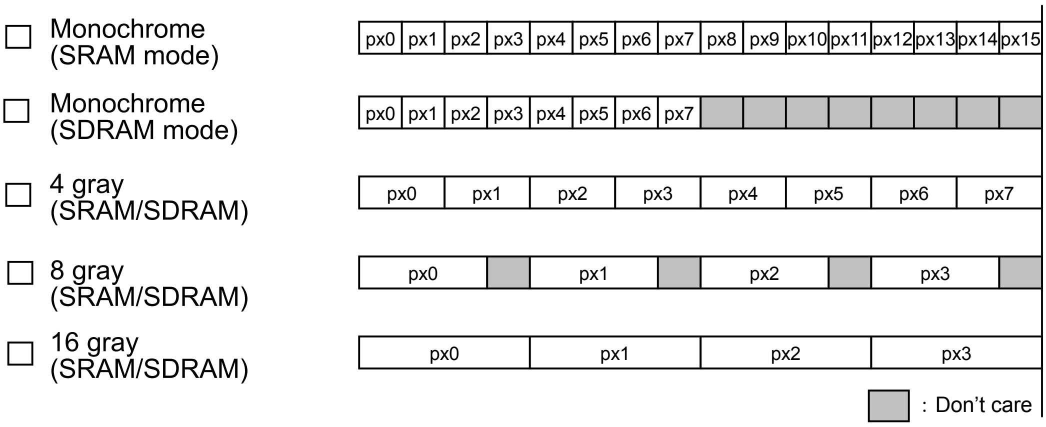

□ Monochrome (SRAM mode)
px0 px1 px2 px3 px4 px5 px6 px7 px8 px9 px10 px11 px12 px13 px14 px15
□ Monochrome (SDRAM mode)
px0 px1 px2 px3 px4 px5 px6 px7
□ 4 gray (SRAM/SDRAM)
px0 px1 px2 px3 px4 px5 px6 px7
□ 8 gray (SRAM/SDRAM)
px0 px1 px2 px3
□ 16 gray (SRAM/SDRAM)
px0 px1 px2 px3
▨ : Don’t care

Figure 3.14.4 Gray display and using bit in memory

And “px” in above Figure 3.14.4 corresponds to the image of LCD panel (Figure 3.14.5). But TMP91C820A outputs data of px0 PE0 (LD0). Therefore PE0 (LD0) should be connected DI7) according to LCD driver you use. Please note th the data differs from LCD controller in TLCS-900/L1 series of TOS TMP91C815, TMP91C016, TMP91C025 etc.).

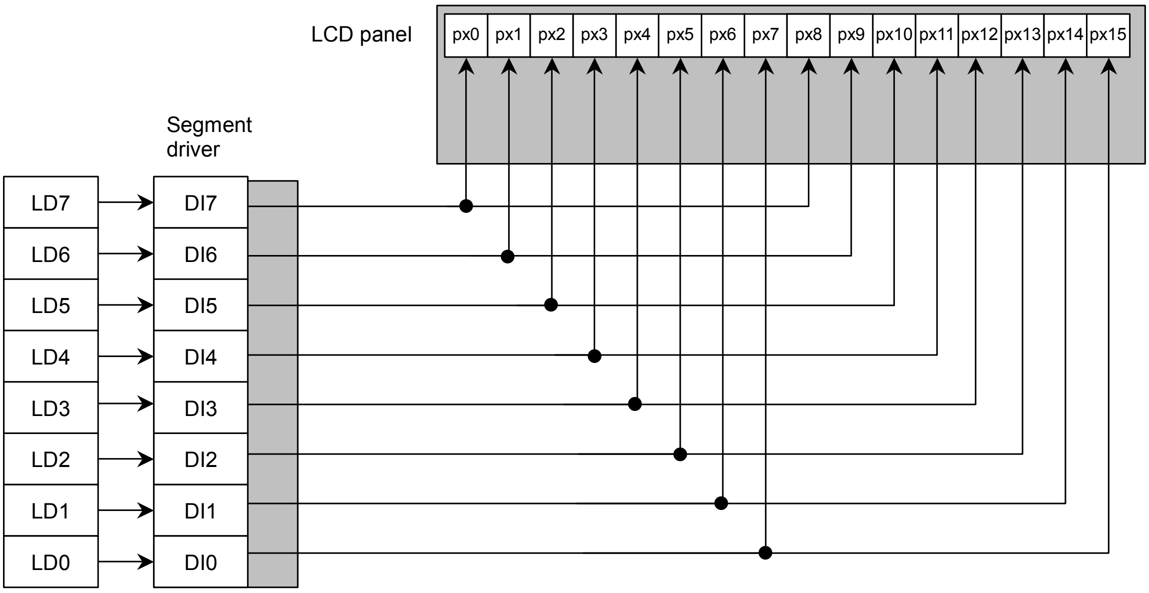

LCD panel
px0 px1 px2 px3 px4 px5 px6 px7 px8 px9 px10 px11 px12 px13 px14 px15
Segment driver
LD7 LD6 LD5 LD4 LD3 LD2 LD1 LD0
DI7 DI6 DI5 DI4 DI3 DI2 DI1 DI0

Figure 3.14.5 Connection between LCD Bus of TMP91C820A and Data Bus of LCDD

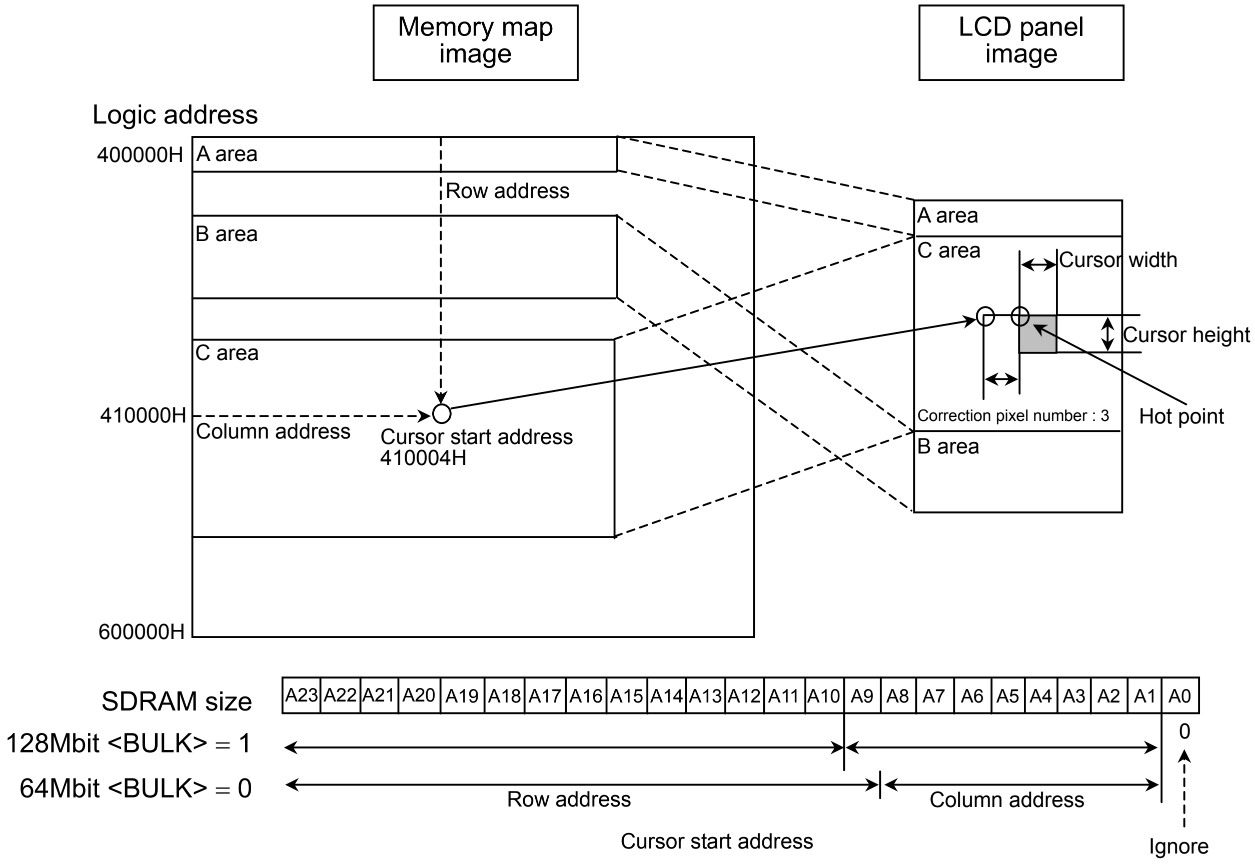
3.14.4.4 Hardware Cursor

TMP91C820A has a cursor that is valid, blinking and size ca
and maximum size is 32X32.

A programmer can control the cursor attributes
registers, for example color (white/black), blin
pixel location. Its space location is specified b
3.14.6.)

The precise location of the hot point is determine
LCDCPM, LCDCPL) and pixel correction number (LCDCP). Fo
pixel for display is fixed setting 2 bit data under 4-gray mode
location of hot point every pixel by setting pixel n
resister (LCDCP).

Cursor image is showed under three area A, B, C setting able, 4-
start address=410004_hex and correction number=3_hex in (LCDCP) follo
figure.



Note: TMP91C820A sets the hardware cursor in the memory address. If panning function is set to enable during hardware cursor displaying, the cursor in the pannel moves, but start address of cursor is not changed.

Figure 3.14.6 Cursor Hot Point Position and Size, Cursor start address

LCD Cursor Setting Register

LCDCM
(04B6H)

	7	6	5	4	3	2	1	0
Bit symbol	CDE	CCS					CBE1	CBE0
Read/Write	R/W	R/W					R/W	R/W
After reset	0	0					0	0
Function	Cursor 0: OFF 1: ON	Cursor color 0: White 1: Black					Cursor blink interval (fs: 32 kHz) 00: Don't blink 01: 2 Hz 10: 1 Hz 11: 0.5 Hz	

Note 1: The function of cursor blink is effective only when low-frequency osdillator is input.

Note 2: The function of cursor blink depends on the low-frequency oscillator (fs) even if you use timer out "TA3OUT" as LCDCK.

LCD Cursor Width Setting Register

LCDCW
(04B7H)

	7	6	5	4	3	2	1	0
Bit symbol				CW4	CW3	CW2	CW1	CW0
Read/Write				R/W	R/W	R/W	R/W	R/W
After reset				0	0	0	0	0
Function				Cursor width 00000: 1 dot (MIN) 11111: 32 dots (MAX)				

LCD Cursor Height Setting Register

LCDCH
(04B8H)

	7	6	5	4	3	2	1	0
Bit symbol				CH4	CH3	CH2	CH1	CH0
Read/Write				R/W	R/W	R/W	R/W	R/W
After reset				0	0	0	0	0
Function				Cursor height 00000: 1 dot (MIN) 11111: 32 dots (MAX)				

LCD Cursor Start Address Setting Register

LCDCPL
(04BAH)

	7	6	5	4	3	2	1	0
Bit symbol	CAP7	CAP6	CAP5	CAP4	CAP3	CAP2	CAP1	CAP0
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	Setting bit7 to bit0 for cursor start address							

LCD Cursor Start Address Setting Register

LCDCPM
(04BBH)

	7	6	5	4	3	2	1	0
Bit symbol	CAP15	CAP14	CAP13	CAP12	CAP11	CAP10	CAP9	CAP8
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	Setting bit15 to bit8 for cursor start address							

LCD Cursor Start Address Setting Register

LCDCPH
(04BCH)

	7	6	5	4	3	2	1	0
Bit symbol	CAP23	CAP22	CAP21	CAP20	CAP19	CAP18	CAP17	CAP16
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	1	0	0	0	0	0	0
Function	Setting bit23 to bit16 for cursor start address							

LCD Cursor Hot Point Pixel correction Setting Register

		7	6	5	4	3	2	1	0
LCDCP (04B9H)	Bit symbol					APB3	APB2	APB1	APB0
	Read/Write					R/W			
	After reset					0			
	Function					Setting bit3 to bit0 for correction of hot point (for 1-dot correction)			

Table 3.14.7 Pixel correct and register setting

In case of monochrome (SRAM mode)	0000: 0 Pixel correct 0111: 7 Pixels correct 1111: 15 Pixels correct	
In case of monochrome (SDRAM mode) and 4 gray (SRAM/SDRAM mode)	x000: 0 Pixel correct x001: 1 Pixel correct x010: 2 Pixels correct x011: 3 Pixels correct	x100: 4 Pixels correct x101: 5 Pixels correct x110: 6 Pixels correct x111: 7 Pixels correct
In case of 8 gray and 16 gray (SRAN/SDRAM mode)	xx00: 0 Pixel correct xx01: 1 Pixel correct	xx10: 2 Pixels correct xx11: 3 Pixels correct

X: Don't care

Here, it is possible to correct pixel from the address set bef
Pixel number should be adjusted in response to the g

For example, When 4-gray and 16-bit BUS mode, cor
pixel because the smallest pixel that can set is 8 pixel by start add
Similarly correction pixel 15 should be less than at monochrome mode, 3-
16-gray mode.

(e.g.) When monochrome mode, correction value = 0011 (LCDCP) curs
size = (8 × 8)

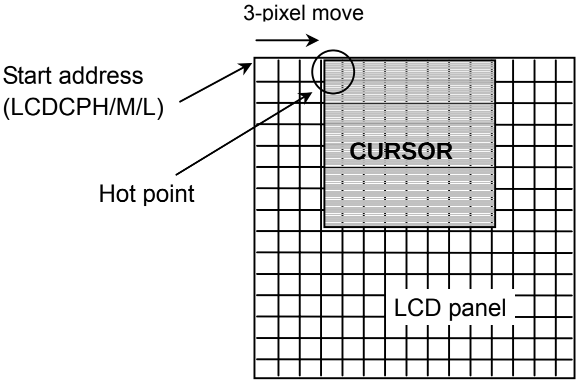


Figure 3.14.7 The Location Hot Point by Setting of Pixel

3.14.4.5 Frame Signal Settlement

TMP91C820A defines so called frame period (Refres
value set in f_{FP}[9:0]. DLEBCD pin outputs pulse every fra
usually outputs the signal inverts polarity every
And TMP91C820A has a special function that can set
polarity irrelevant to above frame frequency for t
of display.

LCD Control Register

LCDCTL
(04B3H)

	7	6	5	4	3	2	1	0
Bit symbol	LCDON	ALL0	FRMON	–	FP9	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	$\overline{\text{DOFF}}$ port 0: Display OFF 1: Display ON	Setting all column ports to 0 0: Normal 1: All display data 0	Divided FR mode 0: Disable 1: Enable	Always write "0".	Setting bit9 for f_{FP} [9:0]	Specify address of LCD driver with built-in RAM 0: OFF 1: ON	Setting bit8 for f_{FP} [9:0]	Start control in SR mode 0: Stop 1: Start

LCD f_{FP} Register

LCDFFP
(04B4H)

	7	6	5	4	3	2	1	0
Bit symbol	FP7	FP6	FP5	FP4	FP3	FP2	FP1	FP0
Read/Write	R/W							
After reset	0							
Function	Setting bit7 to bit0 for f_{FP}							

Divide FRM Register

LCDDVM
(04B1H)

	7	6	5	4	3	2	1	0
Bit symbol	FMN7	FMN6	FMN5	FMN4	FMN3	FMN2	FMN1	FMN0
Read/Write	R/W							
After reset	0							
Function	Setting DVM bit7 to bit0							

(1) Frame frequency setting

Basic frame period; DLEBCD signal is made according to fFP[9:0] setting. Here is s
setting mentioned. However, when fFP[9:0] setting is generally e
number, frame period can be corrected by fFP[9:0] setting increase. This
cannot correct frame frequency by Table 3.14.8. That is ne
frame frequency higher or detailed, please refer to

The equation can calculate frame period.

Frame period = LCDCK / (fFP × D) [Hz]   D: Constant for each commo

fFP: Setting fFP[9:0] register

LCDCK: Source clock of LCD

(Low clock is usually selecte

Please select the value of fFP[9:0] as the frame period you want
3.14.7.

Note: Please make the value set to f_{FP} [9:0] into the following range.

COM (common number) ≤ fFP ≤ 1024

(e.g.) In the case where frame period is set to 72.1

fFP = 240 (COM) + 63 = 303 = 12FH (by Table 3.14.8)

Therefore, LCDCTL<FP8> = 1 and LCDFFP<FP7:0> = 2FH are set up.

(2) Frame invert adjustment function

This mode can prevent the deterioration of display (*Note)
If N is set in (LCDDVM) register while this functi
(LCDCTL)(<FRMON> "1"), D3BFR pin outputs the sig
(D2BLP x N) timing.

If this function isn't necessary, D3BFR pin output
frequency of DLEBCD pin if this function is disable (= (LCDCT
0).

And it is no change waveform of DLEBCD pin by LCDDVM setti

Note: Effects of this function have some differences as the LCD driver or LCD panel you use actually.

(3) Timer out LCDCK

LCD source clock (LCDCK) can use low frequency (fs: 32.768 [kHz]) (TA3OUT) outputs from internal TMRA23.

(e.g.) Here indicates the method that frame peri TA3OUT for source clock (fc = 16 [MHz], LCD 128 (COM))

The next equation calculates frame period.

Frame period (f_{FP}) = $1/(t_{LP} \times 128)$ [Hz] t_{LP}: The period of D2BLP

Source clock for LCDC defines as XT [Hz] and then th

$t_{LP} = D/XT$ D: The value is 3 at 128 C

Therefore if you set the frame period at 70 [Hz] und

XT = 128 × 3 × 70

= 26880 [Hz]

XT should be above value.

In order to make XT = 26880 [Hz] under fc = 16 [MHz] with φT1 of timer 3,

1/XT = T3 × 2 × 8 / fc [s] T3: The value of timer re

in short, T3 = fc/(XT × 2 × 8) [Hz]

However, T3 (TA3REG) is 13.95 after calculate, it's i under a decimal point.

So if T3 (TA3REG) is set 0DH, XT = 28846 [Hz]. And because of D = 3,

Frame period = 28846 / (3 × 128)

= 75.12 [Hz]

Further, if fp is 136 (+8COM) with correction,

Frame period = 28846 / (3 × 136)

= 70.70 [Hz]

Reference: To maintain quality for display, pleas gray scale.

(You have to use frame frequency setting adjustment fu timer out LCDCK.)

Monochrome: Frame period = 70 [Hz]

4 or 8 or 16 gray: Frame period = 140 [Hz]

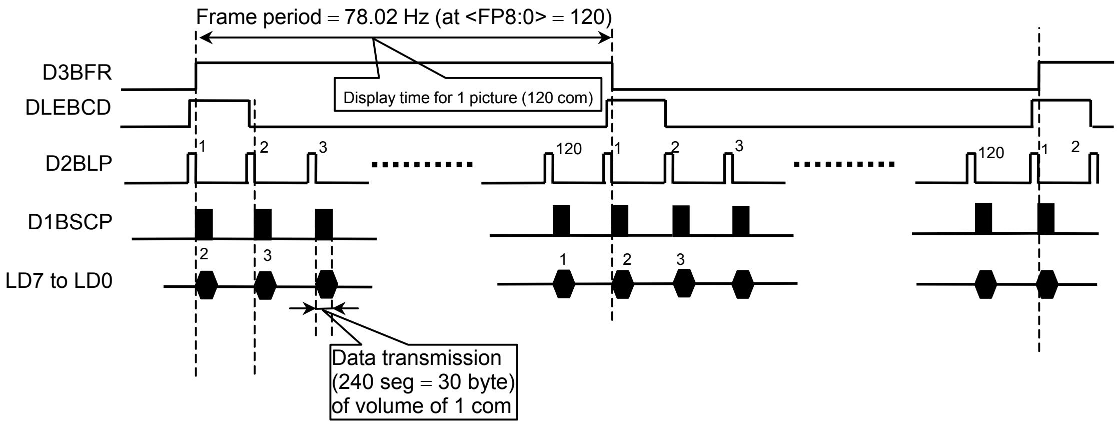

Frame period = 78.02 Hz (at <FP8:0> = 120)
D3BFR
DLEBCD
Display time for 1 picture (120 com)
D2BLP
1
2
3
120
1
2
3
120
1
2
D1BSCP
LD7 to LD0
2
3
1
2
3
Data transmission (240 seg = 30 byte) of volume of 1 com

Figure 3.14.8 Timing Diagram for SR Mode

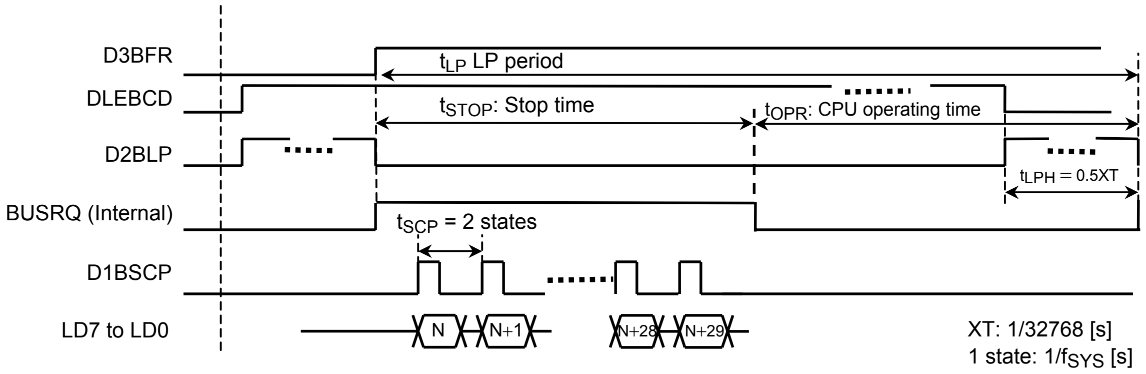


Figure 3.14.9 Timing Diagram for SR Mode (Detail)

D3BFR waveform (in case of [illegible] and LCDDVM [illegible])>

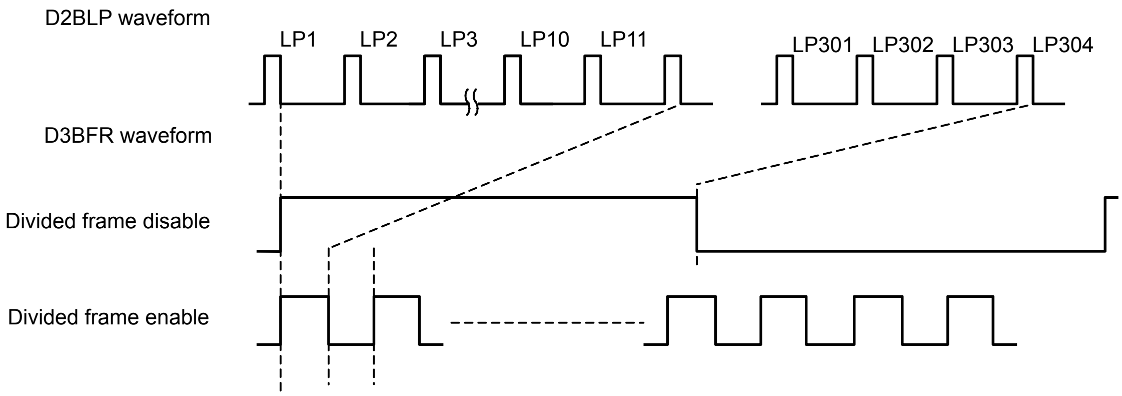


Figure 3.14.10 D2BLP and D3BFR Waveform

Table 3.14.8 f_{FP} Table for Each Common Number (1/2)

D	3	2.5	2	1.5	1.5	1	1	Unit
COM number	128	160	200	240	320	400	480	COM
COM + 0	85.33	81.92	81.92	91.02	68.27	81.92	68.27	Hz
COM + 1	84.67	81.41	81.51	90.64	68.05	81.72	68.12	
COM + 2	84.02	80.91	81.11	90.27	67.84	81.51	67.98	
COM + 3	83.38	80.41	80.71	89.90	67.63	81.31	67.84	
COM + 4	82.75	79.92	80.31	89.53	67.42	81.11	67.70	
COM + 5	82.13	79.44	79.92	89.16	67.22	80.91	67.56	
COM + 6	81.51	78.96	79.53	88.80	67.01	80.71	67.42	
COM + 7	80.91	78.49	79.15	88.44	66.81	80.51	67.29	
COM + 8	80.31	78.02	78.77	88.09	66.60	80.31	67.15	
COM + 9	79.73	77.56	78.39	87.73	66.40	80.12	67.01	
COM + 10	79.15	77.10	78.02	87.38	66.20	79.92	66.87	
COM + 11	78.58	76.65	77.65	87.03	66.00	79.73	66.74	
COM + 12	78.02	76.20	77.28	86.69	65.80	79.53	66.60	
COM + 13	77.47	75.76	76.92	86.35	65.60	79.34	66.47	
COM + 14	76.92	75.33	76.56	86.01	65.41	79.15	66.33	
COM + 15	76.38	74.90	76.20	85.67	65.21	78.96	66.20	
COM + 16	75.85	74.47	75.85	85.33	65.02	78.77	66.06	
COM + 17	75.33	74.05	75.50	85.00	64.82	78.58	65.93	
COM + 18	74.81	73.64	75.16	84.67	64.63	78.39	65.80	
COM + 19	74.30	73.22	74.81	84.34	64.44	78.21	65.67	
COM + 20	73.80	72.82	74.47	84.02	64.25	78.02	65.54	
COM + 21	73.31	72.42	74.14	83.70	64.06	77.83	65.41	
COM + 22	72.82	72.02	73.80	83.38	63.88	77.65	65.27	
COM + 23	72.34	71.62	73.47	83.06	63.69	77.47	65.15	
COM + 24	71.86	71.23	73.14	82.75	63.50	77.28	65.02	
COM + 25	71.39	70.85	72.82	82.44	63.32	77.10	64.89	
COM + 26	70.93	70.47	72.50	82.13	63.14	76.92	64.76	
COM + 27	70.47	70.09	72.18	81.82	62.95	76.74	64.63	
COM + 28	70.02	69.72	71.86	81.51	62.77	76.56	64.50	
COM + 29	69.57	69.35	71.55	81.21	62.59	76.38	64.38	
COM + 30	69.13	68.99	71.23	80.91	62.42	76.20	64.25	
COM + 31	68.70	68.62	70.93	80.61	62.24	76.03	64.13	
COM + 32	68.27	68.27	70.62	80.31	62.06	75.85	64.00	
COM + 33	67.84	67.91	70.32	80.02	61.88	75.68	63.88	
COM + 34	67.42	67.56	70.02	79.73	61.71	75.50	63.75	
COM + 35	67.01	67.22	69.72	79.44	61.54	75.33	63.63	
COM + 36	66.60	66.87	69.42	79.15	61.36	75.16	63.50	
COM + 37	66.20	66.53	69.13	78.86	61.19	74.98	63.38	
COM + 38	65.80	66.20	68.84	78.58	61.02	74.81	63.26	
COM + 39	65.41	65.87	68.55	78.30	60.85	74.64	63.14	
COM + 40	65.02	65.54	68.27	78.02	60.68	74.47	63.02	
COM + 41	64.63	65.21	67.98	77.74	60.51	74.30	62.89	
COM + 42	64.25	64.89	67.70	77.47	60.35	74.14	62.77	
COM + 43	63.88	64.57	67.42	77.19	60.18	73.97	62.65	
COM + 44	63.50	64.25	67.15	76.92	60.01	73.80	62.53	
COM + 45	63.14	63.94	66.87	76.65	59.85	73.64	62.42	
COM + 46	62.77	63.63	66.60	76.38	59.69	73.47	62.30	
COM + 47	62.42	63.32	66.33	76.12	59.52	73.31	62.18	
COM + 48	62.06	63.02	66.06	75.85	59.36	73.14	62.06	
COM + 49	61.71	62.71	65.80	75.59	59.20	72.98	61.94	
COM + 50	61.36	62.42	65.54	75.33	59.04	72.82	61.83	
COM + 51	61.02	62.12	65.27	75.07	58.88	72.66	61.71	

Table 3.14.9 f_{FP} Table for Each Common Number (2/2)

D	3	2.5	2	1.5	1.5	1	1	Unit
COM number	128	160	200	240	320	400	480	COM
COM + 52	60.68	61.83	65.02	74.81	58.72	72.50	61.59	Hz
COM + 53	60.35	61.54	64.76	74.56	58.57	72.34	61.48	
COM + 54	60.01	61.25	64.50	74.30	58.41	72.18	61.36	
COM + 55	59.69	60.96	64.25	74.05	58.25	72.02	61.25	
COM + 56	59.36	60.68	64.00	73.80	58.10	71.86	61.13	
COM + 57	59.04	60.40	63.75	73.55	57.95	71.70	61.02	
COM + 58	58.72	60.12	63.50	73.31	57.79	71.55	60.91	
COM + 59	58.41	59.85	63.26	73.06	57.64	71.39	60.79	
COM + 60	58.10	59.58	63.02	72.82	57.49	71.23	60.68	
COM + 61	57.79	59.31	62.77	72.58	57.34	71.08	60.57	
COM + 62	57.49	59.04	62.53	72.34	57.19	70.93	60.46	
COM + 63	57.19	58.78	62.30	72.10	57.04	70.77	60.35	
COM + 64	56.89	58.51	62.06	71.86	56.89	70.62	60.24	
COM + 65	56.59	58.25	61.83	71.62	56.74	70.47	60.12	
COM + 66	56.30	58.00	61.59	71.39	56.59	70.32	60.01	
COM + 67	56.01	57.74	61.36	71.16	56.45	70.17	59.90	
COM + 68	55.73	57.49	61.13	70.93	56.30	70.02	59.80	
COM + 69	55.45	57.24	60.91	70.70	56.16	69.87	59.69	
COM + 70	55.16	56.99	60.68	70.47	56.01	69.72	59.58	
COM + 71	54.89	56.74	60.46	70.24	55.87	69.57	59.47	
COM + 72	54.61	56.50	60.24	70.02	55.73	69.42	59.36	
COM + 73	54.34	56.25	60.01	69.79	55.59	69.28	59.25	
COM + 74	54.07	56.01	59.80	69.57	55.45	69.13	59.15	
COM + 75	53.81	55.78	59.58	69.35	55.30	68.99	59.04	
COM + 76	53.54	55.54	59.36	69.13	55.16	68.84	58.94	
COM + 77	53.28	55.30	59.15	68.91	55.03	68.70	58.83	
COM + 78	53.02	55.07	58.94	68.70	54.89	68.55	58.72	
COM + 79	52.77	54.84	58.72	68.48	54.75	68.41	58.62	
COM + 80	52.51	54.61	58.51	68.27	54.61	68.27	58.51	

Note: Above value is at fs = 32.768 [kHz].

