TOSHIBA

TOSHIBA Original CMOS 16-Bit Microcontroller

TLCS-900/H1 Series

TMP92C820FG

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 = (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 3 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 32-bit Microcontrollers

TMP92C820FG/JTMP92C820

1. Outline and Device Characteristics

TMP92C820 is high-speed advanced 32-bit microcontroller developed for controlling equipment which processes mass data.

TMP92C820 is a microcontroller which has a high-performance CPU (900/H1 CPU) and various built-in I/Os. TMP92C820FG is housed in a 144-pin flat package. JTMP92C820 is a 144-pad chip product.

Device characteristics are as follows:

(1) CPU: 32-bit CPU (900/H1 CPU)

- Compatible with TLCS-900, 900/L, 900/L1, 900/H's instruction code
- 16 Mbytes of linear address space
- General-purpose register and register banks
- Micro DMA: 8 channels (250 ns/4 bytes at $f_{SYS} = 20$ MHz, best case)

(2) Minimum instruction execution time: 50 ns (at $_{SYS} = 20$ MHz)

(3) Internal memory
- Internal RAM: 8 Kbytes (can use for code section)
- Internal ROM: None

(4) External memory expansion
- Expandable up to 136 Mbytes (Shared with program/data area)
- Can simultaneously support 8-/16-/32-bit width external data bus .... Dynamic data bus sizing
- Separate bus system

(5) Memory controller
- Chip select outputs: 4 channels

(6) 8-bit timers: 4 channels

(7) 16-bit timer/event counter: 1 channel

(8) General-purpose serial interface: 3 channels
- UART/synchronous mode
- IrDA

(9) Serial bus interface: 1 channel
- I^2C bus mode
- Clock synchronous select mode

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

(11) SDRAM controller
- Supported 16-M, 64-M and 128-Mbit SDRAM with 16-/32-bit data bus

(12) Timer for real-time clock (RTC)
- Based on TC8521A
- Separate the power supply

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

(14) 10-bit AD converter: 5 channels

(15) Watchdog timer

(16) 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

(17) MMU
- Expandable up to 136 Mbytes (4 local areas/8 bank methods)

(18) Interrupts: 45 interrupts
- 9 CPU interrupts: Software interrupt instruction and illegal instruction
- 31 internal interrupts: Seven selectable priority levels
- 5 external interrupts: Seven selectable priority levels (4-edge selectable)

(19) Input/output ports: 83 pins (Except Data bus (16bit), Address bus (24bit) and RD pin)

(20) Standby function

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

(21) Triple-clock controller

- Clock gear function: Select a high-frequency clock fc to fc/16
- RTC (fs = 32.768 kHz)

(22) Operating voltage

- DVCC = 3.0 to 3.6 V
- RTCVCC = 2.0 to 3.6 V

(23) Package

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

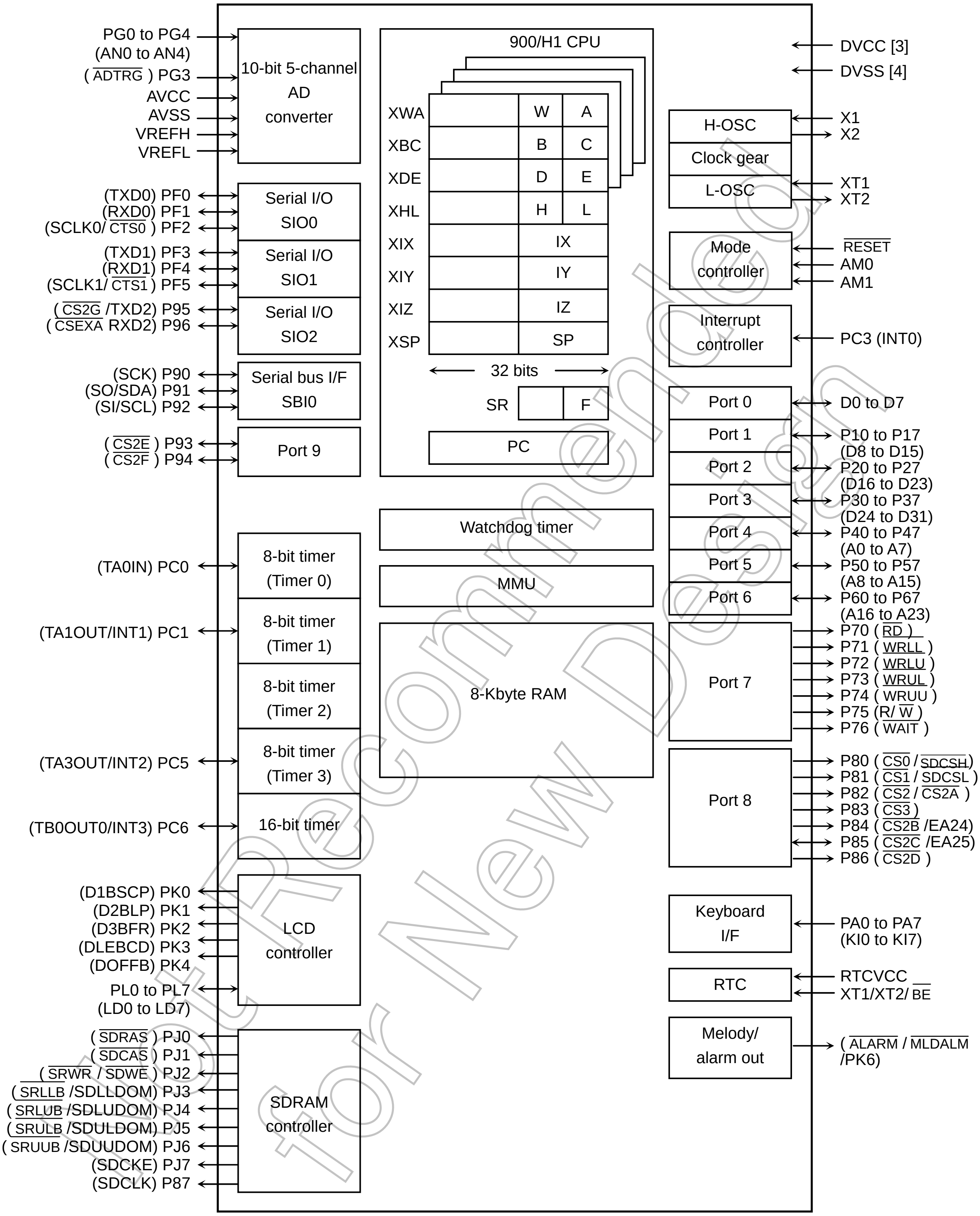


Figure 1.1 TMP92C820 Block Diagram

2. Pin Assignment and Functions

The assignment of input/output pins for the TMP92C820, their names and functions are as follows:

2.1 Pin Assignment

Figure 2.1.1 shows the pin assignment of the TMP92C820FG.

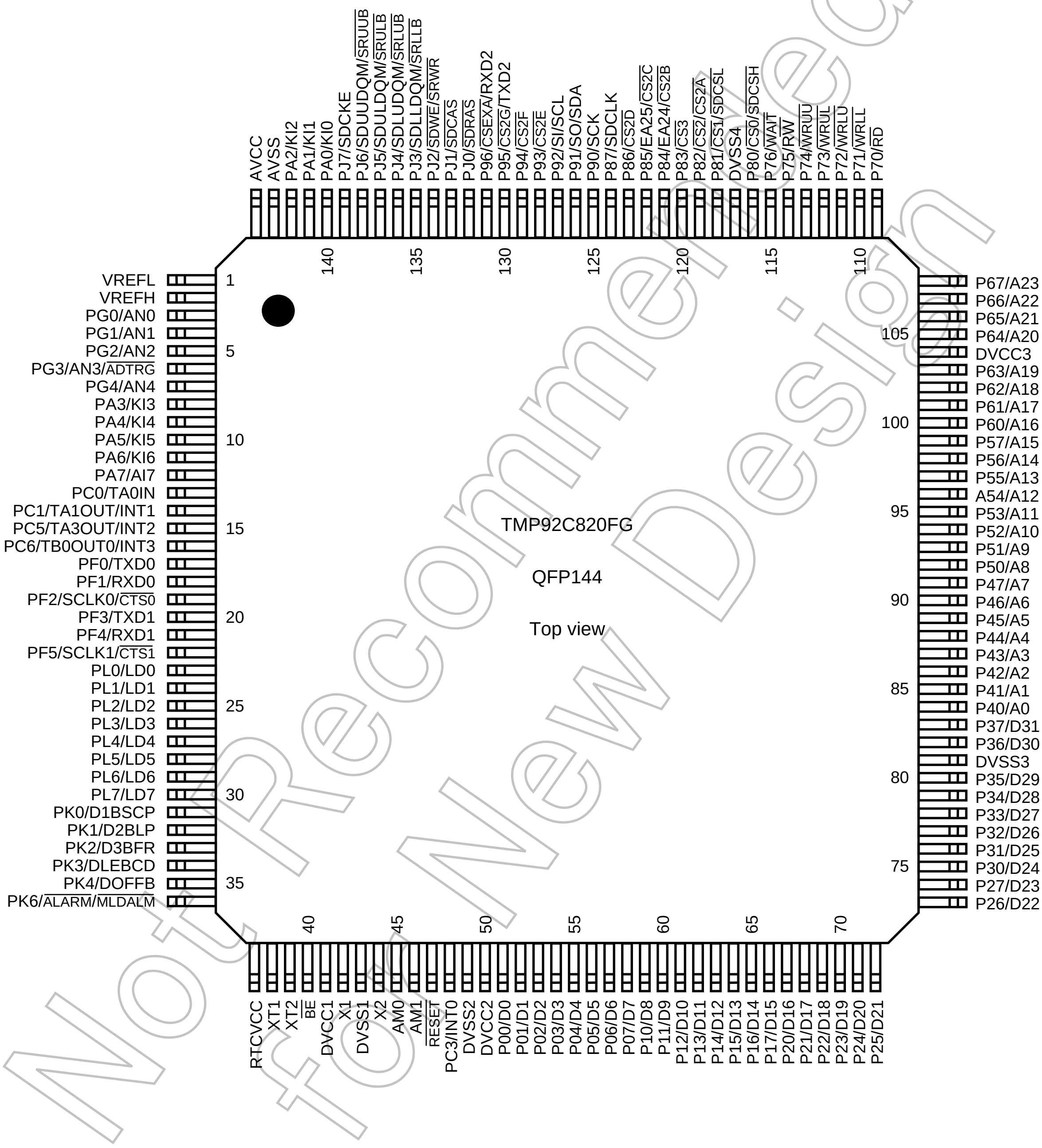


Figure 2.1.1 Pin Assignment Diagram (144-pin QFP)

2.2 PAD Layout

Table 2.2.1 PAD Layout (144-pin chip)

(Chip size 4.68 mm × 4.68 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	VREFL	–2213	1945	49	DVSS2	–440	–2213	97	P55	2211	685
2	VREFH	–2213	1820	50	DVCC2	–340	–2213	98	P56	2211	789
3	PG0	–2213	1694	51	P00	–240	–2213	99	P57	2211	894
4	PG1	–2213	1568	52	P01	–140	–2213	100	P60	2211	1000
5	PG2	–2213	1460	53	P02	–40	–2213	101	P61	2211	1107
6	PG3	–2213	1353	54	P03	59	–2213	102	P62	2211	1213
7	PG4	–2213	1249	55	P04	160	–2213	103	P63	2211	1321
8	PA3	–2213	1050	56	P05	260	–2213	104	DVCC3	2211	1430
9	PA4	–2213	946	57	P06	360	–2213	105	P64	2211	1546
10	PA5	–2213	842	58	P07	460	–2213	106	P65	2211	1672
11	PA6	–2213	739	59	P10	561	–2213	107	P66	2211	1798
12	PA7	–2213	635	60	P11	661	–2213	108	P67	2211	1924
13	PC0	–2213	531	61	P12	761	–2213	109	P70	1925	2211
14	PC1	–2213	427	62	P13	861	–2213	110	P71	1800	2211
15	PC5	–2213	326	63	P14	961	–2213	111	P72	1675	2211
16	PC6	–2213	224	64	P15	1062	–2213	112	P73	1558	2211
17	PF0	–2213	123	65	P16	1162	–2213	113	P74	1448	2211
18	PF1	–2213	23	66	P17	1263	–2213	114	P75	1346	2211
19	PF2	–2213	–77	67	P20	1363	–2213	115	P76	1243	2211
20	PF3	–2213	–179	68	P21	1474	–2213	116	P80	1141	2211
21	PF4	–2213	–284	69	P22	1589	–2213	117	DVSS4	1038	2211
22	PF5	–2213	–388	70	P23	1702	–2213	118	P81	937	2211
23	PL0	–2213	–493	71	P24	1814	–2213	119	P82	835	2211
24	PL1	–2213	–598	72	P25	1926	–2213	120	P83	734	2211
25	PL2	–2213	–704	73	P26	2211	–1924	121	P84	633	2211
26	PL3	–2213	–809	74	P27	2211	–1799	122	P85	532	2211
27	PL4	–2213	–914	75	P30	2211	–1674	123	P86	431	2211
28	PL5	–2213	–1024	76	P31	2211	–1548	124	P87	330	2211
29	PL6	–2213	–1132	77	P32	2211	–1426	125	P90	229	2211
30	PL7	–2213	–1243	78	P33	2211	–1311	126	P91	128	2211
31	PK0	–2213	–1354	79	P34	2211	–1199	127	P92	28	2211
32	PK1	–2213	–1464	80	P35	2211	–1087	128	P93	–72	2211
33	PK2	–2213	–1576	81	DVSS3	2211	–975	129	P94	–173	2211
34	PK3	–2213	–1701	82	P36	2211	–864	130	P95	–274	2211
35	PK4	–2213	–1826	83	P37	2211	–757	131	P96	–375	2211
36	PK6	–2213	–1953	84	P40	2211	–648	132	PJ0	–477	2211
37	RTCVCC	–1962	–2213	85	P41	2211	–541	133	PJ1	–580	2211
38	XT1	–1851	–2213	86	P42	2211	–435	134	PJ2	–684	2211
39	XT2	–1574	–2213	87	P43	2211	–332	135	PJ3	–788	2211
40	$\overline{\text{BE}}$	–1466	–2213	88	P44	2211	–228	136	PJ4	–892	2211
41	DVCC1	–1360	–2213	89	P45	2211	–128	137	PJ5	–996	2211
42	X1	–1257	–2213	90	P46	2211	–28	138	PJ6	–1101	2211
43	DVSS1	–1057	–2213	91	P47	2211	71	139	PJ7	–1208	2211
44	X2	–957	–2213	92	P50	2211	171	140	PA0	–1319	2211
45	AM0	–840	–2213	93	P51	2211	272	141	PA1	–1430	2211
46	AM1	–740	–2213	94	P52	2211	374	142	PA2	–1555	2211
47	$\overline{\text{RESET}}$	–640	–2213	95	P53	2211	477	143	AVSS	–1828	2211
48	PC3	–540	–2213	96	P54	2211	581	144	AVCC	–1955	2211

2.3 Pin Names and Functions

The following table shows the names and functions of the input/output pins.

Table 2.3.1 Pin Names and Functions (1/3)

Pin Names	Number of Pins	I/O	Functions
D0 to D7	8	I/O	Data: Data bus 0 to 7.
P10 to P17 D8 to D15	8	I/O I/O	Port 1: I/O port. Input or output specifiable in units of bits. Data: Data bus 8 to 15.
P20 to P27 D16 to D23	8	I/O I/O	Port 2: I/O port. Input or output specifiable in units of bits. Data: Data bus 16 to 23.
P30 to P37 D24 to D31	8	I/O I/O	Port 3: I/O port. Input or output specifiable in units of bits. Data: Data bus 24 to 31.
P40 to P47 A0 to A7	8	I/O Output	Port 4: I/O port. Input or output specifiable in units of bits. Address: Address bus 0 to 7.
P50 to P57 A8 to A15	8	I/O Output	Port 5: I/O port. Input or output specifiable in units of bits. Address: Address bus 8 to 15.
P60 to P67 A16 to A23	8	I/O Output	Port 6: I/O port. Input or output specifiable in units of bits. Address: Address bus 16 to 23.
P70 $\overline{RD}$	1	Output Output	Port 70: Output port Read: Outputs strobe signal to read external memory.
P71 $\overline{WRLL}$	1	Output Output	Port 71: Output port Write: Output strobe signal for writing data on pins D0 to D7.
P72 $\overline{WRLU}$	1	Output Output	Port 72: Output port Write: Output strobe signal for writing data on pins D8 to D15.
P73 $\overline{WRUL}$	1	Output Output	Port 73: Output port Write: Output strobe signal for writing data on pins D16 to D23.
P74 $\overline{WRUU}$	1	Output Output	Port 74: Output port Write: Output strobe signal for writing data on pins D24 to D31.
P75 R/ $\overline{W}$	1	Output Output	Port 75: Output port Read/Write: 1 represents read or dummy cycle; 0 represents write cycle.
P76 $\overline{WAIT}$	1	I/O Input	Port 76: I/O port Wait: Signal used to request CPU bus wait.
P80 $\overline{CS0}$ $\overline{SDCSH}$	1	Output Output Output	Port 80: Output port Chip select 0: Outputs "low" when address is within specified address area. Chip select for SDRAM: Outputs "0" when address is within SDRAM upper-address area.
P81 $\overline{CS1}$ $\overline{SDCSL}$	1	Output Output Output	Port 81: Output port Chip select 1: Outputs "low" when address is within specified address area. Chip select for SDRAM: Outputs "0" when address is within SDRAM lower-address area.
P82 $\overline{CS2}$ $\overline{CS2A}$	1	Output Output Output	Port 82: Output port Chip select 2: Outputs "low" when address is within specified address area. Expand chip select 2A: Outputs "0" when address is within specified address area.
P83 $\overline{CS3}$	1	Output Output	Port 83: Output port Chip select 3: Outputs "low" when address is within specified address area.
P84 EA24 $\overline{CS2B}$	1	Output Output Output	Port 84: 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.
P85 EA25 $\overline{CS2C}$	1	Output Output Output	Port 85: 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.
P86 $\overline{CS2D}$	1	Output Output	Port 86: Output port Expand chip select 2D: Outputs "0" when address is within specified address area.
P87 SDCLK	1	Output Output	Port 87: Output port Clock for SDRAM

Table 2.3.1 Pin Names and Functions (2/3)

Pin Names	Number of Pins	I/O	Functions
P90 SCK	1	I/O I/O	Port 90: I/O port Serial bus interface clock I/O data at SIO mode.
P91 SO SDA	1	I/O Output I/O	Port 91: I/O port Serial bus interface send data at SIO mode. Serial bus interface send/receive data at I^2C mode. (Open drain/output mode by programmable.)
P92 SI SCL	1	I/O Input I/O	Port 92: I/O port Serial bus interface receive data at SIO mode. Serial bus interface clock I/O data at I^2C mode. (Open drain/output mode by programmable.)
P93 $\overline{CS2E}$	1	I/O Output	Port 93: I/O port Expand chip select 2E: Outputs “0” when address is within specified address area.
P94 $\overline{CS2F}$	1	I/O Output	Port 94: I/O port Expand chip select 2F: Outputs “0” when address is within specified address area.
P95 $\overline{CS2G}$ TXD2	1	I/O Output Output	Port 95: Output port Expand chip select 2G: Outputs “0” when address is within specified address area. Serial transmission data 2. Open drain/output pin by programmable.
P96 RXD2 $\overline{CSEXA}$	1	I/O Input Output	Port 96: Output port Serial receive data 2. Expand chip select EXA: Outputs “0” when address is within specified address area.
PA0 to PA7 KI0 to KI7	8	Input Input	A0 to A7 port: 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.)
PC0 TA0IN	1	I/O Input	Port C0: I/O port 8-bit timer 0 input: Timer 0 input.
PC1 INT1 TA1OUT	1	I/O Input Output	Port C1: I/O port Interrupt request pin1 : Interrupt request pin with programmable rising /falling edge. 8-bit timer 1 output: Timer 1 output.
PC3 INT0	1	I/O Input	Port C3: I/O port Interrupt request pin 0: Interrupt request pin with programmable level/rising/falling edge.
PC5 INT2 TA3OUT	1	I/O Input Output	Port C5: I/O port Interrupt request pin 2 : Interrupt request pin with programmable rising /falling edge. 8-bit timer 3 output: Timer 3 output.
PC6 INT3 TB0OUT0	1	I/O Input Output	Port C6: I/O port Interrupt request pin 3: Interrupt request pin with programmable rising /falling edge. Timer B0 output.
PF0 TXD0	1	I/O Output	Port F0: I/O port Serial 0 send data: Open drain/output pin by programmable.
PF1 RXD0	1	I/O Input	Port F1: I/O port Serial 0 receive data.
PF2 SCLK0 $\overline{CTS0}$	1	I/O I/O Input	Port F2: I/O port Serial 0 clock I/O. Serial 0 data send enable (Clear to send).
PF3 TXD1	1	I/O Output	Port F3: I/O port Serial 1 send data: Open drain/output pin by programmable.
PF4 RXD1	1	I/O Input	Port F4: I/O port Serial 1 receive data.
PF5 SCLK1 $\overline{CTS1}$	1	I/O I/O Input	Port F5: I/O port Serial 1 clock I/O. Serial 1 data send enable (Clear to send).
PG0 to PG4 AN0 to AN4 $\overline{ADTRG}$	5	Input Input Input	Port G0 to G4 port: Pin used to input ports. Analog input 0 to 4: Pin used to Input to AD conveter. AD trigger: Signal used to request AD start (with used to PG3).

Table 2.3.1 Pin Names and Functions (3/3)

Pin Names	Number of Pins	I/O	Functions
PJ0 $\overline{\text{SDRAS}}$	1	Output Output	Port J0: Output port Row address strobe for SDRAM: Outputs “0” when address is within SDRAM address area.
PJ1 $\overline{\text{SDCAS}}$	1	Output Output	Port J1: Output port Column address strobe for SDRAM: Outputs “0” when address is within SDRAM address area.
PJ2 $\overline{\text{SDWE}}$ $\overline{\text{SRWR}}$	1	Output Output Output	Port J2: Output port Write enable for SDRAM. Write for SRAM: Strobe signal for writing data .
PJ3 SDLLDQM $\overline{\text{SRLLB}}$	1	Output Output Output	Port J3: Output port Data enable for SDRAM on pins D0 to D7. Data enable for SRAM on pins D0 to D7.
PJ4 SDLUDQM $\overline{\text{SRLUB}}$	1	Output Output Output	Port J4: Output port Data enable for SDRAM on pins D8 to D15. Data enable for SRAM on pins D8 to D15.
PJ5 SDULDQM $\overline{\text{SRULB}}$	1	Output Output Output	Port J5: Output port Data enable for SDRAM on pins D16 to D23. Data enable for SRAM on pins D16 to D23.
PJ6 SDUUDQM $\overline{\text{SRUUB}}$	1	Output Output Output	Port J6: Output port Data enable for SDRAM on pins D24 to D32. Data enable for SRAM on pins D24 to D32.
PJ7 SDCKE	1	Output Output	Port J7: Output port Clock enable for SDRAM.
PK0 D1BSCP	1	Output Output	Port K0: Output port LCD driver output pin.
PK1 D2BLP	1	Output Output	Port K1: Output port LCD driver output pin.
PK2 D3BFR	1	Output Output	Port K2: Output port LCD driver output pin.
PK3 DLEBCD	1	Output Output	Port K3: Output port LCD driver output pin.
PK4 DOFFB	1	Output Output	Port K4: Output port LCD driver output pin.
PK6 $\overline{\text{ALARM}}$ $\overline{\text{MLDALM}}$	1	Output Output Output	Port K6: Output port RTC alarm output pin. Melody/alarm output pin (Inverted).
PL0 to PL7 LD0 to LD7	8	I/O Output	Port L0 to L7: I/O port Data bus for LCD driver.
$\overline{\text{BE}}$	1	Input	Backup enable.
AM0, AM1	2	Input	Operation mode: Fix to AM1 = “0”, AM0 = “1”: 16-bit external bus or 8-/16-/32-bit dynamic sizing. Fix to AM1 = “1”, AM0 = “0”: 32-bit external bus or 8-/16-/32-bit dynamic sizing.
X1/X2	2	I/O	High-frequency oscillator connection pins.
XT1/XT2	2	I/O	Low-frequency oscillator connection pins.
$\overline{\text{RESET}}$	1	Input	Reset: Initializes TMP92C820 (with pull-up resistor).
VREFH	1	Input	Pin for reference voltage input to AD converter (H).
VREFL	1	Input	Pin for reference voltage input to AD converter (L).
AVCC	1	–	Power supply pin for AD converter.
AVSS	1	–	GND pin for AD converter (0 V).
DVCC	3	–	Power supply pins (All DVCC pins should be connected with the power supply pin).
DVSS	4	–	GND pins (0 V) (All DVSS pins should be connected with GND (0V)).
RTCVCC	1	–	Power supply pin for RTC and low-frequency oscillator.

3. Operation

This section describes the basic components, functions and operation of the TMP92C820.

3.1 CPU

The TMP92C820 contains an advanced high-speed 32-bit CPU (900/H1 CPU). For CPU operation, see the TLCS-900/H1 CPU.

The following describe the unique function of the CPU used in the TMP92C820; these functions are not covered in the TLCS-900/H1 CPU section.

3.1.1 CPU Outline

900/H1 CPU is high-speed and high-performance CPU based on 900/L1 CPU. 900/H1 CPU has expanded 32-bit internal data bus to process instructions more quickly.

Outline of 900/H1 CPU are as follows:

Table 3.1.1 CPU Outline

	900/H1 CPU
Width of CPU address bus	24 bits
Width of CPU data bus	32 bits
Internal operating frequency	20 MHz
Minimum bus cycle	1-clock access (50 ns at 20 MHz)
Data bus sizing	8/16/32 bits
Internal RAM	32 bits 1-clock access
Internal I/O	8-/16-bit 2-clock access 900/H1 I/O 8-/16-bit 5 to 6-clock access 900/L1 I/O
External device	8 bits 2-clock access (can insert some waits.)
Minimum instruction Execution cycle	1 clock (50 ns at 20 MHz)
Conditional jump	2 clocks (100 ns at 20 MHz)
Instruction queue buffer	12 bytes
Instruction set	Compatible with TLCS-900, 900/L, 900/H, 900/L1 and 900/H2 (NORMAL, MAX, MIN and LDX instruction is deleted.)
CPU mode	Only maximum mode
Micro DMA	8 channels

3.1.2 Reset Operation

When resetting the TMP92C820 microcontroller, ensure that the power supply voltage is within the operating voltage range, and that the internal high-frequency oscillator has stabilized. Then hold the $\overline{\text{RESET}}$ input low for at least 20 system clocks (16 μs at 40 MHz).

When the reset has been accepted, the CPU performs the following:

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

  PC<7:0> ← Data in location FFFF00H

  PC<15:8> ← Data in location FFFF01H

  PC<23:16> ← Data in location FFFF02H
- Sets the stack pointer (XSP) to 00000000H.
- Sets bits <IFF0:2> of the status register (SR) to 111 (Thereby setting the interrupt level mask register to level 7).
- Clears bits <RFP0:1> of the status register to 00 (Thereby selecting register bank0).

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

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

- Initializes the internal I/O registers as table of "Table of Special Function Registers (SFRs)" in section 5.
- Sets the port pins, including the pins that also act as internal I/O, to general-purpose input or output port mode.

Internal $\overline{\text{RESET}}$ is released as soon as external reset is released.

The operation of memory controller cannot be insured until power supply becomes stable after power-on reset. The external RAM data provided before turning on the TMP92C820 may be spoiled because the control signals are unstable until power supply becomes stable after power on reset.