Table 3.14.10 Performance Listing for Each Segment and Common Number

64-Mbit SDRAM/BURST 4 GRAY

		COM	128	160	200	240	320	400	480	
	D		3	3	2	2	2	1	1	Unit
	t_{LP}		91.6	76.3	61.0	45.8	45.8	30.5	30.5	μs
SEG										
128	t_{STOP}		0.89	0.89	0.89	0.89	0.89	0.89	0.89	μs
	rate		0.97	1.17	1.46	1.94	1.94	2.91	2.91	%
160	t_{STOP}		1.11	1.11	1.11	1.11	1.11	1.11	1.11	μs
	rate		1.21	1.46	1.82	2.43	2.43	3.64	3.64	%
240	t_{STOP}		1.67	1.67	1.67	1.67	1.67	1.67	1.67	μs
	rate		1.82	2.18	2.73	3.64	3.64	5.46	5.46	%
320	t_{STOP}		2.22	2.22	2.22	2.22	2.22	2.22	2.22	μs
	rate		2.43	2.91	3.64	4.85	4.85	7.28	7.28	%
400	t_{STOP}		2.78	2.78	2.78	2.78	2.78	2.78	2.78	μs
	rate		3.03	3.64	4.55	6.07	6.07	9.10	9.10	%
480	t_{STOP}		3.33	3.33	3.33	3.33	3.33	3.33	3.33	μs
	rate		3.64	4.37	5.46	7.28	7.28	10.92	10.92	%
560	t_{STOP}		3.89	3.89	3.89	3.89	3.89	3.89	3.89	μs
	rate		4.25	5.10	6.37	8.50	8.50	12.74	12.74	%
640	t_{STOP}		4.44	4.44	4.44	4.44	4.44	4.44	4.44	μs
	rate		4.85	5.83	7.28	9.71	9.71	14.56	14.56	%

64-Mbit SDRAM/BURST 8 GRAY/16 GRAY

		COM	128	160	200	240	320	400	480	
	D		3	3	2	2	2	1	1	Unit
	t_{LP}		91.6	76.3	61.0	45.8	45.8	30.5	30.5	
SEG										μs
128	t_{STOP}		2.11	2.11	2.11	2.11	2.11	2.11	2.11	μs
	rate		2.31	2.77	3.46	4.61	4.61	6.92	6.92	%
160	t_{STOP}		2.56	2.56	2.56	2.56	2.56	2.56	2.56	μs
	rate		2.79	3.35	4.19	5.58	5.58	8.37	8.37	%
240	t_{STOP}		3.67	3.67	3.67	3.67	3.67	3.67	3.67	μs
	rate		4.00	4.81	6.01	8.01	8.01	12.01	12.01	%
320	t_{STOP}		4.78	4.78	4.78	4.78	4.78	4.78	4.78	μs
	rate		5.22	6.26	7.83	10.44	10.44	15.66	15.66	%
400	t_{STOP}		5.89	5.89	5.89	5.89	5.89	5.89	5.89	μs
	rate		6.43	7.72	9.65	12.86	12.86	19.30	19.30	%
480	t_{STOP}		7.00	7.00	7.00	7.00	7.00	7.00	7.00	μs
	rate		7.65	9.18	11.47	15.29	15.29	22.94	22.94	%
560	t_{STOP}		8.11	8.11	8.11	8.11	8.11	8.11	8.11	μs
	rate		8.86	10.63	13.29	17.72	17.72	26.58	26.58	%
640	t_{STOP}		9.22	9.22	9.22	9.22	9.22	9.22	9.22	μs
	rate		10.07	12.09	15.11	20.15	20.15	30.22	30.22	%

SRAM MONOCHROME

		COM	128	160	200	240	320	400	480	
	D		3	3	2	2	2	1	1	Unit
	t_{LP}		91.6	76.3	61.0	45.8	45.8	30.5	30.5	µs
SEG										
128	t_{STOP}		0.89	0.89	0.89	0.89	0.89	0.89	0.89	µs
	rate		0.99	1.17	1.46	1.94	1.94	2.91	2.91	%
160	t_{STOP}		1.11	1.11	1.11	1.11	1.11	1.11	1.11	µs
	rate		1.21	1.46	1.82	2.43	2.43	3.64	3.64	%
240	t_{STOP}		1.67	1.67	1.67	1.67	1.67	1.67	1.67	µs
	rate		1.82	2.18	2.73	3.64	3.64	5.46	5.46	%
320	t_{STOP}		2.22	2.22	2.22	2.22	2.22	2.22	2.22	µs
	rate		2.43	2.91	3.64	4.85	4.85	7.28	7.28	%
400	t_{STOP}		2.78	2.78	2.78	2.78	2.78	2.78	2.78	µs
	rate		3.03	3.64	4.55	6.07	6.07	9.10	9.10	%
480	t_{STOP}		3.33	3.33	3.33	3.33	3.33	3.33	3.33	µs
	rate		3.64	4.37	5.46	7.28	7.28	10.92	10.92	%
560	t_{STOP}		3.89	3.89	3.89	3.89	3.89	3.89	3.89	µs
	rate		4.25	5.10	6.37	8.50	8.50	12.74	12.74	%
640	t_{STOP}		4.44	4.44	4.44	4.44	4.44	4.44	4.44	µs
	rate		4.85	5.83	7.28	9.71	9.71	14.56	14.56	%

SRAM 4 GRAY

		COM	128	160	200	240	320	400	480	
	D		3	3	2	2	2	1	1	Unit
	t_{LP}		91.6	76.3	61.0	45.8	45.8	30.5	30.5	µs
SEG										
128	t_{STOP}		1.78	1.78	1.78	1.78	1.78	1.78	1.78	µs
	rate		1.94	2.33	2.91	3.88	3.88	5.83	5.83	%
160	t_{STOP}		2.22	2.22	2.22	2.22	2.22	2.22	2.22	µs
	rate		2.43	2.91	3.64	4.85	4.85	7.28	7.28	%
240	t_{STOP}		3.33	3.33	3.33	3.33	3.33	3.33	3.33	µs
	rate	.	3.64	4.37	5.46	7.28	7.28	10.92	10.92	%
320	t_{STOP}		4.44	4.44	4.44	4.44	4.44	4.44	4.44	µs
	rate		4.85	5.83	7.28	9.71	9.71	14.55	14.56	%
400	t_{STOP}		5.56	5.56	5.56	5.56	5.56	5.56	5.56	µs
	rate		6.07	7.28	9.10	12.14	12.14	18.20	18.20	%
480	t_{STOP}		6.67	6.67	6.67	6.67	6.67	6.67	6.67	µs
	rate		7.28	8.74	10.92	14.56	14.56	21.85	21.85	%
560	t_{STOP}		7.78	7.78	7.78	7.78	7.78	7.78	7.78	µs
	rate		8.50	10.19	12.74	16.99	16.99	25.49	25.49	%
640	t_{STOP}		8.89	8.89	8.89	8.89	8.89	8.89	8.89	µs
	rate		9.71	11.65	14.56	19.42	19.42	29.13	29.13	%

SRAM 8 GRAY/16 GRAY

		COM	128	160	200	240	320	400	480	
	D		3	3	2	2	2	1	1	Unit
	t_{LP}		91.6	76.3	61.0	45.8	45.8	30.5	30.5	μs
SEG										
128	t_{STOP}		3.56	3.56	3.56	3.56	3.56	3.56	3.56	μs
	rate		3.88	4.66	5.83	7.77	7.77	11.65	11.65	%
160	t_{STOP}		4.44	4.44	4.44	4.44	4.44	4.44	4.44	μs
	rate		4.85	5.83	7.28	9.71	9.71	14.56	14.56	%
240	t_{STOP}		6.67	6.67	6.67	6.67	6.67	6.67	6.67	μs
	rate		7.28	8.74	10.92	14.56	14.56	21.85	21.85	%
320	t_{STOP}		8.89	8.89	8.89	8.89	8.89	8.89	8.89	μs
	rate		9.71	11.65	14.56	19.42	19.42	29.13	29.13	%
400	t_{STOP}		11.11	11.11	11.11	11.11	11.11	11.11	11.11	μs
	rate		12.14	14.56	18.20	24.27	24.27	36.41	36.41	%
480	t_{STOP}		13.33	13.33	13.33	13.33	13.33	13.33	13.33	μs
	rate		14.56	17.48	21.85	29.13	29.13	43.69	43.69	%
560	t_{STOP}		15.56	15.56	15.56	15.56	15.56	15.56	15.56	μs
	rate		16.99	20.39	25.49	33.98	33.98	50.97	50.97	%
640	t_{STOP}		17.78	17.78	17.78	17.78	17.78	17.78	17.78	μs
	rate		19.42	23.30	29.13	38.84	38.84	58.25	58.25	%

Over 50%

Table 3.14.11 Possible Panel Size of Panning

Note 1: The value of the Table 3.14.8 is at $f_C = 36$ [MHz].

Note 2: Bus occupation time to CPU; t_{STOP} (in the Figure 3.14.11) is the time which CPU reads the memory of transferring with 0 waits.

Note 3: The following equation can calculate t_{LP} listed below.

$t_{LP} = D/32768$ [s]

(e.g.) If the row is 240 and D = 1.5 by the above table

$t_{LP} = 1.5/32768 = 45.8$ [μs]

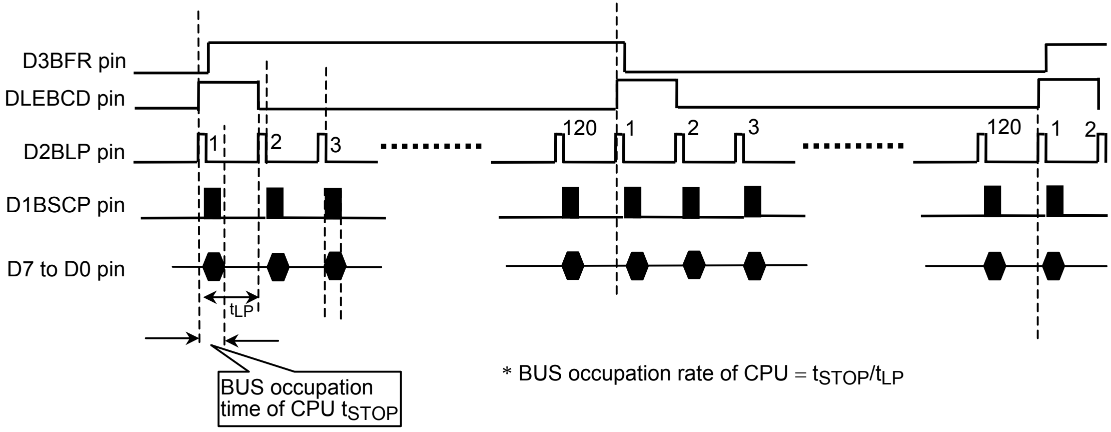

D3BFR pin
DLEBCD pin
D2BLP pin
D1BSCP pin
D7 to D0 pin
1
2
3
120
1
2
3
120
1
2
tLP
BUS occupation time of CPU tSTOP
* BUS occupation rate of CPU = tSTOP/tLP

Figure 3.14.11 Bus occupation time to CPU and BUS Occupation Rate of CPU

3.14.4.6 Timing Charts of Interpreting Memory Codes

TMP91C820A supports different memory. They are SRAM wi SDRAM burst modes, and the size of SDRAM is 16M/64 signals for the LCD panel are shown in Figure 3.14.12. To control the low speed 3 types of SCP ($f_{SYS}/2$, $f_{SYS}/4$ and $f_{SYS}/8$) can be selected. The out to LD0) will be issued from the built-in FIFO at the falling edge of D1BSC FIFO is no empty. The work of the FIFO is in Figure 3.14 buffer size 80 bytes. The LD7 to LD0 is changed at the fal BaseSCP which is shown in Figure 3.14.13 and 3.14.14 for SRAM and SDR respectively. The FIFO is always reset to the empty In BaseSCP mode (e.g. f = 0.0, for DSCP = 1, SCPW = 0), D1BSCP is equal to BaseSCP equal to BaseLD7 to LD0 and Generally, for FIFO, the data input should be greater than the output one.

To make FIFO work correctly, the following relation should be satisfied by set SFR properly.

$$(N/8 + 1) \times t_{cw} + 24 \times 1/f_C < t_{LP} - t_{LPH}$$

Here, N is the segment number, and t_{cw} is D1BSCP clock cycle, t_{LP} is D2BLP cycle and t_{LPH} is High width of D2BLP signal in Figure 3.14.16, we can relation means that the last LD7 to LD0 data must be generated before the rising edge of D2BLP.

For example, in case of f_C = 36 MHz, f_{LP} = 32 kHz, 4 gray, 240 com, 640 SDRAM burst mode, the following table can be obtained mode is impossible and SCPW = base/2/4 clock modes c

SCPW	D1BSCP frequency (MHz)	tcw (ns)	(N/8+1) × tcw + T_busdly + T_busfmax (ns)	$t_{LP} - t_{LPH}$ (ns)	Judgment
Base	18	55.6	5166.1	31250	OK
2 clk	9	111.2	9674.4	31250	OK
4 clk	4.5	222.4	18681.6	31250	OK
8 clk	2.25	444.8	36696	31250	ERROR

Note: The speed of BaseSCP mode is equal to 2clk mode in the 8 or 16 GRAY mode.

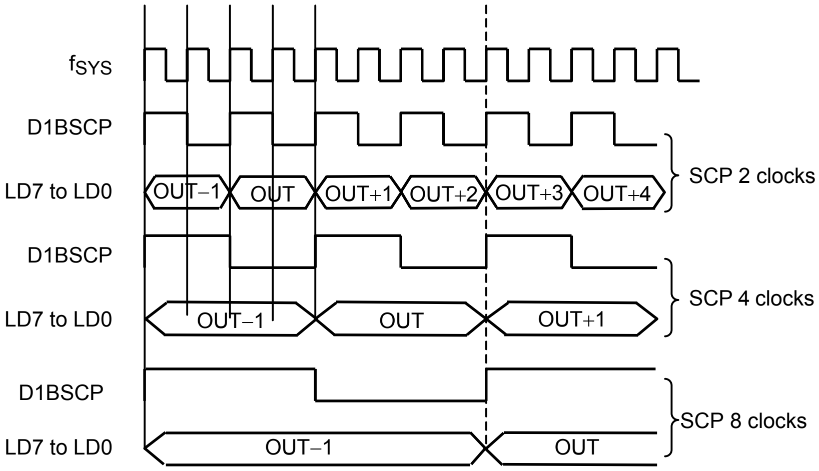
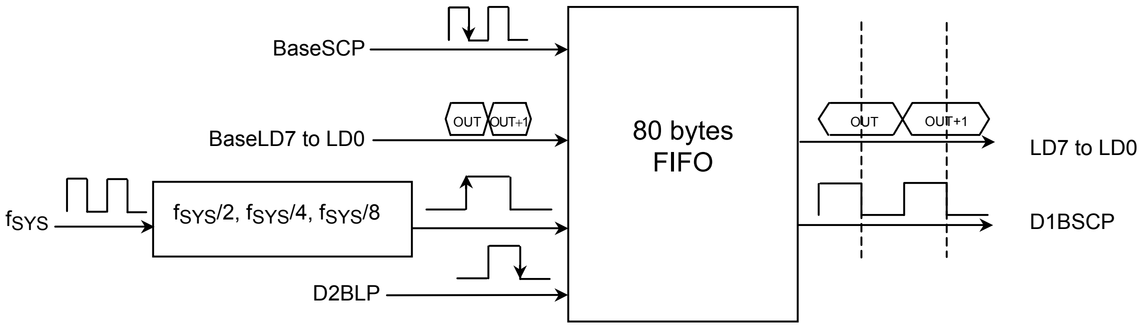


Figure 3.14.12 Timing Diagram for the LCD Panel Access Signals



Note: D1BSCP = BaseSCP and BaseLD7 to LD0 = BaseLD7 to LD0 in BaseSCP mode
(e.g., for SCPW [1:0] = 00)

Figure 3.14.13 Timing Diagram for FIFO

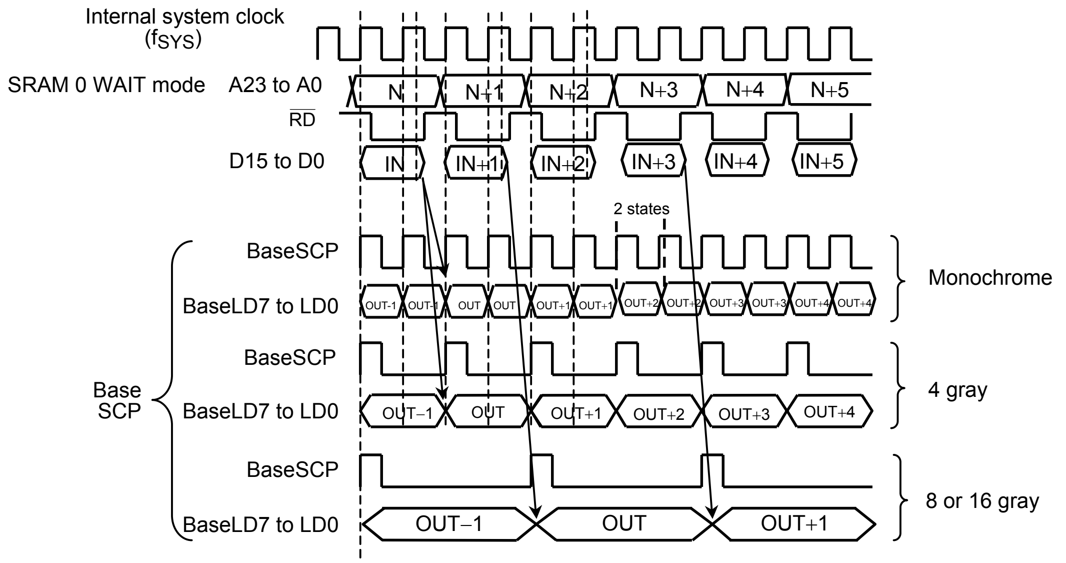


Figure 3.14.14 Timing Diagram for SRAM Mode with BaseSCP

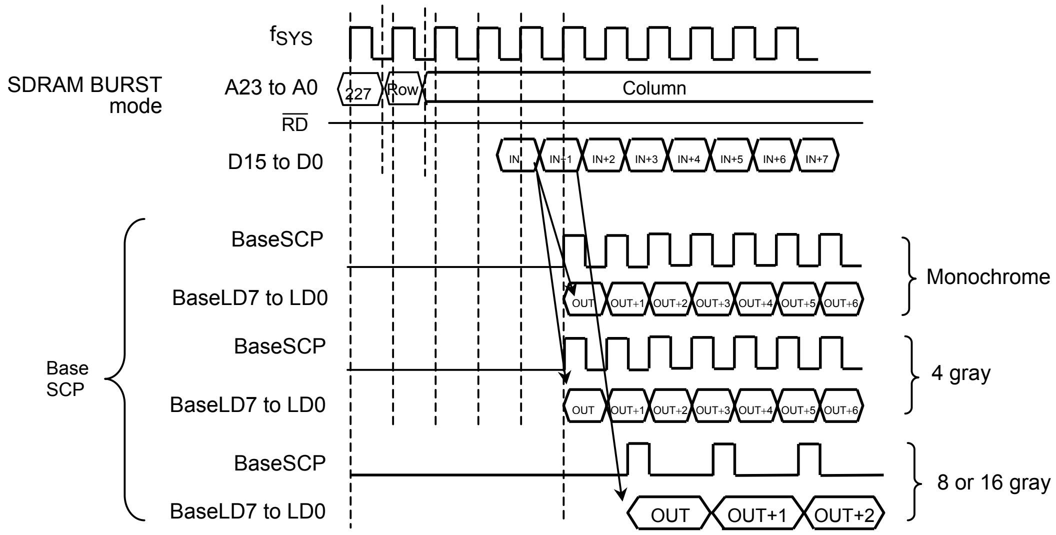
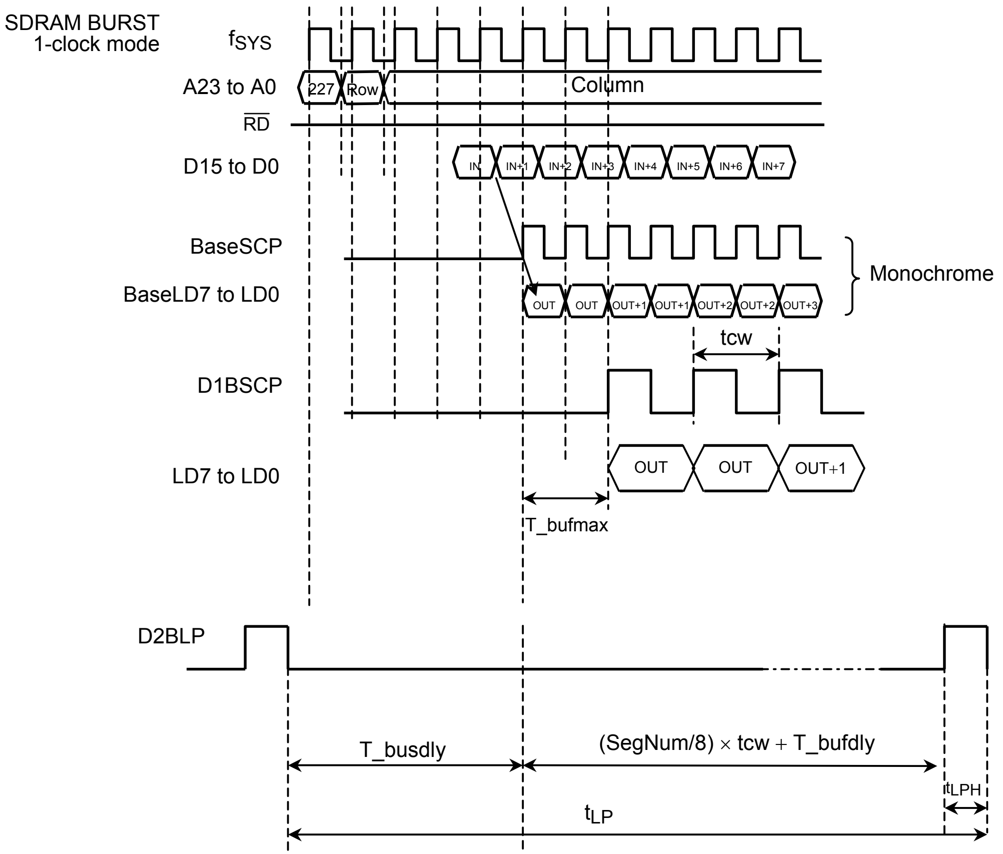


Figure 3.14.15 Timing Diagram for SDRAM BURST Mode with BaseSCP



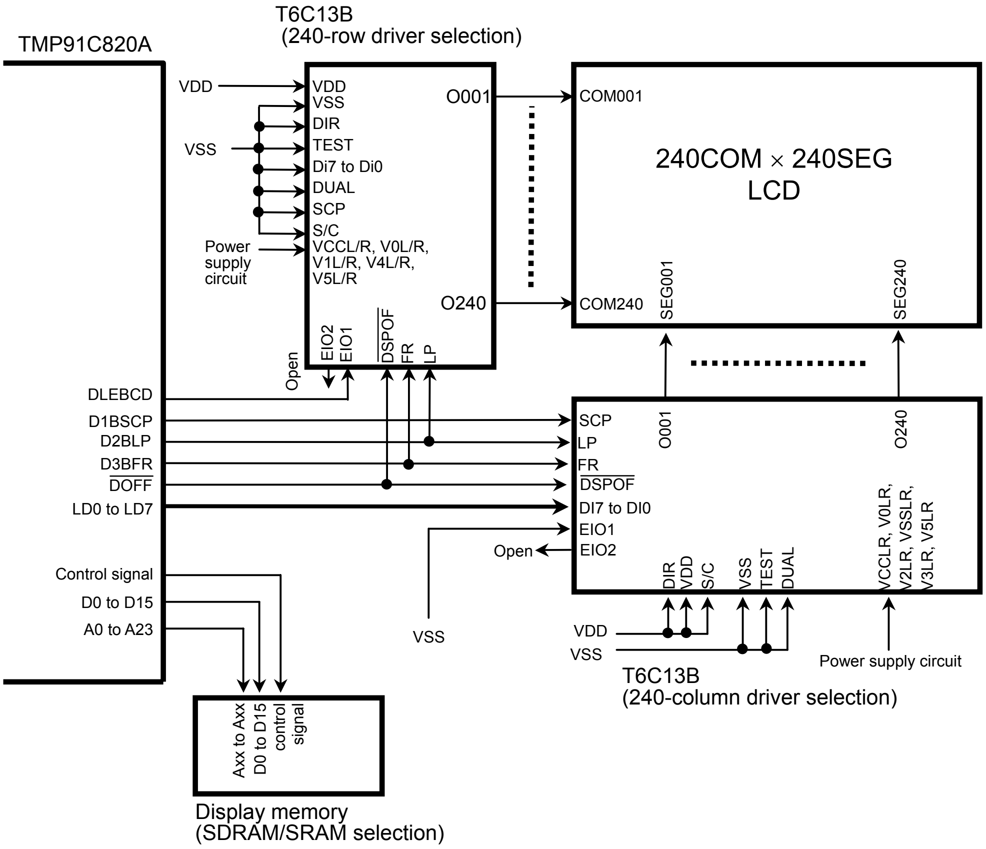
Note 1: If Base SCP, T_bufmax = 0.

Note 2: If except Base, T_bufmax ≤ tcw + 2/fc

Note 3: T_busdly is about 11 times as long as f_{SYS} period (22/fc).

Figure 3.14.16 Timing Diagram for Maximum FIFO Delay Time

3.14.4.7 Interface Examples at SR Mode



Note 1: Display memory should be 16-bit bus.

Note 2: Other circuit is necessary for LCD drive power supply for LCD driver display.

Figure 3.14.17 Interface Example for Shift Register Type LCD Driver

Note: Because the connection between the line of display RAM data and output bus: LD0:7 is just the mirror reverse, please care of connection. The data LSB of display RAM is output from LD7. In the above figure, LD0 shoud be connected to DI7 of LCD driver, and LD1 to DI6.

For detail information, please refer to Figure 3.14.5.

3.14.4.8 Sample Program

- Example:

  In case of use 240 SEG × 240 COM, 4-level gray scale display,
  This sample program operate correctly, LCD panel s

```
;***** SDRAM SET *****
        LD      (sdacr),2bH         ; Add-MUX enable, 64-Mbi
        LD      (sdrcr),01H         ; Interval refresh

;***** GLCDC SET *****
        LD      (lcdmode),17H       ; A/B area off, SDRAM 64 M
                                    ; SCP width 2 clocks
        LD      (lcddvm),11         ; 11-count DVM set
        LD      (lcdsize),32H       ; SEG=240, COM=240
        LD      (lcdctl),20H        ; Divide frame ON, displ
        LD      (lcdffp),240        ; Frame frequency correction (91 Hz)
        LD      (lcdgl),01h         ; 4-level gray
        LD      (lcdcm),0c1H        ; Cursor ON, black, 2 Hz b
        LD      (lcdcw),19          ; Width = 20 dots
        LD      (lcdch),19          ; Height = 20 dots
        LD      (lcdcp),00H         ; Pixel = 0
        LD      (lcdcpl),00H        ; Cursor position
        LD      (lcdcpm),00H        ; Cursor address
        LD      (lcdcph),40H        ; Cursor address
        LD      (lsarch),40H        ; C_area start address
        LD      (lsarcm),00H        ; C_area start address
        LD      (lsarcl),00H        ; C_area start address
;***** 0/4 data write 60 ROW *****
        LD      xix,400000H         ;
        LD      wa,0000H            ; Write data 0/4 gray data (0
loop1:  LD      (xix),wa            ;
        INC     2,xix               ;
        CP      xix,407800H         ; 400000H to 4077FFH: 60 R
        JR      nz,loop1            ;
;***** 2/4 data write 60 ROW *****
        LD      xix,407800H         ;
        LD      wa,05555H           ; Write data 2/4 gray data (0
loop2:  LD      (xix),wa            ;
        INC     2,xix               ;
        CP      xix,40F000H         ; 407800H to 40EFFFH: 60 ROW (Dot)
        JR      nz,loop2            ;
```

```
; ***** 3/4 data write ***** 60 ROW
        LD      xix,40F000H     ;
        LD      wa,0aaaaH       ; Write data 3/4 gray data (
loop3:  LD      (xix),wa        ;
        INC     2,xix           ;
        CP      xix,416800H     ; 40F000H to 4167FFH: 60 R
        JR      nz,loop3        ;
; ***** 4/4 data write ***** 60 ROW
        LD      xix,416800H     ;
        LD      wa,0ffffH       ; Write data 4/4 gray data
loop4:  LD      (xix),wa        ;
        INC     2,xix           ;
        CP      xix,41e000H     ; 416800H tp 41DFFFH: 60 ROW (Dot)
        JR      nz,loop4        ;
; ***** 4-level gray palette ***** pattern set
        LD      (lg0l),00H      ; 0/4 gray scale palette 0
        LD      (lg1l),05H      ; 2/4 gray scale palette 0
        LD      (lg2l),0eH      ; 3/4 gray scale palette 1
        LD      (lg3l),0fH      ; 4/4 gray scale palette 1
; ***** DMA, DISPLAY-ON ***** start
        LD      (lcdctl),01H    ; Display on, divide on
```

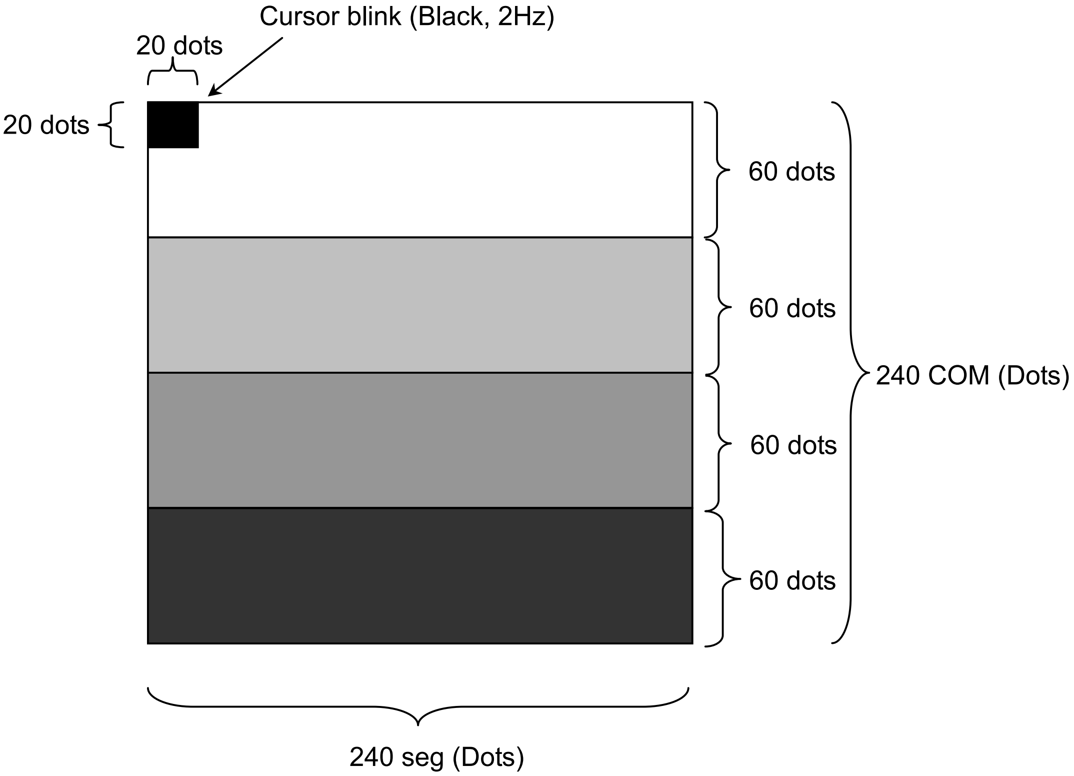


Figure 3.14.18 Display Reference above Sample Program

3.14.5 RAM Built-in Type LCD Driver Control Mode (RAM type)

3.14.5.1 Operation

Data transmission to LCD driver is executed by move instruction of

After setting mode of operation to SFR, when move LCDC outputs chip select signal to LCD driver conne pin (D1BSCP etc.). Therefore control of data trans LCD size is controlled by instruction of CPU. Ther driver in this case, and which is chosen determines

It corresponds to LCD driver which has 1 bit instruction display data register in LCD driver. Please make <M = "0". The transmission place address is set to either of FE0H to FE7 ACCESS TYPE: See Table 3.14.3)

It corresponds to address LCD driver writing at the time of < = "1".

The transmission place address at this time can a 3C0000H to 3FFFFFH to every 64 Kbytes. (RANDOM ACCES See Table 3.14.4)

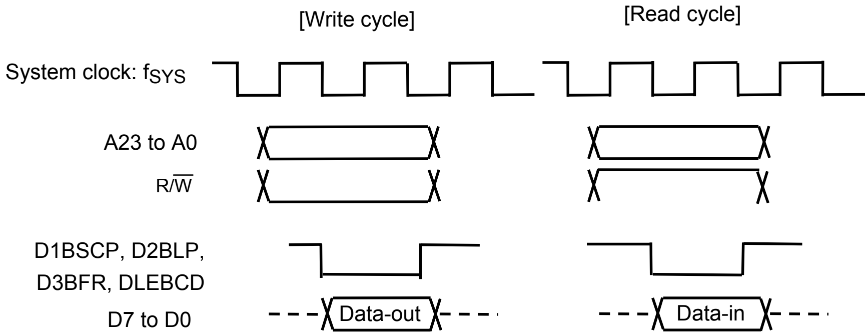
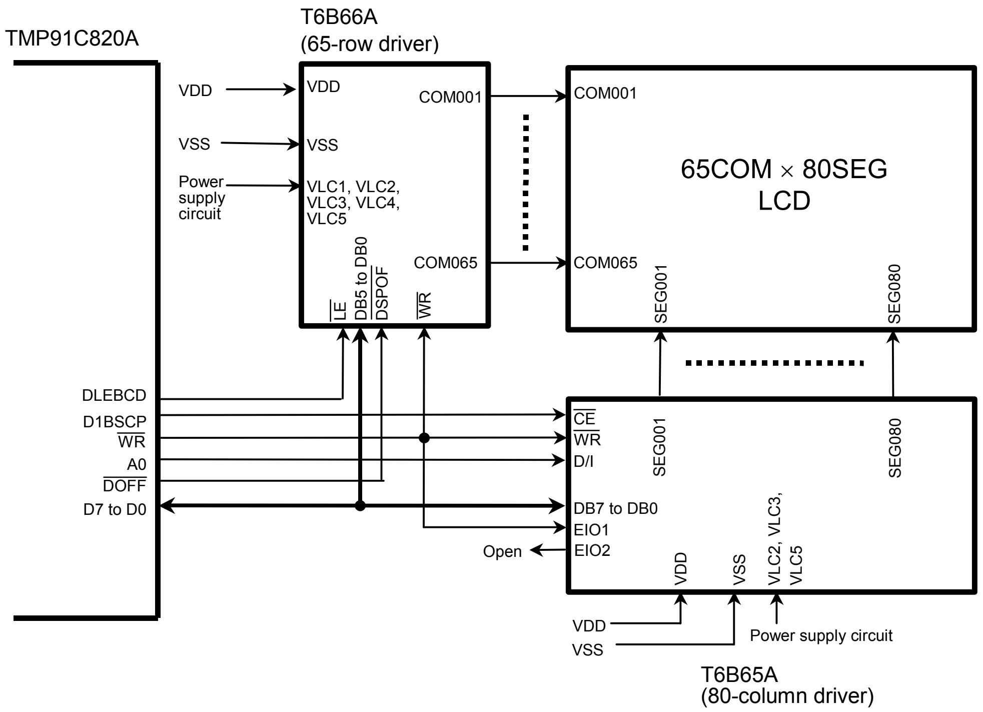


Figure 3.14.19 Example of Access Timing for RAM Built-in Type LCD Driver (Wait = 0)

3.14.5.2 Interface Examples at Internal RAM Mode



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

Figure 3.14.20 Interface Example for RAM Built-in Type Sequential Access Type LCD DRIVER

3.14.5.3 Sample Program

- Example: In case of use 5COM × 80SEG LCD driver.

  Assign external column driver to LCDC1 and row driver to LCDC4.

  This example used LD instruction for transmitting command and used bu
  of micro DMA in transmitting of display data.

In case of store 650 bytes transfer data to LCD driver.

```
;***** Setting external terminal *****
      LD     (pdcr),19H     ; C̅E̅ for LCDC1 : D1BSCP,
                            ; L̅E̅ for LCDL1 : DLEBCD,
                            ; Setting for D̅O̅F̅F̅
;***** Setting for LCDC *****
      LD     (lcdmode),00H  ; Select RAM mode
      LD     (lcdctl),00H   ; MMULCD = 0 (Sequential access mod

;***** Setting for mode of LCDC0/LCDR0 *****
      LD     (lcdc1l),xx    ; Setting instruction for LCDC1
      LD     (lcdc4l),xx    ; Setting instruction for LCDC4

;***** 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
```

3.15 Melody/Alarm Generator (MLD)

TMP91C820A incorporates melody function and alarm function, both of whi
from the MLDALM pin. Five kinds of fixed cycle INTERRUPT is generated by
counter, which is used for alarm generator.

Features are as follows.

Melody generator

The melody function generates signals of any frequ
low-speed clock (32.768 kHz) and outputs the signals

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

Alarm generator

The Alarm function generates eight kinds of alarm waveforms having a m
frequency (4096 Hz) determined by the low-speed clock (32.768 kHz). And thi
able to invert by setting a value to a register.

By connecting a loud speaker outside, alarm tone can

Five kind of fixed cycles (1 Hz, 2 Hz, 64 Hz, 512 Hz, 8 kHz) INTERRUPT be ge
using a counter which is used for alarm generator.

Special mode

It is assigned <TA3LCDE> and <TA3MLDE> to bit0 and bit1, of EMCCR0
(00E7hex). These bits are used when you want to opera
without low-frequency clock (XTIN, XTOUT). A two bits set
clock is supplied each LCDC and MELODY circuit. If y
(Generate by timer 3) is supplied each LCDC and MELODY circuit as same, you sho
set 32 kHz timer 3 frequency. For detail, look AC specification character

This section is constituted as follows.

3.15.1 Block Diagram
3.15.2 Control Registers
3.15.3 Operational Description
3.15.3.1 Melody Generator
3.15.3.2 Alarm Generator

3.15.1 Block Diagram

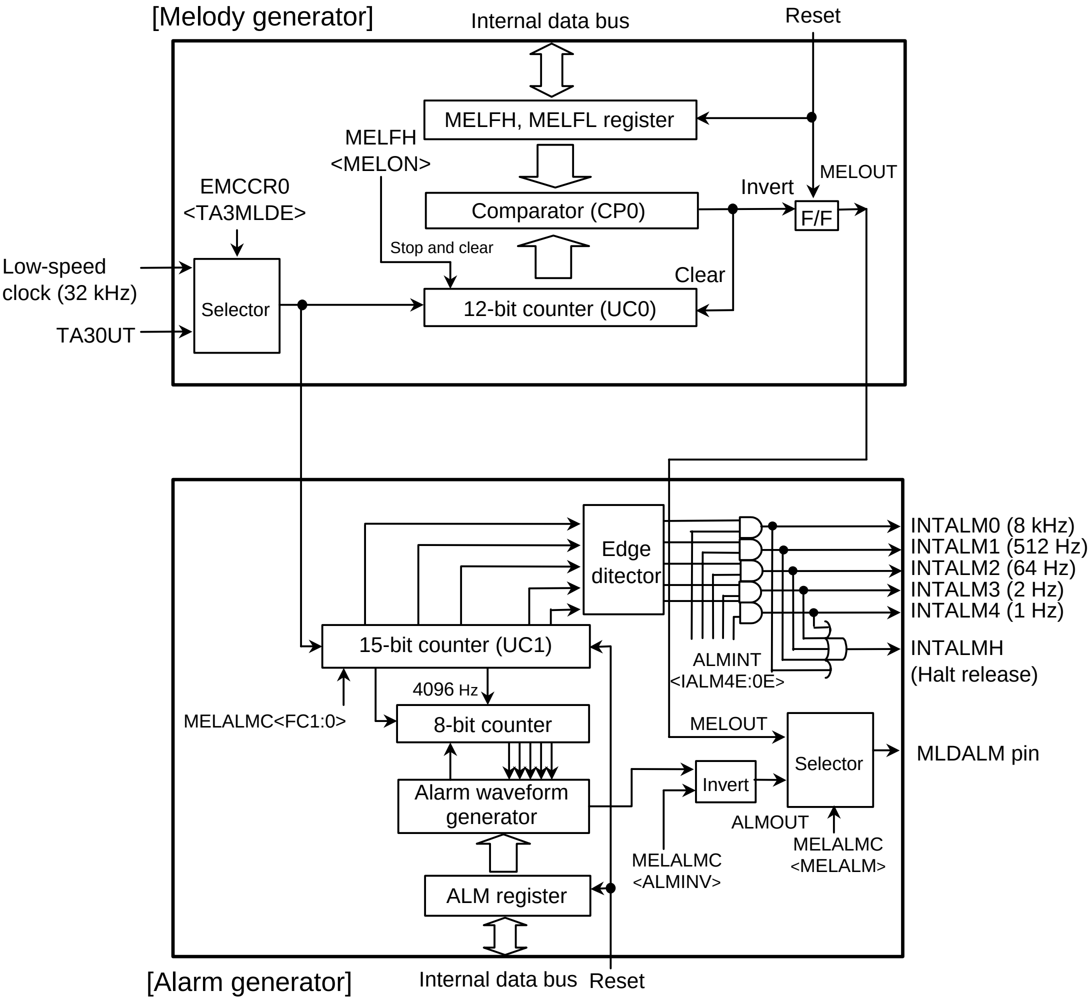


Figure 3.15.1 MLD Block Diagram

3.15.2 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	Write “0”.				Output waveform select 0: Alarm 1: Melody

Note 1: MELALMC<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			Write “0”.	1: Interrupt enable for INTALM4 to INTALM0				

3.15.3 Operational Description

3.15.3.1 Melody Generator

The melody function generates any frequency (4 Hz to 5461 H
low-speed clock (32.768 kHz) and outputs the signa
By connecting a loud speaker outside, melody tone can easily sound

(Operation)

At first, MELALMC<MELALM> have to be set as 1 in order to s
waveform as output waveform from MLDALM. Then melo
be set to 12-bit register MELFH, MELFL.

Followings are setting example of melody output freq

(Formula for calculating of melody waveform frequ

at $f_s = 32.768$ [kHz]

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

setting value for melody $N = (16384/f_{MLD}) - 2$

(Notice: $N = 1$ to 4095 (001H to FFFH), 0 is not ac

(Example program)

In case of outputting a musical scale (440 Hz)

```
LD    (MELALMC), 11X00001B   ; Select melody wa
LD    (MELFL), 23H           ; N = 16384/440 − 2 = 35 = 023H
LD    (MELFH), 80H           ; Start to generate wa
```

(Ref: 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

3.15.3.2 Alarm Generator

The alarm function generates eight kinds of alarm waveform having a m frequency 4096 Hz determined by the low-speed clock (32.768 kH waveform is reversible by setting a value to a regis

By connecting a loud speaker outside, alarm tone c

Five kind of fixed cycles (1 Hz, 64 Hz, 512 Hz, 2 kHz, 8 kHz) INTER generated by using a counter which is used for alarm generator.

(Operation)

At first, MELALMC<MELALM> have to be set as 0 in order to waveform as output waveform from MLDALM. Then al on 8-bit register of ALM. Finally, MELALMC<AC1:0> regi <ALMINV> be set as invert. By setting these values, counter star alarm waveform.

Followings are example program, setting value o of each setting value.

(Setting value of alarm pattern)

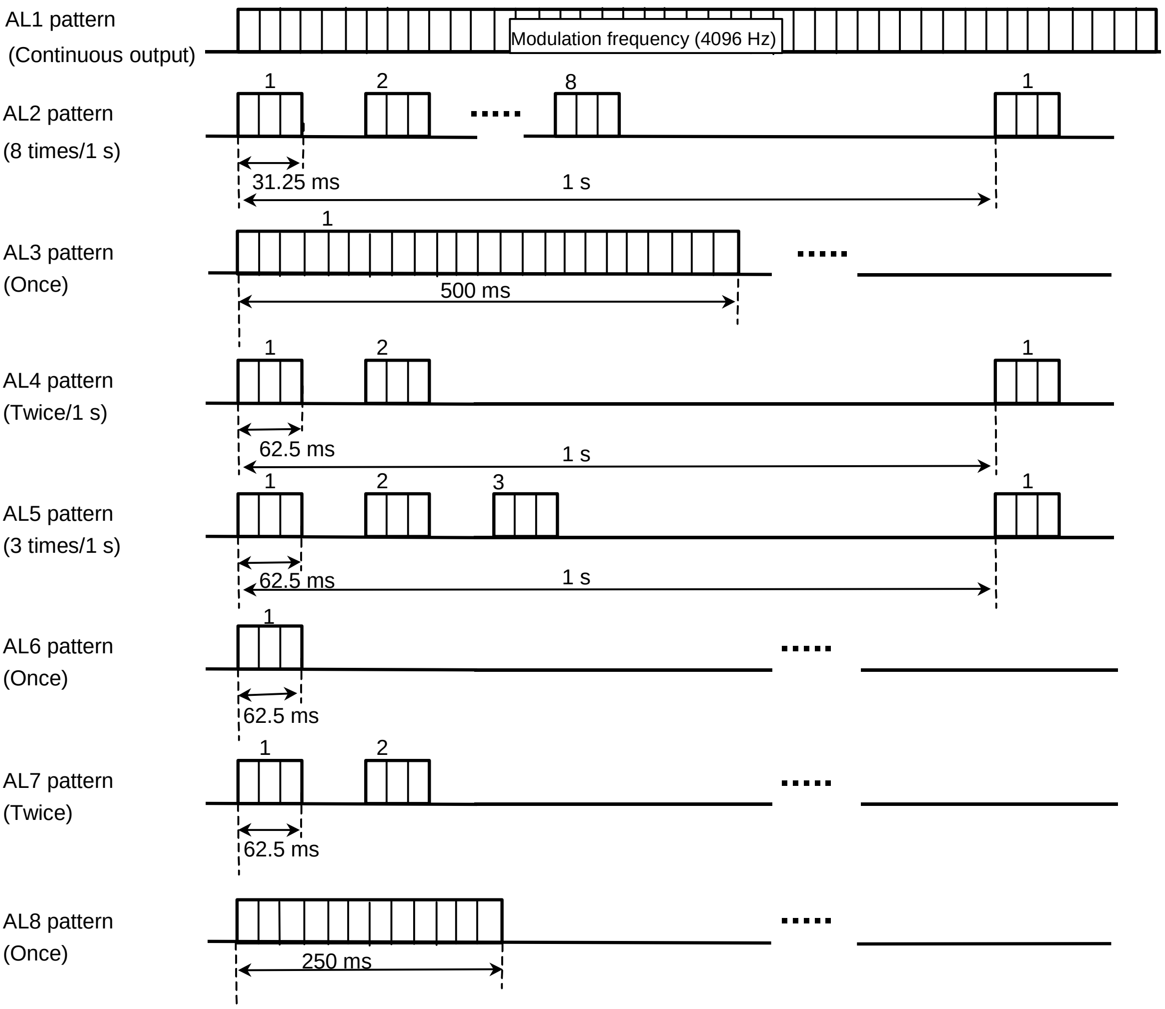
Setting Value for ALM Register	Alarm Waveform
00H	0 fixed
01H	AL1 pattern
02H	AL2 pattern
04H	AL3 pattern
08H	AL4 pattern
10H	AL5 pattern
20H	AL6pattern
40H	AL7 pattern
80H	AL8 pattern
Other	Undefined (Do not set.)

(Example program)

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

```
  LD      (MELALMC), C0H        ; Set output alarm wav
                                ; Free-run counter start
  LD      (ALM), 02H            ; Set AL2 pattern, star
```

Example: Waveform of alarm pattern for each setti



3.16 SDRAM Controller (SDRAMC)

TMP91C820A includes SDRAM controller which supported the access by CPU
The features are as follows.

(1) Support SDRAM

16- or 64- or 128-Mbit SDRAM (×16bit × 2/4 BANKs), not support DDR

(2) Automatic initialize function

- All BANK pre-charge command generate
- Mode register set generate
- 8 times auto refresh

(3) Access mode

	CPU Access	LCDC Access
Burst length	1 word	Full page
Addressing mode	Sequential	Sequential
CAS latency (Clock)	2	2
Write mode	Single write	–

(4) Access cycle

- CPU access (Read/write)

  Read cycle: 4 states (222 ns at 36 MHz)

  Write cycle: 3 states (167 ns at 36 MHz)

  Access data width: 8 bits/16 bits

  Burst length: 1 word only

- LCDC burst access (Read only)

  Read cycle: 1 state (55 ns at 36 MHz)

  Overhead: 4 states (222 ns at 36 MHz)

  Access data width: 16 bits only

  Burst length: Full page only

(5) Refresh cycle auto generate

- Auto refresh is generated during another area acc
- Refresh interval is programmable.
- Self refresh is supported

Notes:

- Display data has to set from the head of each page.
- Program is not operated on SDRAM.
- Following condition is set by setting Chip select controller CS1.
  - WAIT setting: 0 WAIT setting only
  - Bus width: 8/16 bit only
  - Memory area: Optional

3.16.1 Control Registers

Figure 3.16.1 shows the SDRAMC control registers. Setting the registe
operation of SDRAMC.

SDRAM Access Control Register

		7	6	5	4	3	2	1	0
SDACR (04F0H)	Bit symbol	SDINI	SWRC	–	–	SMUXE	SMUXW1	SMUXW0	SMAC
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W		R/W
	After reset	0	0	1	0	0	0	0	0
	Function	Auto initialize 0: Disable 1: Enable	Write recovery 0: 1 clock 1: 2 clock	Always fixed to “10”.		Address mux 0: Disable 1: Enable	SDRAM select 00: 16 Mbits 10: 128 Mbits 01: 64 Mbits 11: Reserved		SDRAM cotroller 0: Disable 1: Enable

SDRAM Refresh Control Register

		7	6	5	4	3	2	1	0
SDRCR (04F1H)	Bit symbol	SFRC	SRS2	SRS1	SRS0	–			SRC
	Read/Write	R/W	R/W			R/W			R/W
	After reset	0	0	0	0	0			0
	Function	Self refresh 0: Exit 1: Entry	Auto refresh interval 000: 78 states 100: 195 states 001: 97 states 101: 210 states 010:124 states 110: 249 states 011:156 states 111: 312 states			Always write “0”.			Auto refresh 0: Disable 1: Enable

Figure 3.16.1 SDRAMC Control Registers

Self refresh operation is controlled by setting SD
become Entry by writing “1” to SDRCR<SFRC>, set “0” to it, Self refresh
Exit.

3.16.2 Operation Description

(1) Memory access control

Access control block is enable when SDACR<SMAC> = 1. And the SDRAM cont
signals (SDCS, SDRAS, SDCAS, SDWE, SDLDQM, SDUDQM, SDCLK and SDC
are output during the time CPU or LCDC accesses CS1 a

In the access cycle, address multiplex outputs row/column address th
A12 pin. The enable/disable setting of address mul
<SMUXE>. And multiplex width is decided by setting
memory size. The relation between multiplex width a

Table 3.16.1 Address Mutiplex

SDRAM	TMP91C820A Address Output			
Address	Column	Row Address		
Pin Name	Address	16 Mbits	64 Mbits	128 Mbits
–	A0	A0	A0	A0
A0	A1	A9	A9	A10
A1	A2	A10	A10	A11
A2	A3	A11	A11	A12
A3	A4	A12	A12	A13
A4	A5	A13	A13	A14
A5	A6	A14	A14	A15
A6	A7	A15	A15	A16
A7	A8	A16	A16	A17
A8	A9	A17	A17	A18
A9	A10	A18	A18	A19
A10	A11	A19	A19	A20
A11	A12		A20	A21
BS0	A13	A20	A21	A22
BS1	A14	–	A22	A23

← Memory size

↑ TMP91C820A address pin name

▨ Effective column address

SDRAM access by CPU is performed by the 1 word bur

SDRAM access by LCDC is performed by 1 page burst mode.

SDRAM access cycle is shown in Figure 3.16.2 to F

The read cycle by CPU is the 4-state fixation, a
fixation.

In the burst read cycle by LCDC, a pre-cha
a refresh cycle are automatically inserted in CPU

Note: In SDRAM access cycle, WAIT setup by the CS/WAIT controller (CS1) is disregarded. The wait setting of CS1 should be 0 waits.