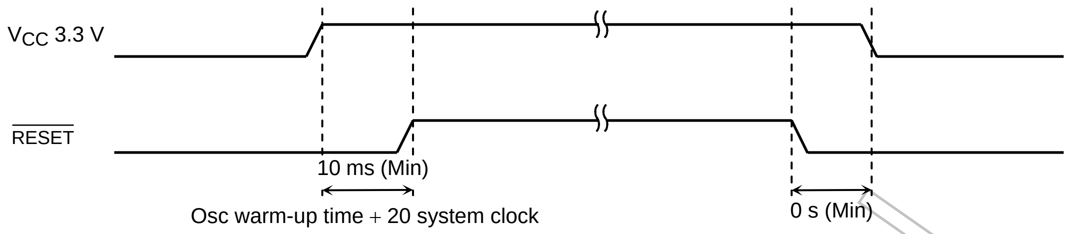


Figure 3.1.1 Power on Reset Timing Example

3.1.3 Setting of AM0 and AM1

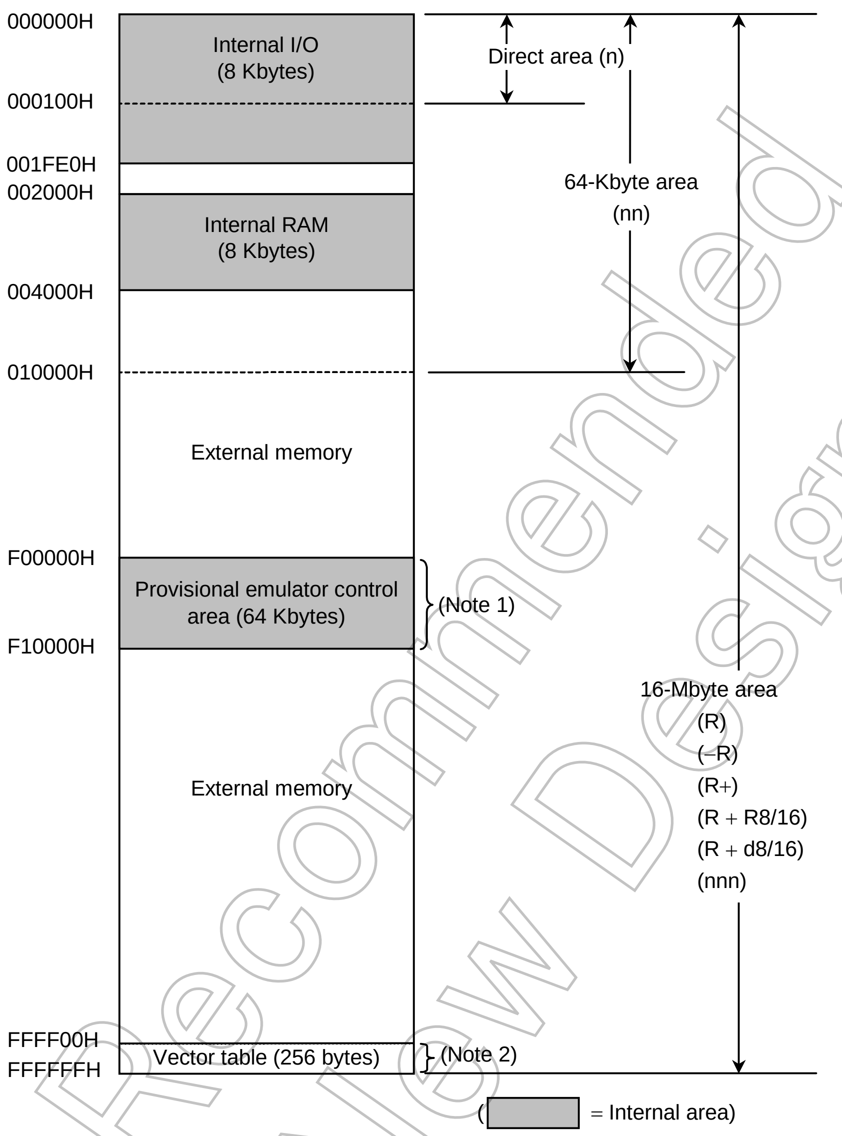
Set AM1 and AM0 pins to “10” to use 32-bit external bus, or set it to “01” to use 16-bit external bus.

Table 3.1.2 Operation Mode Setup Table

Operation Mode	Mode Setup Input Pin		
	RESET	AM1	AM0
16-bit external bus or 8-/16-/32-bit dynamic bus sizing	↑	0	1
32-bit external bus or 8-/16-/32-bit dynamic bus sizing	↑	1	0

3.2 Memory Map

Figure 3.2.1 is a memory map of the TMP92C820.



Note 1: Provisional emulator control area is for emulator, it is mapped F00000H to F10000H address after reset.

Note 2: Don't use the last 16-byte area (FFFFF0H to FFFFFFH). This area is reserved.

Note 3: On emulator $\overline{WR}$ signal and $\overline{RD}$ signal are asserted, when provisional emulator control area is accessed. Be careful to use external memory.

Figure 3.2.1 Memory Map

3.3 Clock Function and Standby Function

TMP92C820 contains (1) Clock gear, (2) Standby controller, and (3) Noise reduction circuit. It is used for low-power, low-noise systems.

This chapter is organized as follows:

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

Figure 3.3.1 shows a transition figure.

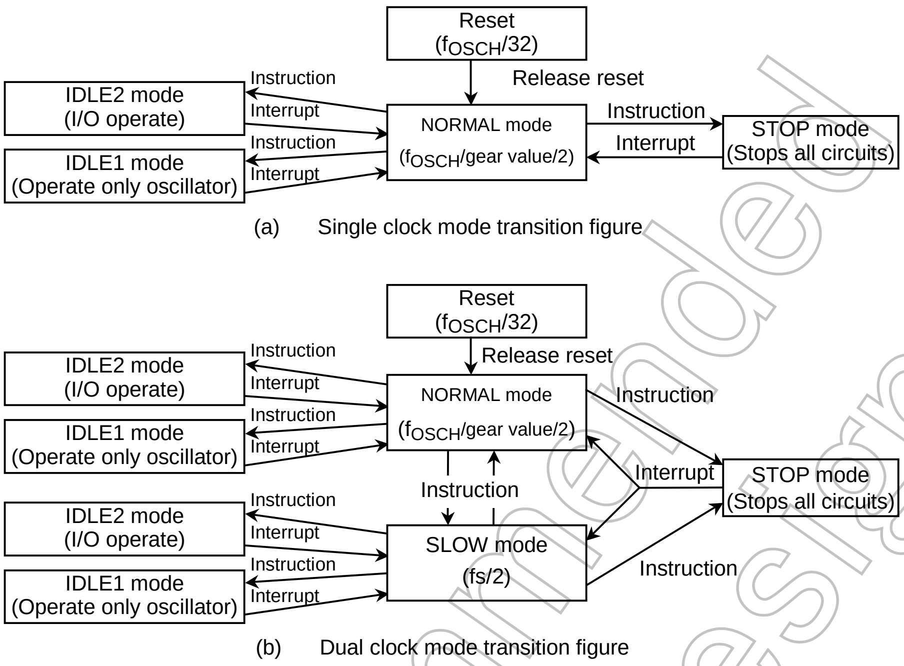


Figure 3.3.1 System Clock Block Diagram

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

3.3.1 Block Diagram of System Clock

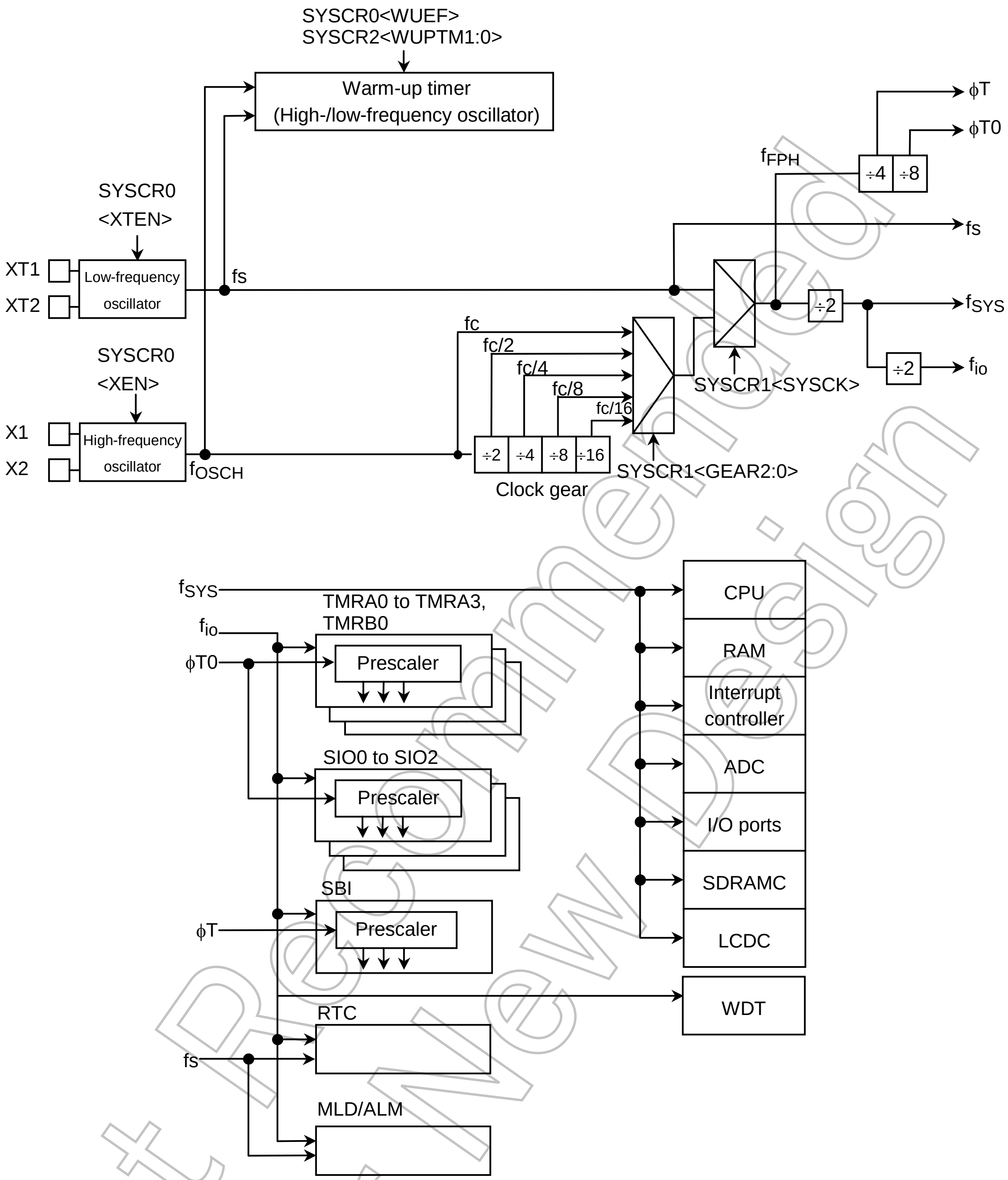


Figure 3.3.2 Block Diagram of System Clock

3.3.2 SFR

Register		7	6	5	4	3	2	1	0
SYSCR0 (10E0H)	Bit symbol	XEN	XTEN				WUEF		
	Read/Write	R/W					R/W		
	After reset	1	1				0		
	Function	High-frequency oscillator (fc) 0: Stop 1: Oscillation	Low-frequency oscillator (fs) 0: Stop 1: Oscillation				Warm-up timer 0: Write Don't care 1: Write start timer 0: Read end warm up 1: Read do not end warm up		
SYSCR1 (10E1H)	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: } 110: } Reserved 111: }		
SYSCR2 (10E2H)	Bit symbol	–		WUPTM1	WUPTM0	HALTM1	HALTM0	SELDRV	DRVE
	Read/Write	R/W		R/W					
	After reset	0		1	0	1	1	0	0
	Function	Always write "0".		Warm-up timer 00: Reserved 01: 2^8/inputted frequency 10: 2^{14}/inputted frequency 11: 2^{16}/inputted frequency		HALT mode 00: Reserved 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		<DRVE> mode select 0: STOP 1: IDLE1	Pin state control in STOP/ IDLE1 mode 0: I/O off 1: Remains the state before halt.

Note 1: The unassigned register, SYSCR0<Bit5:3>, SYSCR0<Bit1:0>, SYSCR1<Bit7:4>, and SYSCR2<Bit7:6> are read as undefined value.

Note 2: By reset, low-frequency oscillator is enabled.

Figure 3.3.3 SFR for System Clock

Register		7	6	5	4	3	2	1	0
EMCCR0 (10E3H)	Bit symbol	PROTECT					EXTIN	DRVOSCH	DRVOSCL
	Read/Write	R					R/W		
	After reset	0					0	1	1
	Function	Protect flag 0: OFF 1: ON					1: fc external clock	fc oscillator drive ability 1: Normal 0: Weak	fs oscillator drive ability 1: Normal 0: Weak
EMCCR1 (10E4H) EMCCR2 (10E5H)	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								

Figure 3.3.4 SFR for Noise-reduction

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

3.3.3 System Clock Controller

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

The combination of settings <XEN> = 1, <XTEN> = 1, <SYSCK> = 0 and <GEAR2:0> = 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 1.25 MHz when the 40 MHz oscillator is connected to the X1 and X2 pins.

(1) Switching from NORMAL mode to SLOW mode

When the resonator is connected to the X1 and X2 pins, or to the XT1 and XT2 pins, the warm-up timer can be used to change the operation frequency after stable oscillation has been attained. The warm-up time can be selected using SYSCR2<WUPTM1:0>. This warm-up timer can be programmed to start and stop as shown in the following examples 1 and 2.

Table 3.3.1 shows the warm-up time.

Note 1: When using an oscillator (other than a resonator) with stable oscillation, a warm-up timer is not needed.

Note 2: The warm-up timer is operated by an oscillation clock. Hence, there may be some variation in warm-up time.

Table 3.3.1 Warm-up Times

Warm-up Time SYSCR2<WUPTM1:0>	Change to NORMAL Mode (fc)	Change to SLOW Mode (fs)
01 (2^8/frequency)	6.4 [μs]	7.8 [ms]
10 (2^{14}/frequency)	409.6 [μs]	500 [ms]
11 (2^{16}/frequency)	1.638 [ms]	2000 [ms]

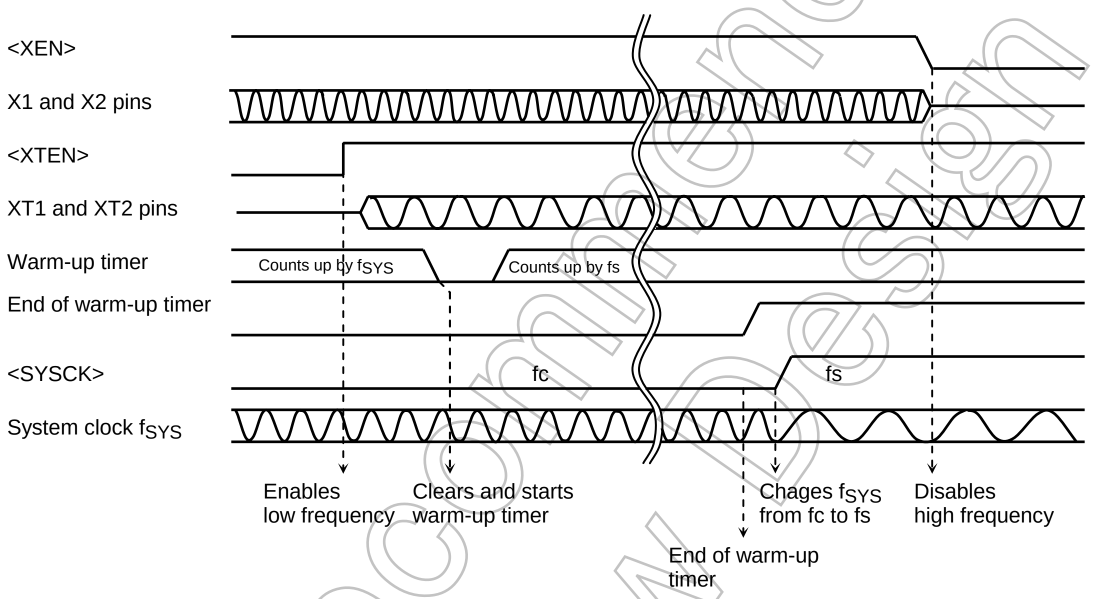
at f_{OSCH} = 40 MHz, fs = 32.768 kHz

Example 1: Setting the clock

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

```
SYSCR0   EQU   10E0H
SYSCR1   EQU   10E1H
SYSCR2   EQU   10E2H
         LD    (SYSCR2), 0X11 - - - - 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

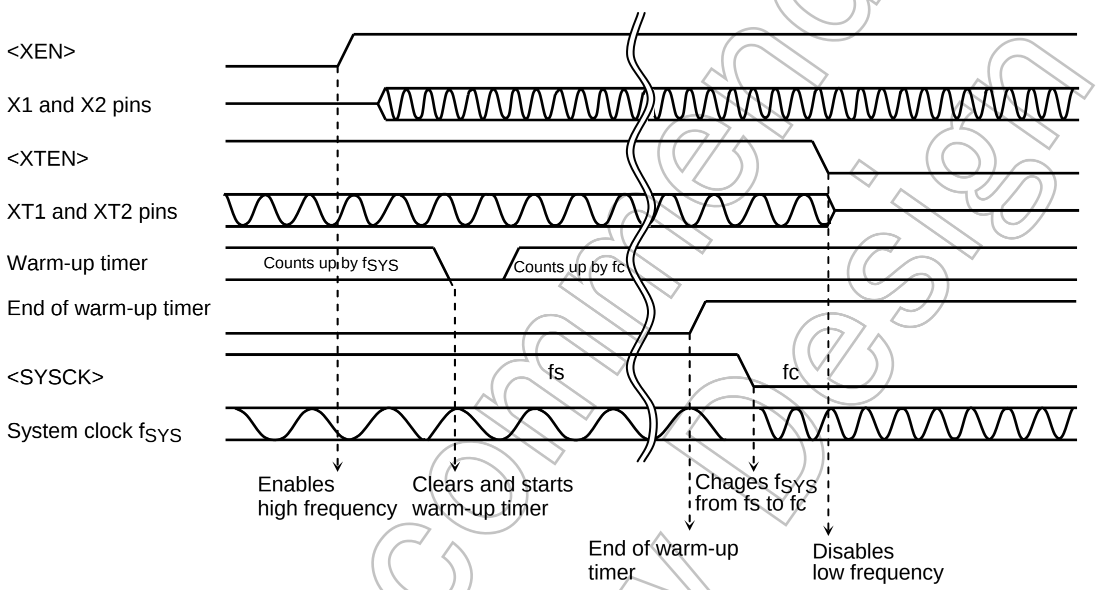


Example 2: Setting the clock

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

```
SYSCR0   EQU   10E0H
SYSCR1   EQU   10E1H
SYSCR2   EQU   10E2H
         LD    (SYSCR2), 0X10 - - - - B   ;  Sets warm-up time to 2^14/fc.
         SET   7, (SYSCR0)                ;  Enables high-frequency oscillation.
         SET   2, (SYSCR0)                ;  Clears and starts warm-up timer.
WUP:     BIT   2, (SYSCR0)                ;  } Detects stopping of warm-up timer.
         JR    NZ, WUP                    ;  }
         RES   3, (SYSCR1)                ;  Changes fSYS from fs to fc.
         RES   6, (SYSCR0)                ;  Disables low-frequency oscillation.
```

X: Don't care, –: No change



(2) Clock gear controller

f_{FPH} is set according to the contents of the clock gear select register SYSCR1<GEAR2:0> to either fc, fc/2, fc/4, fc/8 or fc/16. Using the clock gear to select a lower value of f_{FPH} reduces power consumption.

Example 3:

Changing to a high-frequency gear

```
SYSCR1    EQU    10E1H
          LD     (SYSCR1), XXXX0000B    ;    Changes fSYS to fc/2.
X: Don't care
```

(High-speed clock gear changing)

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

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

```
(Example)
SYSCR1    EQU    10E1H
          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 Noise Reduction Circuits

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

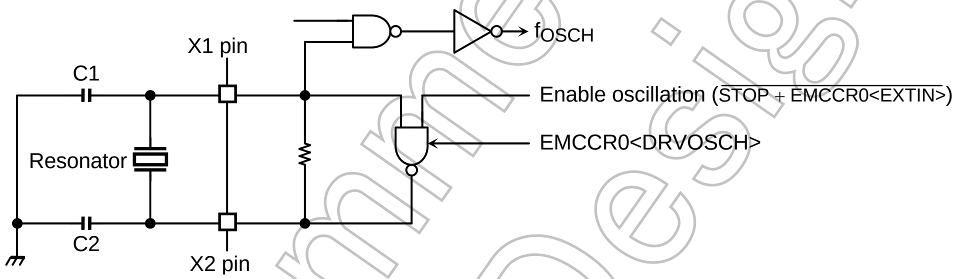
(1) Reduced drivability for high-frequency oscillator
(2) Reduced drivability for low-frequency oscillator
(3) Single drive for high-frequency oscillator
(4) Runaway provision with SFR protection register

(1) Reduced drivability for high-frequency oscillator

(Purpose)

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

(Block diagram)



(Setting method)

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

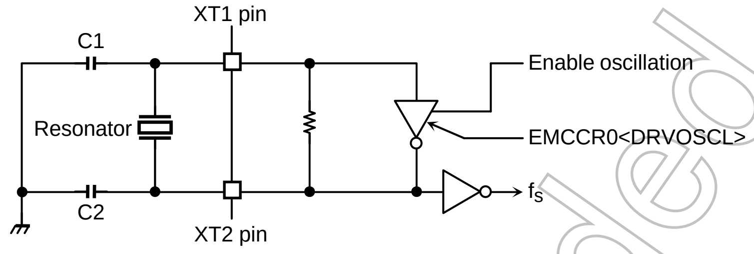
Note: This function (EMCCR0<DRVOSCH> = “0”) is available to use in case ofwhen f_{OSCH} = 6 to 10 MHz condition.

(2) Reduced drivability for low-frequency oscillator

(Purpose)

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

(Block diagram)



(Setting method)

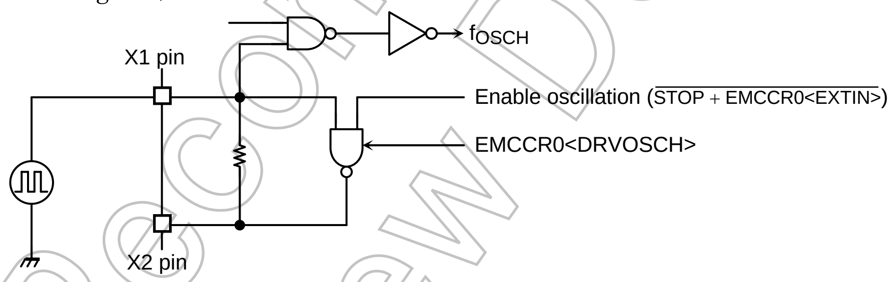
The drivability of the oscillator is reduced by writing 0 to the EMCCR0<DRVOSCL> register. By reset, <DRVOSCL> is initialized to “1”.

(3) Single drive for high-frequency oscillator

(Purpose)

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

(Block diagram)



(Setting method)

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

By reset, <EXTIN> is initialized to “0”.

(4) Runaway provision with SFR protection register

(Purpose)

Provision in runaway of program by noise mixing.

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

And error handling in runaway becomes easy by INTP0 interruption.

Specified SFR list

1.	Memory controller B0CSL/H, B1CSL/H, B2CSL/H, B3CSL/H, BECSL/H MSAR0, MSAR1, MSAR2, MSAR3, MAMR0, MAMR1, MAMR2, MAMR3, PMEMCR
2.	MMU LOCAL 0/1/2/3
3.	Clock gear SYSCR0, SYSCR1, SYSCR2, EMCCR0

(Operation explanation)

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

(Double key)

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

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

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

By reset, protection becomes OFF.

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

3.3.5 Standby Controller

(1) HALT modes

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

The subsequent actions performed in each mode are as follows:

1. IDLE2: Only the CPU halts.

   The internal I/O is available to select operation during IDLE2 mode by setting the following register.
   Table 3.3.2 shows the registers of setting operation during IDLE2 mode.

Table 3.3.2 SFR 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>

2. IDLE1: Only the oscillator, the RTC (Real time clock) and MLD (Melody-alarm generator) continue to operate.
3. STOP: All internal circuits stop operating.

The operation of each of the different HALT modes is described in Table 3.3.3.

Table 3.3.3 I/O Operation during HALT Modes

HALT Modes		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 and Table 3.3.8	
	TMRA, TMRB	Available to select operation block (Note)	Stop	
	SIO, SBI (Note)			
	AD converter			
	WDT			
	LCDC, SDRAMC interrupt controller	Operate		
	RTC, MLD		Operate	

Note: Prohibited in the synchronous mode of SBI circuit.

(2) How to release the HALT mode

These halt states can be released by resetting or requesting an interrupt. The halt release sources are determined by the combination between the states of interrupt mask register <IFF2:0> and the HALT modes. The details for releasing the halt status are shown in Table 3.3.4.

1. Released by requesting an interrupt

The operating released from the HALT mode depends on the interrupt enabled status. When the interrupt request level set before executing the HALT instruction exceeds the value of interrupt mask register, the interrupt due to the source is processed after releasing the HALT mode, and CPU status executing an instruction that follows the HALT instruction. When the interrupt request level set before executing the HALT instruction is less than the value of the interrupt mask register, releasing the HALT mode is not executed. (In non-maskable interrupts, interrupt processing is processed after releasing the HALT mode regardless of the value of the mask register.)

However only for INT0 to INT3, INTKEY, INTRTC, and INTALM0 to INTALM4 interrupts, even if the interrupt request level set before executing the HALT instruction is less than the value of the interrupt mask register, releasing the HALT mode is executed. In this case, interrupt processing, and CPU starts executing the instruction next to the HALT instruction, but the interrupt request flag is held at "1".

Note: Usually, interrupts can release all halts status. However, the interrupts (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 3 clocks of f_{FPH}) with IDLE1 or STOP mode (IDLE2 is not applicable to this case). (In this case, an interrupt request is kept on hold internally.) If another interrupt is generated after it has shifted to HALT mode completely, halt status can be released without difficulty. The priority of this interrupt is compared with that of the interrupt kept on hold internally, and the interrupt with higher priority is handled first followed by the other interrupt.

2. Releasing by resetting

Releasing all halt status is executed by resetting.

When the STOP mode is released by RESET, it is necessary enough resetting time (See Table 3.3.5) to set the operation of the oscillator to be stable.

When releasing the HALT mode by resetting, the internal RAM data keeps the state before the "HALT" instruction is executed. However the other settings contents are initialized. (Releasing due to interrupts keeps the state before the "HALT" instruction is executed.)

Table 3.3.4 Source of Halt State Clearance and Halt Clearance Operation

Status of Received Interrupt		Interrupt Enabled (Interrupt level) ≥ (Interrupt mask)			Interrupt Disabled (Interrupt level) < (Interrupt mask)		
HALT Mode		IDLE2	IDLE1	STOP	IDLE2	IDLE1	STOP
Source of Halt State Clearance: Interrupt	INTWDT	♦	×	×	–	–	–
	INT0 to 3 (Note1)	♦	♦	♦*1	○	○	○*1
	INTALM0 to 4	♦	♦	×	○	○	×
	INTTA0 to 3, INTTB00 to 01	♦	×	×	×	×	×
	INTRX0 to 2, TX0 to 2	♦	×	×	×	×	×
	INTSS0 to 2	♦	×	×	×	×	×
	INTAD	♦	×	×	×	×	×
	INTKEY	♦	♦	♦*1	○	○	○*1
	INTRTC	♦	♦	×	○	○	×
	INTSBE0	♦	×	×	×	×	×
	INTLCD	♦	×	×	×	×	×
Source of Halt State Clearance	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 1: When the HALT mode is cleared by an INT0 interrupt of the level mode in the interrupt enabled status, hold level H until starting interrupt processing. If level L is set before holding level L, interrupt processing is correctly started.

(Example releasing IDLE1 mode)

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

```
Address
8200H   LD    (PCFC), 04H      ;  Sets PC3F to INT0.
8203H   LD    (IIMC), 00H      ;  Selects INT0 interrupt rising edge.
8206H   LD    (INTE0AD), 06H   ;  Sets INT0 interrupt level to 6.
8209H   EI    5                ;  Sets interrupt level to 5 for CPU.
820BH   LD    (SYSCR2), 28H    ;  Sets HALT mode to IDLE1 mode.
820EH   HALT                   ;  Halts CPU.

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

(3) Operation

1. IDLE2 mode

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

Figure 3.3.5 illustrates an example of the timing for clearance of the IDLE2 mode halt state by an interrupt.

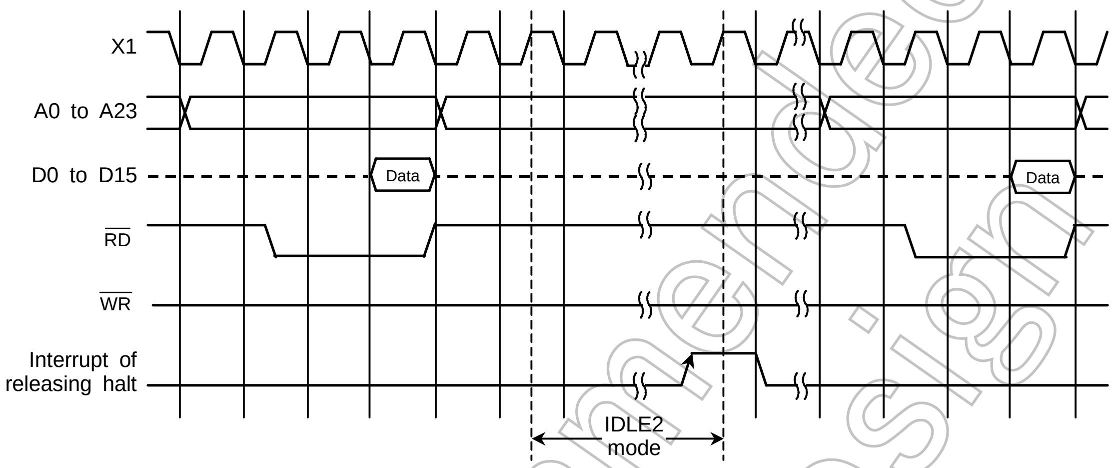


Figure 3.3.5 Timing Chart for IDLE2 Mode Halt State Cleared by Interrupt

2. IDLE1 mode

In IDLE1 mode, only the internal oscillator and the RTC and MLD continue to operate. The system clock in the MCU stops. The pin status in the IDLE1 mode is depended on setting the register SYSCR2<SELDRV, DRVE>. Table 3.3.6 , Table 3.3.7 and Table 3.3.8 summarizes the state of these pins in the IDLE1 mode.

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

Figure 3.3.6 illustrates the timing for clearance of the IDLE1 mode halt state by an interrupt.

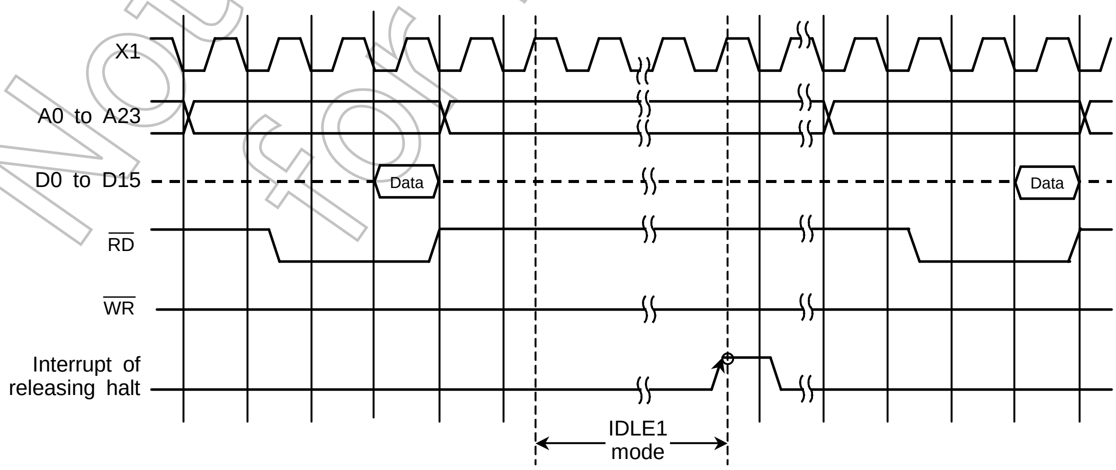


Figure 3.3.6 Timing Chart for IDLE1 Mode Halt State Cleared by Interrupt

3. STOP mode

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

After STOP mode has been cleared system clock output starts when the warm-up time has elapsed, in order to allow oscillation to stabilize.

Figure 3.3.7 illustrates the timing for clearance of the STOP mode halt state by an interrupt.

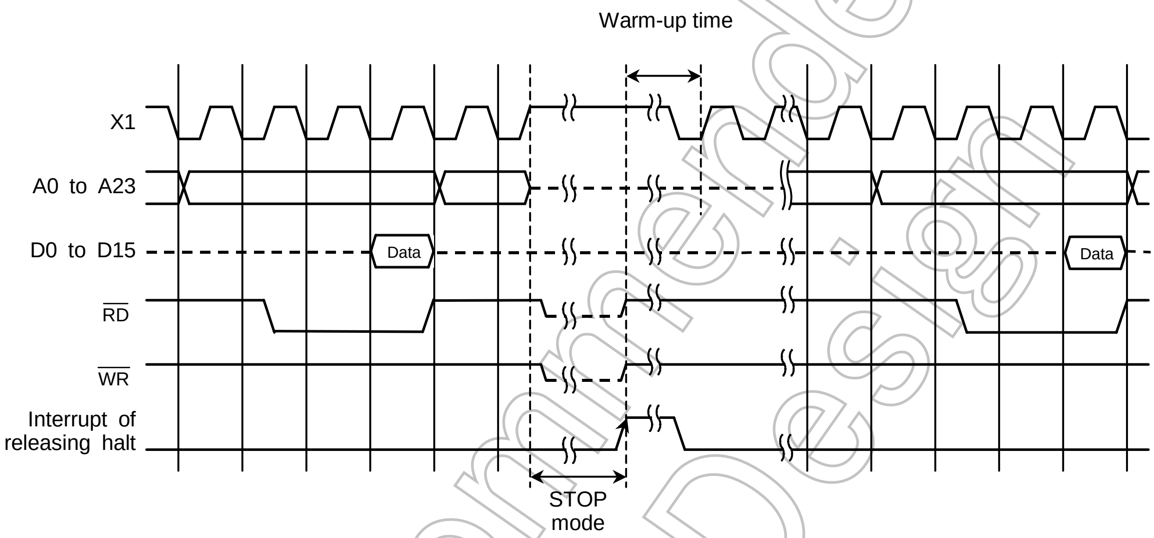


Figure 3.3.7 Timing Chart for STOP Mode Halt State Cleared by Interrupt

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

at f_{OSCH} = 40 MHz, fs = 32.768 kHz

SYSCR0 <RSYSCK>	SYSCR2<WUPTM1:0>		
	01 (2^8)	10 (2^{14})	11 (2^{16})
0 (fc)	6.4 μs	409.6 μs	1.638 ms

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 pin	When used as Function pin	When used as Input pin	When used as Function pin	When used as Input pin	When used as Function pin	When used as Input pin
D0-D7	D0-D7	OFF	ON upon external read	–	ON upon external read of LCDC	–	OFF	–	OFF	–
P10-P17	D8-D15			ON		OFF		OFF		OFF
P20-P27	D16-D23	16-bit start :ON 32-bit start :OFF								
P30-P37	D24-D31									
P40-P47	–	OFF	–		–		–		–	
P50-P57	–									
P60-P67	–									
P76	WAIT	ON	ON		OFF		OFF		OFF	
P90	SCK				ON					
P91	SDA									
P92	SI, SCL									
P93	–		–		–		–		–	
P94	–									
P95	–									
P96	RXD2		ON		ON	ON	OFF		OFF	
PA0-PA7 (*1)	KI0-7						ON	ON	ON	ON
PC0	TA0IN					OFF	OFF	OFF	OFF	OFF
PC1	INT1						ON		ON	
PC3	INT0					ON		ON		ON
PC5	INT2					OFF		OFF		OFF
PC6	INT3									
PF0	–		–		–		–		–	
PF1	RXD0		ON		ON	ON	OFF		OFF	
PF2	SCLK0, CTS0									
PF3	–		–		–	OFF	–		–	
PF4	RXD1		ON		ON	ON	OFF		OFF	
PF5	SCLK1, CTS1									
PG0-PG2, PG4 (*2)	–	OFF	–	ON upon port read	–	OFF	–		–	
PG3 (*2)	ADTRG		ON		ON		ON		ON	
PL0-PL7	–	ON	–	ON	–		–		–	
$\overline{BE}$	–		ON	–	ON	–	ON	–	ON	–
RESET (*1)	–									
AM0, AM1	–									
X1, XT1							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 B	Condition A
0	1	Condition A	
1	0	Condition B	Condition B
1	1		

Table 3.3.7 Output Buffer State Table (1/2)

<table>
<tr><th rowspan="4">Port Name</th><th rowspan="4">Output Function Name</th><th colspan="9">Output Buffer State</th></tr>
<tr><th rowspan="3">During Reset</th><th colspan="2" rowspan="2">When the CPU is Operating</th><th colspan="2" rowspan="2">In HALT mode (IDLE2)</th><th colspan="4">In HALT mode (IDLE1/STOP)</th></tr>
<tr><th colspan="2">Condition A (Note)</th><th colspan="2">Condition B (Note)</th></tr>
<tr><th>When used as Function Pin</th><th>When Used as Output Port</th><th>When Used as Function Pin</th><th>When Used as Output Port</th><th>When Used as Function Pin</th><th>When Used as Output Port</th><th>When Used as Function Pin</th><th>When Used as Output Port</th></tr>
<tr><td>D0-D7</td><td>D0-D7</td><td rowspan="4">OFF</td><td rowspan="4">ON upon external write</td><td>–</td><td rowspan="4">OFF</td><td>–</td><td rowspan="13">OFF</td><td>–</td><td rowspan="4">OFF</td><td>–</td></tr>
<tr><td>P10-P17</td><td>D8-D15</td><td rowspan="33">ON</td><td rowspan="33">ON</td><td rowspan="3">ON</td><td rowspan="33">ON</td></tr>
<tr><td>P20-P27</td><td>D16-D23</td></tr>
<tr><td>P30-P37</td><td>D24-D31</td></tr>
<tr><td>P40-P47</td><td>A0-A7</td><td rowspan="9">ON</td><td rowspan="9">ON</td><td rowspan="9">ON</td><td rowspan="30">OFF</td><td rowspan="9">ON</td></tr>
<tr><td>P50-P57</td><td>A8-A15</td></tr>
<tr><td>P60-P67</td><td>A16-A23</td></tr>
<tr><td>P70</td><td>RD</td></tr>
<tr><td>P71</td><td>WRLL</td></tr>
<tr><td>P72</td><td>WRLU</td></tr>
<tr><td>P73</td><td>WRUL</td></tr>
<tr><td>P74</td><td>WRUU</td></tr>
<tr><td>P75</td><td>R/W</td></tr>
<tr><td>P76</td><td>–</td><td>OFF</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>P80</td><td>CS0, SDCSH</td><td rowspan="8">ON</td><td rowspan="15">ON</td><td rowspan="15">ON</td><td rowspan="15">OFF</td><td rowspan="15">ON</td></tr>
<tr><td>P81</td><td>CS1, SDCSL</td></tr>
<tr><td>P82</td><td>CS2, CS2A</td></tr>
<tr><td>P83</td><td>CS3</td></tr>
<tr><td>P84</td><td>EA24, CS2B</td></tr>
<tr><td>P85</td><td>EA25, CS2C</td></tr>
<tr><td>P86</td><td>CS2D</td></tr>
<tr><td>P87</td><td>SDCLK</td></tr>
<tr><td>P90</td><td>SCK</td><td rowspan="12">OFF</td></tr>
<tr><td>P91</td><td>SO</td></tr>
<tr><td>P92</td><td>SCL</td></tr>
<tr><td>P93</td><td>CS2E</td></tr>
<tr><td>P94</td><td>CS2F</td></tr>
<tr><td>P95</td><td>CS2G TXD2</td></tr>
<tr><td>P96</td><td>CSEXA</td></tr>
<tr><td>PC0</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>PC1</td><td>TA1OUT</td><td>ON</td><td>ON</td><td>OFF</td><td>ON</td></tr>
<tr><td>PC3</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>PC5</td><td>TA3OUT</td><td rowspan="2">ON</td><td rowspan="2">ON</td><td rowspan="2">OFF</td><td rowspan="2">ON</td></tr>
<tr><td>PC6</td><td>TB0OUT</td></tr>
</table>

Table 3.3.8 Output Buffer State Table (2/2)

<table>
<tr><th rowspan="4">Port Name</th><th rowspan="4">Output Function Name</th><th colspan="9">Output Buffer State</th></tr>
<tr><th rowspan="3">During Reset</th><th colspan="2" rowspan="2">When the CPU is Operating</th><th colspan="2" rowspan="2">In HALT mode (IDLE2)</th><th colspan="4">In HALT mode (IDLE1/STOP)</th></tr>
<tr><th colspan="2">Condition A (Note)</th><th colspan="2">Condition B (Note)</th></tr>
<tr><th>When used as Function Pin</th><th>When Used as Output Port</th><th>When Used as Function Pin</th><th>When Used as Output Port</th><th>When Used as Function Pin</th><th>When Used as Output Port</th><th>When Used as Function Pin</th><th>When Used as Output Port</th></tr>
<tr><td>PF0</td><td>TXD0</td><td rowspan="21">OFF</td><td>ON</td><td rowspan="21">ON</td><td>ON</td><td rowspan="21">ON</td><td>OFF</td><td rowspan="21">OFF</td><td></td><td rowspan="21">ON</td></tr>
<tr><td>PF1</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>PF2</td><td>SCLK0</td><td rowspan="2">ON</td><td rowspan="2">ON</td><td rowspan="2">OFF</td><td rowspan="2">ON</td></tr>
<tr><td>PF3</td><td>TXD1</td></tr>
<tr><td>PF4</td><td>–</td><td>–</td><td>–</td><td>–</td><td>–</td></tr>
<tr><td>PF5</td><td>SCLK1</td><td rowspan="18">ON</td><td rowspan="18">ON</td><td rowspan="8">OFF</td><td rowspan="16">ON</td></tr>
<tr><td>PJ0</td><td>SDRAS</td></tr>
<tr><td>PJ1</td><td>SDCAS</td></tr>
<tr><td>PJ2</td><td>SDWE
SRWR</td></tr>
<tr><td>PJ3</td><td>SDLLDQM
SRLLB</td></tr>
<tr><td>PJ4</td><td>SDLUDQM
SRLUB</td></tr>
<tr><td>PJ5</td><td>SDULDQM
SRULB</td></tr>
<tr><td>PJ6</td><td>SDUUDQM
SRUUB</td></tr>
<tr><td>PJ7</td><td>SDCKE</td><td>ON in self refresh cycle</td></tr>
<tr><td>PK0</td><td>D1BSCP</td><td rowspan="7">OFF</td></tr>
<tr><td>PK1</td><td>D2BLP</td></tr>
<tr><td>PK2</td><td>D3BFR</td></tr>
<tr><td>PK3</td><td>DLEBCD</td></tr>
<tr><td>PK4</td><td>DOFFB</td></tr>
<tr><td>PK6</td><td>ALARM
MLDALM</td></tr>
<tr><td>PL0-PL7</td><td>LD0-LD7</td></tr>
<tr><td>X2</td><td>–</td><td rowspan="2">ON</td><td rowspan="2">–</td><td rowspan="2">–</td><td colspan="4">IDLE1: ON, STOP: output “H” level</td></tr>
<tr><td>XT2</td><td>–</td><td colspan="4">IDLE1: ON, STOP: High-Z</td></tr>
</table>

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

OFF: The buffer is always turned off.

-: No applicable

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

Note: Condition A/B are as follos.

<table>
<tr><th colspan="2">SYSCR2 register setting</th><th colspan="2">HALT mode</th></tr>
<tr><th><DRVE></th><th><SELDRV></th><th>IDLE1</th><th>STOP</th></tr>
<tr><td>0</td><td>0</td><td>Condition B</td><td rowspan="2">Condition A</td></tr>
<tr><td>0</td><td>1</td><td>Condition A</td></tr>
<tr><td>1</td><td>0</td><td rowspan="2">Condition B</td><td rowspan="2">Condition B</td></tr>
<tr><td>1</td><td>1</td></tr>
</table>

3.4 Interrupts

Interrupts are controlled by the CPU interrupt mask register <IFF2:0> (Bits 12 to 14 of the status register) and by the built-in interrupt controller.

The TMP92C820 has a total of 45 interrupts divided into the following five types:

Interrupts generated by CPU: 9 sources
- Software interrupts: 8 sources
- Illegal Instruction interrupt: 1 source

Internal interrupts: 31 sources
- Internal I/O interrupts: 23 sources
- Micro DMA transfer end interrupts: 8 sources

External interrupts: 5 sources
- Interrupts on external pins (INT0 to INT3, INTKEY)

A fixed individual interrupt vector number is assigned to each interrupt source.

Any one of six levels of priority can also be assigned to each maskable interrupt. Non-maskable interrupts have a fixed priority level of 7, the highest level.

When an interrupt is generated, the interrupt controller sends the priority of that interrupt to the CPU. When more than one interrupt are generated simultaneously, the interrupt controller sends the priority value of the interrupt is with the highest priority to the CPU. (The highest priority level is 7, the level used for non-maskable interrupts.)

The CPU compares the interrupt priority level which it receives with the value held in the CPU interrupt mask register <IFF2:0>. If the priority level of the interrupt is greater than or equal to the value in the interrupt mask register, the CPU accepts the interrupt.

However, software interrupts and illegal instruction interrupts generated by the CPU are processed irrespective of the value in <IFF2:0>.

The value in the interrupt mask register <IFF2:0> can be changed using the EI instruction (EI num sets <IFF2:0> to num). For example, the command EI3 enables the acceptance of all non-maskable interrupts and of maskable interrupts whose priority level, as set in the interrupt controller, is 3 or higher. The commands EI and EI0 enable the acceptance of all non-maskable interrupts and of maskable interrupts with a priority level of 1 or above (hence both are equivalent to the command EI1).

The DI instruction (Sets <IFF2:0> to 7) is exactly equivalent to the EI7 instruction. The DI instruction is used to disable all maskable interrupts (since the priority level for maskable interrupts ranges from 1 to 6). The EI instruction takes effect as soon as it is executed.

In addition to the general-purpose interrupt processing mode described above, there is also a micro DMA processing mode. In micro DMA mode the CPU automatically transfers data in one-byte, two-byte or four-byte blocks; this mode allows high-speed data transfer to and from internal and external memory and internal I/O ports.

In addition, the TMP92C820 also has a software start function in which micro DMA processing is requested in software rather than by an interrupt.

Figure 3.4.1 is a flowchart showing overall interrupts processing.

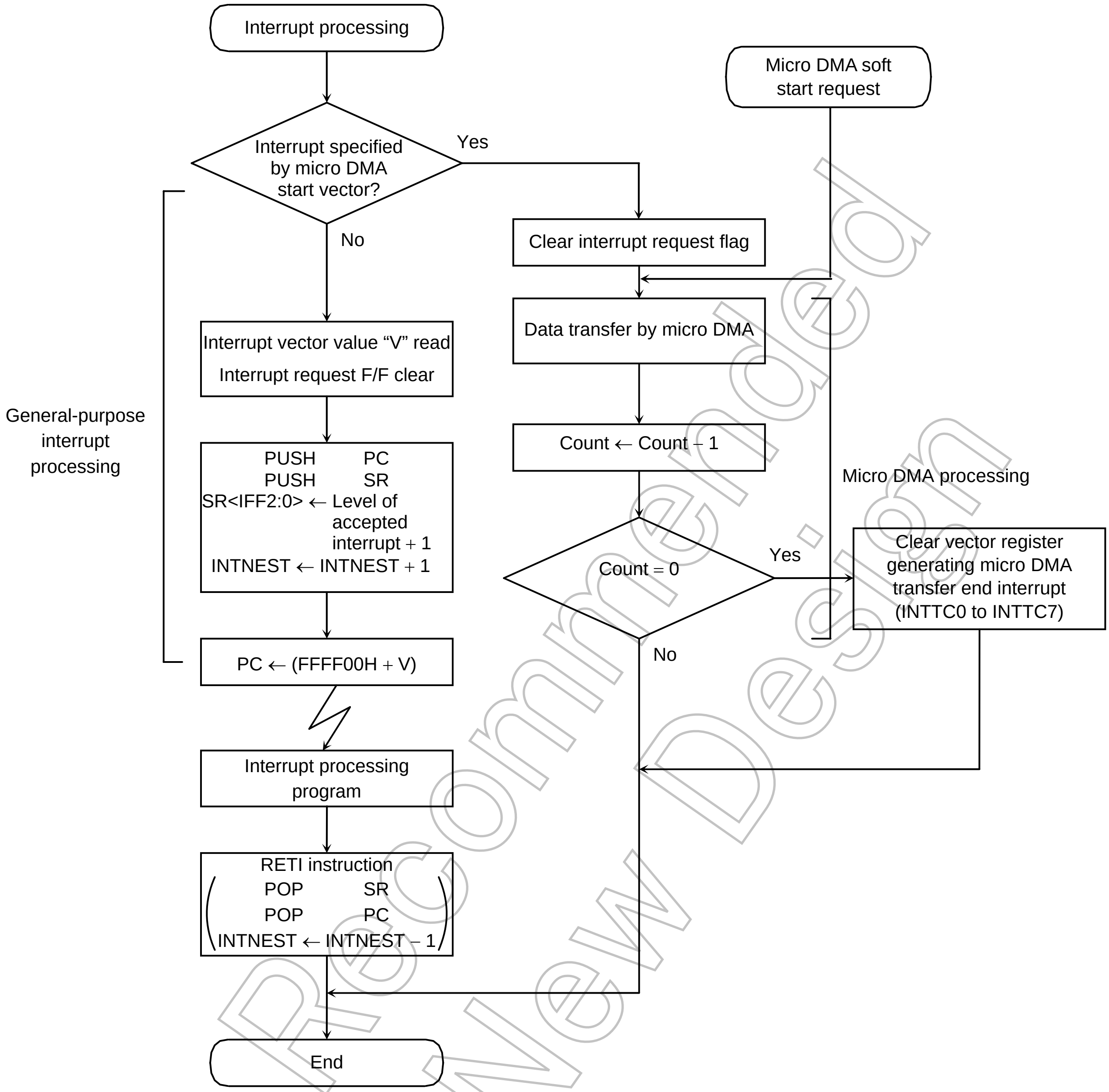


Figure 3.4.1 Interrupt and Micro DMA Processing Sequence

3.4.1 General-purpose Interrupt Processing

When the CPU accepts an interrupt, it usually performs the following sequence of operations. However, in the case of software interrupts and illegal instruction interrupts generated by the CPU, the CPU skips steps (1) and (3), and executes only steps (2), (4), and (5).

(1) The CPU reads the interrupt vector from the interrupt controller.
When more than one interrupt with the same priority level has been generated simultaneously, the interrupt controller generates an interrupt vector in accordance with the default priority and clears the interrupt requests.
(The default priority is determined as follows: The smaller the vector value, the higher the priority.)

(2) The CPU pushes the program counter (PC) and status register (SR) onto the top of the stack (Pointed to by XSP).

(3) The CPU sets the value of the CPU's interrupt mask register <IFF2:0> to the priority level for the accepted interrupt plus 1. However, if the priority level for the accepted interrupt is 7, the register's value is set to 7.

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

(5) The CPU jumps to the address given by adding the contents of address FFFF00H + the interrupt vector, then starts the interrupt processing routine.

On completion of interrupt processing, the RETI instruction is used to return control to the main routine. RETI restores the contents of the program counter and the status register from the stack and decrements the interrupt nesting counter INTNEST by 1.

Non-maskable interrupts cannot be disabled by a user program. Maskable interrupts, however, can be enabled or disabled by a user program. A program can set the priority level for each interrupt source. (A priority level setting of 0 or 7 will disable an interrupt request.) If an interrupt request is received for an interrupt with a priority level equal to or greater than the value set in the CPU interrupt mask register <IFF2:0>, the CPU will accept the interrupt. The CPU interrupt mask register <IFF2:0> is then set to the value of the priority level for the accepted interrupt plus 1.

If during interrupt processing, an interrupt is generated with a higher priority than the interrupt currently being processed, or if, during the processing of a non-maskable interrupt processing, a non-maskable interrupt request is generated from another source, the CPU will suspend the routine which it is currently executing and accept the new interrupt. When processing of the new interrupt has been completed, the CPU will resume processing of the suspended interrupt.

If the CPU receives another interrupt request while performing processing steps (1) to (5), the new interrupt will be sampled immediately after execution of the first instruction of its interrupt processing routine. Specifying DI as the start instruction disables nesting of maskable interrupts.

A reset, initializes the interrupt mask register <IFF2:0> to 111, disabling all maskable interrupts.

Table 3.4.1 shows the TMP92C820 interrupt vectors and micro DMA start vectors. FFFF00H to FFFFFFH (256 bytes) is designated as the interrupt vector area.

Table 3.4.1 TMP92C820 Interrupt Vectors and Micro DMA Start Vectors (1/2)

Default Priority	Type	Interrupt Source and Source of Micro DMA Request	Vector Value	Address Refer to Vector	Micro DMA Start Vector
1	Non maskable	Reset or [SWI0] instruction	0000H	FFFF00H	
2		[SWI1] instruction	0004H	FFFF04H	
3		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		(Reserved)	0020H	FFFF20H	
10		INTWD: Watchdog timer	0024H	FFFF24H	
–	Maskable	Micro DMA	–	–	– (Note 1)
11		INT0: INT0 pin input	0028H	FFFF28H	0AH (Note 2)
12		INT1: INT1 pin input	002CH	FFFF2CH	0BH
13		INT2: INT2 pin input	0030H	FFFF30H	0CH
14		INT3: INT3 pin input	0034H	FFFF34H	0DH
15		(Reserved)	0038H	FFFF38H	0EH
16		INTALM0: ALM0 (8 kHz)	003CH	FFFF3CH	0FH
17		INTALM1: ALM1 (512 Hz)	0040H	FFFF40H	10H
18		INTALM2: ALM2 (64 Hz)	0044H	FFFF44H	11H
19		INTALM3: ALM3 (2 Hz)	0048H	FFFF48H	12H
20		INTALM4: ALM4 (1 Hz)	004CH	FFFF4CH	13H
21		INTP0: Protect 0 (WR to SFR)	0050H	FFFF50H	14H
22		(Reserved)	0054H	FFFF54H	15H
23		INTTA0: 8-bit timer 0	0058H	FFFF58H	16H
24		INTTA1: 8-bit timer 1	005CH	FFFF5CH	17H
25		INTTA2: 8-bit timer 2	0060H	FFFF60H	18H
26		INTTA3: 8-bit timer 3	0064H	FFFF64H	19H
27		INTTB0: 16-bit timer 0	0068H	FFFF68H	1AH
28		INTTB1: 16-bit timer 0	006CH	FFFF6CH	1BH
29		INTKEY: Key wakeup	0070H	FFFF70H	1CH
30		INTRTC: RTC (Alarm interrupt)	0074H	FFFF74H	1DH
31		INTTBO0: 16-bit timer 0 (Overflow)	0078H	FFFF78H	1EH
32		INTLCD: LCDC/LP pin	007CH	FFFF7CH	1FH
33		INTRX0: Serial receive (Channel 0)	0080H	FFFF80H	20H (Note 2)
34		INTTX0: Serial transmission (Channel 0)	0084H	FFFF84H	21H
35		INTRX1: Serial receive (Channel 1)	0088H	FFFF88H	22H (Note 2)
36		INTTX1: Serial transmission (Channel 1)	008CH	FFFF8CH	23H