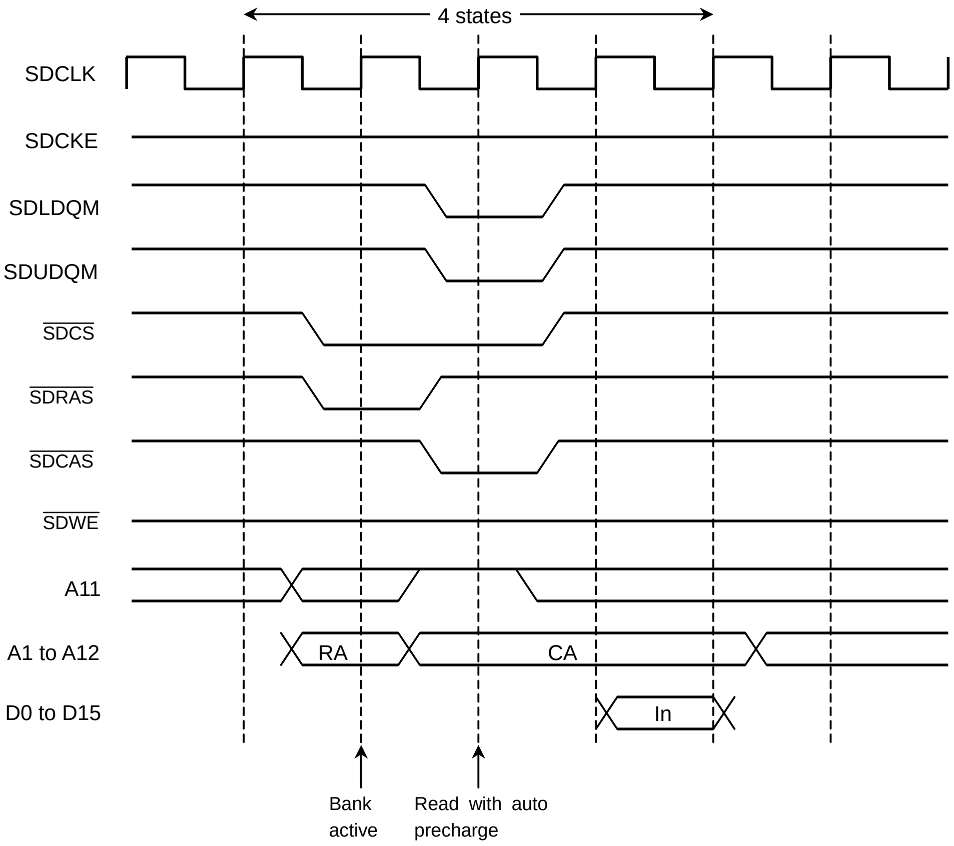


Figure 3.16.2 SDRAM Access Timing (CPU read)

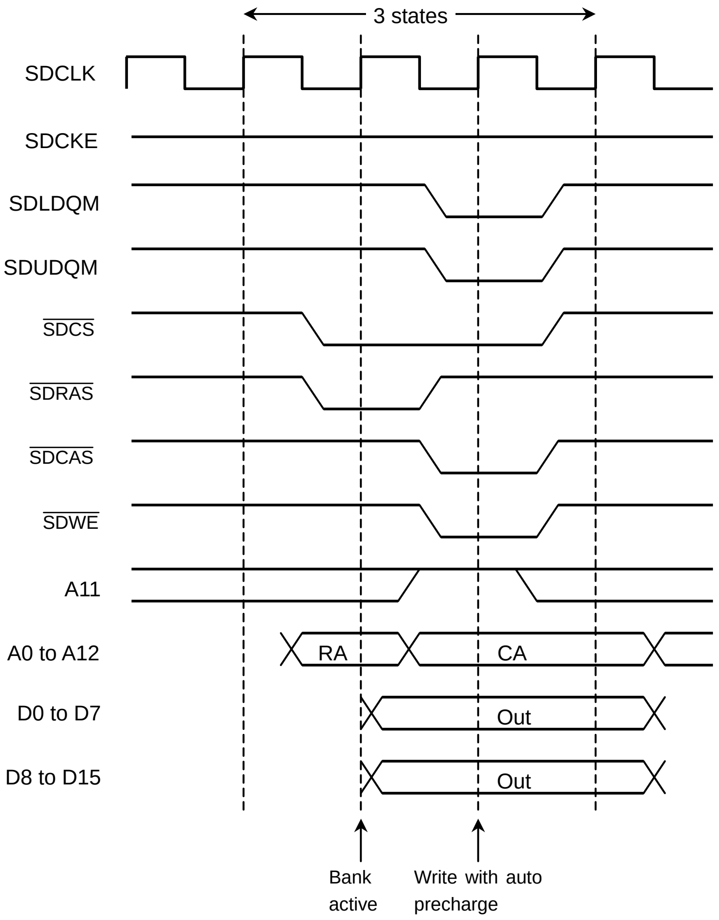


Figure 3.16.3 SDRAM Access Timing (CPU write 16 bits)

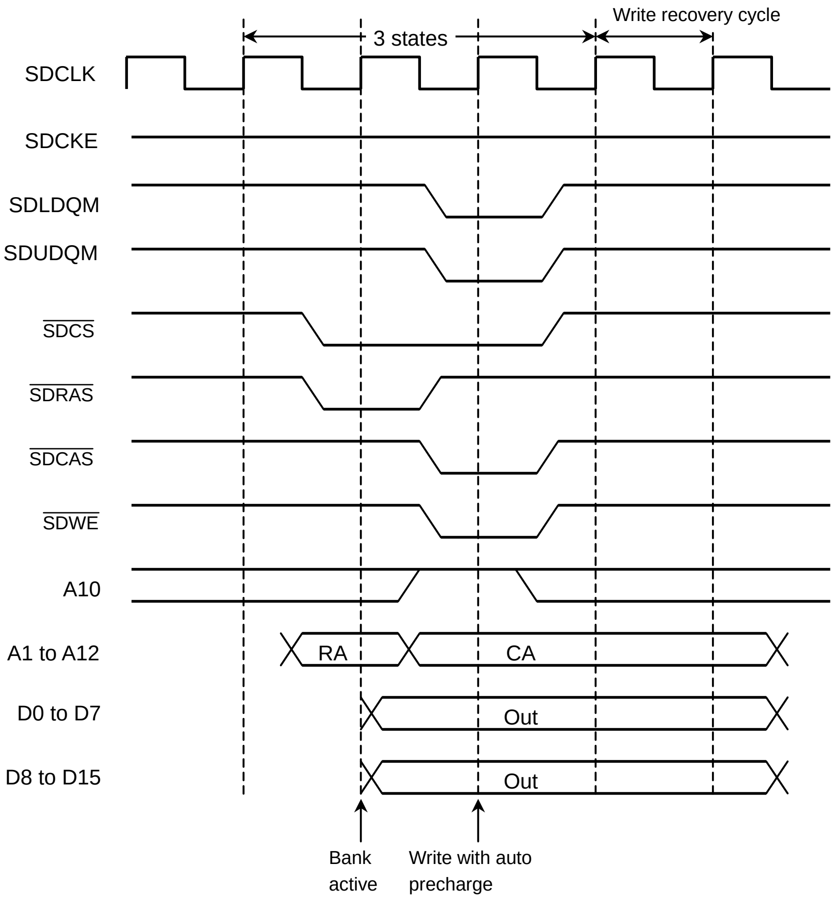


Figure 3.16.4 SDRAM Access Timing (CPU write 16 bits, write recovery: 2 clocks)

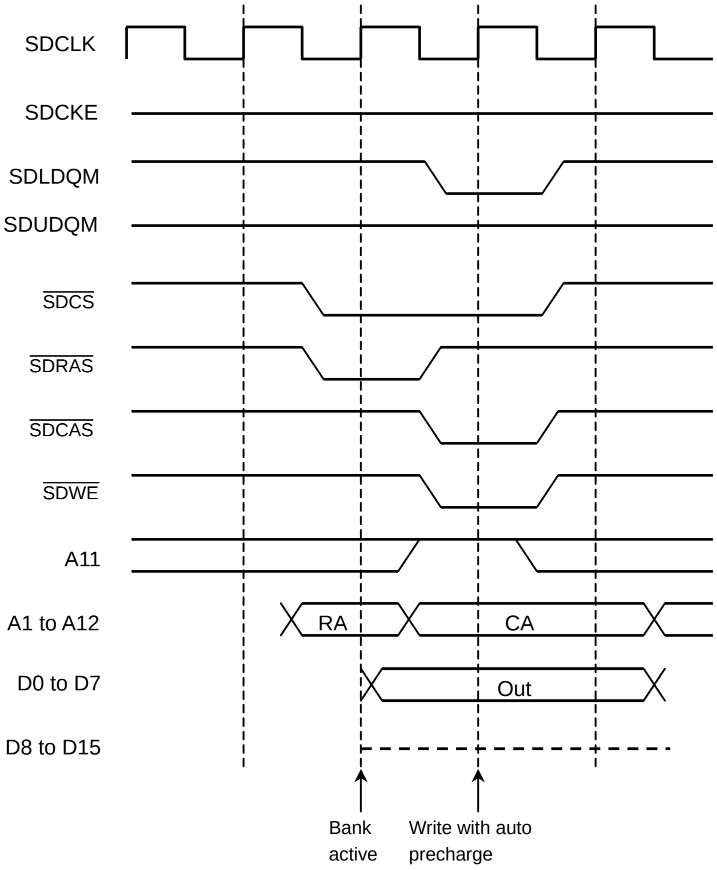


Figure 3.16.5 SDRAM Access Timing (CPU lower byte write)

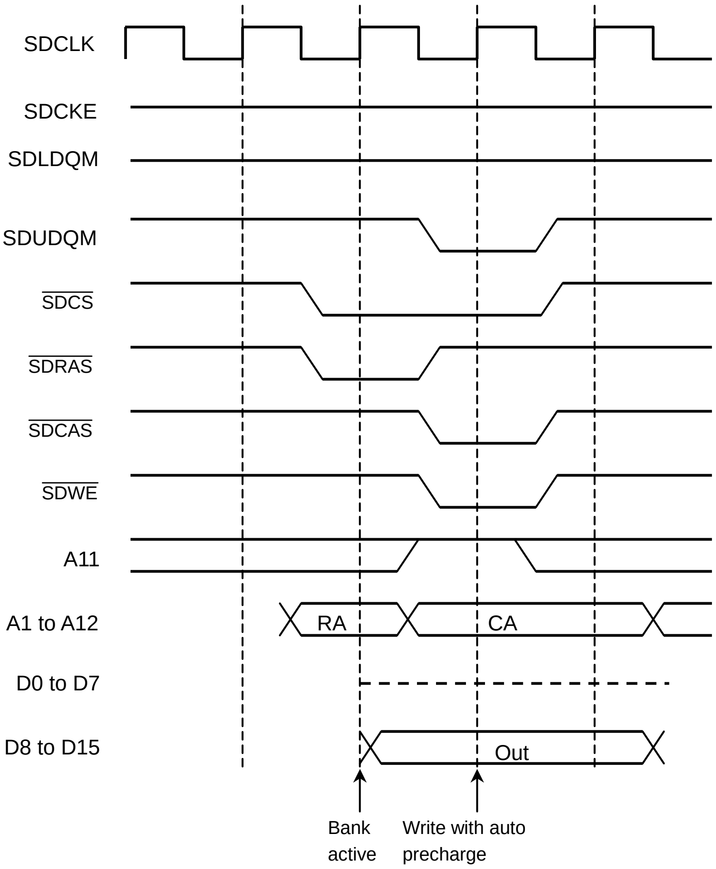


Figure 3.16.6 SDRAM Access Timing (CPU upper byte write)

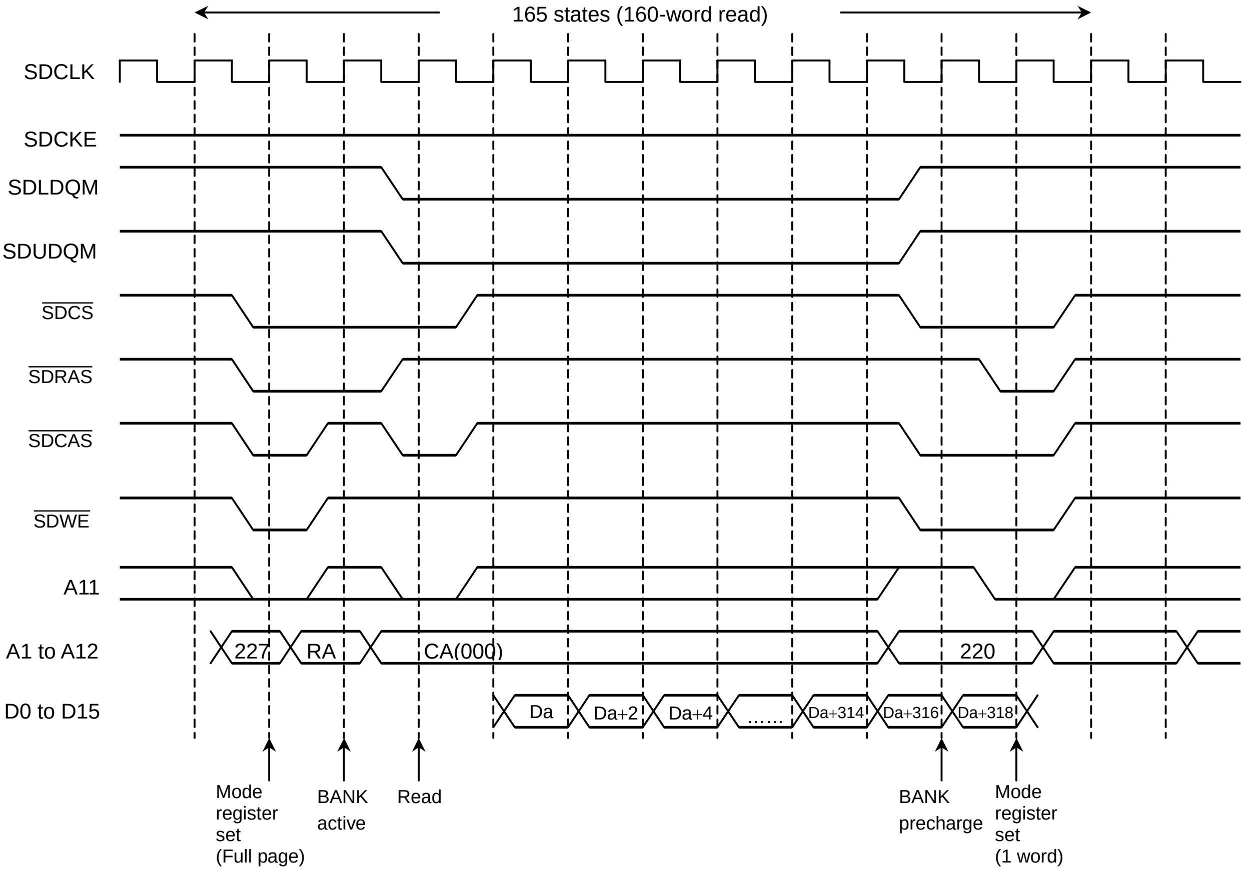


Figure 3.16.7 SDRAM Access Timing (LCDC burst read)

(2) Refresh control

TMP91C820A can generate an auto refresh cycle automatically require
maintenance of SDRAM.

Refreshment interval can be set up by SDRCR<SRS0:2> from the 78 sta
312 states (4.3 µs to 17.3 µs at 36 MHz).

By setting SDRCR<SRC> to 1, a refresh cycle is gene
<SR0:2>.

The generating timing of refresh cycle is one of the access cycle
SDRAM area (CS1) after the interval setup by <SRS0:

The refresh cycle is shown in Figure 3.16.8. More
shown in Table 3.16.2.

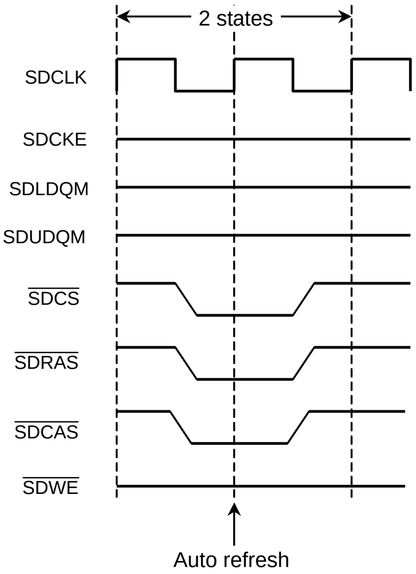


Figure 3.16.8 Refresh Cycle

Table 3.16.2 Auto Refresh Cycle Insertion Interval (Unit: µs)

<SRS2:0> setting			Insertion interval (State)	Frequency (f$_{SYS}$)					
SRS2	SRS1	SRS0		5 MHz	6.25 MHz	8 MHz	10 MHz	12.5 MHz	18MHz
0	0	0	78	15.6	12.5	9.8	7.8	6.2	4.3
0	0	1	97	19.4	15.5	12.1	9.7	7.8	5.4
0	1	0	124	24.8	19.8	15.5	12.4	9.9	6.9
0	1	1	156	31.2	25.0	19.5	15.6	12.5	8.7
1	0	0	195	39.0	31.2	24.4	19.5	15.6	10.8
1	0	1	210	42.0	33.6	26.3	21.0	16.8	11.7
1	1	0	247	49.4	39.5	30.9	24.7	19.8	13.7
1	1	1	312	62.4	49.9	39.0	31.2	25.0	17.3

It does not generate the interval refresh during the bus master access t
LCDC.

The interval refreshment demand generated in the
When it returns to CPU access cycle, a refresh cycle is gene

Furthermore, TMP91C820A can generate self refresh cycle. The self timi
refresh cycle is shown in Figure 3.16.9.

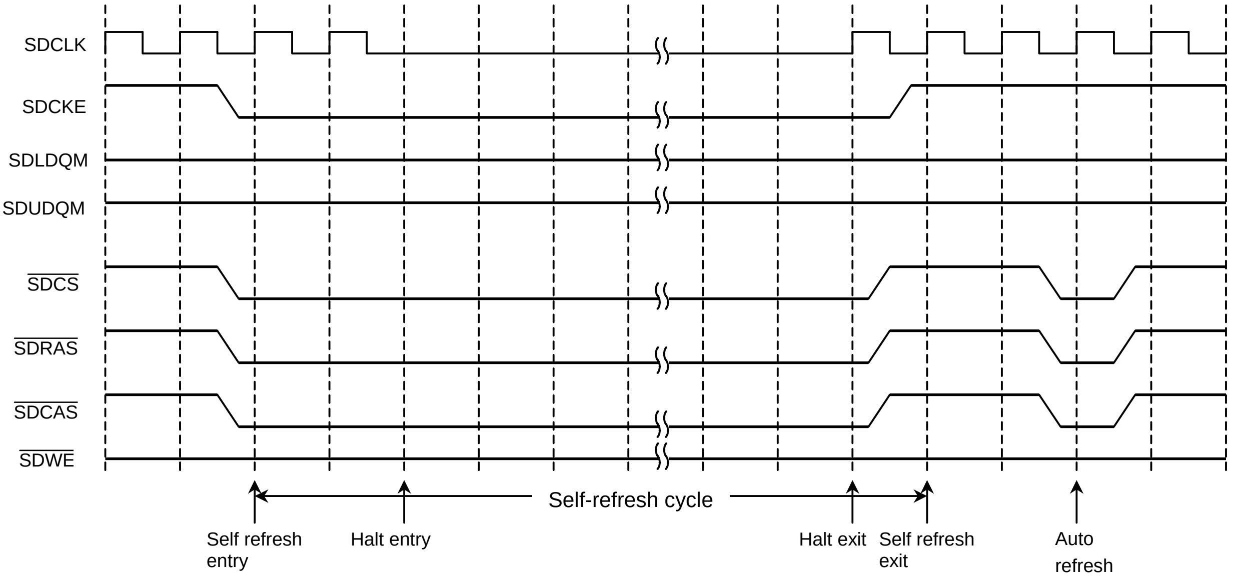


Figure 3.16.9 Self Refresh Cycle

Note 1: SDCLK is output in the IDLE2 mode. Therefore if you stop SDCLK, change PF6 pin to output port before the HALT instruction.

Note 2: Pin condition under the IDLE1/STOP mode depends on the setting of SYSCR2<DRVE>. SDCKE doesn't depend on it but outputs low level.

If SDRCR<SFRC> is set to 1, the self-refresh cycle

The self refreshment mode is used when using the s which an internal clock is stopped such as HALT (STOP, IDLE1) refreshment in the state of enable, please set SDRC

Release of a self refresh cycle is performed automatically by release i mode.

It inserts automatically one interval refresh after the self refreshmen returns to the interval refreshment mode.

(Note: When HALT instruction is cancelled by a reset, the I/O registers are initialized, therefore, refresh is not performed.).

Please do not place the command which accesses SDR which sets 1 to SDRCR<SFRC>. SDRCR<SFRC> set to 1, make sure HALT instruction comes after NOP or some instructio

(3) SDRAM Initialize

TMP91C820A can generate the cycle of the followi
power-supply injection to SDRAM. The cycle is show

1. Precharge of all banks
2. The initial configuration to a mode register
3. The refresh cycle of 8 cycles

The abovementioned cycle is generated by setting

Time after SDACR<SDINI> is set to 1 until an init
changes with following commands.

While performing this cycle, operation (An instru
CPU is stopped.

In addition, even before performing an initializa
SDRAM control signal and an address signal (A1 to A1

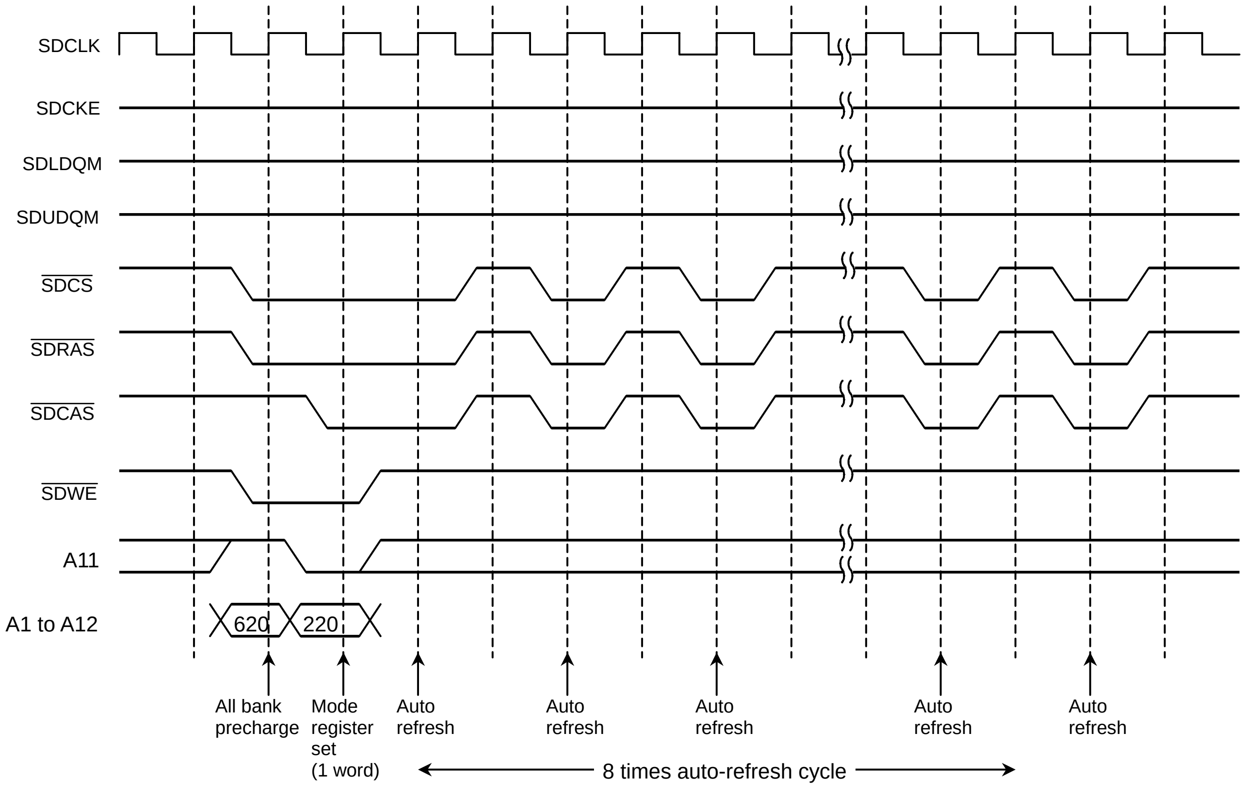
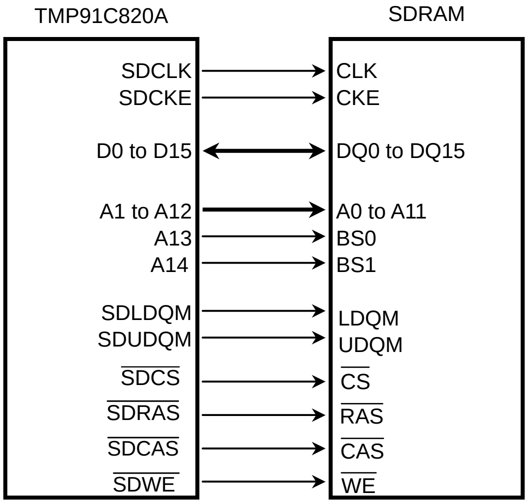


Figure 3.16.10 Initialize Cycle

(4) Connection example

The example of connection with SDRAM is shown in Fi



(1 M word × 4 BANKS × 16 bits)

Figure 3.16.11 Connection with SDRAM

(5) Limitation point to use SDRAM

There are some points to notice when using SDRAMC under below and take care.

1) WAIT access

When using SDRAM, it is added some limitation of

Under the N-WAIT setting of this MCU, it is prohi (1/4 refresh interval time; in Auto Refresh func controller).

2) Execution of SDRAM command instruction before HALT (SR-Entr Initialize, Mode-set)

It requires execution time (a few states) to exe has (SR-Entry, Initialize).

Therefore when executing HALT instruction afte insert over 10 bytes NOP instruction or other 10 byte before HALT

3) AR (Auto Refresh) interval time

When using SDRAM, CPU clock must be set suitab specification that is minimum operating clock an

When using SDRAM under slow mode or down the Cloc system with special care for Auto Refresh interva

And please set Auto Refresh interval time after adding 10 states to Auto Refresh interval time, because it might no SDRAM by stopping Auto Refresh.

(Example of calculation)

Condition:

f_{SYS} = 18MHz, SDRAM specification of distributed A 4096 times/64ms

64ms/4096 times = 15.625μs/1 time = 281.25 state/1 time

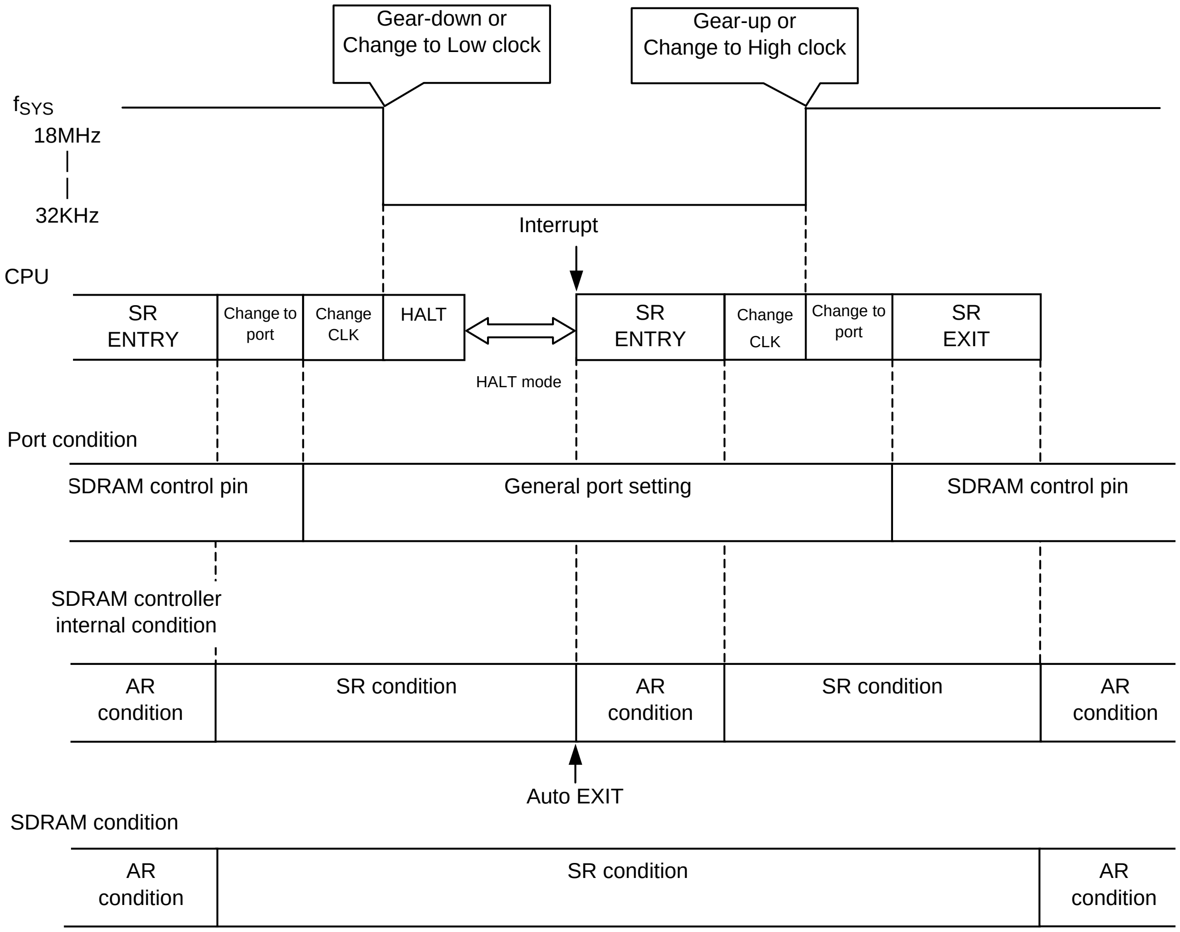
281.25 − 10 = 271.25 state/1 time → less than 271 states is need

4) Auto Exit problem when exiting from Self Refres

When using Self Refresh function together with changing clock, it might not be suit specificati releasing Self Refresh function (Auto Exit func HALT mode.

Following figure shows example for avoid this pr

(Outline concept to control)



*The target ports to change are SDCKE pin and $\overline{\text{SDCS}}$ pin.

*The method of Self refresh Entry includes the condition 4).

* SR : Self refresh , AR : Auto refresh

3.17 16-Bit Timer/Event Counters (TMRB)

The TMP91C820A incorporates one multifunctional 16-
which have the following operation modes:

- Interval timer mode
- Event counter mode

Timer/event counter consists of a 16-bit up counter, two 16-bit timer regist
with a double-buffer structure), two 16-bit capture registers, two comparators,
controller, a timer flip-flop and a control circuit.

Timer/event counter is controlled by an 11-byte control SFR.

This chapter consists of the following items:

3.17.1 Block Diagram

3.17.2 Operation

3.17.3 SFRs

3.17.4 Operation in Each Mode

(1) 16-bit timer mode

(2) 16-bit programmable pulse generation (PPG) ou

Table 3.17.1 Pins and SFR of TMRB0

Spec \ Channel		TMRB0
External pins	External clock/capture trigger input pins	None
	Timer flip-flop output pins	TB0OUT0 (also used as PB6)
SFR (Address)	Timer run register	TB0RUN (0180H)
	Timer mode register	TB0MOD (0182H)
	Timer flip-flop control register	TB0FFCR (0183H)
	Timer register	TB0RG0L (0188H)
		TB0RG0H (0189H)
		TB0RG1L (018AH)
		TB0RG1H (018BH)
	Capture register	TB0CP0L (018CH)
		TB0CP0H (018DH)
		TB0CP1L (018EH)
		TB0CP1H (018FH)

3.17.1 Block Diagram

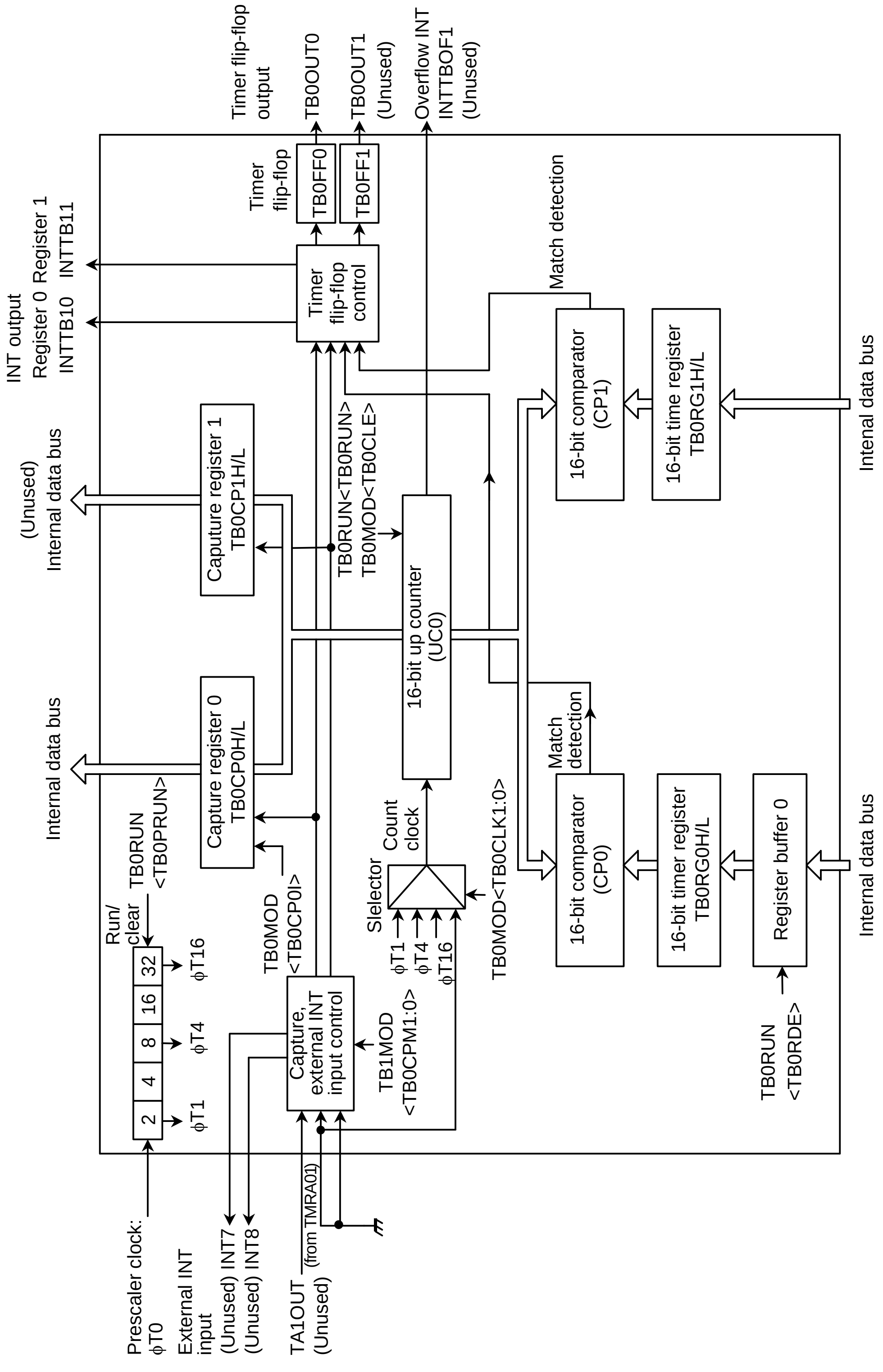


Figure 3.17.1 Block Diagram of TMRB0

3.17.2 Operation

(1) Prescaler

The 5-bit prescaler generates the source clock for TMRB0. The prescaler clock (φT0) is divided clock (Divided by 4) from selected clock of clock gear. This prescaler can be started or st Counting starts when <TB0RUN> is set to 1; the presc operation when <TB0RUN> is set to 0.

Table 3.17.2 Prescaler Clock Resolution

at fc = 16 MHz, fs = 32.768 kHz

System Clock Selection <SYSCK>	Prescaler Clock Selection <PRCK1:0>	Clock Gear Value <GEAR2:0>	Prescaler Clock Resolution		
			φT1	φT4	φT16
1 (fs)	00 (fFPH)	XXX	fs/2^3 (244 μs)	fs/2^5 (977 μs)	fs/2^7 (3.9 μs)
0 (fc)		000 (fc)	fc/2^3 (0.5 μs)	fc/2^5 (2.0 μs)	fc/2^7 (8.0 μs)
		001 (fc/2)	fc/2^4 (1.0 μs)	fc/2^6 (4.0 μs)	fc/2^8 (16 μs)
		010 (fc/4)	fc/2^5 (2.0 μs)	fc/2^7 (8.0 μs)	fc/2^9 (32 μs)
		011 (fc/8)	fc/2^6 (4.0 μs)	fc/2^8 (16 μs)	fc/2^{10} (64 μs)
		100 (fc/16)	fc/2^7 (8.0 μs)	fc/2^9 (32 μs)	fc/2^{11} (128 μs)
	10 (fc/16 clock)	XXX	fc/2^7 (8.0 μs)	fc/2^9 (32 μs)	fc/2^{11} (128 μs)

xxx: Don't care

(2) Up counter (UC0)

UC0 is a 16-bit binary counter which counts up puls by TB0MOD<TB0CLK1:0>.

Any one of the prescaler internal clocks φT1, φT4 and φT16 or an external c input via the TB0IN0 pin can be selected as the input clearing of the counter is controlled by TB0RUN<TB0RUN>.

When clearing is enabled, the up counter UC0 will be cleared to zero e value matches the value in the timer register TB0RG disabled using TB0MOD<TB0CLE>.

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

A timer overflow interrupt (INTTBOF0) is generate

(3) Timer registers (TB0RG0 and TB0RG1)

These two 16-bit registers are used to set the inte
counter UC0 matches the value set in this timer register, the comparato
signal will go active.

Setting data for both upper and lower timer registers is needed
2-byte data transfer instruction or using 1-byte
lower 8 bits and upper 8 bits in order.
The TB0RG0 timer register has a double-buffer structure, which is paired
buffer. The value set in TB0RUN<TB0RDE> determine
structure is enabled or disabled: it is disabled when <TB0RDE> = 0, and enabled whe
<TB0RDE> = 1.

When the double buffer is enabled, data is transfe
timer register when the values in the up counter (UC
match.

After a reset, TB0RG0 and TB0RG1 are undefined. If a 16-bit timer
after a reset, data should be written to it beforeha

On a reset <TB0RDE> is initialized to 0, disabling the double buffe
double buffer, write data to the timer register, se
the register buffer as shown below.

TB0RG0 and the register buffer both have the same memory addresses
and 000189H) allocated to them. If <TB0RDE> = 0, the value is written t
timer register and the register buffer. If <TB0RDE
register buffer only.

The addresses of the timer registers are as follow

TMRB0

TB0RG0		TB0RG1	
Upper 8 bits (TB0RG0H)	Lower 8 bits (TB0RG0L)	Upper 8 bits (TB0RG1H)	Lower 8 bits (TB0RG1L)
000189H	000188H	00018BH	00018AH

The timer registers are write-only registers and thus cannot be read.

(4) Capture registers (TB0CP0H/L)

These 16-bit registers are used to latch the value

Data in the capture registers should be read all 16 bits. For example, u
data load instruction or two 1-byte data load instr
read first, followed by the most significant byte.

The addresses of the capture registers are as foll

TMRB0

TB0CP0	
MS 8 bits (TB0CP0H)	LS 8 bits (TB0CP0L)
00018DH	00018CH

The capture registers are read-only registers and thus cannot be written to.

(5) Capture input control

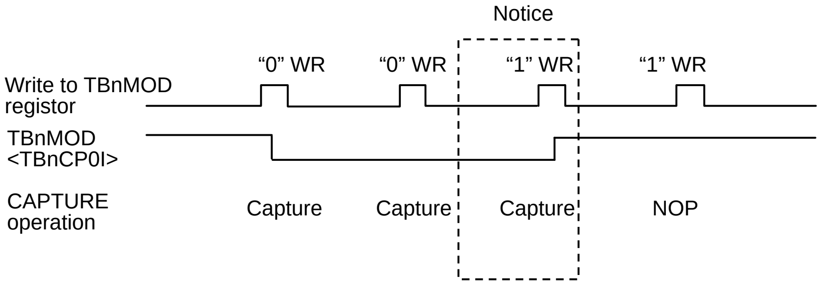
This circuit controls the timing to latch the valu

The value in the up-counter can be loaded into a c
Whenever 0 is written to TB0MOD<TB0CP0I>, the curre
loaded into capture register TB0CP0. It is necessa
mode (e.g., TB0RUN<TB0PRUN> must be held at a value

(6) Comparators (CP0 and CP1)

CP0 and CP1 are 16-bit comparators which compare the value in the up co
with the value set in TB0RG0 or TB0RG1 respectively
match is detected, the comparator generates an in
respectively).

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



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

These flip-flops are inverted by the match detect
the latch signals to the capture registers. Invers
each element using TB0FFCR<TB0E1T1, TB0E0T1>.

After a reset the value of TB0FF0 is undefined. I
<TB0FF0C1:0> or <TB0FF1C1:0>, TB0FF0 will be inve
capture registers, the value of TB0FF0 will be set t
registers, the value of TB0FF0 will be set to 0.

The values of TB0FF0 can be output via the timer ou
shared with PB6). Timer output should be specified

3.17.3 SFRs

TMRB0 Run Register

		7	6	5	4	3	2	1	0
TB0RUN	Bit symbol	TB0RDE	–			I2TB0	TB0PRUN		TB0RUN
(0180H)	Read/Write	R/W	R/W			R/W	R/W		R/W
	After reset	0	0			0	0		0
	Function	Double buffer 0: Disable 1: Enable	Always fixed to “0”.			IDLE2 0: Stop 1: Operate	16-Bit timer run/stop control 0: Stop and clear 1: Run (Count up)		

Count operation

0	Stop and clear
1	Count

I2TB0: Operation during IDLE2 mode
TB0PRUN: Operation of prescaler
TB0RUN: Operation of TMRB0

Note: The 1, 4 and 5 of TB0RUN are read as undefined value.

Figure 3.17.2 Register for TMRB

TMRB0 Mode Register

		7	6	5	4	3	2	1	0
TB0MOD	Bit symbol	–	–	TB0CP01	–	–	TB0CLE	TB0CLK1	TB0CLK0
(0182H)	Read/Write	R/W		W*	R/W	R/W	R/W		
Prohibit read-modify-write	After reset	0		1	0	0	0	0	0
	Function	Always fixed to “0”.		Execute software capture 0: Execute 1: Undefined	Always fixed to “0”.		Control up counter 0: Disable clearing 1: Enable clearing	TMRB0 source clock 00: Reserved 01: ϕT1 10: ϕT4 11: ϕT16	

TMRB0 source clock

00	Reserved
01	ϕT1
10	ϕT4
11	ϕT16

Up counter clear control

0	Disable
1	TB0RG1 clearing on match with TB0RG1.

Software capture

0	The value in the up counter is captured to TB0CP0.
1	Undefined (Note)

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

Figure 3.17.3 Register for TMRB

TMRB0 Flip-flop Control Register

		7	6	5	4	3	2	1	0
TB0FFCR (0183H)	Bit symbol	–	–	–	TB0C0T1	TB0E1T1	TB0E0T1	TB0FF0C1	TB0FF0C0
	Read/Write	W		R/W	R/W			W*	
	After reset	1	1	0	0	0	0	1	1
Prohibit read-modify-write	Function	Always fixed to “11”.		Always fixed to “0”.	TB0FF0 inversion trigger 0: Disable trigger 1: Enable trigger			Control TB0FF0 00: Invert 01: Set 10: Clear 11: Undefined Always read as “11”.	
					Invert when the UC value is loaded in to TB0CP0.	Invert when the UC value matches the value in TB0RG1.	Invert when the UC value matches the value in TB0RG0.		

TB0FF0 control

00	Invert
01	Set to 11
10	Clear to 00
11	Undefined (Always read as “11”)

Inverted when the UC value matches the valued in TB0RG0.

0	Disable trigger
1	Enable trigger

Inverted when the UC value matches the valued in TB0RG1.

0	Disable trigger
1	Enable trigger

Inverted when the UC value is loaded in to TB0CP0.

0	Disable trigger
1	Enable trigger

Figure 3.17.4 Register for TMRB

3.17.4 Operation in Each Mode

(1) 16-bit timer mode

Generating interrupts at fixed intervals

In this example, the interrupt INTTB01 is set to be generated at fixed i
intervals. The interval time is set in the timer register TB0RG1.

		7	6	5	4	3	2	1	0	
TB0RUN	←	0	0	X	X	–	0	X	0	Stop TMRB0.
INTETB01	←	X	1	0	0	X	0	0	0	Enable INTTB01 and set interrupt level 4. Disable INTTB00.
TB0FFCR	←	1	1	0	0	0	0	1	1	Disable the trigger.
TB0MOD	←	0	0	1	0	0	1	*	*	Select internal clock for input and
						(** = 01, 10, 11)				disable the capture function.
TB0RG1	←	*	*	*	*	*	*	*	*	Set the interval time (16 bits).
		*	*	*	*	*	*	*	*	
TB0RUN	←	0	0	X	X	–	1	X	1	Start TMRB0.

X: Don't care, –: No change

(2) 16-bit programmable pulse generation (PPG) outp

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

The PPG mode is obtained by inversion of the timer
enabled by the match of the up counter UC0 with time
and to be output to TB0OUT0. In this mode the following conditions mu

(Value set in TB0RG0) < (Value set in TB0RG1)

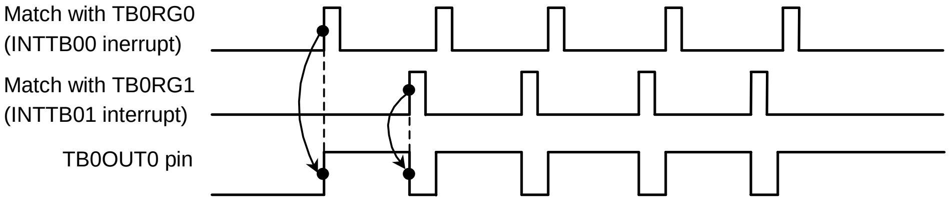


Figure 3.17.5 Programmable Pulse Generation (PPG) Output Waveforms

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

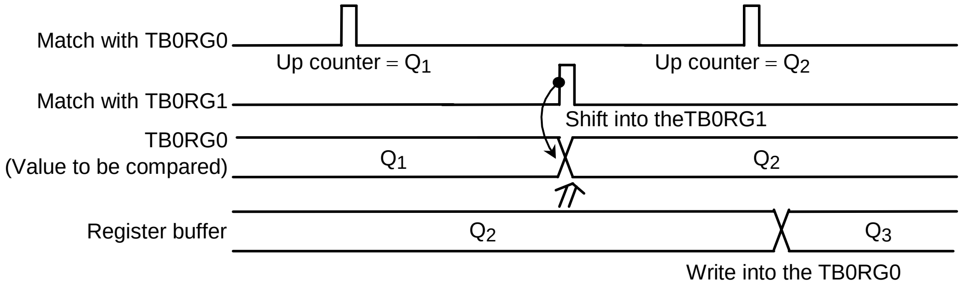


Figure 3.17.6 Operation of Register Buffer

The following block diagram illustrates this mode

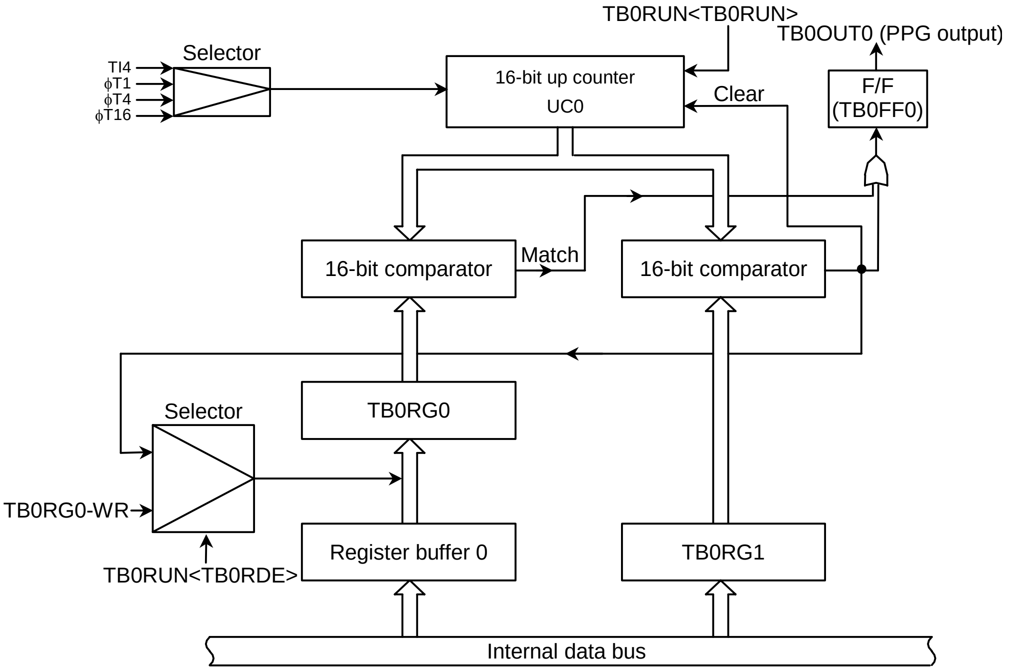


Figure 3.17.7 Block Diagram of 16-Bit Mode

The following example shows how to set 16-bit PPG o

		7	6	5	4	3	2	1	0	
TB0RUN	←	0	0	X	X	–	0	X	0	Disable the TB0RG0 double buffer and stop TMRB0.
TB0RG0	←	*	*	*	*	*	*	*	*	Set the duty ratio (16 bits).
TB0RG1	←	*	*	*	*	*	*	*	*	Set the frequency (16 bits).
TB0RUN	←	1	0	X	X	–	0	X	0	Enable the TB0RG0 double buffer. (The duty and frequency are changed on an INTTB01 interrupt.)
TB0FFCR	←	X	X	0	0	1	1	1	0	Set the mode to invert TB0FF0 at the match with TB0RG0/TB0RG1. Set TB0FF0 to 0.
TB0MOD	←	0	0	1	0	0	1	*	*	Select the internal clock as the input clock and disable
						(** = 01, 10, 11)				the capture function.
PBCR	←	X	1	–	–	–	X	–	–	Set PB6 to function as TB0OUT0.
PBFC	←	X	1	–	–	–	X	–	–	
TB0RUN	←	1	0	X	X	–	1	X	1	Start TMRB0.

X: Don’t care, –: No change

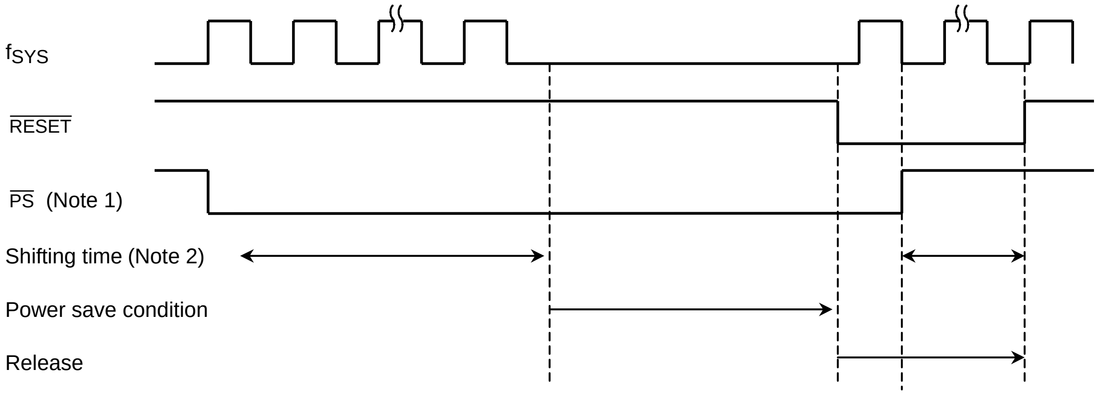
3.18 Hardware Standby Function

TMP91C820A have hardware standby circuit that is able protect from program runaway by supplying power voltage of battery using.

It can be shifted to “PS condition” by PS pin to “low” level fixed

Figure 3.18.1 shows timing diagram of PS condition below.

PS mode can release only external system reset.



Note 1: $\overline{PS}$ pin is effective after RESET because SYSCR2<PSENV> to 0. If you use as $\overline{NMI}$ pin, please write SYSCR2<PSENV> to 1.

Note 2: Shifting time is 2 to 10 clock times of f_{SYS}.

Figure 3.18.1 Hardware Standby Timing Diagram

Table 3.18.1 Power Save Mode Conditions of Each HALT Mode

HALT Mode Setting	IDLE2	IDLE1	STOP
PS condition	IDLE1 mode + High-frequency stop	IDLE1 mode + High-frequency stop	STOP mode

Note: Settings of SYSCR2<DRVE> and <SELDRV> at HALT mode are effective as well as PS condition.

4. Electrical Characteristics

4.1 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	−20 to 70	

Note: The 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 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 maximum rating value will ever be exceeded.

4.2 DC Characteristics (1/2)

Parameter		Symbol	Condition		Min	Typ.	Max	Unit
Power supply voltage (AVCC = DVCC) (AVSS = DVSS = 0 V)		VCC	fc = 4 to 27 MHz	fs = 30 to 34 kHz	2.7	–	3.6	V
			fc = 4 to 36 MHz		3.0	–	3.6	
Input low voltage	D0 to D15	VIL	V_{CC} ≥ 2.7 V		−0.3	–	0.6	V
	P52 to P7 (except PB3, P9)	VIL1	V_{CC} ≥ 2.7 V			–	0.3 Vcc	
	$\overline{\text{RESET}}$, $\overline{\text{NMI}}$, PB3 (INT0), P9	VIL2	V_{CC} ≥ 2.7 V			–	0.25 Vcc	
	AM0 to AM1	VIL3	V_{CC} ≥ 2.7 V			–	0.3	
	X1	VIL4	V_{CC} ≥ 2.7 V			–	0.2 Vcc	
Input high voltage	D0 to D15	VIH	3.6 V ≥ V_{CC} ≥ 2.7 V		2.0	–	Vcc + 0.3	
	P52 to P7 (except PB3, P9)	VIH1	V_{CC} ≥ 2.7 V		0.7 Vcc	–		
	$\overline{\text{RESET}}$, $\overline{\text{NMI}}$, PB3 (INT0), P9	VIH2	V_{CC} ≥ 2.7 V		0.75 Vcc	–		
	AM0 to AM1	VIH3	V_{CC} ≥ 2.7 V		Vcc − 0.3	–		
	X1	VIH4	V_{CC} ≥ 2.7 V		0.8 Vcc	–		
Output low voltage		VOL	IOL = 1.6 mA		–	–	0.45	V
Output high voltage		VOH	IOH = −400 μA		2.4	–	–	

Note: Typical values are for when Ta = 25°C and Vcc = 3.0 V unless otherwise noted.

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	±10	
Power down voltage (at STOP, RAM back up)	VSTOP	VIL2 = 0.2Vcc, VIH2 = 0.8Vcc	1.8	–	3.6	V
$\overline{\text{RESET}}$ pull-up resistor	RRST	3.6 V ≥ V_{CC} ≥ 2.7 V	100	–	400	kΩ
Pin capacitance	CIO	fc = 1 MHz	–	–	10	pF
Schmitt width $\overline{\text{RESET}}$, $\overline{\text{NMI}}$, INT0, KI0 to KI7	VTH	V_{CC} ≥ 2.7 V	0.4	1.0	–	V
Programmable pull-up resistor	RKH	3.6 V ≥ V_{CC} ≥ 2.7 V	100	–	400	kΩ
NORMAL (Note 2)	Icc	Vcc = 3.6 V fc = 36M Hz	–	23.0	35.0	mA
IDLE2			–	16.0	23.0	
IDLE1			–	1.6	3.0	
SLOW (Note 2)		Vcc = 3.6 V fs = 32.768 kHz	–	23.0	45.0	μA
IDLE2			–	14.0	35.0	
IDLE1			–	6.0	25.0	
STOP		Vcc = 3.6 V	–	0.2	15.0	μA

Note 1: Typical values are for when Ta = 25°C and Vcc = 3.0 V unless otherwise noted.

Note 2: Icc measurement conditions (NORMAL, SLOW):
All functions are operational; output pins are open and input pins are fixed. CL = 30 pF loaded on data and address bus.

4.3 AC Characteristics

Vcc = 2.7 to 3.6 V case of f_{FPH} = 27 MHz
Vcc = 3.0 to 3.6 V case of f_{FPH} = 36 MHz

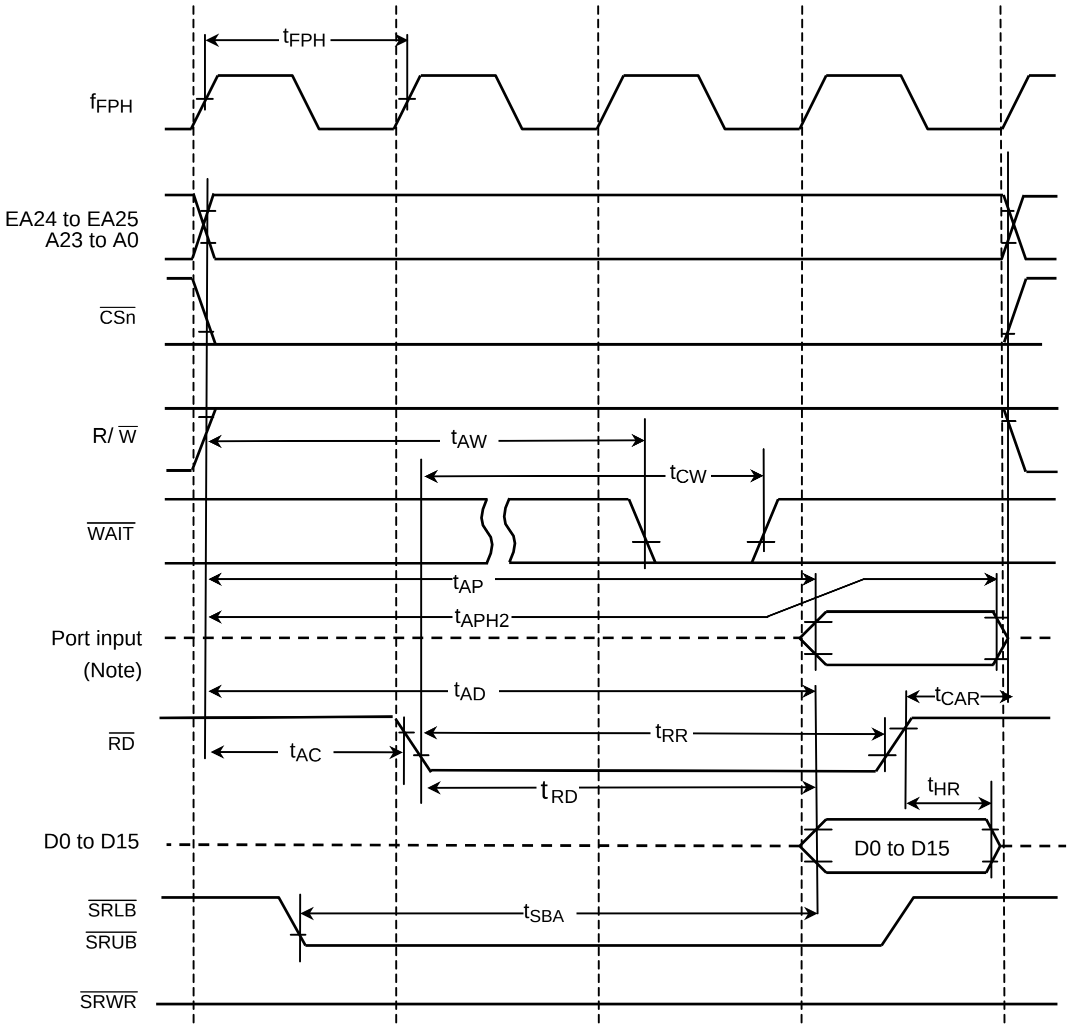
No.	Symbol	Parameter	Variable		f_{FPH} = 27 MHz		f_{FPH} = 36 MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	t_{FPH}	f_{FPH} period (= x)	27.7	31250	37		27.7		ns
2	t_{AC}	A0 to A23 valid → $\overline{RD}$ / $\overline{WR}$ fall	x – 23		14		4.7		ns
3	t_{CAR}	$\overline{RD}$ rise → A0 to A23 hold	0.5x –13		5.5		0.85		ns
4	t_{CAW}	$\overline{WR}$ rise → A0 to A23 hold	x – 13		24		14.7		ns
5	t_{AD}	A0 to A23 valid → D0 to D15 input		3.5x – 24		105.5		72.95	ns
6	t_{RD}	$\overline{RD}$ fall → D0 to D15 input		2.5x – 24		68.5		45.25	ns
7	t_{RR}	$\overline{RD}$ low width	2.5x – 15		77.5		54.25		ns
8	t_{HR}	$\overline{RD}$ rise → D0 to D15 hold	0		0		0		ns
9	t_{WW}	$\overline{WR}$ low width	2x – 15		59		40.4		ns
10	t_{DW}	D0 to D15 valid → $\overline{WR}$ rise	1.5x – 35		20.5		5.5		ns
11	t_{WD}	$\overline{WR}$ rise → D0 to D15 hold	x – 25		12		2.7		ns
12	t_{SBA}	Data byte control access time for SRAM		3x – 39		72		44.1	ns
13	t_{SWP}	Write pulse width for SRAM	2x – 15		59		40.4		ns
14	t_{SBW}	Data byte control to end of write for SRAM	3x – 25		86		58.1		ns
15	t_{SAS}	Address setup time for SRAM	1.5x – 35		20.5		6.55		ns
16	t_{SWR}	Write recovery time for SRAM	0.5x – 13		5.5		0.85		ns
17	t_{SDS}	Data setup time for SRAM	2x – 35		39		20.4		ns
18	t_{SDH}	Data hold time for SRAM	0.5x – 13		5.5		0.85		ns
19	t_{AW}	A0 to A23 valid → $\overline{WAIT}$ input (1 + N) wait		3.5x – 60		69.5		36.95	ns
20	t_{CW}	$\overline{RD}$ / $\overline{WR}$ fall → $\overline{WAIT}$ hold (1 + N) wait	2.5x + 0		92.5		69.25		ns
21	t_{APH}	A0 to A23 valid → PORT input		3.5x – 89		40.5		7.95	ns
22	t_{APH2}	A0 to A23 valid → PORT hold	3.5x		129.5		96.95		ns
23	t_{APO}	A0 to A23 valid → PORT valid		3.5x + 60		189.5		156.9	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