37		INTRX2: Serial receive (Channel 2)	0090H	FFFF90H	24H (Note 2)
38		INTTX2: Serial transmission (Channel 2)	0094H	FFFF94H	25H
39		(Reserved)	0098H	FFFF98H	26H
40		(Reserved)	009CH	FFFF9CH	27H
41		(Reserved)	00A0H	FFFFA0H	28H
42		(Reserved)	00A4H	FFFFA4H	29H
43		(Reserved)	00A8H	FFFFA8H	2AH
44		(Reserved)	00ACH	FFFFACH	2BH
45		(Reserved)	00B0H	FFFFB0H	2CH
46		(Reserved)	00B4H	FFFFB4H	2DH
47		(Reserved)	00B8H	FFFFB8H	2EH
48		INTSBE0: SBI I^2C bus transfer end (Channel 0)	00BCH	FFFFBCH	2FH
49		(Reserved)	00C0H	FFFFC0H	30H
50		(Reserved)	00C4H	FFFFC4H	31H

Table 3.4.1 TMP92C820 Interrupt Vectors and Micro DMA Start Vectors (2/2)

Default Priority	Type	Interrupt Source and Source of Micro DMA Request	Vector Value	Address Refer to Vector	Micro DMA Start Vector
51	Maskable	(Reserved)	00C8H	FFFFC8H	32H
52		INTAD: AD conversion end	00CCH	FFFFCCH	33H
53		INTTC0: Micro DMA end (Channel 0)	00D0H	FFFFD0H	34H
54		INTTC1: Micro DMA end (Channel 1)	00D4H	FFFFD4H	35H
55		INTTC2: Micro DMA end (Channel 2)	00D8H	FFFFD8H	36H
56		INTTC3: Micro DMA end (Channel 3)	00DCH	FFFFDCH	37H
57		INTTC4: Micro DMA end (Channel 4)	00E0H	FFFFE0H	38H
58		INTTC5: Micro DMA end (Channel 5)	00E4H	FFFFE4H	39H
59		INTTC6: Micro DMA end (Channel 6)	00E8H	FFFFE8H	3AH
60		INTTC7: Micro DMA end (Channel 7)	00ECH	FFFFECH	3BH
– to –		(Reserved)	00F0H : 00FCH	FFFFF0H : FFFFFCH	– –

Note 1: Micro DMA default priority.
Micro DMA initiation takes priority over other maskable interrupt.

Note 2: When initiating micro DMA, set at edge detect mode.

3.4.2 Micro DMA processing

In addition to general-purpose interrupt processing, the TMP92C820 also includes a micro DMA function. Micro DMA processing for interrupt requests set by micro DMA is performed at the highest priority level for maskable interrupts (Level 6), regardless of the priority level of the interrupt source.

Because the micro DMA function is implemented though the CPU, when the CPU is placed in a state of standby by HALT instruction, the requirements of the micro DMA will be ignored (Pending).

Micro DMA supports 8 channels and can be transferred continuously by specifying the micro DMA burst function as below.

(1) Micro DMA operation

When an interrupt request is generated by an interrupt source specified by the micro DMA start vector register, the micro DMA triggers a micro DMA request to the CPU at interrupt priority level 6 and starts processing the request. The eight micro DMA channels allow micro DMA processing to be set for up to 8 types of interrupt at once.

When micro DMA is accepted, the interrupt request flip-flop assigned to that channel is cleared. Data in one-byte, two-byte or four-byte blocks, is automatically transferred at once from the transfer source address to the transfer destination address set in the control register, and the transfer counter is decremented by 1. If the value of the counter after it has been decremented is not 0, DMA processing ends with no change in the value of the micro DMA start vector register. If the value of the decremented counter is 0, a micro DMA transfer end interrupt (INTTC0 to INTTC7) is sent from the CPU to the interrupt controller. In addition, the micro DMA start vector register is cleared to 0, the next micro DMA operation is disabled and micro DMA processing terminates.

If micro DMA requests are set simultaneously for more than one channel, priority is not based on the interrupt priority level but on the channel number: The lower the channel number, the higher the priority (Channel 0 thus has the highest priority and channel 7 the lowest).

If an interrupt request is triggered for the interrupt source in use during the interval between the time at which the micro DMA start vector is cleared and the next setting, general-purpose interrupt processing is performed at the interrupt level set. Therefore, if the interrupt is only being used to initiate micro DMA (and not as a general-purpose interrupt), the interrupt level should first be set to 0 (j.e, interrupt requests should be disabled).

If micro DMA and general-purpose interrupts are being used together as described above, the level of the interrupt which is being used to initiate micro DMA processing should first be set to a lower value than all the other interrupt levels. (Note) In this case, edge-triggered interrupts are the only kinds of general interrupts which can be accepted.

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

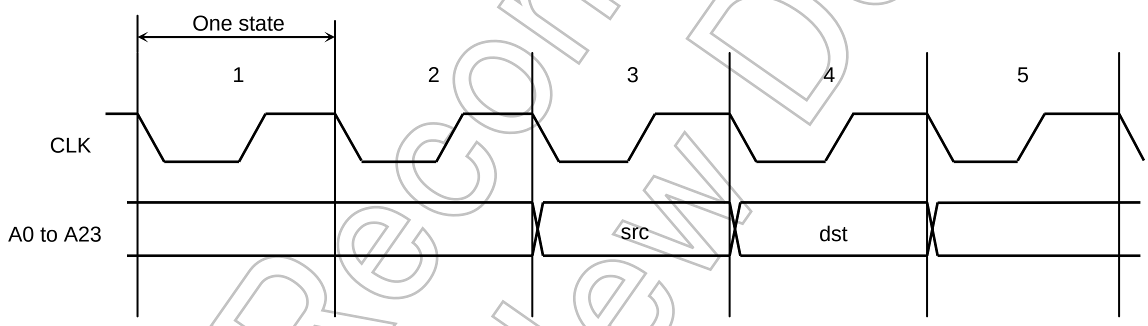
Although the control registers used for setting the transfer source and transfer destination addresses are 32 bits wide, this type of register can only output 24-bit addresses. Accordingly, micro DMA can only access 16 Mbytes (The upper 8 bits of a 32-bit address are not valid).

Three micro DMA transfer modes are supported: One-byte transfer, two-byte (One word) transfers and four-byte transfers. After a transfer in any mode, the transfer source and transfer destination addresses will either be incremented or decremented, or will remain unchanged. This simplifies the transfer of data from I/O to memory, from memory to I/O, and from I/O to I/O. For details of the various transfer modes, see section 3.4.2 (4) "Detailed description of the transfer mode register".

Since a transfer counter is a 16-bit counter, up to 65536 micro DMA processing operations can be performed per interrupt source (Provided that the transfer counter for the source is initially set to 0000H).

Micro DMA processing can be initiated by any one of 34 different interrupts – the 33 interrupts shown in the micro DMA start vectors in Table 3.4.1 and a micro DMA soft start.

Figure 3.4.2 shows a 2-byte transfer carried out using a micro DMA cycle in transfer destination address INC mode (Micro DMA transfers are the same in every mode except counter mode). (The conditions for this cycle are as follows: external 8-bit bus, 0 waits, and even-numbered transfer source and transfer destination addresses).



Note: In fact, src and dst address are not output to A23 to A0 pins because they are internal RAM address

States 1 and 2: Instruction fetch cycle (Prefetches the next instruction code)
If the instruction queue buffer is FULL, this cycle becomes a dummy cycle.
State 3: Micro DMA read cycle.
State 4: Micro DMA write cycle.
State 5: (The same as in state 1, 2.)

Figure 3.4.2 Timing for Micro DMA Cycle

(2) Soft start function

The TMP92C820 can initiate micro DMA either with an interrupt or by using the micro DMA soft start function, in which micro DMA is initiated by a Write cycle which writes to the register DMAR.

Writing 1 to any bit of the register DMAR causes micro DMA to be performed once. (If write “0” to each bit, micro DMA doesn’t operate). On completion of the transfer, the bits of DMAR which support the end channel are automatically cleared to 0.

Only one channel can be set for DMA request at once. (Do not write “1” to plural bits.)

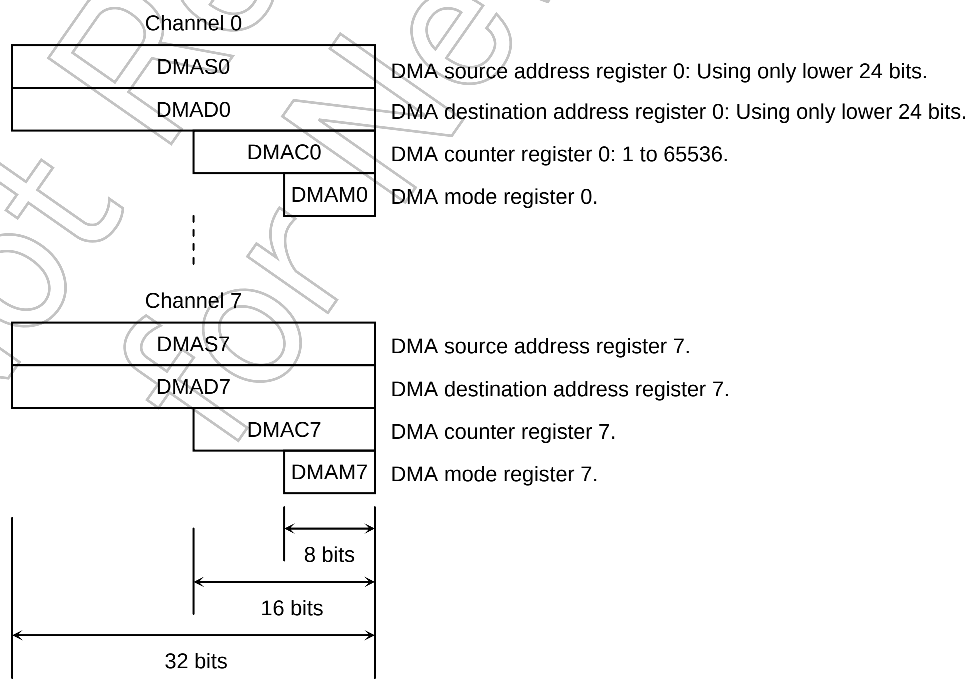
When writing again 1 to the DMAR register, check whether the bit is “0” before writing “1”. If read “1”, micro DMA transfer isn’t started yet.

When a burst is specified by the DMAB register, data is transferred continuously from the initiation of micro DMA until the value in the micro DMA transfer counter is 0. If execatee soft start during micro DMA transfer by interrupt source, micro DMA transfer counter doesn’t change. Don’t use Read-modify-write instruction to avoid writign to other bits by mistake.

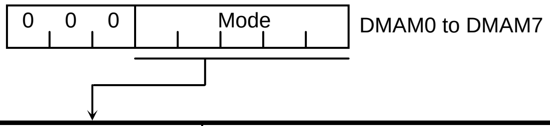
Symbol	Name	Address	7	6	5	4	3	2	1	0
DMAR	DMA request	109H (Prohibit RMW)	DREQ7	DREQ6	DREQ5	DREQ4	DREQ3	DREQ2	DREQ1	DREQ0
			R/W							
			0	0	0	0	0	0	0	0
			1: DMA request in software							

(3) Transfer control registers

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



(4) Detailed description of the transfer mode register



DMAM [4:0]	Mode Description	Execution Time
000ZZ	Destination INC mode (DMADn+) ← (DMASn) DMACn ← DMACn – 1 if DMACn = 0 then INTTCn	5 states
001ZZ	Destination DEC mode (DMADn–) ← (DMASn) DMACn ← DMACn – 1 if DMACn = 0 then INTTCn	5 states
010ZZ	Source INC mode (DMADn) ← (DMASn+) DMACn ← DMACn – 1 if DMACn = 0 then INTTCn	5 states
011ZZ	Source DEC mode (DMADn) ← (DMASn–) DMACn ← DMACn – 1 if DMACn = 0 then INTTCn	5 states
100ZZ	Source and destination INC mode (DMADn+) ← (DMASn+) DMACn ← DMACn – 1 If DMACn = 0 then INTTCn	6 states
101ZZ	Source and destination DEC mode (DMADn–) ← (DMASn–) DMACn ← DMACn – 1 If DMACn = 0 then INTTCn	6 states
110ZZ	Destination and fixed mode (DMADn) ← (DMASn) DMACn ← DMACn – 1 If DMACn = 0 then INTTCn	5 states
11100	Counter mode DMASn ← DMASn + 1 DMACn ← DMACn – 1 if DMACn = 0 then INTTCn	5 states

ZZ: 00 = 1-byte transfer
01 = 2-byte transfer
10 = 4-byte transfer
11 = Reserved

Note 1: The execution time is measured at 1 states = 50 ns (Operation at internal 20 MHz).

Note 2: n stands for the micro DMA channel number (0 to 7).

DMADn+/DMASn+: Post increment (Register value is incremented after transfer).

DMADn–/DMASn–: Post decrement (Register value is decremented after transfer).

"I/O" signifies fixed memory addresses; "memory" signifies incremented or decremented memory addresses.

Note2: The transfer mode register should not be set to any value other than those listed above.

3.4.3 Interrupt Controller Operation

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

For each of the 52 interrupt channels there is an interrupt request flag (consisting of a flip-flop), an interrupt priority setting register and a micro DMA start vector register. The interrupt request flag latches interrupt requests from the peripherals. The flag is cleared to zero in the following cases: when a reset occurs, when the CPU reads the channel vector of an interrupt it has received, when the CPU receives a micro DMA request (when micro DMA is set), when a micro DMA burst transfer is terminated, and when an instruction that clears the interrupt for that channel is executed (by writing a micro DMA start vector to the INTCLR register).

An interrupt priority can be set independently for each interrupt source by writing the priority to the interrupt priority setting register (e.g., INTE0AD or INTE12). Six interrupt priorities levels (1 to 6) are provided. Setting an interrupt source's priority level to 0 (or 7) disables interrupt requests from that source.

The priority of non-maskable interrupt (Watchdog timer interrupts) is fixed at 7. If more than one interrupt request with a given priority level are generated simultaneously, the default priority (The interrupt with the lowest priority or, in other words, the interrupt with the lowest vector value) is used to determine which interrupt request is accepted first.

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

If several interrupts are generated simultaneously, the interrupt controller sends the interrupt request for the interrupt with the highest priority and the interrupt's vector address to the CPU. The CPU compares the mask value set in <IFF2:0> of the status register (SR) with the priority level of the requested interrupt; if the latter is higher, the interrupt is accepted. Then the CPU sets SR<IFF2:0> to the priority level of the accepted interrupt + 1. Hence, during processing of the accepted interrupt, new interrupt requests with a priority value equal to or higher than the value set in SR<IFF2:0> (e.g., interrupts with a priority higher than the interrupt being processed) will be accepted.

When interrupt processing has been completed (e.g., after execution of a RETI instruction), the CPU restores to SR<IFF2:0> the priority value which was saved on the stack before the interrupt was generated.

The interrupt controller also includes eight registers which are used to store the micro DMA start vector. Writing the start vector of the interrupt source for the micro DMA processing (See Table 3.4.1 and Table 3.4.), enables the corresponding interrupt to be processed by micro DMA processing. The values must be set in the micro DMA parameter registers (e.g., DMAS and DMAD) prior to micro DMA processing.

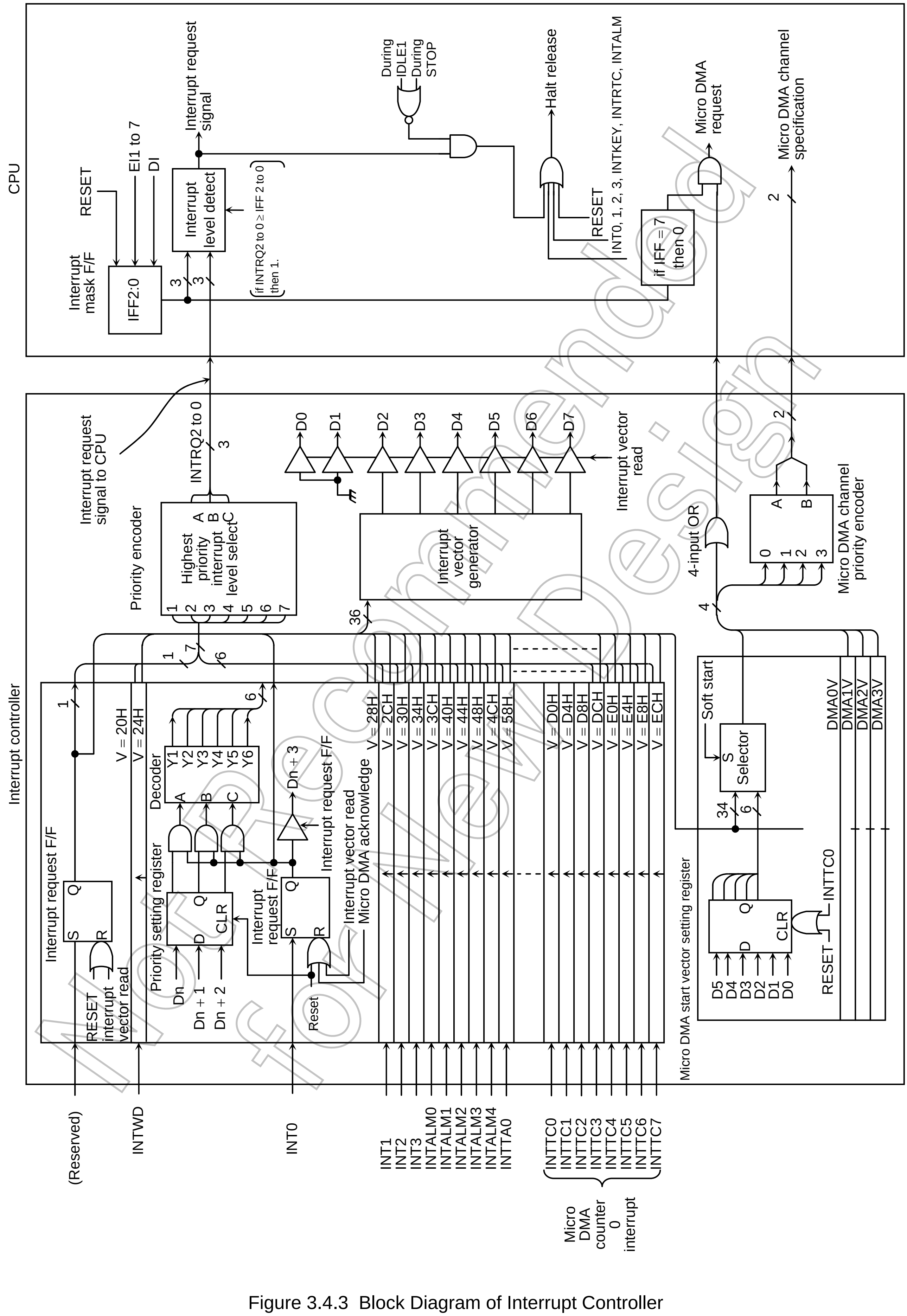


Figure 3.4.3 Block Diagram of Interrupt Controller

(1) Interrupt priority setting registers

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE0AD	INT0& INTAD enable	F0H	INTAD				INT0			
			IADC	IADM2	IADM1	IADM0	I0C	I0M2	I0M1	I0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTE12	INT1&INT2 enable	D0H	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	INT3 enable	D1H	–				INT3			
			–	–	–	–	I3C	I3M2	I3M1	I3M0
			–	–			R	R/W		
			Always write “0”.				0	0	0	0
INTETA01	INTTA0& INTTA1 enable	D4H	INTTA1 (TMRA1)				INTTA0 (TMRA0)			
			ITA1C	ITA1M2	ITA1M1	ITA1M0	ITA0C	ITA0M2	ITA0M1	ITA0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETA23	INTTA2& INTTA3 enable	D5H	INTAT3 (TMRA3)				INTAT2 (TMRA2)			
			ITA3C	ITA3M2	ITA3M1	ITA3M0	ITA2C	ITA2M2	ITA2M1	ITA2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETB01	INTTB0& INTTB1 enable	D8H	INTTB1 (TMRB1)				INTTB0 (TMRB0)			
			ITB1C	ITB1M2	ITB1M1	ITB1M0	ITB0C	ITB0M2	ITB0M1	ITB0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETBO0	INTTBO0 (Overflow) enable	DAH	–				INTTBO0			
			–	–	–	–	ITBO0C	ITBO0M2	ITBO0M1	ITBO0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTES0	INTRX0& INTTX0 enable	DBH	INTTX0				INTRX0			
			ITX0C	ITX0M2	ITX0M1	ITX0M0	IRX0C	IRX0M2	IRX0M1	IRX0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTES1	INTRX1& INTTX1 enable	DCH	INTTX1				INTRX1			
			ITX1C	ITX1M2	ITX1M1	ITX1M0	IRX1C	IRX1M2	IRX1M1	IRX1M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTESB0	INTSBE0 enable	E3H	–				INTSBE0			
			–	–	–	–	ISBE0C	ISBE0M2	ISBE0M1	ISBE0M0
			–	–			R	R/W		
			Always write “0”.				0	0	0	0
INTEALM 01	INTALM0& INTALM1 enable	E5H	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& INTALM3 enable	E6H	INTALM3				INTALM2			
			IA3C	IA3M2	IA3M1	IA3M0	IA2C	IA2M2	IA2M1	IA2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTEALM4	INTALM4 enable	E7H	–				INTALM4			
			–	–	–	–	IA4C	IA4M2	IA4M1	IA4M0
			–	–			R	R/W		
			Always write “0”.				0	0	0	0
INTERTC	INTRTC enable	E8H	–				INTRTC			
			–	–	–	–	IRC	IRM2	IRM1	IRM0
			–	–			R	R/W		
			Always write “0”.				0	0	0	0
INTECKEY	INTKEY enable	E9H	–				INTKEY			
			–	–	–	–	IKC	IKM2	IKM1	IKM0
			–	–			R	R/W		
			Always write “0”.				0	0	0	0
INTLCD	INTLCD enable	EAH	–				INTLCD			
			–	–	–	–	ILCD1C	ILCDM2	ILCDM1	ILCDM0
			–	–			R	R/W		
			Always write “0”.				0	0	0	0
INTES2	INTRX2& INTTX2 enable	EDH	INTTX2				INTRX2			
			ITX2C	ITX2M2	ITX2M1	ITX2M0	IRX2C	IRX2M2	IRX2M1	IRX2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTEP0	INTP0 enable	EEH	–				INTP0			
			–	–	–	–	IP0C	IP0M2	IP0M1	IP0M0
			–	–			R	R/W		
			Always write “0”.				0	0	0	0

Interrupt request flag

IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Disables interrupt requests
0	0	1	Sets interrupt priority level to 1
0	1	0	Sets interrupt priority level to 2
0	1	1	Sets interrupt priority level to 3
1	0	0	Sets interrupt priority level to 4
1	0	1	Sets interrupt priority level to 5
1	1	0	Sets interrupt priority level to 6
1	1	1	Disables interrupt requests

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTETC01	INTTC0& INTTC1 enable	F1H	INTTC1 (DMA1)				INTTC0 (DMA0)			
			ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC23	INTTC2& INTTC3 enable	F2H	INTTC3 (DMA3)				INTTC2 (DMA2)			
			ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC45	INTTC4& INTTC5 enable	F3H	INTTC5 (DMA5)				INTTC4 (DMA4)			
			ITC5C	ITC5M2	ITC5M1	ITC5M0	ITC4C	ITC4M2	ITC4M1	ITC4M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTETC67	INTTC6& INTTC7 enable	F4H	INTTC7 (DMA7)				INTTC6 (DMA6)			
			ITC7C	ITC7M2	ITC7M1	ITC7M0	ITC6C	ITC6M2	ITC6M1	ITC6M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
INTWDT	INTWD	F7H	–				INTWD			
			–	–	–	–	ITCWD	–	–	–
			–	–			R	–		
			Always write “0”.				0	–	–	–

Interrupt request flag

IxxM2	IxxM1	IxxM0	Function (Write)
0	0	0	Disables interrupt requests
0	0	1	Sets interrupt priority level to 1
0	1	0	Sets interrupt priority level to 2
0	1	1	Sets interrupt priority level to 3
1	0	0	Sets interrupt priority level to 4
1	0	1	Sets interrupt priority level to 5
1	1	0	Sets interrupt priority level to 6
1	1	1	Disables interrupt requests

(2) External interrupt control

Symbol	Name	Address	7	6	5	4	3	2	1	0
IIMC	Interrupt input mode control	F6H (Prohibit RMW)			I3EDGE	I2EDGE	I1EDGE	I0EDGE	I0LE	–
					W	W	W	W	R/W	R/W
					0	0	0	0	0	0
					INT3EDGE 0: Rising 1: Falling	INT2EDGE 0: Rising 1: Falling	INT1EDGE 0: Rising 1: Falling	INT0EDGE 0: Rising 1: Falling	INT0 0: Edge mode 1: Level mode	Always write “0”.

*INT0 level enable

0	Edge detect INT
1	“H” level INT

Note 1: Disable INT0 request before changing INT0 pin mode from level sense to edge sense.

Setting example:

```
DI
LD    (IIMC), XXXXXX0 - B    ;  Switches from level to edge.
LD    (INTCLR), 0AH          ;  Clears interrupt request flag.
EI
```

Note 2: X: Don’t care, –: No change

Note 3: See electrical characteristics in section 4 for external interrupt input pulse width.

Settings of External Interrupt Pin Function

Interrupt	Pin Name	Mode	Setting Method
INT0	PC3	Rising edge	IIMC<I0LE> = 0, INT0EDGE = 0
		Falling edge	IIMC<I0LE> = 0, INT0EDGE = 1
		High level	IIMC<I0LE> = 1
INT1	PC1	Rising edge	INT1EDGE = 0
		Falling edge	INT1EDGE = 1
INT2	PC5	Rising edge	INT2EDGE = 0
		Falling edge	INT2EDGE = 1
INT3	PC6	Rising edge	INT3EDGE = 0
		Falling edge	INT3EDGE = 1

(3) SIO receive interrupt control

Symbol	Name	Address	7	6	5	4	3	2	1	0
SIMC	SIO interrupt mode control	F5H (Prohibit RMW)						IR2LE	IR1LE	IR0LE
								W	W	W
								1	1	1
								0: INTRX2 edge mode 1: INTRX2 level mode	0: INTRX1 edge mode 1: INTRX1 level mode	0: INTRX0 edge mode 1: INTRX0 level mode

INTRX0 rising edge enable

0	Rising edge detect INTRX0
1	"H" level INTRX0

INTRX1 level enable

0	Rising edge detect INTRX1
1	"H" level INTRX1

INTRX2 level enable

0	Rising edge detect INTRX2
1	"H" level INTRX2

(4) Interrupt request flag clear register

The interrupt request flag is cleared by writing the appropriate micro DMA start vector, as given in Table 3.4.1 to the register INTCLR.

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

INTCLR ← 0AH ; Clears interrupt request flag INT0.

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTCLR	Interrupt clear control	F8H (Prohibit RMW)	CLRV7	CLRV6	CLRV5	CLRV4	CLRV3	CLRV2	CLRV1	CLRV0
			W							
			0	0	0	0	0	0	0	0
			Interrupt vector							

(5) Micro DMA start vector registers

These registers assign micro DMA processing to an sets which source corresponds to DMA. The interrupt source whose micro DMA start vector value matches the vector set in one of these registers is designated as the micro DMA start source.

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

If the same vector is set in the micro DMA start vector registers of more than one channel, the lowest numbered channel takes priority.

Accordingly, if the same vector is set in the micro DMA start vector registers for two different channels, the interrupt generated on the lower-numbered channel is executed until micro DMA transfer is complete. If the micro DMA start vector for this channel has not been set in the channel's micro DMA start vector register again, micro DMA transfer for the higher-numbered channel will be commenced. (This process is known as micro DMA chaining.)

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMA0V	DMA0 start vector	100H			DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
					R/W					
					0	0	0	0	0	0
					DMA0 start vector					
DMA1V	DMA1 start vector	101H			DMA1V5	DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
					R/W					
					0	0	0	0	0	0
					DMA1 start vector					
DMA2V	DMA2 start vector	102H			DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
					R/W					
					0	0	0	0	0	0
					DMA2 start vector					
DMA3V	DMA3 start vector	103H			DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
					R/W					
					0	0	0	0	0	0
					DMA3 start vector					
DMA4V	DMA4 start vector	104H			DMA4V5	DMA4V4	DMA4V3	DMA4V2	DMA4V1	DMA4V0
					R/W					
					0	0	0	0	0	0
					DMA4 start vector					
DMA5V	DMA5 start vector	105H			DMA5V5	DMA5V4	DMA5V3	DMA5V2	DMA5V1	DMA5V0
					R/W					
					0	0	0	0	0	0
					DMA5 start vector					
DMA6V	DMA6 start vector	106H			DMA6V5	DMA6V4	DMA6V3	DMA6V2	DMA6V1	DMA6V0
					R/W					
					0	0	0	0	0	0
					DMA6 start vector					
DMA7V	DMA7 start vector	107H			DMA7V5	DMA7V4	DMA7V3	DMA7V2	DMA7V1	DMA7V0
					R/W					
					0	0	0	0	0	0
					DMA7 start vector					

(6) Specification of a micro DMA burst

Specifying the micro DMA burst function causes micro DMA transfer, once started, to continue until the value in the transfer counter register reaches zero. Setting any of the bits in the register DMAB which correspond to a micro DMA channel (as shown below) to 1 specifies that any micro DMA transfer on that channel will be a burst transfer.

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMAB	DMA burst	108H	DBST7	DBST6	DBST5	DBST4	DBST3	DBST2	DBST1	DBST0
			R/W							
			0	0	0	0	0	0	0	0
			1: DMA request on Burst mode							

(7) Notes

The instruction execution unit and the bus interface unit in this CPU operate independently. Therefore, if immediately before an interrupt is generated, the CPU fetches an instruction which clears the corresponding interrupt request flag, the CPU may execute this instruction in between accepting the interrupt and reading the interrupt vector. In this case, the CPU will read the default vector 0004H and jump to interrupt vector address FFFF04H.

To avoid this, an instruction which clears an interrupt request flag should always be preceded by a DI instruction. And in the case of setting an interrupt enable again by EI instruction after the execution of clearing instruction, execute EI instruction after clearing and more than 3-instructions (e.g., “NOP” × 3 times). If placed EI instruction without waiting NOP instruction after execution of clearing instruction, interrupt will be enable before request flag is cleared.

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

In addition, please note that the following two circuits are exceptional and demand special attention.

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

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

INTRX	In edge mode (The register SIMC<IRxLE> set to “0”), the interrupt request flip-flop can only be cleared by a reset or by reading the serial channel receive buffer. It cannot be cleared by writing INTCLR register.

Note: The following instructions or pin input state changes are equivalent to instructions which 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 Function of Ports

TMP92C820 has I/O port pins that are shown in Table 3.5.1. In addition to functioning as general-purpose I/O ports, these pins are also used by internal CPU and I/O functions. Table 3.5.2 lists I/O registers and their specifications.

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	I/O	R	I/O Setting	Pin Name for Built-in Function
Port 1	P10 to P17	8	I/O	–	Bit	D8 to D15
Port 2	P20 to P27	8	I/O	–	Bit	D16 to D23
Port 3	P30 to P37	8	I/O	–	Bit	D24 to D31
Port 4	P40 to P47	8	I/O*	–	Bit*	A0 to A7
Port 5	P50 to P57	8	I/O*	–	Bit*	A8 to A15
Port 6	P60 to P67	8	I/O*	–	Bit*	A16 to A23
Port 7	P70	1	Output	–	(Fixed)	$\overline{RD}$
	P71	1	Output	–	(Fixed)	$\overline{WRLL}$
	P72	1	Output	–	(Fixed)	$\overline{WRLU}$
	P73	1	Output	–	(Fixed)	$\overline{WRUL}$
	P74	1	Output	–	(Fixed)	$\overline{WRUU}$
	P75	1	Output	–	(Fixed)	R/ $\overline{W}$
	P76	1	I/O	–	Bit	$\overline{WAIT}$
Port 8	P80	1	Output	–	(Fixed)	$\overline{CS0}$, $\overline{SDCSH}$
	P81	1	Output	–	(Fixed)	$\overline{CS1}$, $\overline{SDCSL}$
	P82	1	Output	–	(Fixed)	$\overline{CS2}$, $\overline{CS2A}$
	P83	1	Output	–	(Fixed)	$\overline{CS3}$
	P84	1	Output	–	(Fixed)	EA24, $\overline{CS2B}$
	P85	1	Output	–	(Fixed)	EA25, $\overline{CS2C}$
	P86	1	Output	–	(Fixed)	$\overline{CS2D}$
	P87	1	Output	–	(Fixed)	SDCLK
Port 9	P90	1	I/O	–	Bit	SCK
	P91	1	I/O	–	Bit	SO, SDA
	P92	1	I/O	–	Bit	SI, SCL
	P93	1	I/O	–	Bit	$\overline{CS2E}$
	P94	1	I/O	–	Bit	$\overline{CS2F}$
	P95	1	I/O	–	Bit	$\overline{CS2G}$, TXD2
	P96	1	I/O	–	Bit	$\overline{CSEXA}$, RXD2
Port A	PA0 to PA7	8	Input	U	(Fixed)	KI0 to KI7
Port C	PC0	1	I/O	–	Bit	TA0IN
	PC1	1	I/O	–	Bit	INT1, TA1OUT
	PC3	1	I/O	–	Bit	INT0
	PC5	1	I/O	–	Bit	INT2, TA3OUT
	PC6	1	I/O	–	Bit	INT3, TB0OUT0
Port F	PF0	1	I/O	–	Bit	TXD0
	PF1	1	I/O	–	Bit	RXD0
	PF2	1	I/O	–	Bit	SCLK0, $\overline{CTS0}$
	PF3	1	I/O	–	Bit	TXD1
	PF4	1	I/O	–	Bit	RXD1
	PF5	1	I/O	–	Bit	SCLK1, $\overline{CTS1}$

*: When these ports are used as general-purpose I/O port, each bit can be set individually for input or output. However, each bit cannot be set individually for input or output even if 1bit or more bits are used as address bus in same port.
All of general-purpose I/O ports except for port that used as address bus are operated as output port. Please be careful when using this setting.

Table 3.5.1 Port Functions (2/2)

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

Port Name	Pin Name	Number of Pins	I/O	R	I/O Setting	Pin Name for Built-in Function
Port G	PG0 to PG4	5	Input	–	(Fixed)	AN0 to AN4, $\overline{\text{ADTRG}}$ (PG3)
Port J	PJ0	1	Output	–	(Fixed)	$\overline{\text{SDRAS}}$
	PJ1	1	Output	–	(Fixed)	$\overline{\text{SDCAS}}$
	PJ2	1	Output	–	(Fixed)	$\overline{\text{SDWE}}$, $\overline{\text{SRWR}}$
	PJ3	1	Output	–	(Fixed)	$\overline{\text{SDLLDQM}}$, $\overline{\text{SRLLB}}$
	PJ4	1	Output	–	(Fixed)	$\overline{\text{SDLUDQM}}$, $\overline{\text{SRLUB}}$
	PJ5	1	Output	–	(Fixed)	$\overline{\text{SDULDQM}}$, $\overline{\text{SRULB}}$
	PJ6	1	Output	–	(Fixed)	$\overline{\text{SDUUDQM}}$, $\overline{\text{SRUUB}}$
	PJ7	1	Output	–	(Fixed)	SDCKE
Port K	PK0	1	Output	–	(Fixed)	D1BSCP
	PK1	1	Output	–	(Fixed)	D2BLP
	PK2	1	Output	–	(Fixed)	D3BFR
	PK3	1	Output	–	(Fixed)	DLEBCD
	PK4	1	Output	–	(Fixed)	DOFFB
	PK6	1	Output	–	(Fixed)	$\overline{\text{ALARM}}$, $\overline{\text{MLDALM}}$
Port L	PL0 to PL7	8	I/O	–	Bit	LD0 to LD7

Table 3.5.2 I/O Registers and Specifications (1/3)

Port	Pin Name	Specification	I/O Register				
			Pn	PnCR	PnFC	PnFC2	PnODE
Port 1	P10 to P17	Input port	X	0	0	None	None
		Output port	X	1			
		D8 to D15 bus	X	X	1		
Port 2	P20 to P27	Input port	X	0	0	None	None
		Output port	X	1			
		D16 to D23 bus	X	X	1		
Port 3	P30 to P37	Input port	X	0	0	None	None
		Output port	X	1			
		D24 to D31 bus	X	X	1		
Port 4	P40 to P47	Input port*	X	0*	0	None	None
		Output port*	X	1*			
		A0 to A7 output	X	0	1		
Port 5	P50 to P57	Input port*	X	0*	0	None	None
		Output port*	X	1*			
		A8 to A15 output	X	0	1		
Port 6	P60 to P67	Input port*	X	0*	0	None	None
		Output port*	X	1*			
		A16 to A23 output	X	0	1		
Port 7	P70 to P75	Output port	X	None	0	None	None
	P70	$\overline{\text{RD}}$ output	X	None	1		
	P71	$\overline{\text{WRLL}}$ output					
	P72	$\overline{\text{WRLU}}$ output					
	P73	$\overline{\text{WRUL}}$ output					
	P74	$\overline{\text{WRUU}}$ output					
	P75	R/ $\overline{\text{W}}$ output					
	P76	Input port	X	0	0		
		Output port	X	1	0		
		$\overline{\text{WAIT}}$ input	X	0	1		
Port 8	P80 to P87	Output port	X	None	0	0	None
	P80	$\overline{\text{CS0}}$ output	X		1	0	
	P81	$\overline{\text{CS1}}$ output	X		1	0	
		$\overline{\text{SDCS}}$ output	X		X	1	
	P82	$\overline{\text{CS2}}$ output	X		1	0	
		$\overline{\text{CS2A}}$ output	X		X	1	
	P83	$\overline{\text{CS3}}$ output	X		1	0	
	P84	EA24 output	X		1	0	
		$\overline{\text{CS2B}}$ output	X		X	1	
	P85	EA25 output	X		1	0	
		$\overline{\text{CS2C}}$ output	X		X	1	
	P86	$\overline{\text{CS2D}}$ output	X		X	1	
	P87	SDCLK output	X		1	0	

X: Don't care

*: When these ports are used as general-purpose I/O port, each bit can be set individually for input or output. However, each bit cannot be set individually for input or output even if 1bit or more bits are used as address bus in same port.
All of general-purpose I/O ports except for port that used as address bus are operated as output port. Please be careful when using this setting.

Table 3.5.2 I/O Registers and Specifications (2/3)

Port	Pin Name	Specification	I/O Register				
			Pn	PnCR	PnFC	PnFC2	PnODE
Port 9	P90 to P96	Input port	X	0	0	None	0
		Output port	X	1	0		0
	P90	SCK input	X	0	0		0
		SCK output	X	X	1		0/1
	P91	SO output	X	1	1		0/1
		SDA	X	X	1		1
	P92	SI input	X	0	0		0
		SCL	X	X	1		1
	P93	$\overline{CS2E}$ output	X	1	1		X
		SSCMD input	X	0	1		X
		SSCMD output	X	0	1		0
		SSCMD (Open drain)	X	0	1		1
	P94	$\overline{CS2F}$ output	X	1	1		X
		SSDAT input	X	0	1		X
		SSDAT output	X	0	1		0
		SSDAT (Open drain)	X	0	1		1
	P95	$\overline{CS2G}$ output	X	1	1		X
		TXD2 output	X	0	1		0
		TXD2 (Open drain)	X	0	1		1
	P96	$\overline{CSEXA}$ output	X	1	1		X
		RXD2 input	X	0	1		X
Port A	PA0 to PA7	Input port	X	None	0	None	None
		KI0 to KI7 input	X		1		
Port C	PC0, PC1, PC3 PC5, PC6	Input port	X	0	0	None	None
		Output port	X	1	0		
	PC0	TA0IN input	X	X	1		
	PC1	TA1OUT output	X	1	1		
		INT1 input	0	0	1		
	PC3	INT0 input	X	0	1		
	PC5	INT2 input	0	0	1		
		TA3OUT	1	1	1		
	PC6	INT3 input	0	0	1		
		TB0OUT0	1	1	1		
Port F	PF0 to PF5	Input port	X	0	0	None	None
		Output port	X	1	0		
	PF0	TXD0	1	0	1		
		TXD0 (Open drain)	1	1	1		
	PF1	RXD0 input	X	0	None		
	PF2	SCLK0 input/output	1	0/1	1		
		$\overline{CTS0}$ input	1	0	1		
	PF3	TXD1	1	0	1		
		TXD1 (Open drain)	1	1	1		
	PF4	RXD1 input	X	0	None		
	PF5	SCLK1 input/output	1	0/1	1		
		$\overline{CTS1}$ input	1	0	1		
Port G	PG0 to PG4	Input port	X	None	None	None	None
		AN0 to AN4 input	X				
	PG3	$\overline{ADTRG}$ input	X				

X: Don't care

Table 3.5.2 I/O Registers and Specifications (3/3)

Port	Pin Name	Specification	I/O Register				
			Pn	PnCR	PnFC	PnFC2	PnODE
Port J	PJ0 to PJ7	Output port	X	None	0	0	None
	PJ0	$\overline{\text{SDRAS}}$ output	X		1	0	
	PJ1	$\overline{\text{SDCAS}}$ output	X		1	0	
	PJ2	$\overline{\text{SDWE}}$ output	X		1	0	
		$\overline{\text{SRWR}}$ output	X		X	1	
	PJ3	$\overline{\text{SDLLDQM}}$ output	X		1	0	
		$\overline{\text{SRLLB}}$ output	X		X	1	
	PJ4	$\overline{\text{SDLUDQM}}$ output	X		1	0	
		$\overline{\text{SRLUB}}$ output	X		X	1	
	PJ5	$\overline{\text{SDULDQM}}$ output	X		1	0	
		$\overline{\text{SRULB}}$ output	X		X	1	
	PJ6	$\overline{\text{SDUUDQM}}$ output	X		1	0	
		$\overline{\text{SRUUB}}$ output	X		X	1	
	PJ7	SDCKE output	X		1	0	
Port K	PK0 to PK6	Output port	X	None	0	None	None
	PK0	D1BSCP output	X		1		
	PK1	D2BLP output	X		1		
	PK2	D3BFR output	X		1		
	PK3	DLEBCD output	X		1		
	PK4	DOFFB output	X		1		
	PK6	$\overline{\text{ALARM}}$ output	1		1		
		$\overline{\text{MLDALM}}$ output	0		1		
Port L	PL0 to PL7	Input port	X	0	0	None	None
		Output port	X	1	0		
		LD0 to LD7 output	X	X	1		

X: Don't care

After a reset the port pins listed below function as general-purpose I/O port pins. A reset sets I/O pins, which can be programmed for either input, or output to be input ports pins. Setting the port pins for internal function use must be done in software.

3.5.1 Port 1 (P10 to P17)

Port 1 is an 8-bit general-purpose I/O port.

Bits can be individually set as either inputs or outputs by control register P1CR and function register P1FC. In addition to functioning as a general-purpose I/O port, port 1 can also function as a data bus (D8 to D15).

AM1	AM0	Function Setting after Reset is Released
0	0	Don't use this setting
0	1	Data bus (D8 to D15)
1	0	Data bus (D8 to D15)
1	1	Don't use this setting

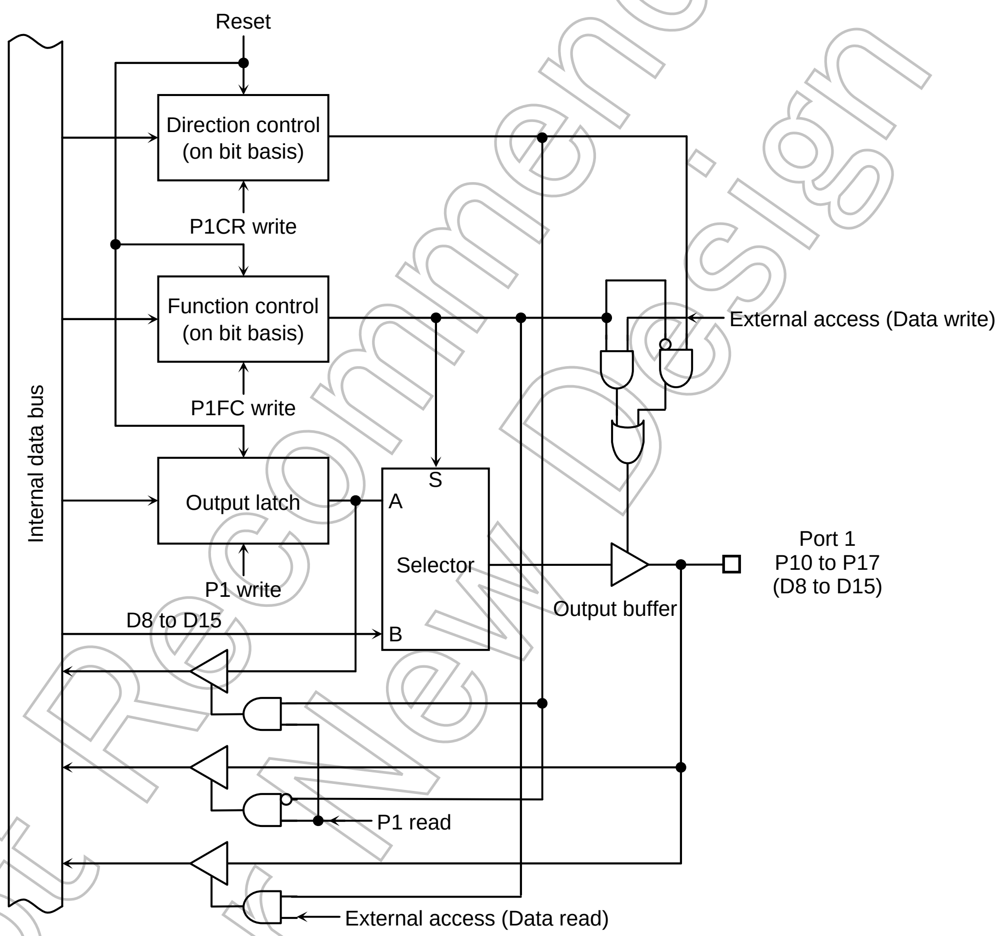


Figure 3.5.1 Port 1

Port 1 Register

P1 (0004H)

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

Port 1 Control Register

P1CR (0006H)

	7	6	5	4	3	2	1	0
Bit symbol	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Refer to port 1 function setting							

Port 1 Function Register

P1FC (0007H)

	7	6	5	4	3	2	1	0
Bit symbol								P1F
Read/Write								W
After reset								1
Function								Refer to port 1 function setting

Note 1:Read-modify-write is prohibited for the registers P1CR and P1FC.
Note 2:<P1XC> show X bit of P1CR register.

Port 1 function register

P1CR<P1xC> \ P1FC<P1xF>	0	1
0	Input port	Data bus (D15 to D8)
1	Output port	

Figure 3.5.2 Register for Port 1

3.5.2 Port 2 (P20 to P27)

Port 2 is an 8-bit general-purpose I/O port.

Bits can be individually set as either inputs or outputs by control register P2CR and function register P2FC. In addition to functioning as a general-purpose I/O port, port 2 can also function as a data bus (D16 to D23).

AM1	AM0	Function Setting after Reset is Released
0	0	Don't use this setting
0	1	Input port
1	0	Data bus (D16 to D23)
1	1	Don't use this setting

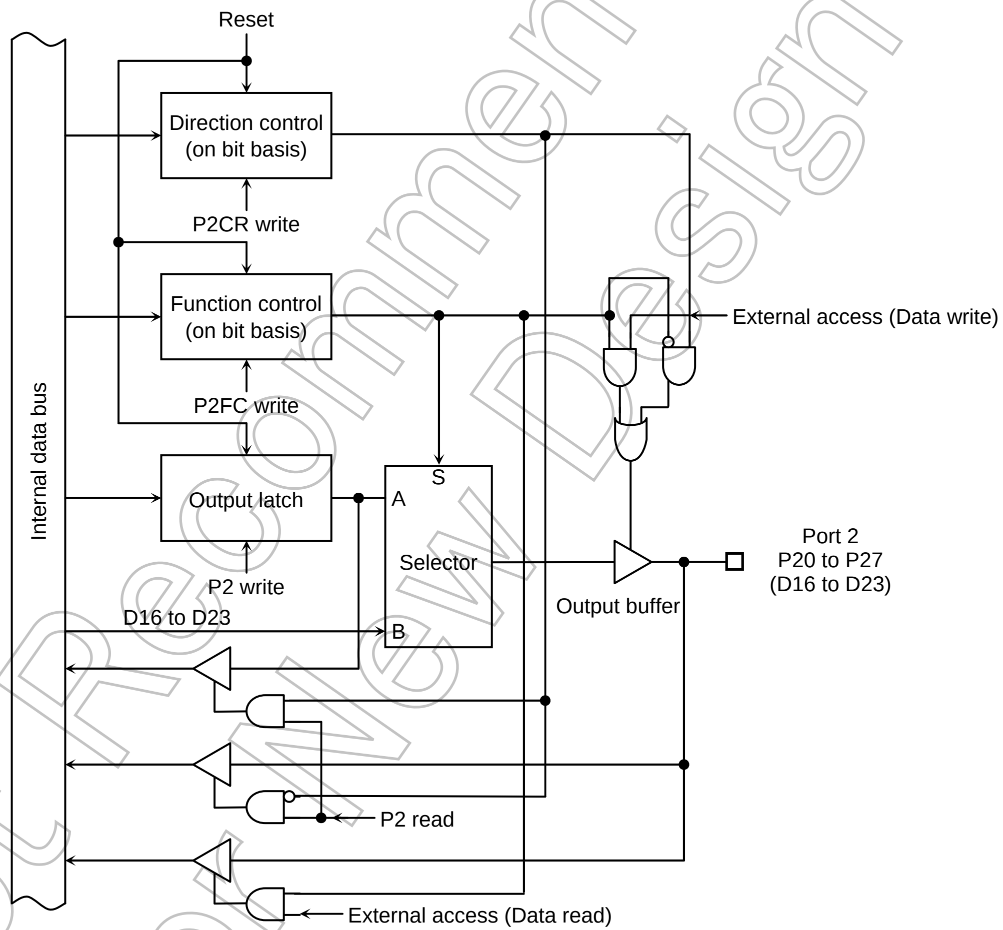


Figure 3.5.3 Port 2

Port 2 Register

P2 (0008H)

	7	6	5	4	3	2	1	0
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

P2CR (000AH)

	7	6	5	4	3	2	1	0
Bit symbol	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output							

Port 2 Function Register

P2FC (000BH)

	7	6	5	4	3	2	1	0
Bit symbol								P2F
Read/Write								W
After reset								0/1 Note2
Function								0: Port 1: Data bus (D16 to D23)

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

Note 2: It is set to “Port” or “Data bus” by AM pin setting.

Note 3:<P2XC> show X bit of P2CR register.

Port 2 function register

P2CR<P2xC> \ P2FC<P2xF>	0	1
0	Input port	Data bus (D16 to D23)
1	Output port	

Figure 3.5.4 Register for Port 2

3.5.3 Port 3 (P30 to P37)

Port 3 is an 8-bit general-purpose I/O port.

Bits can be individually set as either inputs or outputs by control register P3CR and function register P3FC. In addition to functioning as a general-purpose I/O port, port 3 can also function as a data bus (D24 to D31).

AM1	AM0	Function Setting after Reset is Released
0	0	Don't use this setting
0	1	Input port
1	0	Data bus (D24 to D31)
1	1	Don't use this setting

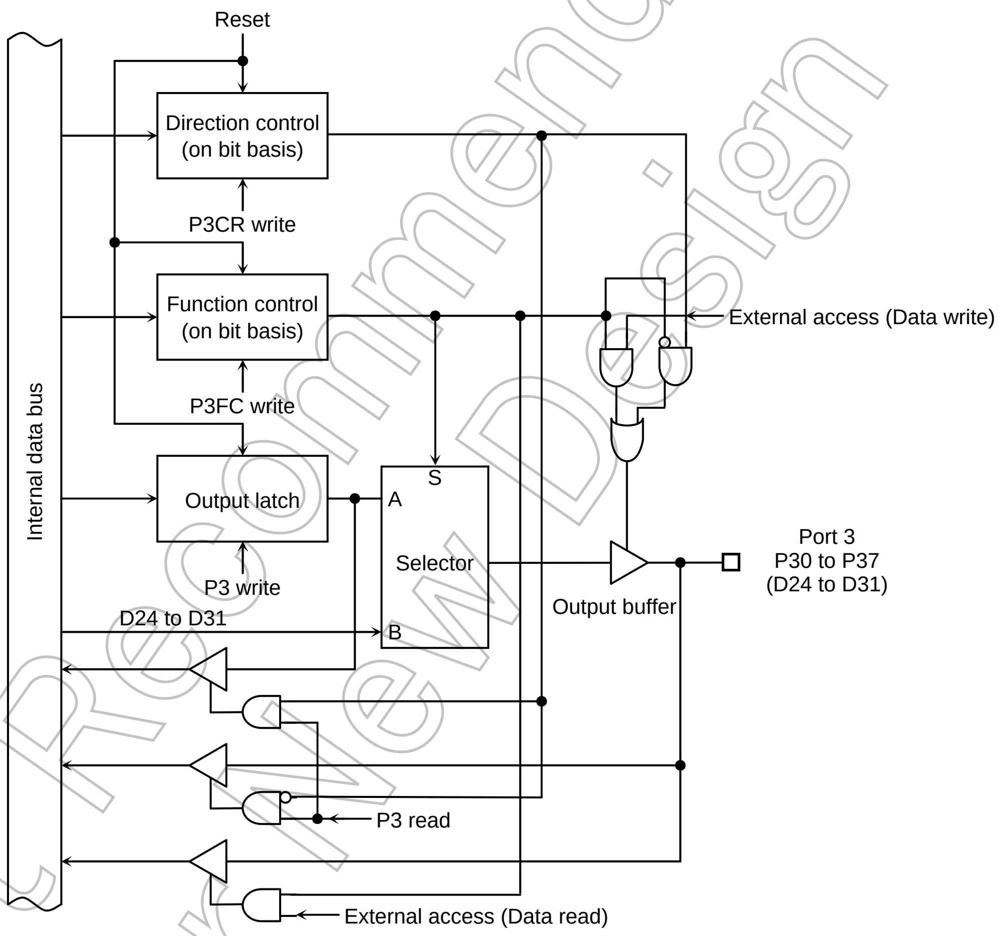


Figure 3.5.5 Port 3

Port 3 Register

P3 (000CH)		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 (000EH)		7	6	5	4	3	2	1	0
	Bit symbol	P37C	P36C	P35C	P34C	P33C	P32C	P31C	P30C
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Input 1: Output							

Port 3 Function Register

P3FC (000FH)		7	6	5	4	3	2	1	0
	Bit symbol								P3F
	Read/Write								W
	After reset								0/1 Note2
	Function								0: Port 1: Data bus (D24 to D31)

Note 1:Read-modify-write is prohibited for the registers P3CR and P3FC.

Note 2: It is set to “Port” or “Data bus” by AM pin setting.

Note 3:<P3XC> show X bit of P3CR register.

Port 3 function register

P3CR<P3xC> \ P3FC<P3xF>	0	1
0	Input port	Data bus (D24 to D31)
1	Output port	Data bus (D24 to D31)

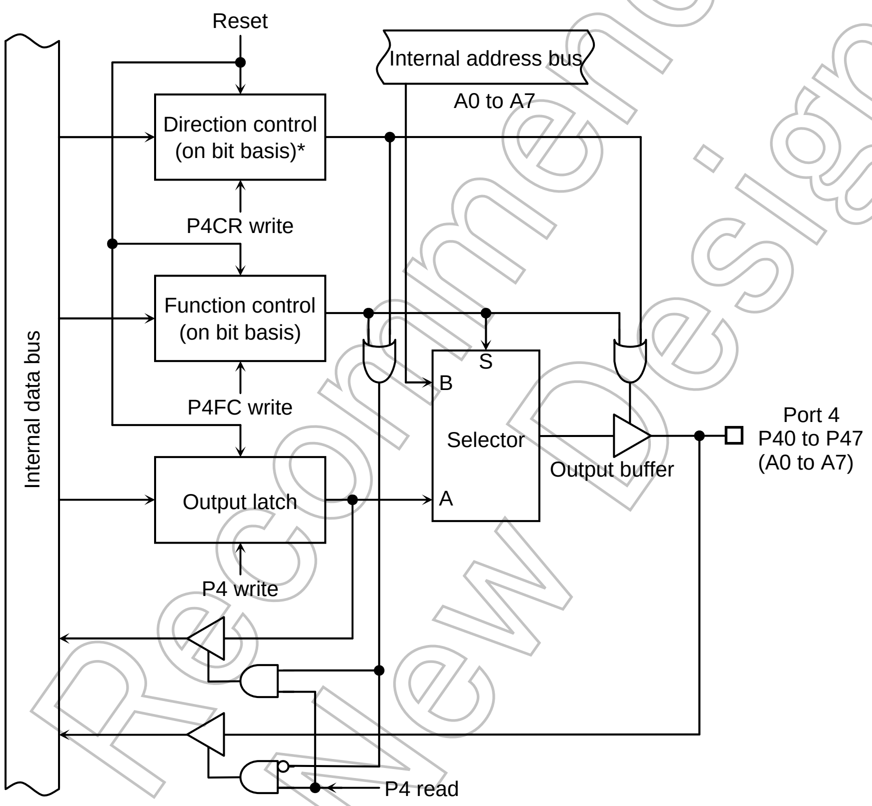
Figure 3.5.6 Register for Port 3

3.5.4 Port 4 (P40 to P47)

Port 4 is an 8-bit general-purpose I/O ports*.

Bits can be individually set as either inputs or outputs by control register P4CR and function register P4FC*. In addition to functioning as a general-purpose I/O port, port 4 can also function as an address bus (A0 to A7).

AM1	AM0	Function Setting after Reset is Released
0	0	Don't use this setting
0	1	Address bus (A0 to A7)
1	0	Address bus (A0 to A7)
1	1	Don't use this setting



*: When these ports are used as general-purpose I/O port, each bit can be set individually for input or output. However, each bit cannot be set individually for input or output even if 1bit or more bits are used as address bus in same port.
All of general-purpose I/O ports except for port that used as address bus are operated as output port. Please be careful when using this setting.

Figure 3.5.7 Port 4

Port 4 Register

P4 (0010H)

	7	6	5	4	3	2	1	0
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

P4CR (0012H)

	7	6	5	4	3	2	1	0
Bit symbol	P47C	P46C	P45C	P44C	P43C	P42C	P41C	P40C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output (Note2)							

Port 4 Function Register

P4FC (0013H)

	7	6	5	4	3	2	1	0
Bit symbol	P47F	P46F	P45F	P44F	P43F	P42F	P41F	P40F
Read/Write	W							
After reset	1	1	1	1	1	1	1	1
Function	0: Port 1: Address bus (A0 to A7) (Note2)							

Note1: Read-modify-write is prohibited for the registers P4CR and P4FC.

Note2: When these ports are used as general-purpose I/O port, each bit can be set individually for input or output. However, each bit cannot be set individually for input or output even if 1bit or more bits are used as address bus in same port. All of general-purpose I/O ports except for port that used as address bus are operated as output port. Please be careful when using this setting.

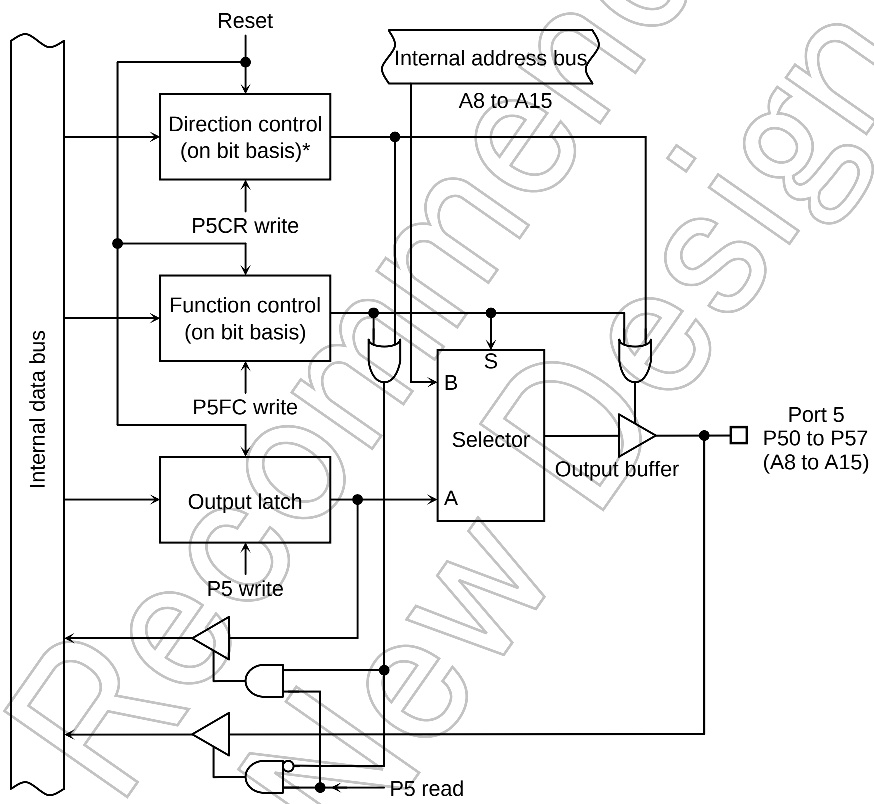
Figure 3.5.8 Port 4 Registers

3.5.5 Port 5 (P50 to P57)

Port 5 is an 8-bit general-purpose I/O ports*.

Bits can be individually set as either inputs or outputs by control register P5CR and function register P5FC*. In addition to functioning as a general-purpose I/O port, port 5 can also function as an address bus (A8 to A15).

AM1	AM0	Function Setting after Reset is Released
0	0	Don't use this setting
0	1	Address bus (A8 to A15)
1	0	Address bus (A8 to A15)
1	1	Don't use this setting



*: When these ports are used as general-purpose I/O port, each bit can be set individually for input or output. However, each bit cannot be set individually for input or output even if 1bit or more bits are used as address bus in same port.
All of general-purpose I/O ports except for port that used as address bus are operated as output port.
Please be careful when using this setting.

Figure 3.5.9 Port 5

Port 5 Register

P5 (0014H)		7	6	5	4	3	2	1	0
	Bit symbol	P57	P56	P55	P54	P53	P52	P51	P50
	Read/Write	R/W							
	After reset	Data from external port (Output latch register is cleared to 0)							

Port 5 Control Register

P5CR (0016H)		7	6	5	4	3	2	1	0
	Bit symbol	P57C	P56C	P55C	P54C	P53C	P52C	P51C	P50C
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Input 1: Output (Note2)							

Port 5 Function Register

P5FC (0017H)		7	6	5	4	3	2	1	0
	Bit symbol	P57F	P56F	P55F	P54F	P53F	P52F	P51F	P50F
	Read/Write	W							
	After reset	1	1	1	1	1	1	1	1
	Function	0: Port 1: Address bus (A8 to A15) (Note2)							

Note1: Read-modify-write is prohibited for the registers P5CR and P5FC.

Note2: When these ports are used as general-purpose I/O port, each bit can be set individually for input or output. However, each bit cannot be set individually for input or output even if 1bit or more bits are used as address bus in same port. All of general-purpose I/O ports except for port that used as address bus are operated as output port. Please be careful when using this setting.

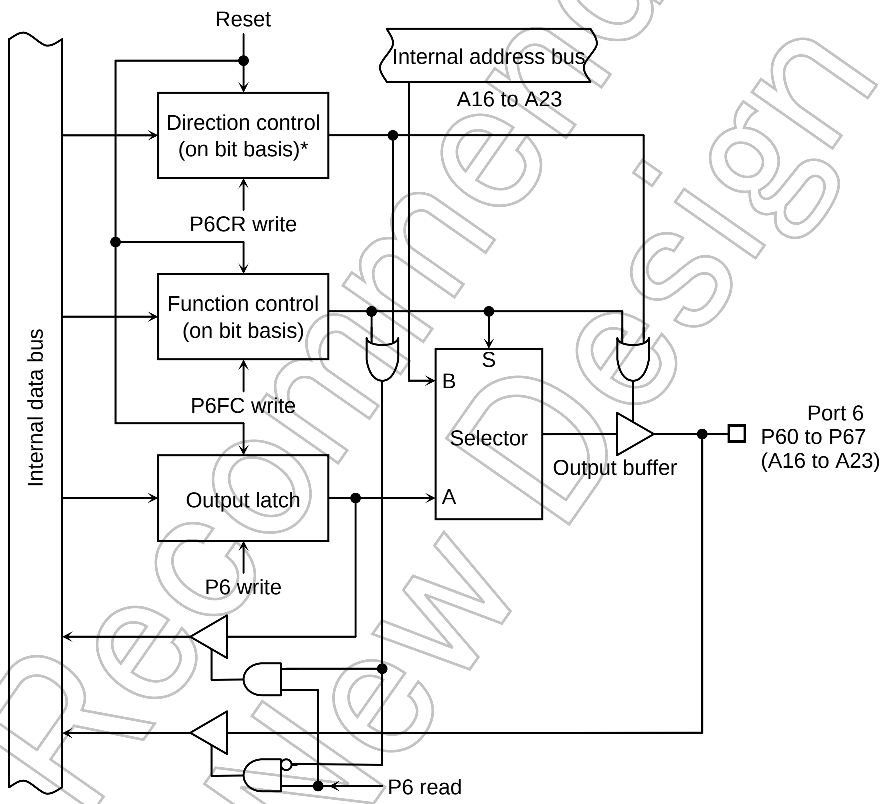
Figure 3.5.10 Register for Port 5

3.5.6 Port 6 (P60 to P67)

Port 6 is an 8-bit general-purpose I/O ports*.

Bits can be individually set as either inputs or outputs by control register P6CR and function register P6FC*. In addition to functioning as a general-purpose I/O port, port 6 can also function as an address bus (A16 to A23).

AM1	AM0	Function Setting after Reset is Released
0	0	Don't use this setting
0	1	Address bus (A16 to A23)
1	0	Address bus (A16 to A23)
1	1	Don't use this setting



*: When these ports are used as general-purpose I/O port, each bit can be set individually for input or output. However, each bit cannot be set individually for input or output even if 1bit or more bits are used as address bus in same port.
All of general-purpose I/O ports except for port that used as address bus are operated as output port.
Please be careful when using this setting.

Figure 3.5.11 Port 6

Port 6 Register

P6 (0018H)		7	6	5	4	3	2	1	0
	Bit symbol	P67	P66	P65	P64	P63	P62	P61	P60
	Read/Write	R/W							
	After reset	Data from external port (Output latch register is cleared to 0)							

Port 6 Control Register

P6CR (001AH)		7	6	5	4	3	2	1	0
	Bit symbol	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Input 1: Output (Note2)							

Port 6 Function Register

P6FC (001BH)		7	6	5	4	3	2	1	0
	Bit symbol	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
	Read/Write	W							
	After reset	1	1	1	1	1	1	1	1
	Function	0: Port 1: Address bus (A16 to A23) (Note2)							

Note1: Read-modify-write is prohibited for the registers P6CR and P6FC.

Note2: When these ports are used as general-purpose I/O port, each bit can be set individually for input or output. However, each bit cannot be set individually for input or output even if 1bit or more bits are used as address bus in same port. All of general-purpose I/O ports except for port that used as address bus are operated as output port. Please be careful when using this setting.

Figure 3.5.12 Port 6 Registers

3.5.7 Port 7 (P70 to P76)

Port 7 is a 7-bit general-purpose I/O port (P70 to P75 are used for output only).

Bits can be individually set as either inputs or outputs by control register P7CR and function register P7FC.

In addition to functioning as a general-purpose I/O port, P70 to P75 pins can also function as read/write strobe signals to connect with an external memory. P76 pin can also function as wait input. A reset initializes P70 to P75 pins to output port mode, and P76 pin to input port mode.

AM1	AM0	Function Setting after Reset is Released
0	0	Don't use this setting
0	1	$\overline{RD}$ pin
1	0	$\overline{RD}$ pin
1	1	Don't use this setting

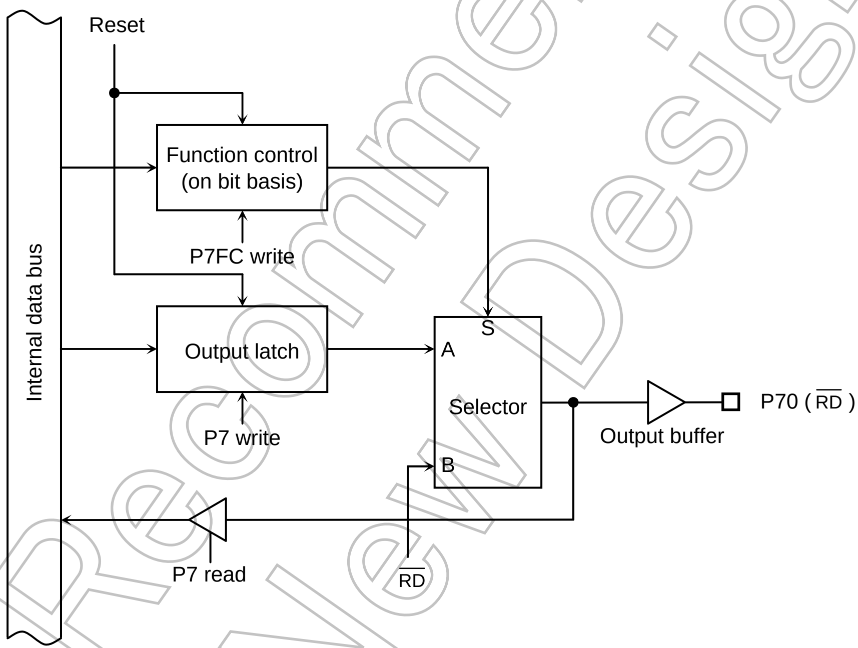


Figure 3.5.13 Port 7 (P70)

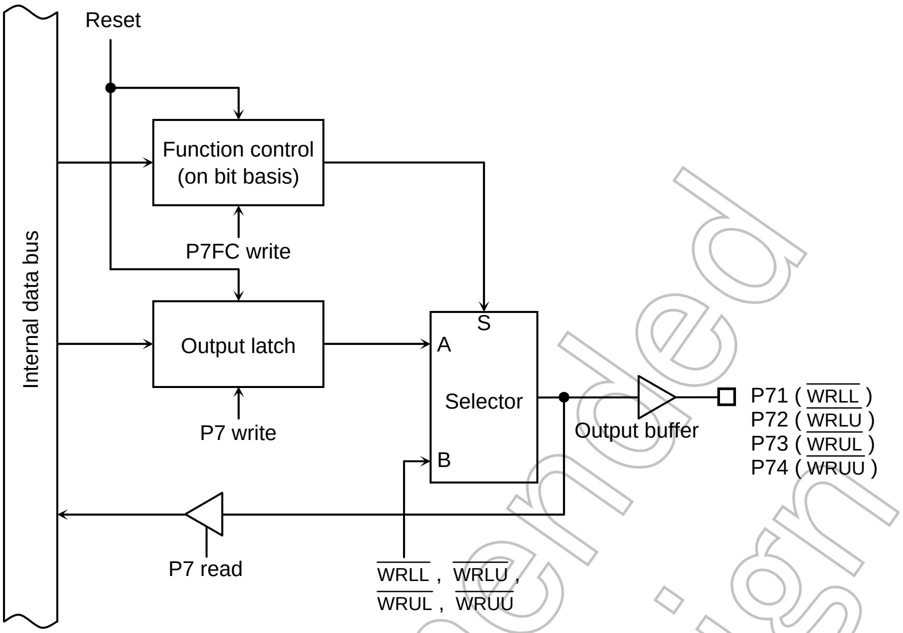


Figure 3.5.14 Port 7 (P71 to P74)

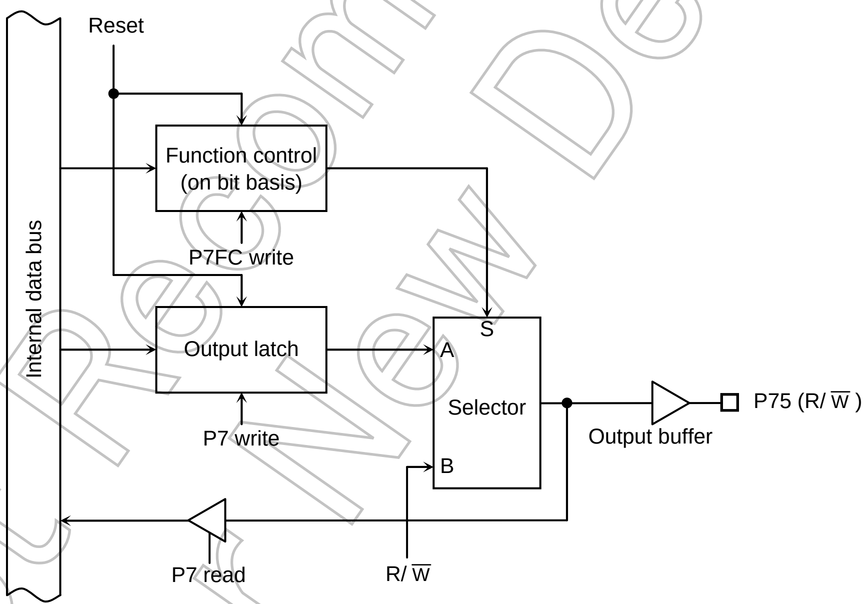


Figure 3.5.15 Port 7 (P75)

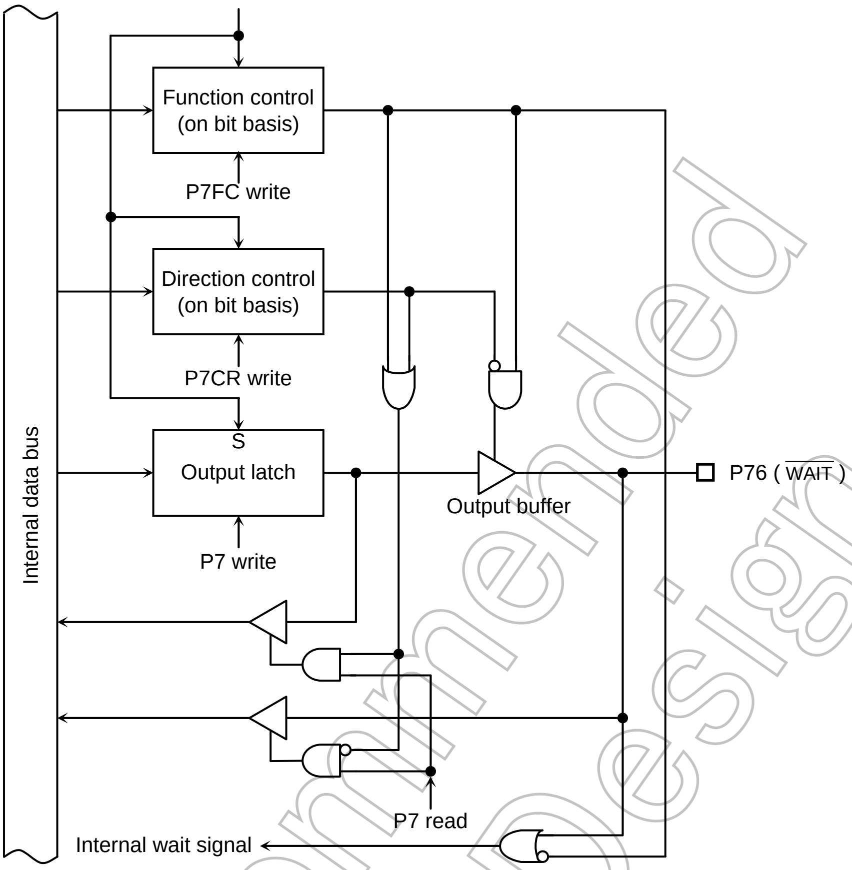


Figure 3.5.16 Port 7 (P76)

Port 7 Register

P7 (001CH)		7	6	5	4	3	2	1	0
	Bit symbol		P76	P75	P74	P73	P72	P71	P70
	Read/Write		R/W						
	After reset		Data from external port (Note)	1	1	1	1	1	1

Note: Output latch register is cleared to 0.

Port 7 Control Register

P7CR (001EH)		7	6	5	4	3	2	1	0
	Bit symbol		P76C						
	Read/Write		W						
	After reset		0						
	Function		0: Input 1: Output						

Port 7 Function Register

P7FC (001FH)		7	6	5	4	3	2	1	0
	Bit symbol		P76F	P75F	P74F	P73F	P72F	P71F	P70F
	Read/Write		W						
	After reset		0	0	0	0	0	0	1
	Function		0: Port 1: $\overline{\text{WAIT}}$	0: Port 1: R/$\overline{\text{W}}$	0: Port 1: $\overline{\text{WRUU}}$	0: Port 1: $\overline{\text{WRUL}}$	0: Port 1: $\overline{\text{WRLU}}$	0: Port 1: $\overline{\text{WRLL}}$	0: Port 1: $\overline{\text{RD}}$

Note: Read-modify-write is prohibited for the registers P7CR and P7FC.

Figure 3.5.17 Register for Port 7

3.5.8 Port 8 (P80 to P87)

Ports 80 to 87 are 8-bit output ports. Resetting sets output latch of P82 to “0” and output latches of P80 to P81, P83 to P87 to “1”.

Port 8 also function as chip-select output ($\overline{CS0}$ to $\overline{CS3}$), extend address output (EA24, EA25), extend chip-select output ($\overline{CS2A}$, $\overline{CS2B}$, $\overline{CS2C}$, $\overline{CS2D}$), port 8 also function as output pin for SDRAM controller ($\overline{SDCSL}$, $\overline{SDCSH}$, SDCLK), Above setting is used the function register P8FC. Writing “1” in the corresponding bit of P8FC, P8FC2 enables the respective functions.

Resetting resets P87F of P8FC to “1”, P80F to P86F of P8FC to “0”, and P8FC2 to “0”, sets all bits to output ports.

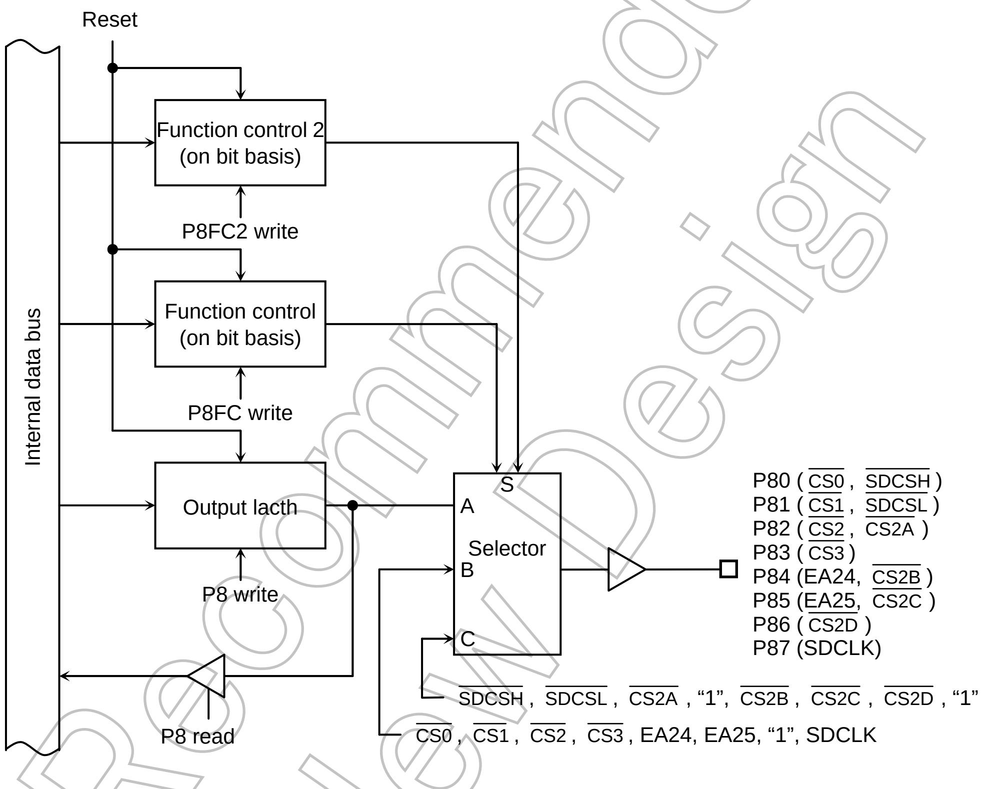


Figure 3.5.18 Port 8

Port 8 Register

P8 (0020H)

	7	6	5	4	3	2	1	0
Bit symbol	P87	P86	P85	P84	P83	P82	P81	P80
Read/Write	R/W							
After reset	1	1	1	1	1	0	1	1

Port 8 Function Register

P8FC (0023H)

	7	6	5	4	3	2	1	0
Bit symbol	P87F	–	P85F	P84F	P83F	P82F	P81F	P80F
Read/Write	W							
After reset	1	0	0	0	0	0	0	0