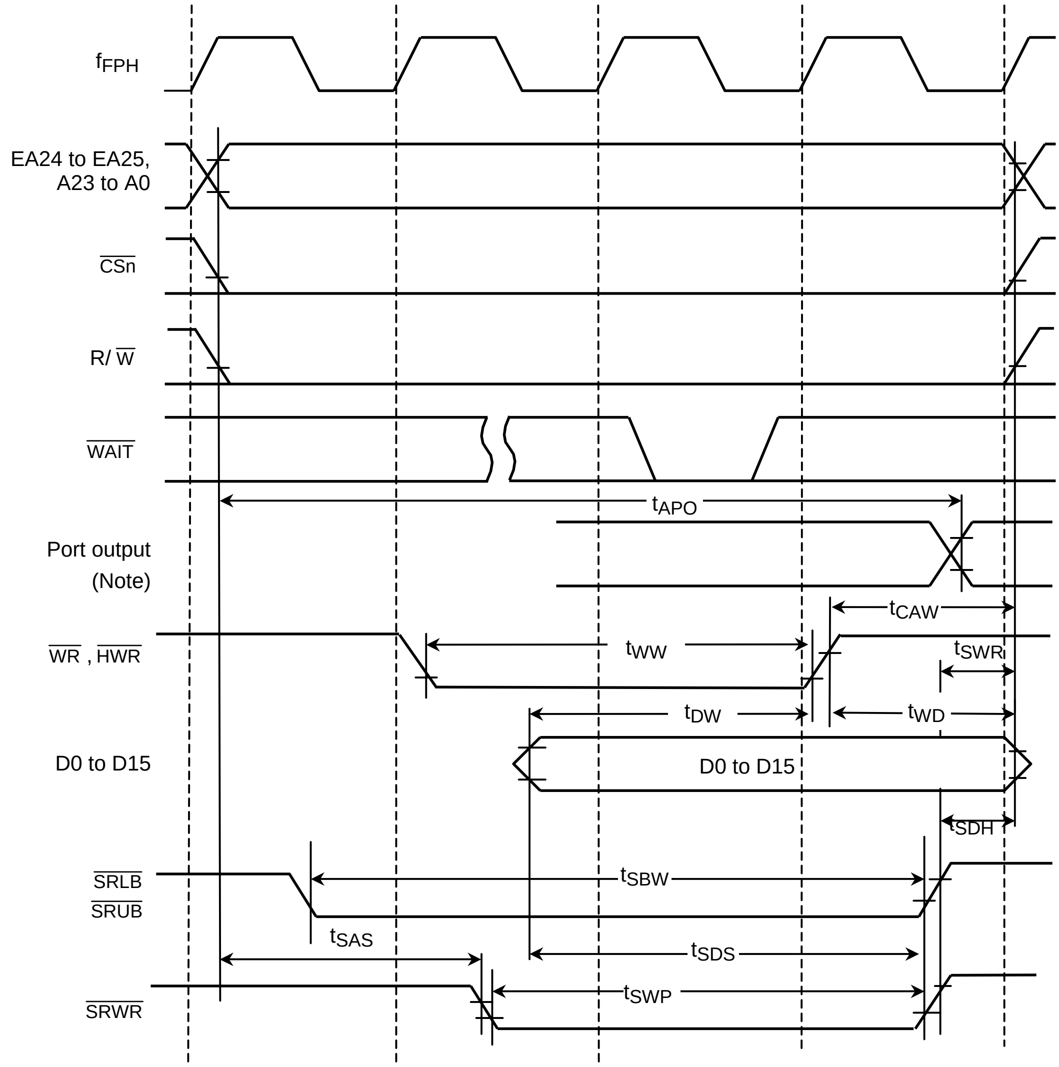
Note: Symbol x in the above table means the period of clock f_{FPH}, it's half period of the system clock f_{SYS} for CPU core. The period of f_{FPH} depends on the clock gear setting or the selection of high-/low-frequency oscillator.

(1) 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.

(2) 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.

4.4 SDRAM Controller AC Electrical Characteristics

Vcc = 2.7 to 3.6 V case of f_{FPH} = 27 MHz
Vcc = 3.0 to 3.6 V case of f_{FPH} = 36 MHz

No.	Symbol	Parameter	Variable		27 MHz		36 MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	t_{RC}	Ref/active to ref/active command period	4X		148		27.7		ns
2	t_{RAS}	Active to precharge command period	4X	12210	148	12210	111.1	12210	ns
3	t_{RCD}	Active to read/write command delay time	2X		74		55.6		ns
4	t_{RP}	Precharge to active command period	2X		74		55.6		ns
5	t_{RRD}	Active to active command period	6X		222		166.7		ns
6	t_{WR}	Write recovery time (CL* = 2)	2X		74		55.6		ns
7	t_{WR2}	Write recovery time	3X		111		83		
8	t_{CK}	CLK cycle time (CL* = 2)	2X		74		55.6		ns
9	t_{CH}	CLK high level width	1X−15		22		12.8		ns
10	t_{CL}	CLK low level width	1X−15		22		12.8		ns
11	t_{AC}	Access time from CLK (CL* = 2)		1X−25		12		2.8	ns
12	t_{OH}	Output data hold time	0		0		0		ns
13	t_{DS}	Data-in setup time	2X−35		39		20.6		ns
14	t_{DH}	Data-in hold time	2.5X−20		72		49.4		ns
15	t_{AS}	Address setup time	1.5X−35		20		6.7		ns
16	t_{AH}	Address hold time	0.5X−13		5		0.9		ns
17	t_{CKS}	CKE setup time	1X−15		22		12.8		ns
18	t_{CMS}	Command setup time	1X−15		22		12.8		ns
19	t_{CMH}	Command hold time	1X−15		22		12.8		ns
20	t_{RSC}	Mode register set cycle time	2X		74		55.6		ns

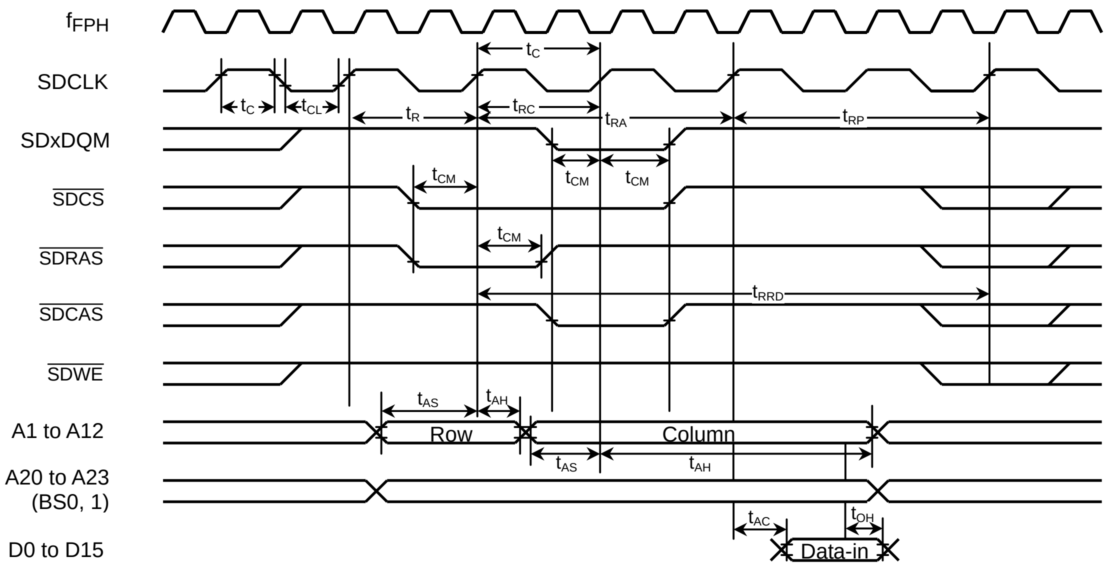
* CL is $\overline{\text{CAS}}$ latency.

AC measuring conditions

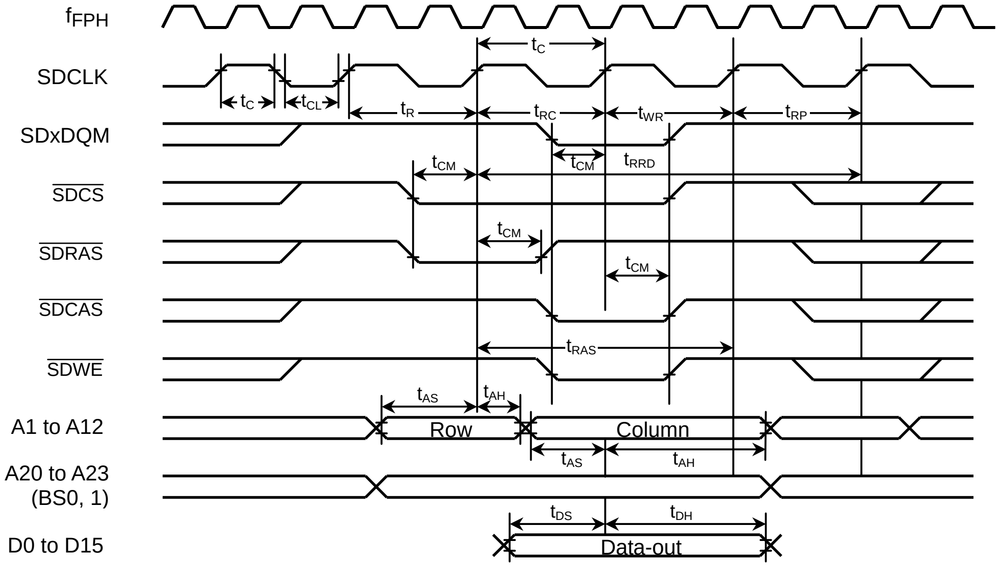
- Output lev=e0l.:7HVicgch=,0L.o3wVc=e5,0CpF
- Input lev=e0l.:9HVi ogd=,0L.o1wVcc

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-frequency oscillator.

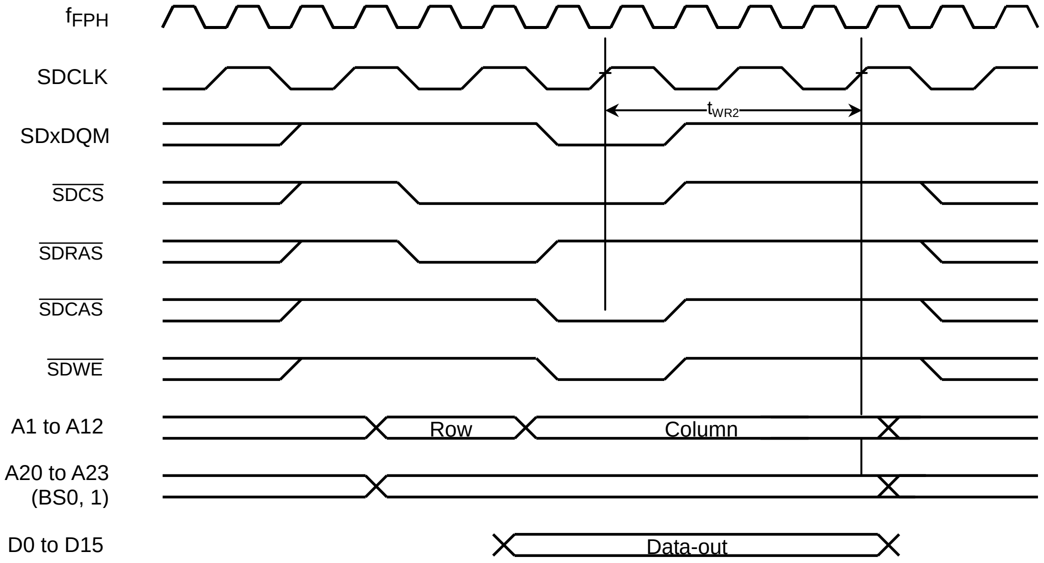
- SDRAM read timing (CPU access)



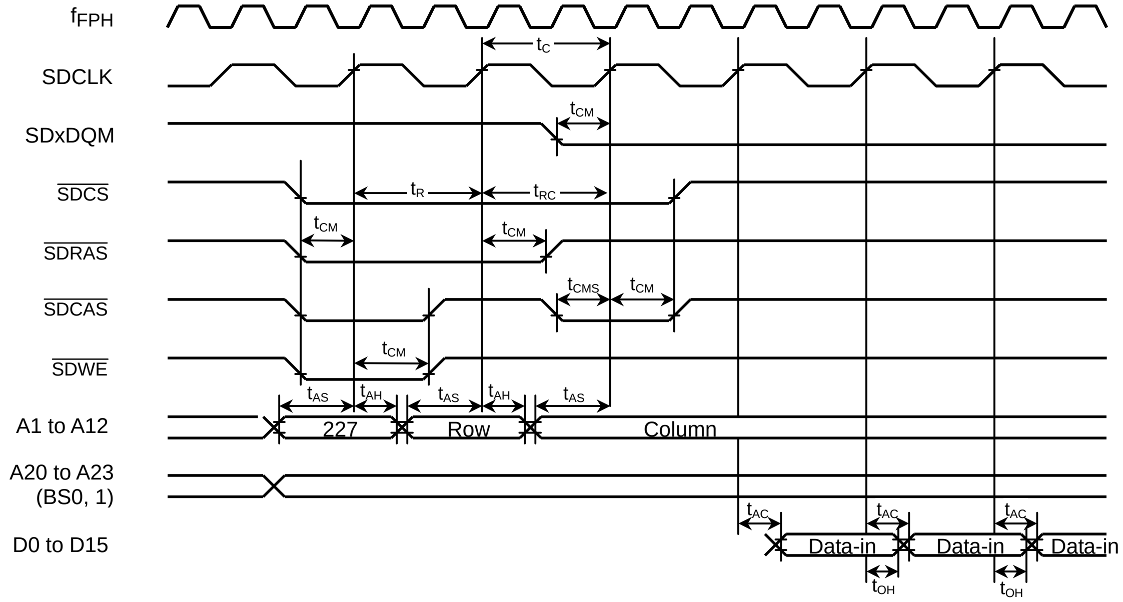
- SDRAM write timing (CPU access)



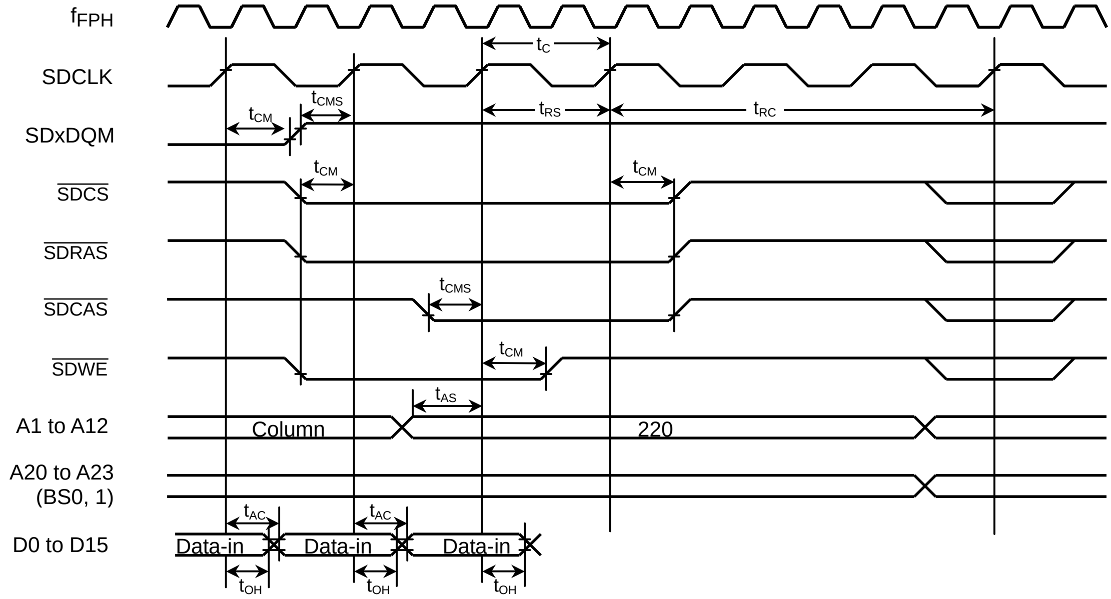
- SDRAM write timing (CPU access, write recivery enable)



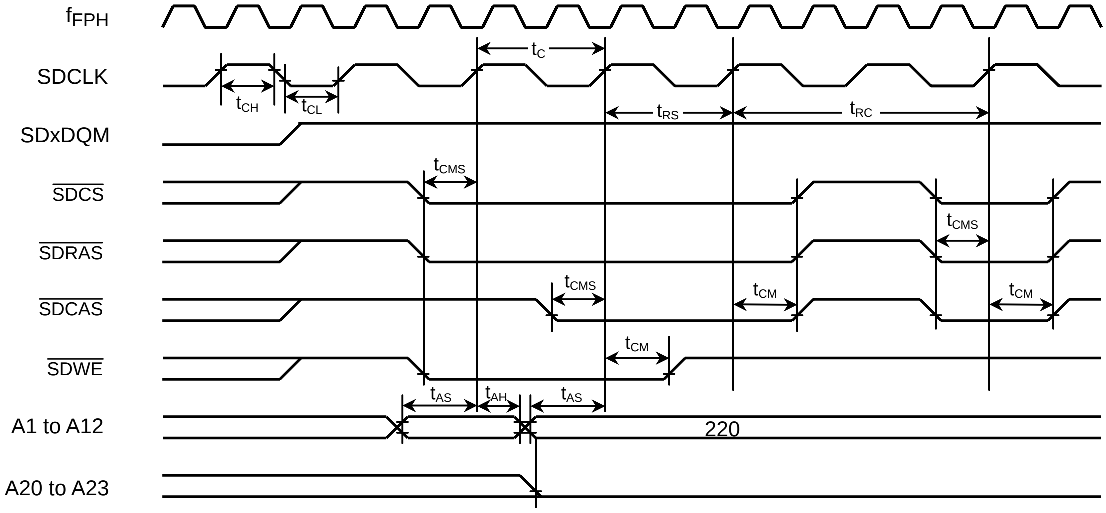
- SDRAM burst read timing (Head of burst cycle)



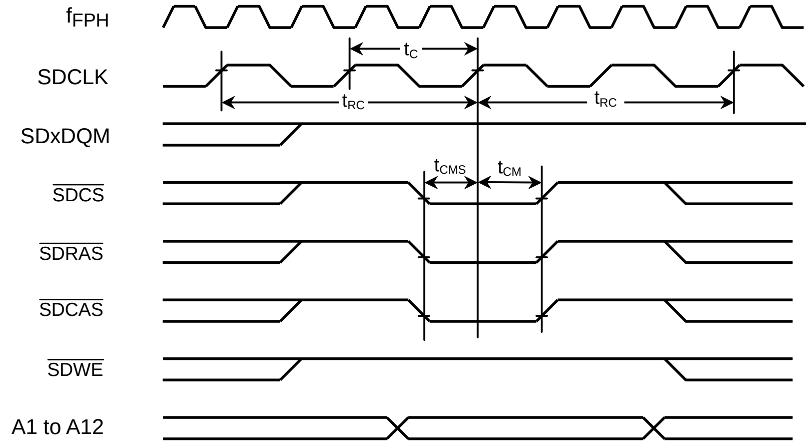
- SDRAM burst read timing (End of burst cycle)



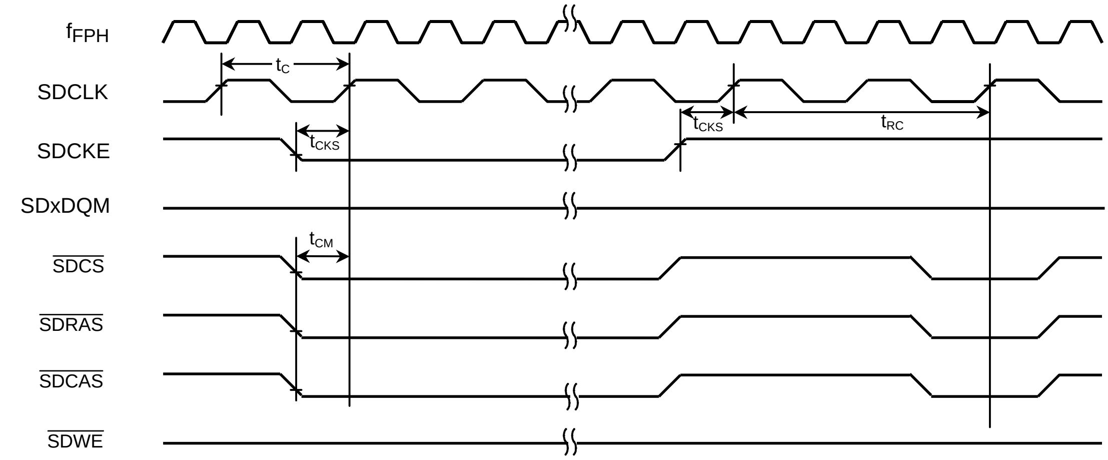
- SDRAM initialize timing



- SDRAM refresh timing



- SDRAM self-refresh timing



4.5 AD Conversion Characteristics

AVcc = Vcc, AVss = Vss

Symbol	Parameter	Condition	Min	Typ.	Max	Unit
VREFH	Analog reference voltage (+)	VCC = 2.7 V to 3.6 V	Vcc − 0.2 V	Vcc	Vcc	V
VREFL	Analog reference voltage (−)		Vss	Vss	Vss + 0.2 V	
VAIN	Analog input voltage range		V_{REFL}		V_{REFH}	
IREF (VREFL = 0V)	Analog current for analog reference voltage <VREFON> = 1			0.94	1.20	mA
	<VREFON> = 0			0.02	5.0	μA
−	Error (Not including quantizing errors)			±1.0	±4.0	LSB

Note 1: 1 LSB = (VREFH − VREFL)/1024 [V].

Note 2: The operation above is guaranteed for f_{FPH} ≥ 4 MHz.

Note 3: The value of Icc includes the current which flows through the AVCC pin.

4.6 Serial Channel Timing (I/O internal mode)

(1) SCLK input mode

Vcc = 2.7 to 3.6 V case of f_{FPH} = 27 MHz
Vcc = 3.0 to 3.6 V case of f_{FPH} = 36 MHz

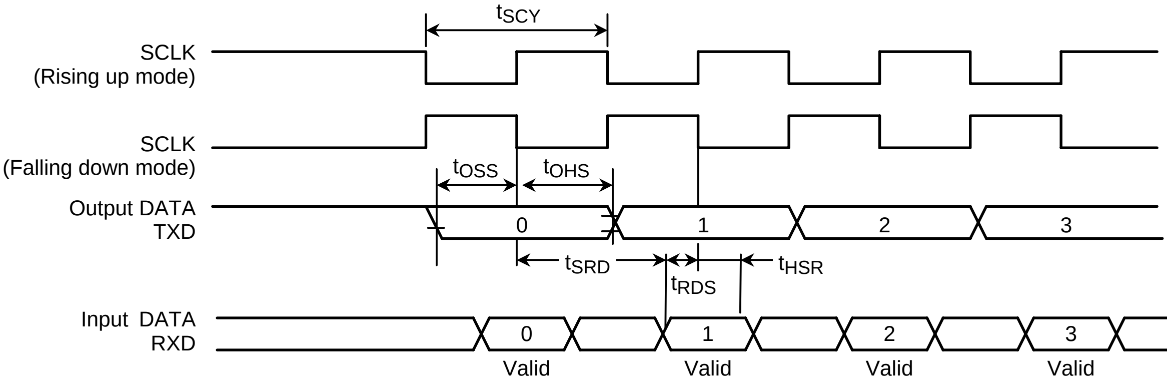
Symbol	Parameter	Variable		27 MHz		36 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{SCY}	SCLK period	16X		0.59		0.44		μs
t_{OSS}	Output data → SCLK rising/falling edge*	t_{SCY}/2 − 4X-110		38		0		ns
t_{OHS}	SCLK rising/falling edge* → Output data hold	t_{SCY}/2 + 2X + 0		370		277		ns
t_{HSR}	SCLK rising/falling edge* → Input data hold	3X + 10		121		93		ns
t_{SRD}	SCLK rising/falling edge* → Valid data input		t_{SCY} − 0		592		443	ns
t_{RDS}	SCLK rising/falling edge* → Valid data input	0		0		0		ns

(2) SCLK output mode

Symbol	Parameter	Variable		27 MHz		36 MHz		Unit
		Min	Max	Min	Max	Min	Max	
t_{SCY}	SCLK period	16X	8192X	0.59	303	0.44	227	μs
t_{OSS}	Output data → SCLK rising/falling edge*	t_{SCY}/2 − 40		256		181		ns
t_{OHS}	SCLK rising/falling edge* → Output data hold	t_{SCY}/2 − 40		256		181		ns
t_{HSR}	SCLK rising/falling edge* → Input data hold	0		0		0		ns
t_{SRD}	SCLK rising/falling edge* → Valid data input		t_{SCY} − 1X − 180		375		235	ns
t_{RDS}	SCLK rising/falling edge* → Valid data input	1X + 180		217		207.7		ns

SCLK rising/falling edge*: The rising edge is used in SCLK rising mode.
The Falling edge is used in SCLK falling mode.

Note: Above table's data values at 27 MHz and 36 MHz are calculated from t_{SCY} = 16x base.



4.7 Event Counter (TA0IN)

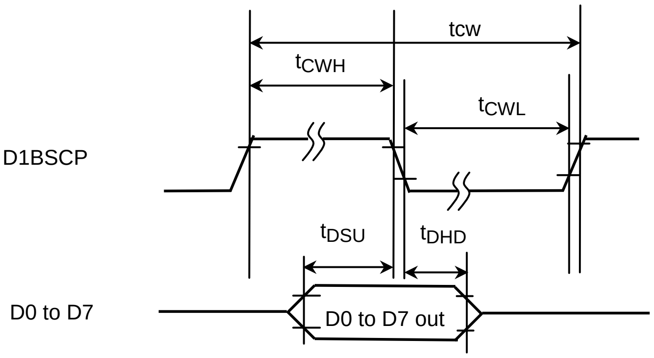
Symbol	Parameter	Variable		27 MHz (Vcc = 2.7 to 3.6 V)		36 MHz (Vcc = 3.0 to 3.6 V)		Unit
		Min	Max	Min	Max	Min	Max	
t_{VCK}	Clock period	8X + 100		396		321		ns
t_{VCKL}	Clock low level width	4X + 40		188		151		ns
t_{VCKH}	Clock high level width	4X + 40		188		151		ns

4.8 Interrupt, Capture

(1) $\overline{\text{NMI}}$, INT0 to INT3 interrupts

Symbol	Parameter	Variable		27 MHz (Vcc = 2.7 to 3.6 V)		36 MHz (Vcc = 3.0 to 3.6 V)		Unit
		Min	Max	Min	Max	Min	Max	
t_{INTAL}	$\overline{\text{NMI}}$, INT0 to INT3 low level width	4X + 40		188		151		ns
t_{INTAH}	$\overline{\text{NMI}}$, INT0 to INT3 high level width	4X + 40		188		151		ns

4.9 LCD Controller SR Mode



Vcc = 2.7 to 3.6 V case of f_{FPH} = 27 MHz
Vcc = 3.0 to 3.6 V case of f_{FPH} = 36 MHz

No.	Symbol	Parameter	Variable		f_{FPH} = 27 MHz (Case: tm = 0)		f_{FPH} = 36 MHz (Case: tm = 0)		Unit
			Min	Max	Min	Max	Min	Max	
1	t_{DSU}	Data valid → D1BSCP fall	x − 20 + tm		17		7.7		ns
2	t_{DHD}	D1BSCP fall → Data hold	x − 5 + tm		32		22		ns
3	t_{CWH}	D1SBCP → Clock high width	x − 10 + tm		27		17.7		ns
4	t_{CWL}	D1BSCP → Clock low width	x − 10 + tm		27		17.7		ns
5	t_{CW}	D1BSCP → Clock cycle	2x + 2tm		27		55.4		ns

tm = $(2^{SCPW} - 1)$ X SCPW: Setting of (LCDMODE)<SCPW1:0>

X: $1/f_{FPH}$

Example: If SCPW = 3 (8 clock mode), f_{FPH} = 36 [MHz]

tm = $(2^3 - 1)$ * 1/36 MHz = 194.4 [ns]

Table 4.9.1 tm at f_{FPH} = 36 [MHz]

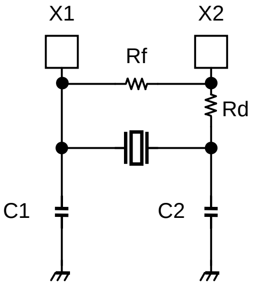
SCP Width	<SCPW1:0>	tm
Base SCP	00	0 ns
2 clocks	01	27.77 ns
4 clocks	10	83.31 ns
8 clocks	11	194.4 ns

4.10 Recommended Crystal Oscillation Circuit

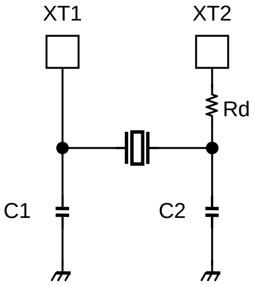
TMP91C820A is evaluated by below oscillator vender. W
use of this information.