Function	0: Port 1: SDCLK	Always write “0”.	0: Port 1: EA25	0: Port 1: EA24	0: Port 1: $\overline{CS3}$	0: Port 1: $\overline{CS2}$	0: Port 1: $\overline{CS1}$	0: Port 1: $\overline{CS0}$

Port 8 Function Register 2

P8FC2 (0021H)

	7	6	5	4	3	2	1	0
Bit symbol	–	P86F2	P85F2	P84F2	–	P82F2	P81F2	P80F2
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Always write “0”.	0: <P86F> 1: $\overline{CS2D}$	0: <P85F> 1: $\overline{CS2C}$	0: <P84F> 1: $\overline{CS2B}$	Always write “0”.	0: <P82F> 1: $\overline{CS2A}$	0: <P81F> 1: $\overline{SDCSL}$	0: <P80F> 1: $\overline{SDCSH}$

Note :Read-modify-write is prohibited for P8FC and P8FC2 .

Figure 3.5.19 Registers for Port 8

3.5.9 Port 9 (P90 to P96)

P90 to P96 are 7-bit general-purpose I/O port. I/O can be set on bit basis using the control register.

Resetting sets port 9 to input port and all bits of output latch to “1”.

Writing in the corresponding bit of P9FC enables the respective functions.

Resetting resets the P9FC to “0”, and sets all bits to input ports.

(1) Port 90 (SCK), port 91 (SO/SDA), and port 92 (SI/SCL)

Ports 90 to 92 are general-purpose I/O port. It is also used as SCK (Clock signal for SIO mode), SO (Data output for SIO mode), SDA (Data input for I²C mode), SI (Data input for SIO mode), and SCL (Clock input/output for I²C mode) for serial bus interface.

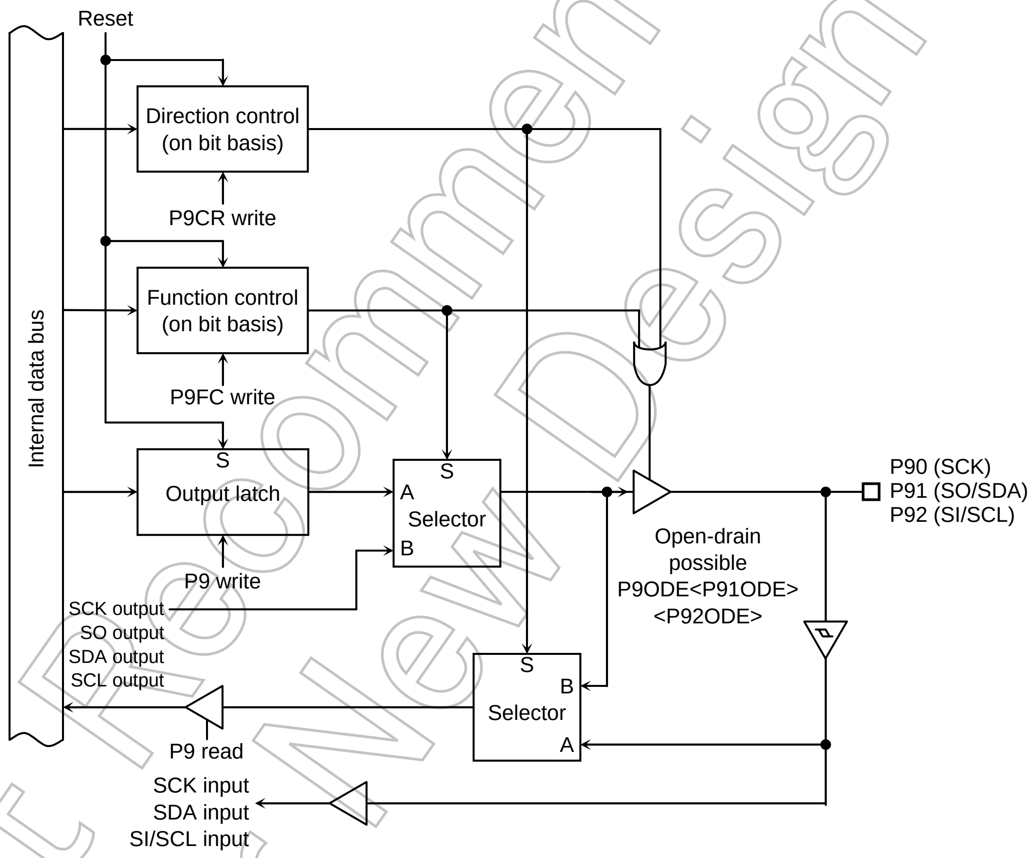


Figure 3.5.20 Port 9 (P90 to P92)

(2) Ports 93 ($\overline{CS2E}$), 94 ($\overline{CS2F}$), 95 (TXD2, $\overline{CS2G}$), and 96 (RXD2, $\overline{CSEXA}$)

Ports 93 to 96 are general-purpose I/O ports.

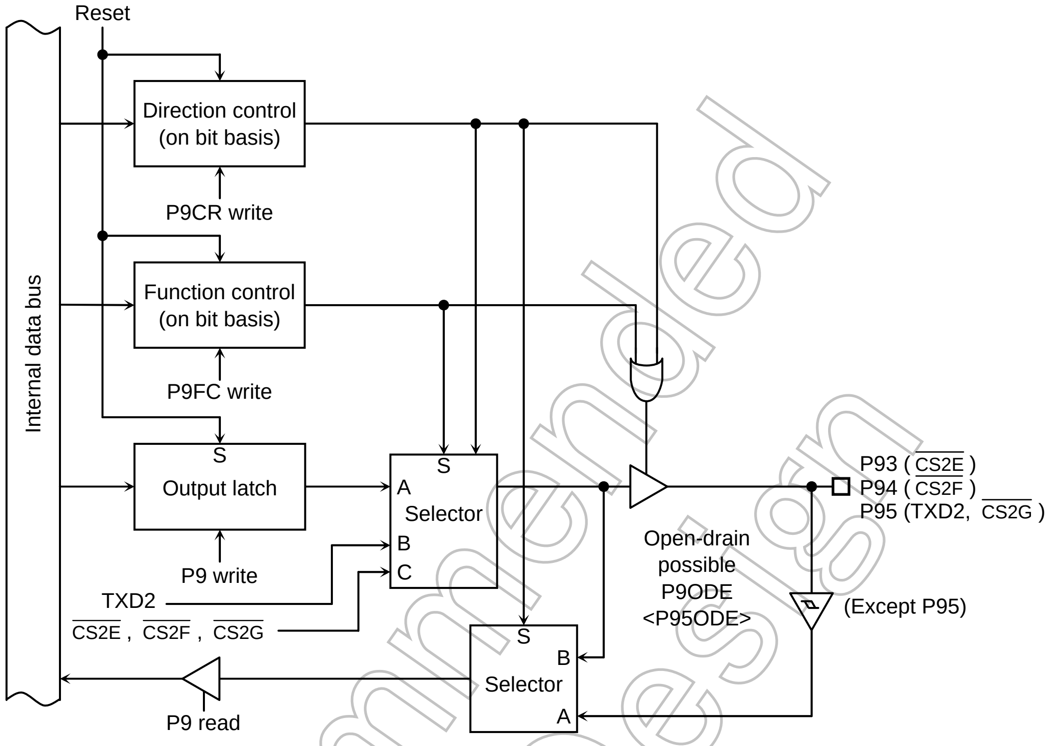


Figure 3.5.21 Port 9 (P93 to P95)

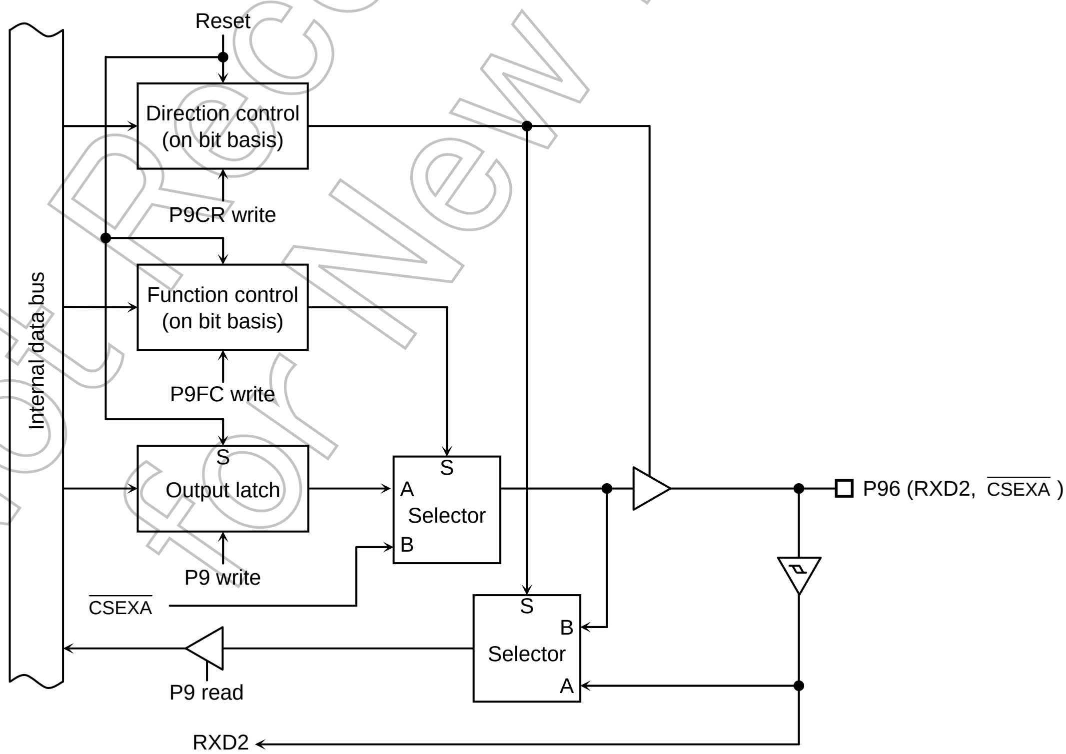


Figure 3.5.22 Port 9 (P96)

Port 9 Register

P9 (0024H)		7	6	5	4	3	2	1	0
	Bit symbol		P96	P95	P94	P93	P92	P91	P90
	Read/Write		R/W						
	After reset		Data from external port (Output latch register is set to 1)						

Port 9 Control Register

P9CR (0026H)		7	6	5	4	3	2	1	0
	Bit symbol		P96C	P95C	P94C	P93C	P92C	P91C	P90C
	Read/Write		W						
	After reset		0	0	0	0	0	0	0
	Function		0: Input 1: Output						

Port 9 Function Register

P9FC (0027H)		7	6	5	4	3	2	1	0
	Bit symbol		P96F	P95F	P94F	P93F	P92F	P91F	P90F
	Read/Write		W						
	After reset		0	0	0	0	0	0	0
	Function		0: Port 1: RXD2, $\overline{CSEXA}$	0: Port 1: TXD2, $\overline{CS2G}$	0: Port 1: $\overline{CS2F}$	0: Port 1: $\overline{CS2E}$	0: Port, SI, 1: SCL Note 2	0: Port 1: SO, SDA	0: Port, SCK input 1: SCK Output Note 2

$\overline{CS2E}$ setting

<P93F> \ <P93C>	0	1
0	Input port	Output port
1	(Reserved)	$\overline{CS2E}$

$\overline{CS2F}$ setting

<P94F> \ <P94C>	0	1
0	Input port	Output port
1	(Reserved)	$\overline{CS2F}$

TXD2, $\overline{CS2G}$ setting

<P95F> \ <P95C>	0	1
0	Input port	Output port
1	TXD2	$\overline{CS2G}$

Port 9 ODE Register

P9ODE (0025H)		7	6	5	4	3	2	1	0
	Bit symbol			P95ODE	–	–	P92ODE	P91ODE	
	Read/Write			W	W	W	W	W	
	After reset			0	0	0	0	0	
	Function			0: 3 states 1: Open drain	Always write "0".	Always write "0".	0: 3 states 1: Open drain	0: 3 states 1: Open drain	

Note 1: Read-modify-write is prohibited for P9CR, P9FC, and P9ODE.

Note 2: When using SI and SCK input function, set P9FC<P92F,P90F> to "0" (Function setting).

Figure 3.5.23 Register for Port 9

3.5.10 Port A (PA0 to PA7)

Ports A0 to A7 are 8-bit input ports with pull-up resistor. In addition to functioning as general-purpose I/O ports, ports A0 to A7 can also key-on wakeup function as keyboard interface. The various functions can each be enabled by writing a "1" to the corresponding bit of the port A function register (PAFC).

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

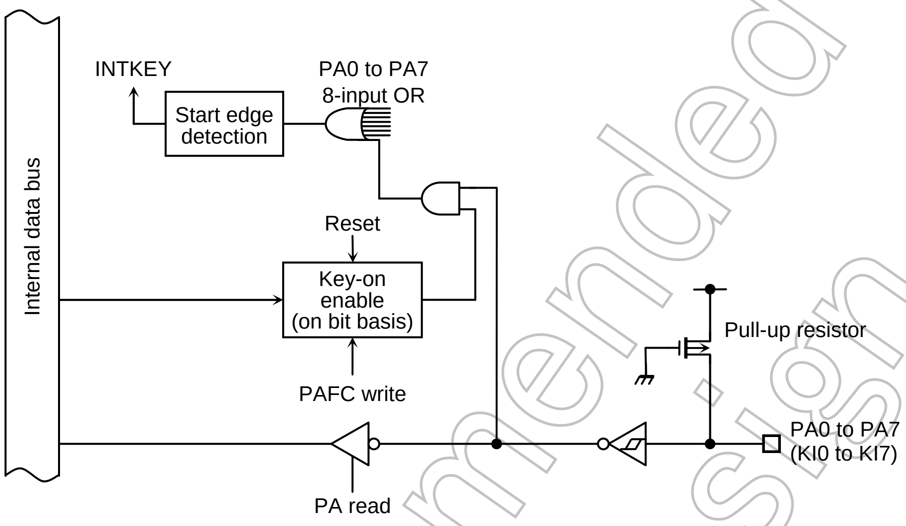


Figure 3.5.24 Port A

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

Port A Register

PA (0028H)		7	6	5	4	3	2	1	0
	Bit symbol	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
	Read/Write	R							
	After reset	Data from external port							

Port A Function Register

PAFC (002BH)		7	6	5	4	3	2	1	0
	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: KEY-IN disable 1: KEY-IN enable							

Key-IN of Port A

0	Disable
1	Enable

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

Figure 3.5.25 Register for Port A

3.5.11 Port C (PC0, PC1, PC3, PC5 and PC6)

Port C is 5-bit general-purpose I/O port. Each bit can be set individually for input or output. Resetting sets port C to be an input port.

In addition to functioning as a general-purpose I/O port, port C can also functions as I/O pin for timers (TA0IN, TA1OUT, TA3OUT, TB0OUT0), input pin for external interruption (INT0 to INT3). Above setting is used the function register PCFC and PCCR register. Edge select of external interruption establishes it with IIMC register, which there is in interruption controller. Resetting resets bits of the register PCCR and PCFC to "0" and sets all pins to be input port.

(1) PC0 (TA0IN)

In addition to function as I/O port, port 0 can also function as input pin TA0IN of timer channel 0.

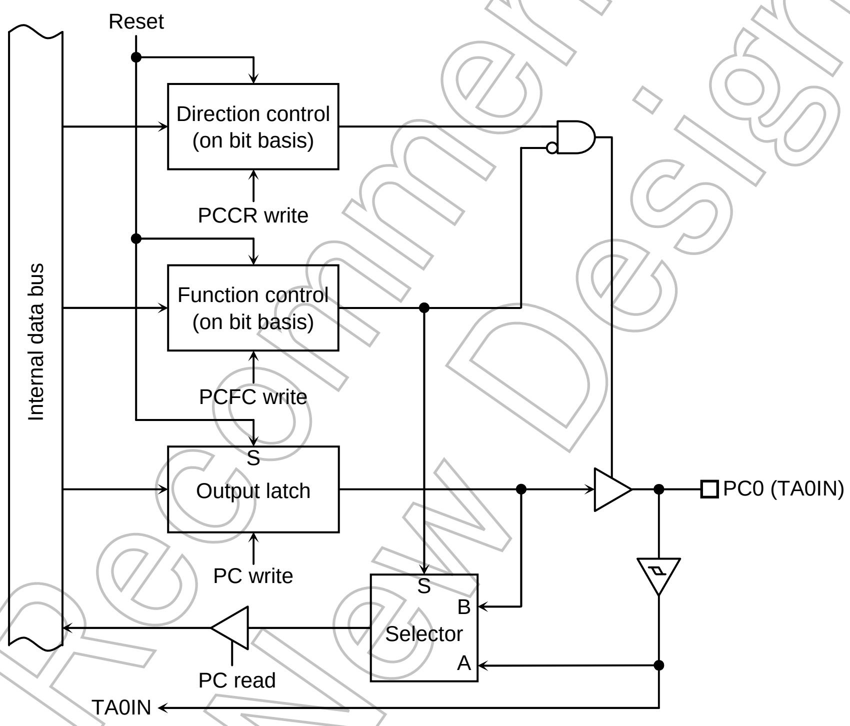


Figure 3.5.26 Port C (PC0)

Note: Cannot read the output latch data when output mode.

(2) PC1 (INT1, TA1OUT), PC5 (INT2, TA3OUT) and PC6 (INT3, TB0OUT0)

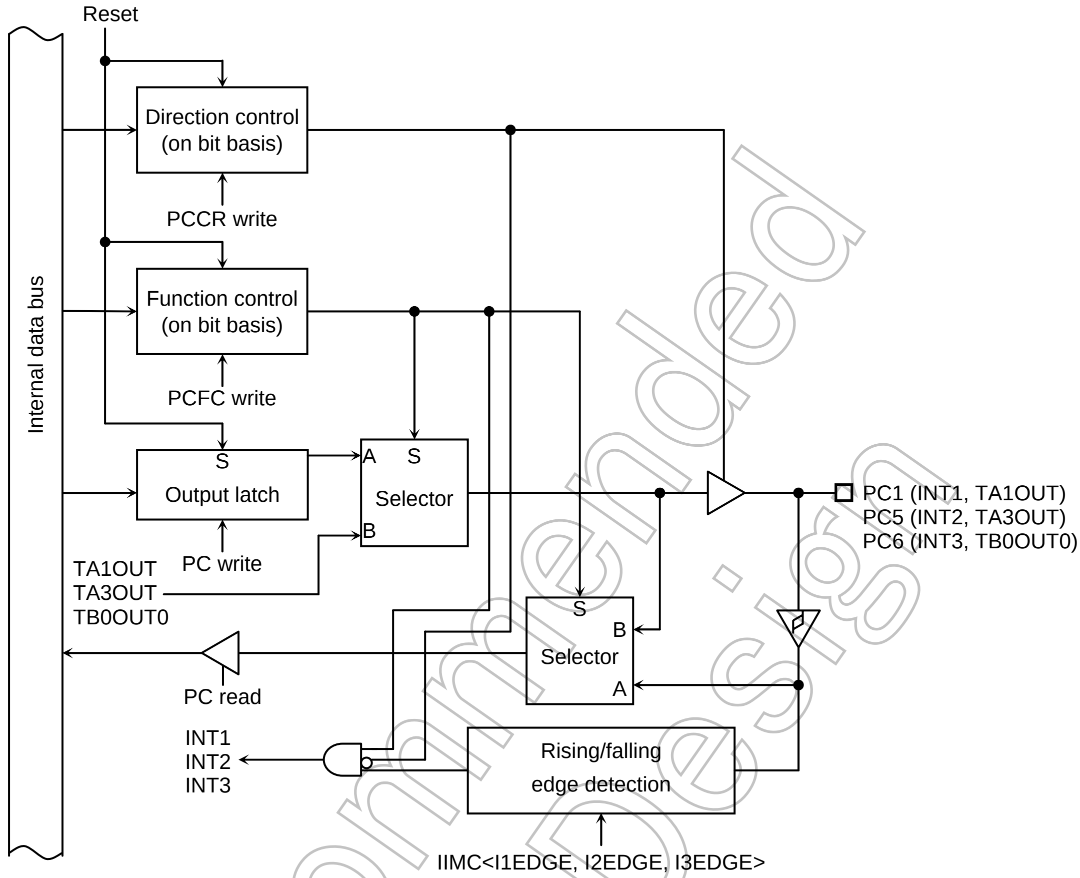


Figure 3.5.27 Port C (PC1, PC5, PC6)

Note: Cannot read the output latch data when output mode.

(3) PC3 (INT0)

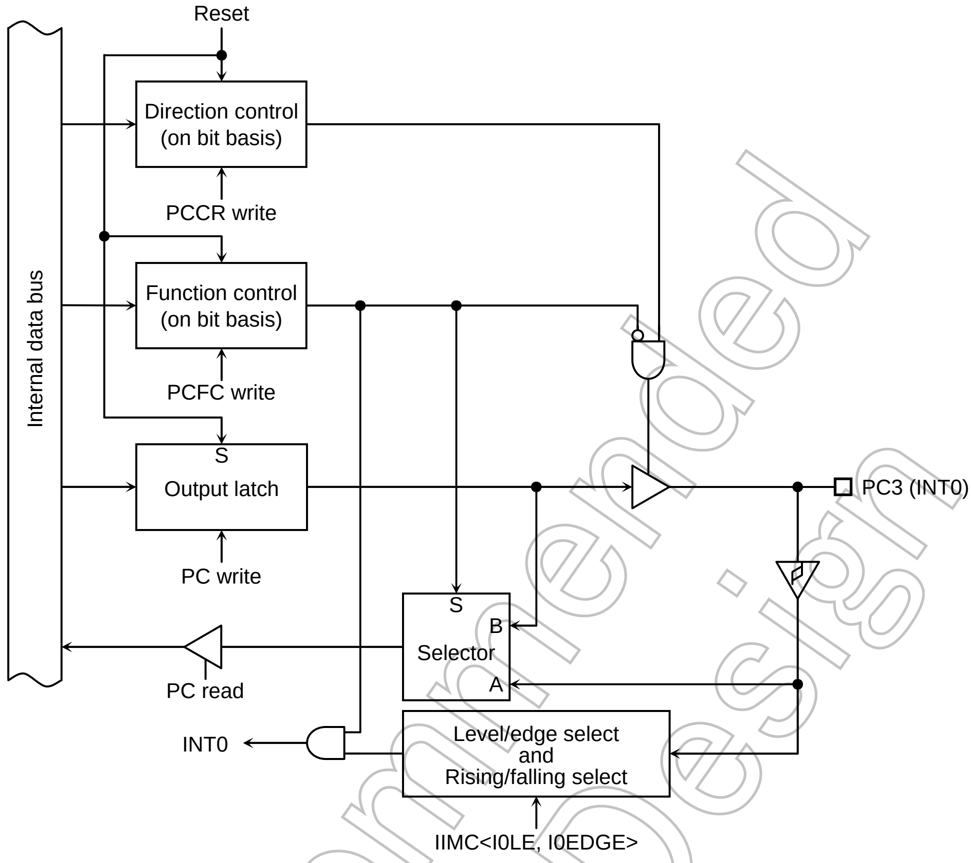


Figure 3.5.28 Port C (PC3)

Port C Register

PC (0030H)

	7	6	5	4	3	2	1	0
Bit symbol		PC6	PC5		PC3		PC1	PC0
Read/Write		R/W			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)		Data from external port (Output latch register is set to 1)	

Port C Control Register

PCCR (0032H)

	7	6	5	4	3	2	1	0
Bit symbol		PC6C	PC5C		PC3C		PC1C	PC0C
Read/Write		W			W		W	
After reset		0	0		0		0	0
Function		0: Input 1: Output			0: Input 1: Output		0: Input 1: Output	

Port C Function Register

PCFC (0033H)

	7	6	5	4	3	2	1	0
Bit symbol		PC6F	PC5F		PC3F		PC1F	PC0F
Read/Write		W			W		W	
After reset		0	0		1		0	0
Function		0: Port 1: INT3 TB0OUT0	0: Port 1: INT2 TA3OUT		0: Port 1: INT0		0: Port 1: INT1 TA1OUT	0: Port 1: TA0IN

INT1, TA1OUT setting

<PC1F> \ <PC1C>	0	1
0	Input port	Output port
1	INT1	TA1OUT

INT2, TA3OUT setting

<PC5F> \ <PC5C>	0	1
0	Input port	Output port
1	INT2	TA3OUT

INT3, TB0OUT0 setting

<PC6F> \ <PC6C>	0	1
0	Input port	Output port
1	INT3	TB0OUT0

Note 1: Read-modify-write is prohibited for the registers PCCR and PCFC.

Note 2: PC0/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: Cannot read the output latch data when PC0, PC1, PC5, and PC6 are output mode.

Figure 3.5.29 Register for Port C

3.5.12 Port F (PF0 to PF5)

Ports F0 to F5 are 6-bit general-purpose I/O ports. Each bit can be set individually for input or output. Resetting sets PF0 to PF5 to be an input ports. It also sets all bits of the output latch register to "1".

In addition to functioning as general-purpose I/O port pins, PF0 to PF5 can also function as the I/O for serial channels 0 and 1. A pin can be enabled for I/O by writing a "1" to the corresponding bit of the port F function register (PFFC).

By resetting, clears all bits of the registers PFCR and PFFC to 0 and sets all pins to be input ports.

(1) Ports PF0 (TXD0) and PF3 (TXD1)

As well as functioning as I/O port pins, port PF0 and PF3 can also function as serial channel TXD output pins.

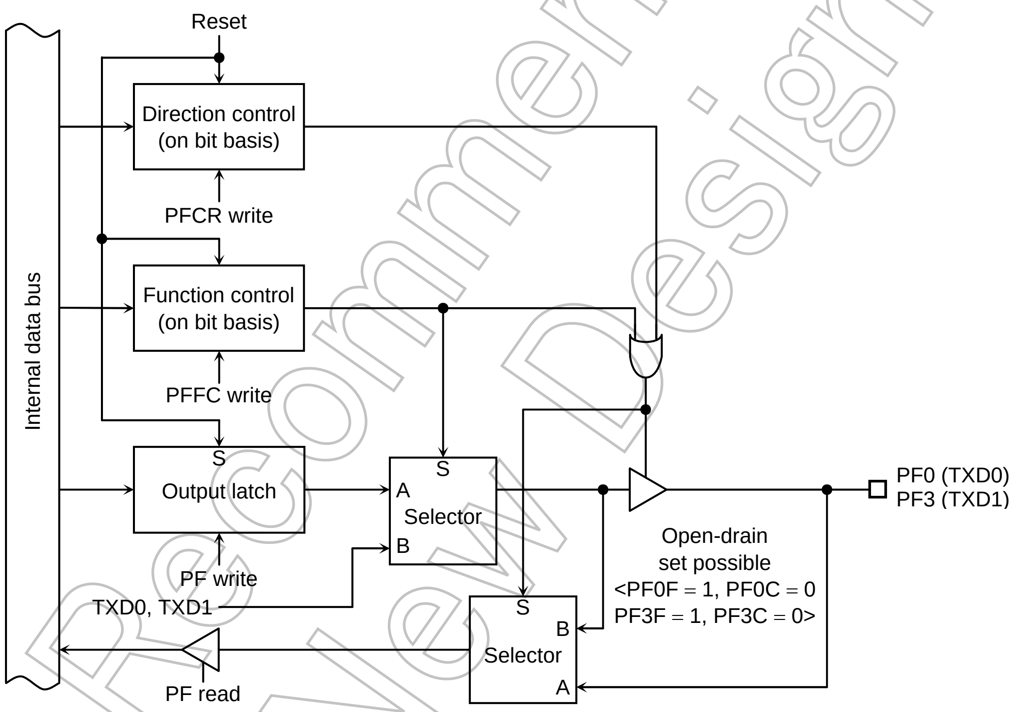


Figure 3.5.30 Port F (PF0 and PF3)

(2) Ports PF1 and PF4 (RXD0, RXD1)

Ports PF1 and PF4 are I/O port pins and can also is used as RXD input for the serial channels.

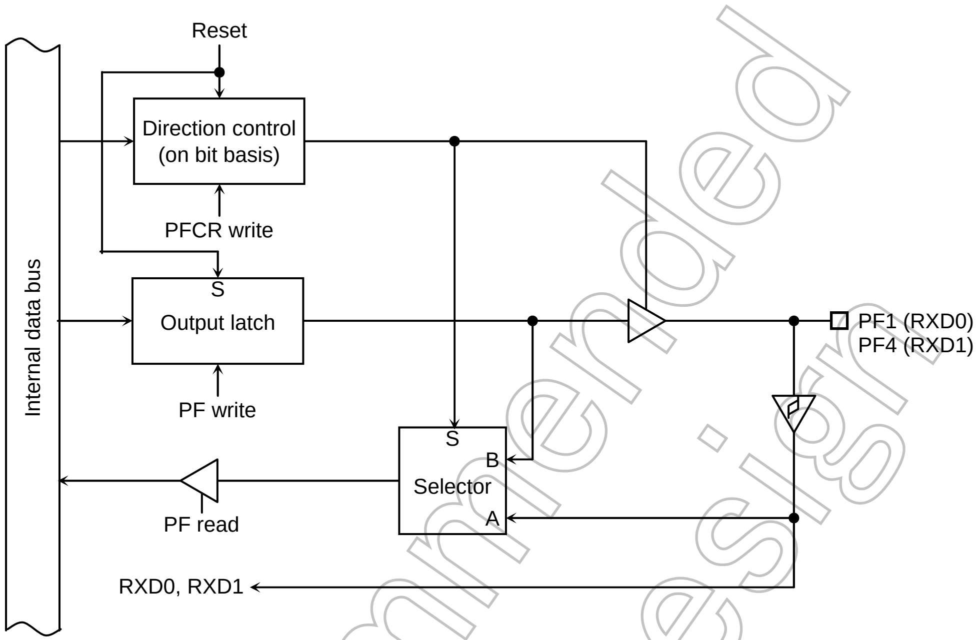


Figure 3.5.31 Port F (PF1 and PF4)

(3) Ports PF2 ($\overline{\text{CTS0}}$, SCLK0) and PF5 ($\overline{\text{CTS1}}$, SCLK1)

Ports PF2 and PF5 are I/O port pins and can also be used as $\overline{\text{CTS}}$ input or SCLK input/output for the serial channels.

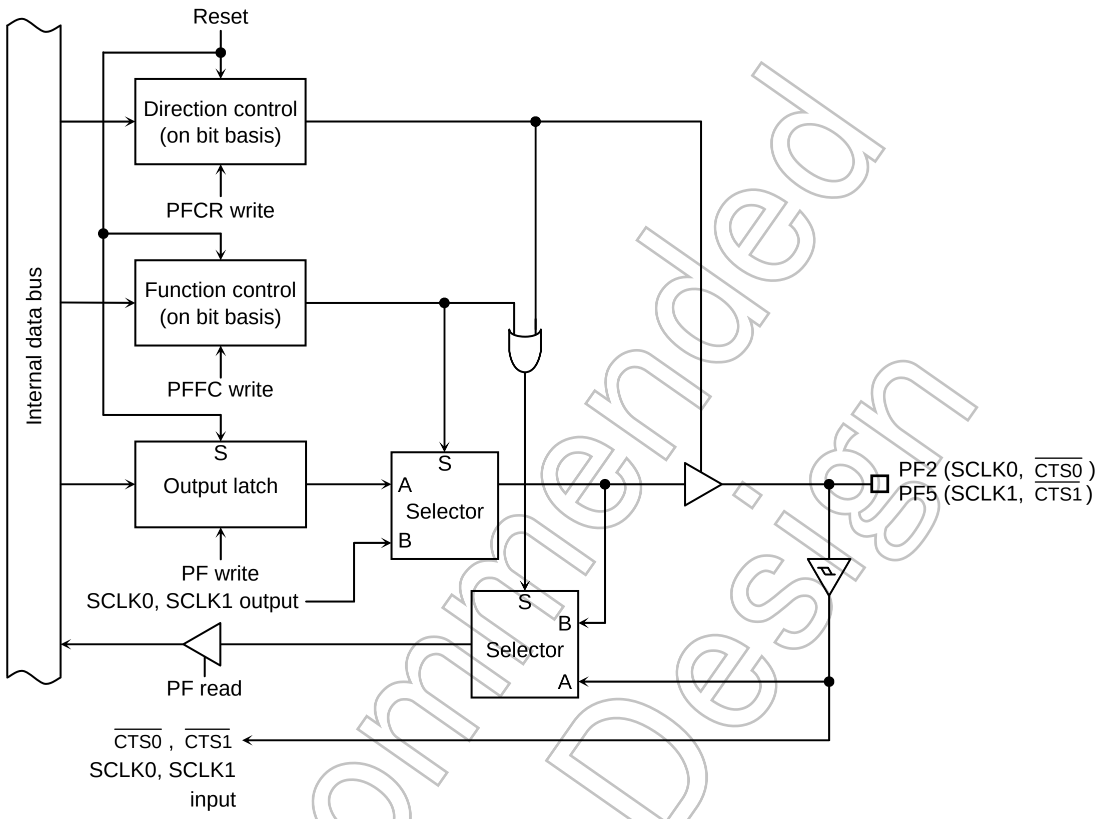


Figure 3.5.32 Port F (PF2 and PF5)

Port F Register

PF (003CH)		7	6	5	4	3	2	1	0
	Bit symbol			PF5	PF4	PF3	PF2	PF1	PF0
	Read/Write			R/W					
	After reset			Data from external port (Output latch register is set to 1)					

Port F Control Register

PFCR (003EH)		7	6	5	4	3	2	1	0
	Bit symbol			PF5C	PF4C	PF3C	PF2C	PF1C	PF0C
	Read/Write			W					
	After reset			0	0	0	0	0	0
	Function			0: Input 1: Output					

Port F Function Register

PFFC (003FH)		7	6	5	4	3	2	1	0
	Bit symbol			PF5F		PF3F	PF2F		PF0F
	Read/Write			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

3 states, Open-drain setting

<PF3F> \ <PF3C>	0	1
0	Input port	Output port
1	TXD1 (Open drain)	TXD1 (3 states)

<PF1F> \ <PF1C>	0	1
0	Input port	Output port
1	TXD0 (Open drain)	TXD0 (3 states)

Note 1: Read-modify-write is prohibited for the registers PFCR and PFFC.

Note 2: PF1/RXD0 and PF4/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.

Note 3: PF1 and PF3 pins dose not have a register changing 3 states/Open drain.

Figure 3.5.33 Register for Port F

3.5.13 Port G (PG0 to PG4)

PG0 to PG4 are 5-bit input port and can also be used as the analog input pins for the internal AD converter. PG3 can also be used as ADTRG pin for the AD converter.

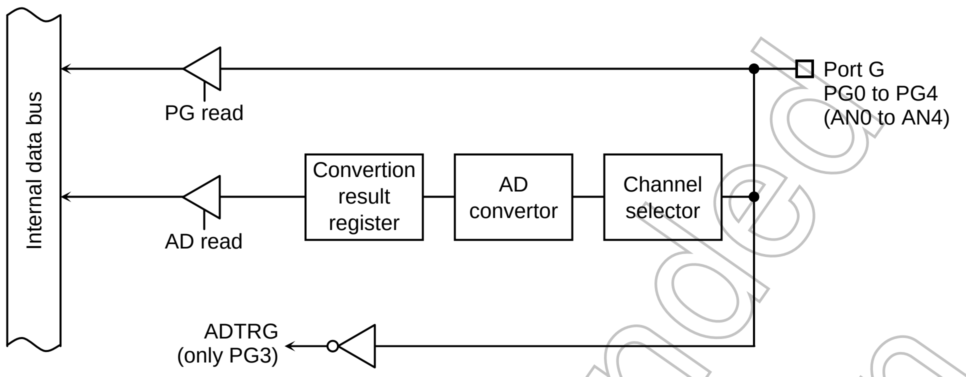


Figure 3.5.34 Port G

Port G Register

		7	6	5	4	3	2	1	0
PG (0040H)	Bit symbol				PG4	PG3	PG2	PG1	PG0
	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 3.5.35 Register for Port G

3.5.14 Port J (PJ0 to PJ7)

PJ0 to PJ7 are 8-bit output port. Resetting sets the output latch PJ to “1” and PJ0 to PJ7 pins output “1”.

In addition to functioning as output port, port J also functions as output pins for SDRAM ($\overline{\text{SDRAS}}$, $\overline{\text{SDCAS}}$, $\overline{\text{SDWE}}$, SDLLDQM, SDLUDQM, SDULDQM, SDUUDQM, SDCKE) and SRAM ($\overline{\text{SRWR}}$, $\overline{\text{SRLLB}}$, $\overline{\text{SRLUB}}$, $\overline{\text{SRULB}}$, $\overline{\text{SRUUB}}$). Above setting is used the function register PJFC.

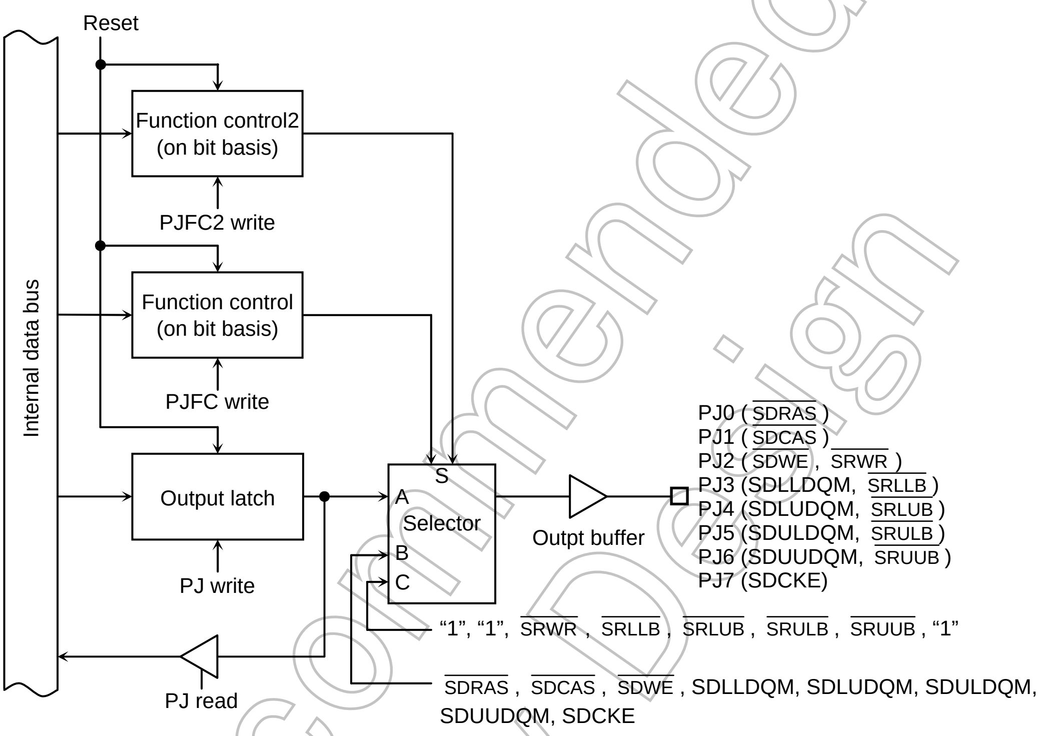


Figure 3.5.36 Port J

Port J Register

PJ (004CH)		7	6	5	4	3	2	1	0
	Bit symbol	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1

Port J Function Register

PJFC (004FH)		7	6	5	4	3	2	1	0
	Bit symbol	PJ7F	PJ6F	PJ5F	PJ4F	PJ3F	PJ2F	PJ1F	PJ0F
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	0: Port 1: SDCKE	0: Port 1: SDUUDQM	0: Port 1: SDULDQM	0: Port 1: SDLUDQM	0: Port 1: SDLLDQM	0: Port 1: $\overline{\text{SDWE}}$	0: Port 1: $\overline{\text{SDCAS}}$	0: Port 1: $\overline{\text{SDRAS}}$

Port J Function Register 2

PJFC2 (004DH)		7	6	5	4	3	2	1	0
	Bit symbol	–	PJ6F2	PJ5F2	PJ4F2	PJ3F2	PJ2F2	–	–
	Read/Write	W							
	After reset	0	0	0	0	0	0	0	0
	Function	Always write "0".	0: <PJ6F> 1: $\overline{\text{SRUUB}}$	0: <PJ5F> 1: $\overline{\text{SRULB}}$	0: <PJ4F> 1: $\overline{\text{SRLUB}}$	0: <PJ3F> 1: $\overline{\text{SRLLB}}$	0: <PJ2F> 1: $\overline{\text{SRWR}}$	Always write "0".	Always write "0".

Note: Read-modify-write is prohibited for the registers PJFC and PJFC2.

Figure 3.5.37 Register for Port J

3.5.15 Port K (PK0 to PK4, PK6)

Port K is 6-bit output port. Resetting sets the output latch PK to “1”, and port K pins output to “1”.

In addition to functioning as output ports, port K also functions as output pins for LCD controller (D1BSCP, D2BLP, D3BFR, DLEBCD and DOFFB), output pins for RTC alarm ($\overline{\text{ALARM}}$) and output pin for melody/alarm generator (MLDALM, $\overline{\text{MLDALM}}$). Above setting is used the function register PKFC.

Only PK6 has two output function which $\overline{\text{ALARM}}$ and $\overline{\text{MLDALM}}$. This selection is used PK<PK6>. Resetting resets the function register PKFC to “0”, and sets all ports to output ports.

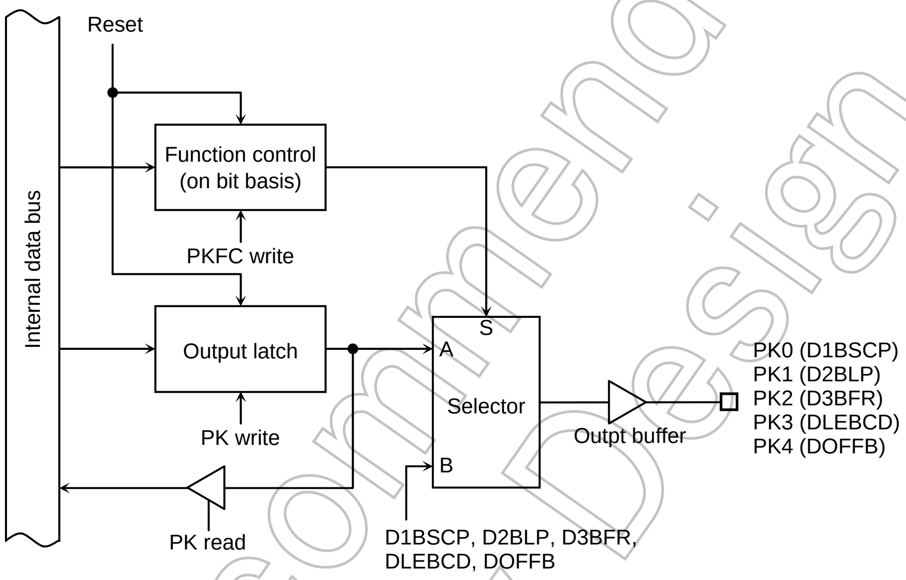


Figure 3.5.38 Port K (PK0 to PK4)

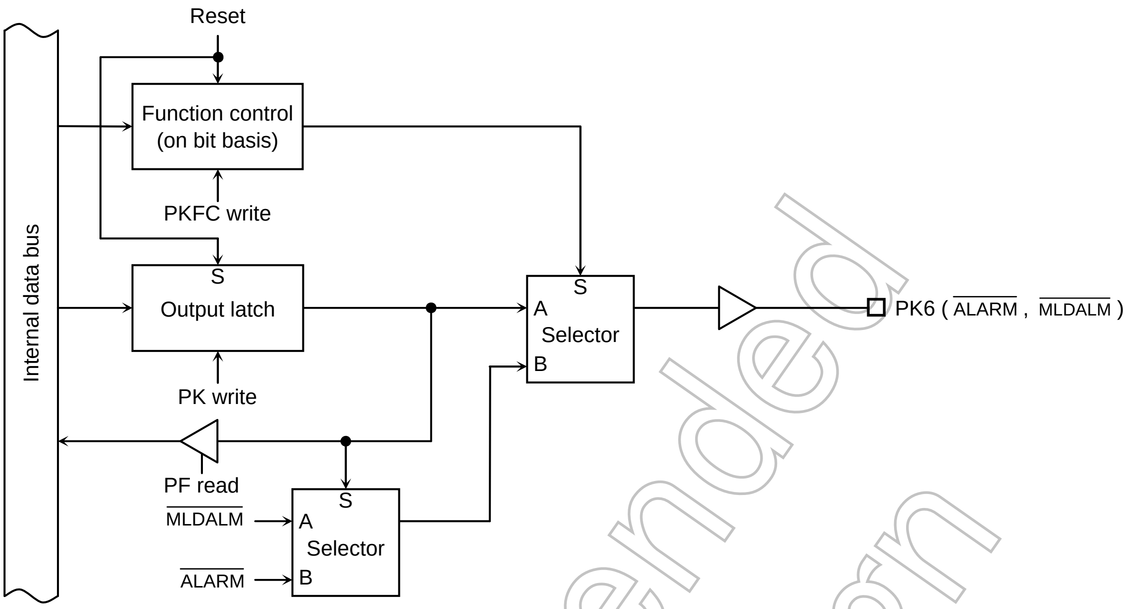


Figure 3.5.39 Port K (PK6)

Port K Register

PK (0050H)		7	6	5	4	3	2	1	0
	Bit symbol		PK6		PK4	PK3	PK2	PK1	PK0
	Read/Write		R/W		R/W				
	After reset		1		1	1	1	1	1

Port K Function Register

PKFC (0053H)		7	6	5	4	3	2	1	0
	Bit symbol		PK6F		PK4F	PK3F	PK2F	PK1F	PK0F
	Read/Write		W		W				
	After reset		0		0	0	0	0	0
	Function		0: Port 1: $\overline{\text{ALARM}}$ at <PK6> = 1 1: $\overline{\text{MLDALM}}$ at <PK6> = 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 register PKFC.

Figure 3.5.40 Register for Port K

3.5.16 Port L (PL0 to PL7)

PL0 to PL7 are 8-bit general-purpose I/O ports.

Each bit can be set individually for input or output using the control register PLCR. Resetting, the control register PLCR to "0" and sets port L to input ports.

It also sets all bits of the output latch register to "1". In addition to functioning as a general-purpose I/O port, port L can also function as a data bus for LCD controller (LD0 to LD7). Above setting is used the function register PLFC.

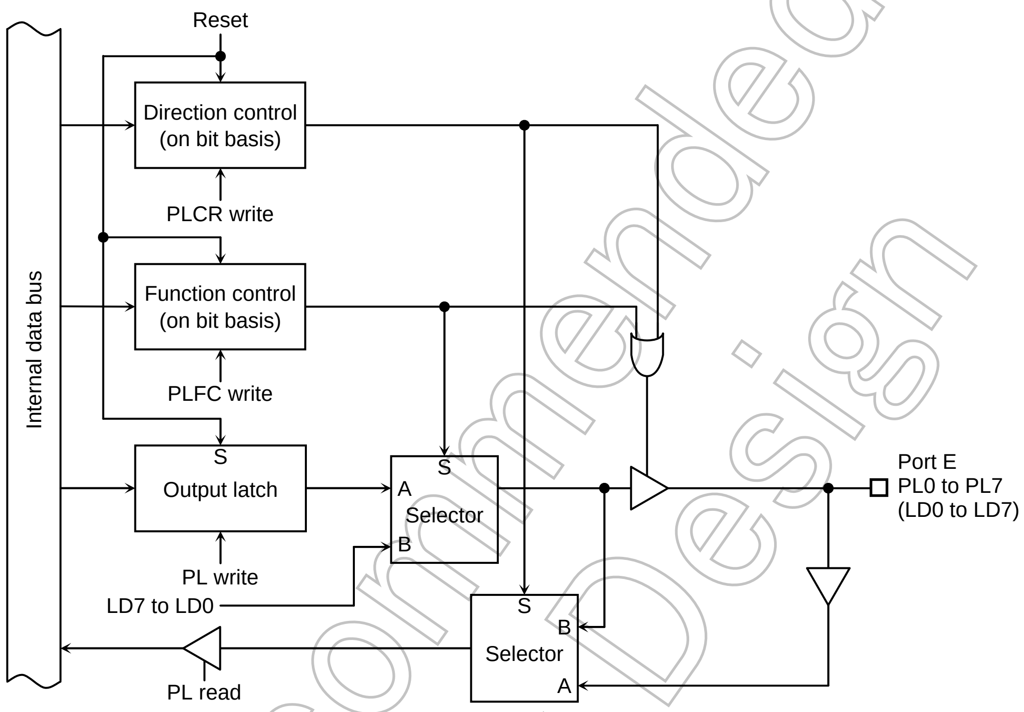


Figure 3.5.41 Port L

Port L Register

PL (0054H)

	7	6	5	4	3	2	1	0
Bit symbol	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
Read/Write	R/W							
After reset	Data from external port (Output latch register is set to 1)							

Port L Control Register

PLCR (0056H)

	7	6	5	4	3	2	1	0
Bit symbol	PL7C	PL6C	PL5C	PL4C	PL3C	PL2C	PL1C	PL0C
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0: Input 1: Output							

Port L Function Register

PLFC (0057H)

	7	6	5	4	3	2	1	0
Bit symbol	PL7F	PL6F	PL5F	PL4F	PL3F	PL2F	PL1F	PL0F