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) TMP91C820A recommended oscillator: Murata Manufactu

Circuit parameter recommended

MCU	Oscillation Frequency [MHZ]	Item of Oscillator Upper: Old Lower: New	Parameter of Elements				Running Condition	
			C1 [pF]	C2 [pF]	Rf [Ω]	Rd [Ω]	Voltage of Power [V]	Tc [°C]
TMP91C820A	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	CSTS1000MG03 *CSTLS10M0G53-B0	(15)	(15)	Open	0		
	12.50	CSA12.5MTZ093 *CSALA12M5T55093-B0	30	30	Open	0		
		CST12.0MTW093 *CSTLA12M5T55093-B0	(30)	(30)	Open	0		

MCU	Oscillation Frequency [MHz]	Item of Oscillator Upper: Old Lower: New	Parameter of Elements				Running Condition	
			C1 [pF]	C2 [pF]	Rf [Ω]	Rd [Ω]	Voltage of Power [V]	Tc [°C]
TMP91C820A	4.00	CSTS0400MG06 *CSTLS4M00G56-B0	(47)	(47)	Open	0	2.7 to 3.6	−40 to +85
	6.750	CSTS0675MG06 *CSTLS6M75G56-B0	(47)	(47)	Open	0		
	12.50	CSA12.5MTZ *CSALA12M5T55-B0	30	30	Open	0		
		CST12.0MTW *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		

Note: In CST ***type oscillator, capacitance C1, C2 is built in.

- The product numbers and specifications of the resona
  Ltd. are subject to change. For up-to-date information, please refer to the following UR
  http://www.murata.co.jp/search/index.html

5. Table of SFRs

The special function registers (SFRs) include the I/O
allocated to the 4-Kbyte address space from 000000H to 00

(1) I/O ports

(2) I/O port control

(3) Interrupt control

(4) Chip select/wait control

(5) Clock gear

(6) DFM (Clock doubler)

(7) 8-bit timer

(8) UART/serial channel

(9) I²C bus/serial interface

(10) AD converter

(11) Watchdog timer

(12) RTC (Real time clock)

(13) Melody/alarm generator

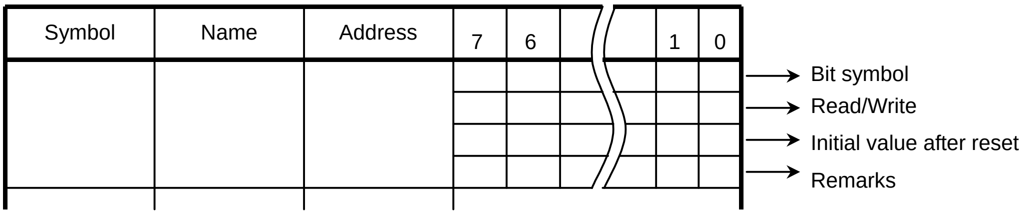
(14) MMU

(15) LCD controller

(16) SDRAM controller

(17) 16-bit timer

Table layout



Note: "Prohibit RMW" in the table means that you cannot use RMW instructions on these register.

Example: When setting bit0 only of the register P0CR,
cannot be used. The LD (Transfer) instruction must be used to write all

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 r

Prohibit RMW: Read-modify-write is prohibited when controlling this register. (The EX, ADD
SBC, INC, DEC, AND, OR, XOR, STCF, RES, SET, CHG, TSET, RLC
RL, RR, SLA, SRA, SLL, SRL, RLD and RRD i
read-modify-write instructions.)

Prohibit RMW*: Read-modify-write is prohibited when contro

Table 5.1 SFR Address Map (1/4)

[1], [2] PORT

Address	Name
0000H	P0
1H	P1
2H	P0CR
3H	
4H	P1CR
5H	P1FC
6H	P2
7H	P3
8H	P2CR
9H	P2FC
AH	P3CR
BH	P3FC
CH	P4
DH	P5
EH	P4CR
FH	P4FC

Address	Name
0010H	P5CR
1H	
2H	P6
3H	P7
4H	
5H	P6FC
6H	P7CR
7H	P7FC
8H	P8
9H	P9
AH	
BH	P6FC2
CH	P7FC2
DH	P9FC
EH	PA
FH	P7ODE

Address	Name
0020H	
1H	PAFC
2H	PB
3H	PC
4H	PBCR
5H	PBFC
6H	PCCR
7H	PCFC
8H	PCODE
9H	PD
AH	PDFC
BH	PBODE
CH	PE
DH	PECR
EH	PEFC
FH	

Address	Name
0030H	PF
1H	
2H	PFFC
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Name
0070H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	PZ
EH	PZCR
FH	PZFC

[3] INTC

Address	Name
0080H	DMA0V
1H	DMA1V
2H	DMA2V
3H	DMA3V
4H	
5H	
6H	
7H	
8H	INTCLR
9H	DMAR
AH	DMAB
BH	
CH	IIMC
DH	
EH	
FH	

Address	Name
0090H	INTE0AD
1H	INTE12
2H	INTE3ALM4
3H	INTEALM01
4H	INTEALM23
5H	INTETA01
6H	INTETA23
7H	INTERTCKEY
8H	INTES0
9H	INTES1
AH	INTES2LCD
BH	INTETC01
CH	INTETC23
DH	INTEP01
EH	INTESS01
FH	INTESS2

Address	Name
00A0H	INTES3
1H	INTETB0
2H	
3H	
4H	
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 5.2 SFR Address Map (2/4)

[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	
8H	DFMCR0
9H	DFMCR1
AH	
BH	
CH	
DH	
EH	
FH	

[7] TMRA

Address	Name
0100H	TA01RUN
1H	
2H	TA0REG
3H	TA1REG
4H	TA01MOD
5H	TA01FFCR
6H	
7H	
8H	TA23RUN
9H	
AH	TA2REG
BH	TA3REG
CH	TA23MOD
DH	TA3FFCR
EH	
FH	

[8] UART/SIO

Address	Name
0200H	SC0BUF
1H	SC0CR
2H	SC0MOD0
3H	BR0CR
4H	BR0ADD
5H	SC0MOD1
6H	
7H	SIRCR
8H	SC1BUF
9H	SC1CR
AH	SC1MOD0
BH	BR1CR
CH	BR1ADD
DH	SC1MOD1
EH	
FH	

Address	Name
0210H	SC2BUF
1H	SC2CR
2H	SC2MOD0
3H	BR2CR
4H	BR2ADD
5H	SC2MOD1
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[9] I^{2}C bus/SIO

Address	Name
0240H	SBI0CR1
1H	SBI0DBR
2H	I2C0AR
3H	SBI0CR2/SBI0SR
4H	SBI0BR0
5H	SBI0BR1
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Note: Do not access to the unnamed addresses (e.g., addresses to which no register has been allocated).

Table 5.3 SFR Address Map (3/4)

[10] 10-bit ADC

Address	Name
02A0H	ADREG04L
1H	ADREG04H
2H	ADREG15L
3H	ADREG15H
4H	ADREG26L
5H	ADREG26H
6H	ADREG37L
7H	ADREG37H
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Address	Name
02B0H	ADMOD0
1H	ADMOD1
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[11] WDT

Address	Name
0300H	WDMOD
1H	WDCR
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[12] RTC

Address	Name
0320H	SECR
1H	MINR
2H	HOURR
3H	DAYR
4H	DATER
5H	MONTHR
6H	YEARR
7H	PAGER
8H	RESTR
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[13] MLD

Address	Name
0330H	ALM
1H	MELALMC
2H	MELFL
3H	MELFH
4H	ALMINT
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[14] MMU

Address	Name
0350H	LOCAL0
1H	LOCAL1
2H	LOCAL2
3H	LOCAL3
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Note: Do not access to the unnamed addresses (e.g., addresses to which no register has been allocated).

Table 5.4 SFR Address Map (4/4)

[15] LCDC

Address	Name
04B0H	LCDMODE
1H	LCDDVM
2H	LCDSIZE
3H	LCDCTL
4H	LCDFFP
5H	LCDGL
6H	LCDCM
7H	LCDCW
8H	LCDCH
9H	LCDCP
AH	LCDCPL
BH	LCDCPM
CH	LCDCPH
DH	
EH	
FH	

Address	Name
04C0H	LSARAM
1H	LSARAH
2H	LEARAM
3H	LEARAH
4H	LSARBM
5H	LSARBH
6H	LEARBM
7H	LEARBH
8H	LSARCL
9H	LSARCM
AH	LSARCH
BH	
CH	
DH	
EH	
FH	

Address	Name
04D0H	LG0L
1H	LG0H
2H	LG1L
3H	LG1H
4H	LG2L
5H	LG2H
6H	LG3L
7H	LG3H
8H	LG4L
9H	LG4H
AH	LG5L
BH	LG5H
CH	LG6L
DH	LG6H
EH	LG7L
FH	LG7H

Address	Name
04E0H	LG8L
1H	LG8H
2H	LG9L
3H	LG9H
4H	LGAL
5H	LGAH
6H	LGBL
7H	LGBH
8H	LGCL
9H	LGCH
AH	LGDL
BH	LGDH
CH	LGEL
DH	LGEH
EH	LGFL
FH	LGFH

[16] SDRAMC

Address	Name
04F0H	SDACR
1H	SDRCR
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[17] TMRB

Address	Name
0180H	TB0RUN
1H	
2H	TB0MOD
3H	TB0FFCR
4H	
5H	
6H	
7H	
8H	TB0RG0L
9H	TB0RG0H
AH	TB0RG1L
BH	TB0RG1H
CH	TB0CP0L
DH	TB0CP0H
EH	TB0CP1L
FH	TB0CP1H

Note: Do not access to the unnamed addresses (e.g., address to which no register has been allocated).

(1) I/O ports (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P0	PORT0	00H	P07	P06	P05	P04	P03	P02	P01	P00
			R/W							
			Data from external port (Output latch register is cleared to 0.)							
P1	PORT1	01H	P17	P16	P15	P14	P13	P12	P11	P10
			R/W							
			Data from external port (Output latch register is cleared to 0.)							
P2	PORT2	06H	P27	P26	P25	P24	P23	P22	P21	P20
			R/W							
			Data from external port (Output latch register is cleared to 0.)							
P3	PORT3	07H	P37	P36	P35	P34	P33	P32	P31	P30
			R/W							
			Data from external port (Output latch register is cleared to 0.)							
P4	PORT4	0CH	P47	P46	P45	P44	P43	P42	P41	P40
			R/W							
			Data from external port (Output latch register is cleared to 0.)							
PZ	PORTZ	7DH (Prohibit RMW*)					PZ3	PZ2	PZ1	PZ0
							R/W			
							Data from external port (Output latch register is set to 1.)		1	1
							0: Pull-up resistor OFF 1: Pull-up resistor ON		–	
P5	PORT5	0DH (Prohibit RMW*)		P56						
				R/W						
				Data from external port (Output latch register is set to 1.)						
				0: Pull-up resistor OFF 1: Pull-up resistor ON						
P6	PORT6	12H	P67	P66	P65	P64	P63	P62	P61	P60
			R/W							
			1	1	1	1	1	0	1	1
P7	PORT7	13H	P77	P76	P75	P74	P73	P72	P71	P70
			R/W							
			Data from external port (Output latch register is set to 1.)							
P8	PORT8	18H	P87	P86	P85	P84	P83	P82	P81	P80
			R							
			Data from external port							
P9	PORT9	19H	P97	P96	P95	P94	P93	P92	P91	P90
			R							
			Data from external port							
PA	PORTA	1EH	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
			R/W							
			1							

I/O ports (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
PB	PORTB	22H		PB6	PB5	PB4	PB3		PB1	PB0
				R/W					R/W	
				Data from external port (Output latch register is set to 1.)					Data from external port (Output latch register is set to 1.)	
PC	PORTC	23H			PC5	PC4	PC3	PC2	PC1	PC0
					R/W					
					Data from external port (Output latch register is set to 1.)					
PD	PORTD	29H	PD7	PD6		PD4	PD3	PD2	PD1	PD0
			R/W			R/W				
			1	1		1	1	1	1	1
PE	PORTE	2CH	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
			R/W							
			Data from external port (Output latch register is set to 1.)							
PF	PORTF	30H	PF7	PF6	PF5	PF4	PF3	PF2	PF1	PF0
			R/W							
			1	1	1	1	1	1	1	1

(2) I/O port control (1/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P0CR	PORT0 control	02H (Prohibit RMW)	P07C	P06C	P05C	P04C	P03C	P02C	P01C	P00C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P1CR	PORT1 control	04H (Prohibit RMW)	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P1FC	PORT1 function	05H (Prohibit RMW)	P17F	P16F	P15F	P14F	P13F	P12F	P11F	P10F
			W							
			0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
			0: Port, 1: Data bus (D15 to D8)							
P2CR	PORT2 control	08H (Prohibit RMW)	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
			W							
			0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
			0: Input 1: Output							
P2FC	PORT2 function	09H (Prohibit RMW)	P27F	P26F	P25F	P24F	P23F	P22F	P21F	P20F
			W							
			0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
			0: Port, 1: Address bus (A23 to A16)							
P3CR	PORT3 control	0AH (Prohibit RMW)	P37C	P36C	P35C	P34C	P33C	P32C	P31C	P30C
			W							
			0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
			0: Input 1: Output							
P3FC	PORT3 function	0BH (Prohibit RMW)	P37F	P36F	P35F	P34F	P33F	P32F	P31F	P30F
			W							
			0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
			0: Port, 1: Address bus (A15 to A8)							
P4CR	PORT4 control	0EH (Prohibit RMW)	P47C	P46C	P45C	P44C	P43C	P42C	P41C	P40C
			W							
			0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
			0: Input 1: Output							
P4FC	PORT4 function	0FH (Prohibit RMW)	P47F	P46F	P45F	P44F	P43F	P42F	P41F	P40F
			W							
			0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1
			0: Port, 1: Address bus (A7 to A0)							

I/O port control (2/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
PZCR	PORTZ control	7EH (Prohibit RMW)					PZ3C	PZ2C		
							W			
							0	0		
							0: Input 1: Output			
P5CR	PORT5 control	10H (Prohibit RMW)		P56C						
				W						
				0						
				0: Input 1: Output						
PZFC	PORTZ function	7FH (Prohibit RMW)					PZ3F	PZ2F	PZ1F	PZ0F
							W			
							0			
							0: Port 1: R/W, $\overline{\text{SRWE}}$	0: Port 1: $\overline{\text{HWR}}$	0: Port 1: $\overline{\text{WR}}$	0: Port 1: $\overline{\text{RD}}$
P6FC	PORT6 function	15H (Prohibit RMW)	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
			W							
			0							
			0: Port 1: $\overline{\text{SRUB}}$	0: Port 1: $\overline{\text{SRLB}}$	0: Port 1: EA25	0: Port 1: EA24	0: Port 1: $\overline{\text{CS3}}$	0: Port 1: $\overline{\text{CS2}}$	0: Port 1: $\overline{\text{CS1}}$	0: Port 1: $\overline{\text{CS0}}$
P6FC2	PORT6 function 2	1BH (Prohibit RMW)	P67F2	P66F2	P65F2	P64F2	–	P62F2	P61F2	–
			W							
			0							
			0: <P67F> 1: $\overline{\text{CS2E}}$	0: <P66F> 1: $\overline{\text{CS2D}}$	0: <P65F> 1: $\overline{\text{CS2C}}$	0: <P64F> 1: $\overline{\text{CS2B}}$	Always fixed to "0".	0: <P62F> 1: $\overline{\text{CS2A}}$	0: <P61F> 1: $\overline{\text{SDCS}}$	Always fixed to "0".
P7CR	PORT7 control	16H (Prohibit RMW)	P77C	P76C	P75C	P74C	P73C	P72C	P71C	P70C
			W							
			0							
			0: Input 1: Output							
P7FC	PORT7 function	17H (Prohibit RMW)	P77F	P76F	P75F	P74F	P73F	P72F	P71F	P70F
			W							
			0							
			0: Port 1: VEECLK	MSK logic select 0: CLK by 1 1: CLK by 0	0: Port	0: Port	0: Port	0: Port 1: SCL	0: Port 1: SDA/SO	0: Port 1: SCK
P7FC2	PORT7 function 2	1CH (Prohibit RMW)	–	–	P75F2	P74F2	P73F2	–	P71F2	P70F2
			W							
			0							
			Always fixed to "0".	Always fixed to "0".	0: <P75F> 1: $\overline{\text{CSEXA}}$	0: <P74F> 1: $\overline{\text{CS2G}}$	0: <P73F> 1: $\overline{\text{CS2F}}$	Always fixed to "0".	0: <P71F> 1: OPTTX0	SIO0/RXD0 PIN SELECT 0: RXD0 (PC1) 1: OPTRX0 (P70)

I/O port control (3/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P7ODE	PORT7 open drain	1FH (Prohibit RMW)	–	–				ODEP72	ODEP71	
			W					W		
			0					0	0	
			Always fixed to “0”.					0: 3 states 1: Open drain		
P9FC	PORT9 function	1DH (Prohibit RMW)	P97F	P96F	P95F	P94F	P93F	P92F	P91F	P90F
			W							
			0							
			0: Key-in disable 1:Key-in enable							
PAFC	PORTA function	21H (Prohibit RMW)	PA7F	PA6F	PA5F	PA4F	PA3F	PA2F	PA1F	PA0F
			W							
			0							
			0: CMOS output 1: Open-drain output							
PBCR	PORTB control	24H (Prohibit RMW)		PB6C	PB5C	PB4C	PB3C		PB1C	PB0C
				W					W	
				0					0	
				0: Input 1: Output					0: Input 1: Output	
PBFC	PORTB function	25H (Prohibit RMW)		PB6F	PB5F	PB4F	PB3F		PB1F	PB0F
				W					W	
				0					0	
				0: Port 1: INT3, TB0OUT0	0: Port 1: INT2 TA3OUT	0: Port 1: INT1	0: Port 1: INT0		0: Port 1: TA1OUT	0: Port 1: TXD2
PBODE	PORTB open drain	2BH (Prohibit RMW)								ODEPB0
										W
										0
										0: CMOS 1: Open drain
PCCR	PORTC control	26H (Prohibit RMW)			PC5C	PC4C	PC3C	PC2C	PC1C	PC0C
					W					
					0					
					0: Input 1: Output					
PCFC	PORTC function	27H (Prohibit RMW)			PC5F		PC3F	PC2F		PC0F
					W		W			W
					0		0			0
					0: Port 1: SCLK1		0: Port 1: TXD1	0: Port 1: SCLK0		0: Port 1: TXD0
PCODE	PORTC open drain	28H (Prohibit RMW)					ODEPC3			ODEPC0
							W			W
							0			0
							0: CMOS 1: Open drain			0: CMOS 1: Open drain

I/O port control (4/4)

Symbol	Name	Address	7	6	5	4	3	2	1	0
PDFC	PORTD function	2AH (Prohibit RMW)	PD7F	PD6F		PD4F	PD3F	PD2F	PD1F	PD0F
			W			W				
			0			0				
			0: Port 1:MLDALM	0: Port 1: $\overline{\text{ALARM}}$ $\overline{\text{MLDALM}}$		0: Port 1: DOFFB	0: Port 1: DLEBCD	0: Port 1:D3BFR	0: Port 1: D2BLP	0: Port 1: D1BSCP
PECR	PORTE control	2DH (Prohibit RMW)	PE7C	PE6C	PE5C	PE4C	PE3C	PE2C	PE1C	PE0C
			W							
			0							
			0: Input 1: Output							
PEFC	PORTE function	2EH (Prohibit RMW)	PE7F	PE6F	PE5F	PE4F	PE3F	PE2F	PE1F	PE0F
			W							
			0							
			0: Port 1: LD7 to LD0 for LCD driver							
PFFC	PORTF function	32H (Prohibit RMW)	–	PF6F	PF5F	PF4F	PF3F	PF2F	PF1F	PF0F
			W							
			0	1	0					
			Always fixed to “0”.	0: Port 1: SDCLK	0: Port 1: SDCKE	0: Port 1: SDUDQM	0: Port 1: SDLDQM	0: Port 1: $\overline{\text{SDWE}}$	0: Port 1: $\overline{\text{SDCAS}}$	0: Port 1: $\overline{\text{SDRAS}}$

(3) Interrupt control (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE 0AD	Interrupt enable 0 and AD	90H	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTAD	Interrupt level			1: INT0	Interrupt level		
INTE12	Interrupt enable 2/1	91H	INT2				INT1			
			I2C	I2M2	I2M1	I2M0	I1C	I1M2	I1M1	I1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INT2	Interrupt level			1: INT1	Interrupt level		
INTE3 ALM4	Interrupt enable 3 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		
INTES2LCD	Interrupt enable SBI/LCD	9AH	INTLCD				INTSBI			
			ILCD2C	ILCDM2	ILCDM1	ILCDM0	ISBIC	ISBIM2	ISBIM1	ISBIM0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTLCD	Interrupt level			1: INTSBI	Interrupt level		
INTETC01	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
INTETC23	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
INTES3	Interrupt enable serial 3	A0H	INTTX2				INTRX2			
			ITX2C	ITX2M2	ITX2M1	ITX2M0	IRX2C	IRX2M2	IRX2M1	IRX2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTX2	Interrupt level			1: INTRX2	Interrupt level		
INTETB0	Interrupt enable TMRB0	A1H	INTTB01				INTTB00			
			ITB01C	ITB01M2	ITB01M1	ITB01M0	ITB00C	ITB00M2	ITB00M1	ITB00M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTB01	Interrupt level			1: INTTB00	Interrupt level		

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

(4) Chip select/wait control (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
B0CS	Block 0 CS/WAIT control register	C0H (Prohibit RMW)	B0E		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
									00: 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 set
					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: }
					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	

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 V9	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 V9	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	PSENV		WUPTM1	WUPTM0	HALTM1	HALTM0	SELDRV	DRVE
			R/W		R/W					
			0		1	0	1	1	0	0
			1: Disable 0: Power save mode enable		Warm-up time 00: Reserved 01: 2^8 input frequency 10: 2^{14} 11: 2^{16}		00: Reserved 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		<Drive> mode select 1: STOP 0: 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	TA3LCE	AHOLD	TA3MLE	−	EXTIN	DRVOSCH	DRVOSCL
			R	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	1	1
			Protection flag 0: OFF 1: ON	LCDC Source clock 0: 32 kHz 1: TA3OUT	Address hold 0: Disable 1: Enable	Melody/ alarm source clock 0: 32 kHz 1: TA3OUT	Write “0”.	1: fc is external clock	fc oscillator driver ability 1: Normal 0: Weak	fs oscillator driver ability 1: Normal 0: Weak
EMCCR1	EMC control register 1	E4H	Switching the protect ON/OFF by writing following 1ST-KEY, 2ND-KEY. Continuation writes in 1ST-KEY: EMCCR1 = 5AH,EMCCR2 = A5H. Continuation writes in 2ND-KEY: EMCCR1 = A5H, EMCCR2 = 5AH.							
EMCCR2	EMC control register 2	E5H								
EMCCR3	EMC control register 3	E6H		ENFROM	ENDROM	ENPROM		FFLAG	DFLAG	PFLAG
				R/W	R/W	R/W		R/W	R/W	R/W
				0	0	0		0	0	0
				CS1A area 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		

(6) DFM (Clock doubler)

Symbol	Name	Address	7	6	5	4	3	2	1	0
DFMCR0	DFM control register 0	E8H	ACT1	ACT0	DLUPFG	DLUPTM				
			R/W	R/W	R	R/W				
			0	0	0	0				
			DFM LUP f_{FPH} 00 STOP STOP f_{OSCH} 01 RUN RUN f_{OSCH} 10 RUN STOP f_{DFM} 11 RUN STOP f_{OSCH}		Lock-up flag 0: End LUP 1: Do not end LUP	Lockup time 0: $2^{12}/f_{OSCH}$ 1: $2^{10}/f_{OSCH}$				
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
			Write 0BH before starting lockup operation.							

(7) 8-bit timer

(7-1) TMRA01

Symbol	Name	Address	7	6	5	4	3	2	1	0
TA01 RUN	Timer RUN	100H	TA0RDE				I2TA01	TA01PRUN	TA1RUN	TA0RUN
			R/W				R/W	R/W	R/W	R/W
			0				0	0	0	0
			Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	8-bit timer run/stop control 0: Stop 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: Operate	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: 2^6 PWM 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 inversion

(8) UART/serial channel (1/3)

(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	SCLKS	IOC
			R	R/W		R (Cleared to 0 by reading)			R/W	
			Undefined	0	0	0	0	0	0	0
			Receiving data bit8	Parity 0: Odd 1: Even	1: Parity enable	1: Error			0:SCLK0↑ 1:SCLK0↓	1: Input SCLK0 pin
						Overrun	Parity	Framing		
SC0 MOD0	Serial channel 0 mode0	202H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Transmission data bit8	1: CTS enable	1: Receive enable	1: Wakeup enable	00: I/O Interface 01: UART 7 bits 10: UART 8 biits 11: UART 9 bits		00: TA0TRG 01: Baud rate generator 10: Internal clock f_{SYS} 11: External clock SCLK0	
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 divided enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Set the dividing value. 0 to F			
BR0 ADD	Serial channel 0 K setting register	204H					BR0K3	BR0K2	BR0K1	BR0K0
							R/W			
							0	0	0	0
							Baud rate 0 K 1 to F			
SC0 MOD1	Serial channel 0 mode1	205H	I2S0	FDPX0						
			R/W	R/W						
			0	0						
			IDLE2 0: Stop 1: Operate	I/O interface 1: Full duplex 0: Half duplex						

(8-2) IrDA

Symbol	Name	Address	7	6	5	4	3	2	1	0
SIRCR	IrDA control register	207H	PLSEL	RXSEL	TXEN	RXEN	SIRWD3	SIRWD2	SIRWD1	SIRWD0
			R/W	R/W	R/W	R/W	R/W			
			0	0	0	0	0	0	0	0
			Transmission pulse width 0: 3/16 1: 1/16	Receiving data 0: H pulse 1: L pulse	Transmission 0: Disable 1: Enable	Receiving 0: Disable 1: Enable	Set the effective SIRRxD pulse width Pulse width more than 2x × (Set value + 1) + 100 ns Possible: 1 to 14 Not possible: 0, 15			

UART/serial channel (2/3)

(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 bit8	Parity 0: Odd 1: Even	1: Parity enable	1: Error			0: SCLK1↑ 1: SCLK1↓	1: 1: Input SCLK1 pin
						Overrun	Parity	Framing		
SC1 MOD0	Serial channel 1 mode	20AH	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Transmission data bit8	1: CTS enable	1: Receive enable	1: Wakeup enable	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: 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 divided enable	00: φT0 01: φT2 10: φT8 11: φT32		Dividing value 0 to F			
BR1 ADD	Serial channel 1 K setting register	20CH					BR1K3	BR1K2	BR1K1	BR1K0
							R/W			
							0	0	0	0
							Set the frequency divisor K 1 to F			
SC1 MOD1	Serial channel 1 mode1	20DH	I2S1	FDPX1						
			R/W	R/W						
			0	0						
			IDLE2 0: Stop 1: Operate	I/O interface mode 1: Full duplex 0: Half duplex						

UART/serial channel (3/3)

(8-4) UART/SIO channel 2

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC2BUF	Serial channel 2 buffer	210H (Prohibit RMW)	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	RB0/TB0
			R (Receiving)/W (Transmission)							
			Undefined							
SC2CR	Serial channel 2 control	211H	RB8	EVEN	PE	OERR	PERR	FERR	–	–
			R	R/W		R (Cleared to 0 by reading)			R/W	
			Undefined	0	0	0	0	0	0	0
			Receiving data bit8	Parity 0: Odd 1: Even	1: Parity enable	1: Error			Always write “0”.	Always write “0”.
						Overrun	Parity	Framing		
SC2 MOD0	Serial channel 2 mode	212H	TB8	–	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Transmission data bit8	Always write “0”.	1: Receive enable	1: Wakeup enable	00: Reserved 01: UART 7 bits 10: UART 8 bits 11: UART 9 bits		00: TA0TRG 01: Baud rate generater 10: Internal clock f_{SYS} 11: External clock SCLK1	
BR2CR	Baud rate control	213H	–	BR2ADDE	BR2CK1	BR2CK0	BR2S3	BR2S2	BR2S1	BR2S0
			R/W							
			0	0	0	0	0	0	0	0
			Always write “0”.	1: (16 − K)/16 divided enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Dividing value 0 to F			
BR2 ADD	Serial channel 2 K setting register	214H					BR1K3	BR1K2	BR1K1	BR1K0
							R/W			
							0	0	0	0
							Set the frequency divisor K 1 to F			
SC2 MOD1	Serial channel 2 mode1	215H	I2S2	FDPX2						
			R/W	R/W						
			0	0						
			IDLE2 0: Stop 1: Operate	I/O interface mode 1: Full duplex 0: Half duplex						

(9) I^{2}C bus/serial interface (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SBI0CR1	Serial bus interface control register 1	240H (I^{2}C bus mode) (Prohibit RMW)	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0 /SWRMON
			W			R/W		W	W	R/W
			0	0	0	0		0	0	0/1
			Number of transfer bits 000: 8 001: 1 010: 2 011: 3 100: 4 101: 5 110: 6 111: 7			Acknowledge mode 0: Disable 1: Enable		Setting for the devisor value n 000: 5 001: 6 010: 7 011: 8 100: 9 101: 10 110: 11 111: (Reserved)		
		240H (SIO mode) (Prohibit RMW)	SIOS	SIOINH	SIOM1	SIOM0		SCK2	SCK1	SCK0
			W	W	W	W		W	W	W
			0	0	0	0		0	0	0
			Transfer 0: Stop 1: Start	Transfer 0: Continue 1: Abort	Transfer mode 00: 8-bit transmit mode 10: 8-bit transmit/ receive mode 11: 8-bit received mode			Setting for the divisor value n 000: 4 001: 5 010: 6 011: 7 100: 8 101: 9 110: 10 111: SCK pin		
SBI0 DBR	SBI buffer register	241H (Prohibit RMW)	RB7/TB7	RB6/TB6	RB5/TB5	RB4/TB4	RB3/TB3	RB2/TB2	RB1/TB1	PB0/TB0
			R (Receiving)/W (Transmission)							
			Undefined							
I2C0AR	I2CBUS address register	242H (Prohibit RMW)	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
			W	W	W	W	W	W	W	W
			0	0	0	0	0	0	0	0
			Setting slave address							Address recognition 0: Enable 1: Disable
When read SBI0SR	Serial bus interface status register	243H (I^{2}C bus mode) (Prohibit RMW)	MST	TRX	BB	PIN	AL/SBIM1	AAS/SBIM0	AD0/ SWRST	LRB/ SWRST0
			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	1	0	0	0	0
			0: Slave 1: Master	0: Receiver 1: Transmit	Bus status monitor 0: Free 1: Busy	INTSBI request monitor 0: Request 1: Cancel	Arbitration lost detection monitor 1: Detect	Slave address match detection monitor 1: Detect	GENERAL CALL detection monitor 1: Detect	Lost receive bit monitor 0: 0 1: 1
When write SBI0CR 2	Serial bus interface control register 2				Start/stop condition generation 0: Start condition 1: Stop condition		Serial bus interface operating mode selection 00: Port mode 01: SIO mode 10: I^{2}C bus mode 11: (Reserved)		Software reset generate write “10” and “01”, then an internal reset signal is generated.	
When read SBI0SR	Serial bus interface status register	243H (SIO mode) (Prohibit RMW)					SIOF/SBIM1	SEF/SBIM2	–	–
							R	R	W	W
							0	0	0	0
							Transfer status monitor 0: Stopped 1: Terminated in process	Shift operation status monitor 0: Stopped 1: Terminated in process		
When write SBI0CR 2	Serial bus interface control register 2						Serial bus interface operating mode selection 00: Port mode 01: SIO mode 10: I^{2}C bus mode 11: (Reserved)		Always write “0”.	Always write “0”.

I^2C bus/serial interface (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SBI0 BR0	Serial bus interface baud rate register 0	244H (Prohibit RMW)	–	I2SBI0						
			W	R/W						
			0	0						
			Always write “0”.	IDLE2 0: Abort 1: Operate						
SBI0 BR1	Serial bus interface baud rate register 1	245H (Prohibit RMW)	P4EN	–						
			W	W						
			0	0						
			Clock control 0: Abort 1: Operate	Always write “0”.						

(10) AD converter

Symbol	Name	Address	7	6	5	4	3	2	1	0
ADMOD0	AD mode register 0	2B0H	EOCF	ADBF	–	–	ITM0	REPEAT	SCAN	ADS
			R		R/W	R/W	R/W	R/W	R/W	R/W
			0	0	0	0	0	0	0	0
			1: End	1: Busy	Always write “0”.	Always write “0”.	Interrupt in repeat mode	1: Repeat	1: Scan	1: Start
ADMOD1	AD mode register 1	2B1H	VREFON	I2AD			ADTRGE	ADCH2	ADCH1	ADCH0
			R/W	R/W			R/W	R/W		
			0	0			0	0	0	0
			1: VREF on	IDLE2 0: Abort 1: Operate			1: Enable for external start	Input channel 000: AN0 AN0 001: AN1 AN0 → AN1 010: AN2 AN0 → AN1 → AN2 011: AN3 AN0 → AN1 → AN2 → AN3 100: AN4 AN4 101: AN5 AN4 → AN5 110: AN6 AN4 → AN5 → AN6 111: AN7 AN4 → AN5 → AN6 → AN7		
AD REG04L	AD result register 0/4 low	2A0H	ADR01	ADR00						ADR0RF
			R							R
			Undefined							0
AD REG04H	AD result register 0/4 high	2A1H	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
			R							
			Undefined							
AD REG15L	AD result register 1/5 low	2A2H	ADR11	ADR10						ADR1RF
			R							R
			Undefined							0
AD REG15H	AD result register 1/5 high	2A3H	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
			R							
			Undefined							
AD REG26L	AD result register 2/6 low	2A4H	ADR21	ADR20						ADR2RF
			R							R
			Undefined							0
AD REG26H	AD result register 2/6 high	2A5H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
			R							
			Undefined							
AD REG37L	AD result register 3/7 low	2A6H	ADR31	ADR30						ADR3RF
			R							R
			Undefined							0
AD REG37H	AD result register 3/7 high	2A7H	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
			R							
			Undefined							

(11) Watchdog timer

Symbol	Name	Address	7	6	5	4	3	2	1	0
WDMOD	WDT mode 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: Abort 1: Operate	1: RESET connect internally WDT out to reset pin	Always write “0”.
WDCR	WD control	301H (Prohibit RMW)	–							
			W							
			–							
			B1H: WDT disable 4EH: WDT clear							

(12) RTC (Real time clock)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SECR	Second 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	1min
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	0	20 day	10 day	8 day	4 day	2 day	1 day
MONTHR	Month register	325H				MO4	MO3	MO2	MO1	MO0
						R/W				
						Undefined				
		PAGE0	“0” is read.			10 month	8 month	4 month	2 month	1 month
		PAGE1	“0” is read.							0: Indicator for 12 hours 1: Indicator for 24 hours
YEARR	Year register	326H	YE7	YE6	YE5	YE4	YE3	YE2	YE1	YE0
			R/W							
			Undefined							
		PAGE0	80 year	40 year	20 year	10 year	8 year	4 year	2 year	1 year
		PAGE1	“0” is read.						Leap year setting	
PAGER	Page register	327H (Prohibit RMW)	INTENA			ADJUST	ENATMR	ENAALM		PAGE
			R/W			W	R/W			R/W
			0			Undefined	Undefined			Undefined
			INTRTC ENABLE	“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”.			

(13) 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 set							
MELALMC	Melody/ alarm control register	331H	FC1	FC0	ALMINV	–	–	–	–	MELALM
			R/W		R/W	R/W	R/W	R/W	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 frequency invert 1: Invert	Always write “0”.				Output frequency 0: Alarm 1: Melody
MELFL	Melody frequency register-L	332H	ML7	ML6	ML5	ML4	ML3	ML2	ML1	ML0
			R/W							
			0							
			Melody frequency set (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 set (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 alarm interrupt enable				

(14) 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		
			0: Disable 1: Enable					LOCAL0 area BANK set		
LOCAL1	LOCAL1 control register	351H	L1E					L1EA23	L1EA22	L1EA21
			R/W					R/W		
			0					0		
			0: Disable 1: Enable					LOCAL1 area BANK set		
LOCAL2	LOCAL2 control register	352H	L2E					L2EA23	L2EA22	L2EA21
			R/W					R/W		
			0					0		
			0: Disable 1: Enable					LOCAL2 area BANK set		
LOCAL3	LOCAL3 control register	353H	L3E			L3EA26	L3EA25	L3EA24	L3EA23	L3EA22
			R/W			R/W				
			0			0				
			0: Disable 1: Enable			LOCAL3 area BANK set				

(15) LCD controller (1/5)

Symbol	Name	Address	7	6	5	4	3	2	1	0
LCD MODE	LCD mode register	04B0H	BAE	AAE	SCPW1	SCPW0	–	BULK	RAMTYPE	MODE
			R/W							
			0	0	1	0	0	0	0	0
			Used by B AREA 0: Disable 1: Enable	Used by A AREA 0: Disable 1: Enable	SCP width 00: Base mode 01: 2 clocks 10: 4 clocks 11: 8 clocks		Always write “0”.	SDRAM bank selection 0: 64 Mbit 1: 128 Mbit	Display RAM Selection 0: SRAM 1: SDRAM	Mode selection 0: RAM 1: SR
LCD DVM	Divide Frame register	04B1	FMN7	FMN6	FMN5	FMN4	FMN3	FMN2	FMN1	FMN0
			R/W							
			0	0	0	0	0	0	0	0
			Setting Frame invert adjust function bit7 to bit0							
LCD SIZE	LCD size register	04B2H	COM3	COM2	COM1	COM0	SEG3	SEG2	SEG1	SEG0
			R/W							
			0	0	0	0	0	0	0	0
			Setting the LCD common number for SR type 0000: 128 0101: 400 0001: 160 0110: 480 0010: 200 0011: 240 0100: 320 Other: Reserve				Setting the LCD segment number for SR type 0000: 128 0101: 480 0001: 160 0110: 560 0010: 240 0111: 640 0011: 320 0100: 400 Other: Reserve			
LCDCTL	LCD control register	04B3H	LCDON	ALL0	FRMON	–	FP9	MMULCD	FP8	START
			R/W							
			0	0	0	0	0	0	0	0
			$\overline{\text{DOFF}}$ port 0: OFF 1: ON	Transfer data of exclusive bus for LCD 0:Normal 1:All display data 0	Divided FR mode 0: Disable 1: Enable	Always write “0”.	Setting bit9 for f_{FP} [9:0]	Specify address of LCD driver with built-in RAM 0: OFF 1: ON	Setting bit8 for f_{FP} [9:0]	Start control in SR type 0: Stop 1: Start
LCD FFP	LCD FRAME FREQUENCY register	04B4H	FP7	FP6	FP5	FP4	FP3	FP2	FP1	FP0
			R/W							
			0							
			f_{FP} set value bit7 to bit0							
LCDGL	LCD gray level register	04B5H							GRAY1	GRAY0
									R/W	R/W
									0	0
									00: Monochrome 01: 4 levels 10: 8 levels 11: 16 levels	

LCD controller (2/5)

Symbol	Name	Address	7	6	5	4	3	2	1	0
LCDCM	LCD cursor mode register	04B6H	CDE	CCS					CBE1	CBE0
			R/W						R/W	
			0	0					0	0
			Cursor 0: OFF 1: ON	Cursor color 0: White 1: Black					Cursor blink interval 00: Don't blink 01: 2 Hz 10: 1 Hz 11: 0.5 Hz	
LCDCW	LCD cursor width register	04B7H				CW4	CW3	CW2	CW1	CW0
						R/W				
						0	0	0	0	0
						Cursor width (X size) 00000: 1 dot (Min) 11111: 32 dots (Max)				
LCDCH	LCD cursor height register	04B8H				CW4	CW3	CW2	CW1	CW0
						R/W				
						0	0	0	0	0
						Cursor height (Y size) 00000: 1 dot (Min) 11111: 32 dots (Max)				
LCDCP	LCD cursor APB register	04B9H					APB 3	APB 2	APB 1	APB 0
							R/W			
							0	0	0	0
							Setting bit3 to bit0 for cursor absolute position			
LCDCPL	LCD cursor AP register-L	04BAH	CAP 7	CAP 6	CAP 5	CAP 4	CAP 3	CAP 2	CAP 1	CAP 0
			R/W							
			0	0	0	0	0	0	0	0
			Setting bit7 to bit0 for cursor absolute position							
LCDCPM	LCD cursor AP register-M	04BBH	CAP 15	CAP 14	CAP 13	CAP 12	CAP 11	CAP 10	CAP 9	CAP 8
			R/W							
			0	0	0	0	0	0	0	0
			Setting bit15 to bit8 for cursor absolute position							
LCDCPH	LCD cursor AP register-H	04BCH	CAP 23	CAP 22	CAP 21	CAP 20	CAP 19	CAP 18	CAP 17	CAP 16
			R/W							
			0	1	0	0	0	0	0	0
			Setting bit23 to bit16 for cursor absolute position							

LCD controller (3/5)

Symbol	Name	Address	7	6	5	4	3	2	1	0
LSARAM	A area start address register-M	04C0H	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8
			R/W							
			0	0	0	0	0	0	0	0
			Setting start address A15 to A8 for the source data memory in A area.							
LSARAH	A area start address register-H	04C1H	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16
			R/W							
			0	1	0	0	0	0	0	0
			Setting start address A23 to A16 for the source data memory in A area.							
LEARAM	A area end address register-M	04C2H	EA15	EA14	EA13	EA12	EA11	EA10	EA9	EA8
			R/W							
			0	0	0	0	0	0	0	0
			Setting end address A15 to A8 for the source data memory in A area.							
LEARAH	A area end address register-H	04C3H	EA23	EA22	EA21	EA20	EA19	EA18	EA17	EA16
			R/W							
			0	1	0	0	0	0	0	0
			Setting end address A23 to A16 for the source data memory in A area.							
LSARBM	B area start address register-M	04C4H	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8
			R/W							
			0	0	0	0	0	0	0	0
			Setting start address A15 to A8 for the source data memory in B area.							
LSARBH	B area start address register-H	04C5H	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16
			R/W							
			0	1	0	0	0	0	0	0
			Setting start address A23 to A16 for the source data memory in B area.							
LEARBM	B area end address register-M	04C6H	EA15	EA14	EA13	EA12	EA11	EA10	EA9	EA8
			R/W							
			0	0	0	0	0	0	0	0
			Setting end address A15 to A8 for the source data memory in B area.							
LEARBH	B area end address register-H	04C7H	EA23	EA22	EA21	EA20	EA19	EA18	EA17	EA16
			R/W							
			0	1	0	0	0	0	0	0
			Setting end address A23 to A16 for the source data memory in B area.							
LSARCL	C area start address register-L	04C8H	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0
			R/W							
			0	0	0	0	0	0	0	0
			Setting start address A7 to A0 for the source data memory in C area.							
LSARCM	C area start address register-M	04C9H	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8
			R/W							
			0	0	0	0	0	0	0	0
			Setting start address A15 to A8 for the source data memory in C area.							
LSARCH	C area start address register-H	04CAH	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16
			R/W							
			0	1	0	0	0	0	0	0
			Setting start address A23 to A16 for the source data memory in C area.							

LCD controller (4/5)

Symbol	Name	Address	7	6	5	4	3	2	1	0
LG0L	LCD gray level data setting register-L	04D0H								
			R/W							
			0	0	0	0	0	0	0	0
LG0H	LCD gray level data setting register-H	04D1H								
			R/W							
			0	0	0	0	0	0	0	0
LG1L	LCD gray level data setting register-L	04D2H								
			R/W							
			0	0	0	0	0	0	0	0
LG1H	LCD gray level data setting register-H	04D3H								
			R/W							
			1	0	0	0	0	0	0	0
LG2L	LCD gray level data setting register-L	04D4H								
			R/W							
			1	0	0	0	0	0	0	0
LG2H	LCD gray level data setting register-H	04D5H								
			R/W							
			1	0	0	0	0	0	0	0
LG3L	LCD gray level data setting register-L	04D6H								
			R/W							
			1	0	0	0	0	0	0	0
LG3H	LCD gray level data setting register-H	04D7H								
			R/W							
			1	0	0	0	1	0	0	0
LG4L	LCD gray level data setting register-L	04D8H								
			R/W							
			1	0	0	0	1	0	0	0
LG4H	LCD gray level data setting register-H	04D9H								
			R/W							
			1	0	0	0	1	0	0	0
LG5L	LCD gray level data setting register-L	04DAH								
			R/W							
			1	0	0	0	1	0	1	0
LG5H	LCD gray level data setting register-H	04DBH								
			R/W							
			1	0	0	0	1	0	0	0
LG6L	LCD gray level data setting register-L	04DCH								
			R/W							
			1	0	0	0	1	0	1	0
LG6H	LCD gray level data setting register-H	04DDH								
			R/W							
			1	0	0	0	1	0	1	0
LG7L	LCD gray level data setting register-L	04DEH								
			R/W							
			1	0	1	0	1	0	1	0
LG7H	LCD gray level data setting register-H	04DFH								
			R/W							
			1	0	0	0	1	0	1	0

LCD controller (5/5)

Symbol	Name	Address	7	6	5	4	3	2	1	0
LG8L	LCD gray level data setting register-L	04E0H								
			R/W							
			1	0	1	0	1	0	1	0
LG8H	LCD gray level data setting register-H	04E1H								
			R/W							
			1	0	1	0	1	0	1	0
LG9L	LCD gray level data setting register-L	04E2H								
			R/W							
			0	1	0	1	0	1	0	1
LG9H	LCD gray level data setting register-H	04E3H								
			R/W							
			1	1	0	1	0	1	0	1
LGAL	LCD gray level data setting register-L	04E4H								
			R/W							
			1	1	0	1	0	1	0	1
LGAH	LCD gray level data setting register-H	04E5H								
			R/W							
			1	1	0	1	0	1	0	1
LGBL	LCD gray level data setting register-L	04E6H								
			R/W							
			1	1	0	1	0	1	0	1
LGBH	LCD gray level data setting register-H	04E7H								
			R/W							
			1	1	0	1	1	1	0	1
LGCL	LCD gray level data setting register-L	04E8H								
			R/W							
			1	1	0	1	1	1	0	1
LGCH	LCD gray level data setting register-H	04E9H								
			R/W							
			1	1	0	1	1	1	0	1
LGDL	LCD gray level data setting register-L	04EAH								
			R/W							
			1	1	0	1	1	1	0	1
LGDH	LCD gray level data setting register-H	04EBH								
			R/W							
			1	1	1	1	1	1	0	1
LGEL	LCD gray level data setting register-L	04ECH								
			R/W							
			1	1	0	1	1	1	0	1
LGEH	LCD gray level data setting register-H	04EDH								
			R/W							
			1	1	0	1	1	1	0	1
LGFL	LCD gray level data setting register-L	04EEH								
			R/W							
			1	1	1	1	1	1	1	1
LGFH	LCD gray level data setting register-H	04EFH								
			R/W							
			1	1	1	1	1	1	1	1

(16) SDRAM controller

Symbol	Name	Address	7	6	5	4	3	2	1	0
SDACR	SDRAM address control	04F0H	SDINI	SWRC	–	–	SMUXE	SMUXW1	SMUXW0	SMAC
			R/W	R/W	R/W	R/W	R/W			
			0	0	1	0	0	0	0	0
			Auto initialize 0: Disable 1: Enable	Write recovery 0: 1 clock 1: 2 clocks	Always fixed to “10”.		Address mux 0: Disable 1: Enable	SDRAM select 00:16 Mbits 10: 128 Mbits 01: 64 Mbits 11: Reserved		Access cycle 0: Disable 1: Enable
SDRCR	SDRAM refresh control	04F1H	SFRC	SRS2	SRS1	SRS0	SASFRC			SRC
			R/W							R/W
			0	0	0	0	0			0
			Self refresh 0: Disable 1: Enable	Auto refresh interval 000: 78 states 100: 195 states 001: 97 states 101: 210 states 010: 124 states 110: 249 states 011: 156 states 111: 312 states			Auto self refresh 0: Disable 1: Enable			Auto refresh 0: Disable 1: Enable

(17) 16-bit timer

Symbol	Name	Address	7	6	5	4	3	2	1	0
TB0RUN	Timer control	180H	TB0RDE	–			I2TB0	TB0PRUN		TB0RUN
			R/W	R/W			R/W	R/W		R/W
			0	0			0	0		0
			Double buffer 0: Disable 1: Enable	Always write “0”.			IDLE2 0: Stop 1: Operate	16-bit timer run/stop control 0: Stop and clear 1: Run (Count up)		
TB0MOD	16-bit timer source CLK and mode	182H (Prohibit RMW)	–	–	TB0CPOI	–	–	TB0CLE	TB0CLK1	TB0CLK0
			R/W		W*	R/W				
			0	0	1	0	0	0	0	0
			Always fixed to “0”.		0: Soft capture 1: Undefined	Always fixed to “0”.		1: UC0 clear enable	Source clock 00: Reserved 01: φT1 10: φT4 11: φT16	
TB0FFCR	16-bit timer flip-flop control	183H (Prohibit RMW)	–	–	–	TB0C0T1	TB0E1T1	TB0E0T1	TB0FF0C1	TB0FF0C0
			W		R/W				W*	
			1	1	0	0	0	0	0	0
			Always fixed to “11”.		Always fixed “0”.	TB0FF0 invert trigger 0: Trigger disable 1: Trigger enable			00: Invert TB0FF0 01: Set 10: Clear 11: Don’t care Always read as “11”.	
						Invert when the UC value is loaded in to TB0CP0.	Invert when the UC value matches the value in TB0RG1.	Invert when the UC value matches the value in TB0RG0.		
TB0RG0L	16-bit timer register 0-L	188H (Prohibit RMW)	–							
			W							
			Undefined							
TB0RG0H	16-bit timer register 0-H	189H (Prohibit RMW)	–							
			W							
			Undefined							
TB0RG1L	16-bit timer register 1-L	18AH (Prohibit RMW)	–							
			W							
			Undefined							
TB0RG1H	16-bit timer register 1-H	18BH (Prohibit RMW)	–							
			W							
			Undefined							
TB0CP0L	Capture register 0-L	18CH	–							
			R							
			Undefined							
TB0CP0H	Capture register 0-H	18DH	–							
			R							
			Undefined							
TB0CP1L	Capture register 1-L	18EH	–							
			R							
			Undefined							
TB0CP1H	Capture register 1-H	18FH	–							
			R							
			Undefined							

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 memo
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 TL

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, f_{FPH}, f_{SYS} and one state

The clock frequency input on X1 and X2 is called f_{OSCH}. The clock selecte
DFMCR0<ACT1:0> is called fc.

The clock selected by SYSCR1<SYSCK> is called f_{FPH}. The clock frequency gi
divided by 2 is called f_{SYS}.

One cycle of f_{SYS} is referred to as one state.

(2) Points of note

a. AM0 and AM1 pins

This pin is connected to the V_{CC} pin. Do not alter the level wh active.

b. EMU0 and EMU1

Open pins.

c. Reserved address areas

The TMP91C820A does not have any reserved areas.

d. Warm-up counter

The warm-up counter operates when STOP mode is released, even if the sy an external oscillator. As a result a time equivalent to the warm-up time el input of the release request and output of the system c

e. Programmable pull-up resistance

The programmable pull-up resistor can be turned ON/O are set for use as input ports. When the ports are set fo turned ON/OFF by a program.

The data registers (e.g., P5) are used to turn the pu Consequently read-modify-write instructions are pro

f. Watchdog timer

The watchdog timer starts operation immediately af watchdog timer is not to be used, disable it.

g. AD converter

The string resistor between the VREFH and VREFL pins c reduce power consumption. When STOP mode is used, d program before the HALT instruction is executed.

h. CPU (Micro DMA)

Only the "LDC cr, r" and "LDC r, cr" instructions ca 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 wh

j. POP SR instruction

Please execute the POP SR instruction during DI cond

k. Releasing the HALT mode by requesting an interruptio

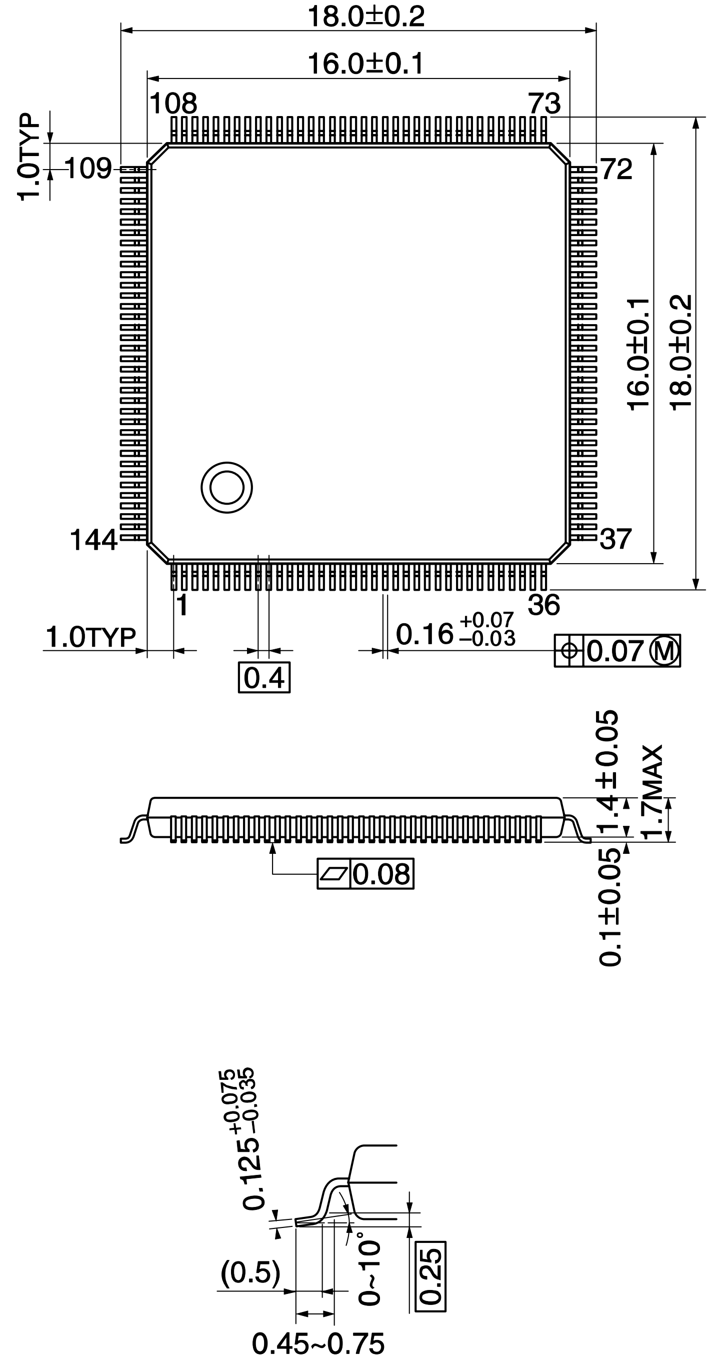
Usually, interrupts can release all halts status. Ho $\overline{\text{NMI}}$, INT0 to INT3, INTKEY, INTRTC, INTALM0 to INTALM4) which can re not be able to do so if they are input during the period (for about 5 clocks of f_{FPH}) with IDLE1 or STOP mode (IDLE2 is not app (In this case, an interrupt request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

If another interrupt is generated after it has shift status can be released without difficulty. The priority of this interrupt i that of the interrupt kept on hold internally, and the interrupt with high handled first followed by the other interrupt.

7. Package Dimensions

P-LQFP144-1616-0.40C

Unit: mm


18.0±0.2
16.0±0.1
108
73
1.0TYP
109
72
16.0±0.1
18.0±0.2
144
37
1
36
1.0TYP
0.16 +0.07 −0.03
0.4
⌖ 0.07 Ⓜ
1.4±0.05
1.7MAX
0.1±0.05
▱ 0.08
0.125 +0.075 −0.035
(0.5)
0~10°
0.25
0.45~0.75