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	0: Port 1: Data bus for LCDC (LD7 to LD0)							

Figure 3.5.42 Register for Port L

3.6 Memory Controller

3.6.1 Functions

TMP92C820 has a memory controller with a variable 4-block address area that controls as follows.

(1) 4-block address area support

Specifies a start address and a block size for 4-block address area (Block 0 to block 5).

(2) Connecting memory specifications

Specifies SRAM, ROM as memories to connect with the selected address areas.

(3) Data bus size selection

Whether 8 bits, 16 bits or 32 bits is selected as the data bus size of the respective block address areas.

(4) Wait control

Wait specification bit in the control register and $\overline{\text{WAIT}}$ input pin control the number of waits in the external bus cycle. Read cycle and write cycle can specify the number of waits individually. The number of waits is controlled in five mode mentioned below.

0 waits, 1 wait, 2 waits, 3 waits N waits (control with $\overline{\text{WAIT}}$ pin)

3.6.2 Control Register and Operation after Reset Release

This section describes the registers to control the memory controller, the state after reset release and necessary settings.

(1) Control register

The control registers of the memory controller are as follows.

- Control register: BnCSH/BnCSL (n = 0 to 3, EX)
  Sets the basic functions of the memory controller, that is the connecting memory type, the number of waits to be read and written.
- Memory start address register: MSARn (n = 0 to 3)
  Sets a start address in the selected address areas.
- Memory address mask register: MAMR (n = 0 to 3)
  Sets a block size in the selected address areas.

In addition to setting of the above-mentioned registers, it is necessary to set the following registers to control ROM page mode access.

- Page ROM control register: PMEMCR
  Sets to executed ROM page mode accessing.

(2) Operation after reset release

The start data bus size is determined depending on the state of AM1/AM0 pins just after reset release. Then, the external memory is accessed as follows:

AM1	AM0	Start Mode
0	0	Don't use this setting
0	1	Start with 16-bit data bus
1	0	Start with 32-bit data bus
1	1	Don't use this setting

AM1/AM0 pins are valid only just after reset release. In the other cases, the data bus width is set to the value set to BnBUS bit of the control register.

After reset, only control register (B2CSH/B2CSL) of the block address area 2 is automatically valid. The data bus width which is specified by AM1/AM0 pin is loaded to the bit to specify the bus width of the control register in the block address area 2. The block address area 2 is set to address 000000H to FFFFFFH after reset.

After reset release, the block address areas are specified by the memory start address register (MSARn) and the memory address mask register (MAMRn). Then the control register (BnCS) is set.

Set the enable bit (BnE) of the control register to "1" to enable the setting.

3.6.3 Basic Functions and Register Setting

In this section, setting of the block address area, the connecting memory, and the number of waits out of the memory controller's functions are described.

(1) Block address area specification

The block address area is specified by two registers.

The memory start address register (MSARn) sets the start address of the block address areas. The memory controller compares between the register value and the address every bus cycles. The address bit which is masked by the memory address mask register (MAMRn) is not compared by the memory controller. The block address area size is determined by setting the memory address mask register. The set value in the register is compared with the block address area on the bus. If the compared result is a match, the memory controller sets the chip select signal ($\overline{CSn}$) to "low".

(i) Setting memory start address register

The MS23 to MS16 bits of the memory start address register respectively correspond with addresses A23 to A16. The lower start address A15 to A0 are always set to address 0000H. Therefore the start address of the block address area are set to addresses 000000H to FF0000H every 64 Kbytes.

(ii) Setting memory address mask registers

The memory address mask register sets whether an address bit is compared or not. Set the register to "0" to compare, or to "1" not to compare.

The address bit to be set is depended on the block address area.

Block address area 0: A20 to A8

Block address area 1: A21 to A8

Block address area 2 to 3: A22 to A15

The above-mentioned bits are always compared. The block address area size is determined by the compared result.

The size to be set depending on the block address area is as follows.

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

Note: After reset release, only the control register of the block address area 2 is valid. The control register of the block address area 2 has <B2M> bit. Setting <B2M> bit to "0" sets the block address area 2 to addresses 000000H to FFFFFFH. Setting <B2M> bit to "1" specifies the start address and the address area size as it is in the other block address area.

(iii) Example of register setting

To set the block address area 512 bytes from address 110000H, set the register as follows.

MSAR1 Register

	7	6	5	4	3	2	1	0
Bit symbol	M1S23	M1S22	M1S21	M1S20	M1S19	M1S18	M1S17	M1S16
Specified value	0	0	0	1	0	0	0	1

M1S23 to M1S16 bits of the memory start address register MSAR1 correspond with address A23 to A16. A15 to A0 are set to "0". Therefore setting MSAR1 to the above-mentioned value specifies the start address of the block address area to address 110000H.

The start address is set as it is in the other block address areas.

MAMR1 Register

	7	6	5	4	3	2	1	0
Bit symbol	M1V21	M1V20	M1V19	M1V18	M1V17	M1V16	M1V15 to M1V9	M1V8
Specified value	0	0	0	0	0	0	0	1

M1V21 to M1V16 and M1V8 bits of the memory address mask register MAMR1 set whether address A21 to A16 and A8 are compared or not. Set the register to "0" to compare, or to "1" not to compare. M1V15 to M1V9 bits set whether address A15 to A9 are compared or not with 1 bit. A23 and A22 are always compared.

Setting the above-mentioned compares A23 to A9 with the values set as the start addresses. Therefore 512 bytes of addresses 110000H to 1101FFH are set as the block address area 1, and compared with the addresses on the bus. If the compared result is a match, the chip select signal $\overline{\text{CS1}}$ is set to "low".

The other block address area sizes are specified like this.

Similarly, A23 is always compared in block address areas 2 to 3. Whether A22 to A15 are compared or not is set to register.

Note: When the set block address area overlaps with the built-in memory area, or both two address areas overlap, the block address area is processed according to priority as follows.

Built-in I/O > Built-in memory > Block address area 0 > 1 > 2 > 3 > CSEX

also that any accessed areas outside the address spaces set by $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ are processed as the CSEX space. Therefore, settings of CSEX apply for the control of wait cycles, data bus width, etc,.

(2) Connection memory specification

Setting the BnOM1 to 0 bit of the control register (BnCSH) specifies the memory type to be connected with the block address areas. The interface signal is output according to the set memory as follows

BnOM1, BnOM0 Bit (BnCSH register)

BnOM1	BnOM0	Function
0	0	SRAM/ROM (Default)
0	1	(Reserved)
1	0	(Reserved)
1	1	SDRAM

SDRAM is set only in block address are 1.

(3) Data bus width specification

The data bus width is set for every block address area. The bus size is set by the BnBUS1 and BnBUS0 bits of the control register (BnCSH) as follows.

BnBUS Bit (BnCSH register)

BnBUS1	BnBUS0	Function
0	0	8-bit bus mode (Default)
0	1	16-bit bus mode
1	0	32-bit bus mode
1	1	(Reserved)

This way of changing the data bus size depending on the address being accessed is called "dynamic bus sizing". The part where the data is output to is depended on the data size, the bus width and the start address.

Note: Since there is a possibility of abnormal writing/reading of the data if two memories with different bus width are put in consecutive address, do not execute a access to both memories with one command.

Operand Data Size (Bit)	Operand Start Address	Memory Data Size (Bit)	CPU Address	CPU Data			
				D32 to D24	D23 to D16	D15 to D8	D7 to D0
8	4n + 0	8/16/32	4n + 0	xxxxx	xxxxx	xxxxx	b7 to b0
	4n + 1	8	4n + 1	xxxxx	xxxxx	xxxxx	b7 to b0
		16/32	4n + 1	xxxxx	xxxxx	b7 to b0	xxxxx
	4n + 2	8/16	4n + 2	xxxxx	xxxxx	xxxxx	b7 to b0
		32	4n + 2	xxxxx	b7 to b0	xxxxx	xxxxx
	4n + 3	8	4n + 3	xxxxx	xxxxx	xxxxx	b7 to b0
		16	4n + 3	xxxxx	xxxxx	b7 to b0	xxxxx
		32	4n + 3	b7 to b0	xxxxx	xxxxx	xxxxx
16	4n + 0	8	(1) 4n + 0	xxxxx	xxxxx	xxxxx	b7 to b0
			(2) 4n + 1	xxxxx	xxxxx	xxxxx	b15 to b8
		16/32	4n + 0	xxxxx	xxxxx	b15 to b8	b7 to b0
	4n + 1	8	(1) 4n + 1	xxxxx	xxxxx	xxxxx	b7 to b0
			(2) 4n + 2	xxxxx	xxxxx	xxxxx	b15 to b8
		16	(1) 4n + 1	xxxxx	xxxxx	b7 to b0	xxxxx
			(2) 4n + 2	xxxxx	xxxxx	xxxxx	b15 to b8
		32	4n + 1	xxxxx	b15 to b8	b7 to b0	xxxxx
	4n + 2	8	(1) 4n + 2	xxxxx	xxxxx	xxxxx	b7 to b0
			(2) 4n + 1	xxxxx	xxxxx	xxxxx	b15 to b8
		16	4n + 2	xxxxx	xxxxx	b15 to b8	b7 to b0
		32	4n + 2	b15 to b8	b7 to b0	xxxxx	xxxxx
	4n + 3	8	(1) 4n + 3	xxxxx	xxxxx	xxxxx	b7 to b0
			(2) 4n + 4	xxxxx	xxxxx	xxxxx	b15 to b8
		16	(1) 4n + 3	xxxxx	xxxxx	b7 to b0	xxxxx
			(2) 4n + 4	xxxxx	xxxxx	xxxxx	b15 to b8
		32	(1) 4n + 3	b7 to b0	xxxxx	xxxxx	xxxxx
			(2) 4n + 4	xxxxx	xxxxx	xxxxx	b15 to b8
32	4n + 0	8	(1) 4n + 0	xxxxx	xxxxx	xxxxx	b7 to b0
			(2) 4n + 1	xxxxx	xxxxx	xxxxx	b15 to b8
			(3) 4n + 2	xxxxx	xxxxx	xxxxx	b23 to b16
			(4) 4n + 3	xxxxx	xxxxx	xxxxx	b31 to b24
		16	(1) 4n + 0	xxxxx	xxxxx	b15 to b8	b7 to b0
			(2) 4n + 2	xxxxx	xxxxx	b31 to b24	b23 to b16
		32	4n + 0	b31 to b24	b23 to b16	b15 to b8	b7 to b0
	4n + 1	8	(1) 4n + 0	xxxxx	xxxxx	xxxxx	b7 to b0
			(2) 4n + 1	xxxxx	xxxxx	xxxxx	b15 to b8
			(3) 4n + 2	xxxxx	xxxxx	xxxxx	b23 to b16
			(4) 4n + 3	xxxxx	xxxxx	xxxxx	b31 to b24
		16	(1) 4n + 1	xxxxx	xxxxx	b7 to b0	xxxxx
			(2) 4n + 2	xxxxx	xxxxx	b23 to b16	b15 to b8
			(3) 4n + 4	xxxxx	xxxxx	xxxxx	b31 to b24
		32	(1) 4n + 1	b23 to b16	b15 to b8	b7 to b0	xxxxx
			(2) 4n + 4	xxxxx	xxxxx	xxxxx	b31 to b24
	4n + 2	8	(1) 4n + 2	xxxxx	xxxxx	xxxxx	b7 to b0
			(2) 4n + 3	xxxxx	xxxxx	xxxxx	b15 to b8
			(3) 4n + 4	xxxxx	xxxxx	xxxxx	b23 to b16
			(4) 4n + 5	xxxxx	xxxxx	xxxxx	b31 to b24
		16	(1) 4n + 2	xxxxx	xxxxx	b15 to b8	b7 to b0
			(2) 4n + 4	xxxxx	xxxxx	b31 to b24	b23 to b16
		32	(1) 4n + 2	b15 to b8	b7 to b0	xxxxx	xxxxx
			(2) 4n + 4	xxxxx	xxxxx	b31 to b24	b23 to b16
	4n + 3	8	(1) 4n + 3	xxxxx	xxxxx	xxxxx	b7 to b0
			(2) 4n + 4	xxxxx	xxxxx	xxxxx	b15 to b8
			(3) 4n + 5	xxxxx	xxxxx	xxxxx	b23 to b16
			(4) 4n + 6	xxxxx	xxxxx	xxxxx	b31 to b24
		16	(1) 4n + 3	xxxxx	xxxxx	b7 to b0	xxxxx
			(2) 4n + 4	xxxxx	xxxxx	b23 to b16	b15 to b8
			(3) 4n + 6	xxxxx	xxxxx	xxxxx	b31 to b24
		32	(1) 4n + 3	b7 to b0	xxxxx	xxxxx	xxxxx
			(2) 4n + 4	xxxxx	b31 to b24	b23 to b16	b15 to b8

xxxxx: During a read, data input to the bus is ignored. At write, the bus is at high impedance and the write strobe signal remains to non active.

(4) Wait control

The external bus cycle completes a wait of two states at least (100 ns at 20 MHz). Setting the BnWW2 to BnWW0 and BnWR2 to BnWR0 of the control register (BnCSL) specifies the number of waits in the read cycle and the write cycle. BnWW is set with the same method as BnWR.

BnWW/BnWR Bit (BnCSL register)

BnWW2 BnWR2	BnWW1 BnWR1	BnWW0 BnWR0	Function
0	0	1	2states (0 waits) access fixed mode
0	1	0	3states (1 wait) access fixed mode (Default)
1	0	1	4states (2 waits) access fixed mode
1	1	0	5states (3 waits) access fixed mode
1	1	1	6states (4 waits) access fixed mode
0	1	1	$\overline{\text{WAIT}}$ pin input mode
Others			(Reserved)

Note: When SDRAM is specified as a connecting memory, setting should be 4 states (2 waits) in RD cycle and 3 states (1 wait) in WR cycle.

(i) Waits number fixed mode

The bus cycle is completed with the set states. The number of states is selected from 2 states (0 waits) to 5 states (3 waits).

(ii) $\overline{\text{WAIT}}$ pin input mode

This mode samples the $\overline{\text{WAIT}}$ input pins. It continuously samples the $\overline{\text{WAIT}}$ pin state and inserts a wait if the pin is active. The bus cycle is minimum 2 states. The bus cycle is completed when the wait signal is non active ("High" level) at 2 states. The bus cycle extends if the wait signal is active at 2 states and more.

(5) Insert recovery cycle

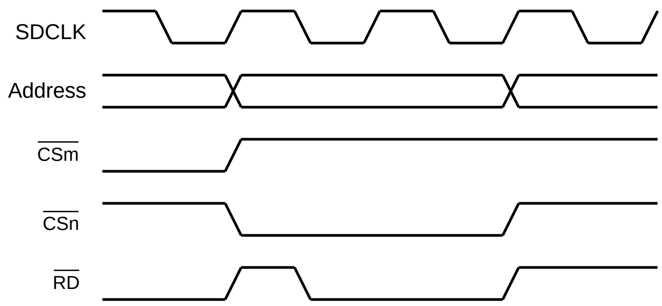
If a lot of connected pertain ROM and etc. (Much data output floating time (t_{DF})), each other's data-bus-output-recovery-time is trouble. However, by setting BnREC of control register (BnCSH), can insert dummy cycle of 1 state just before first bus cycle of starting access another block address.

BnREC Bit (BnCSH register)

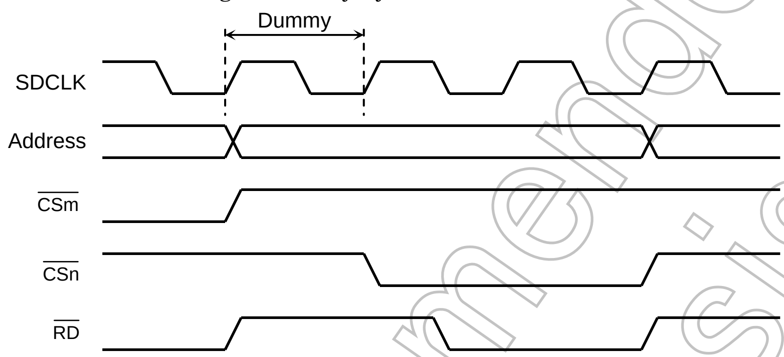
0	No dummy cycle is inserted (Default).
1	Dummy cycle is inserted.

Note: When use MMU, built-in RAM type LCDD, this function cannot use.

- When not inserting a dummy cycle (0 waits)

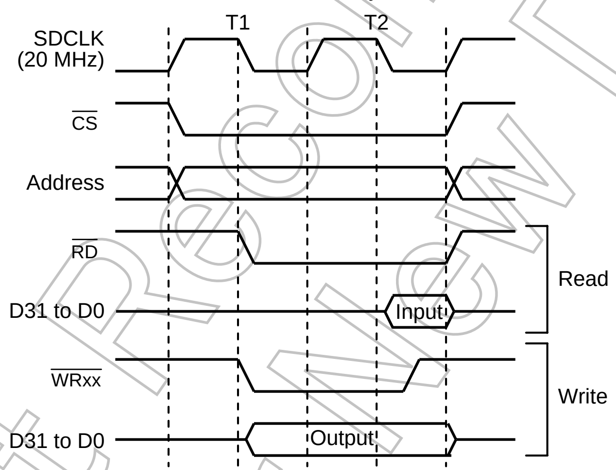


- When inserting a dummy cycle (0 waits)

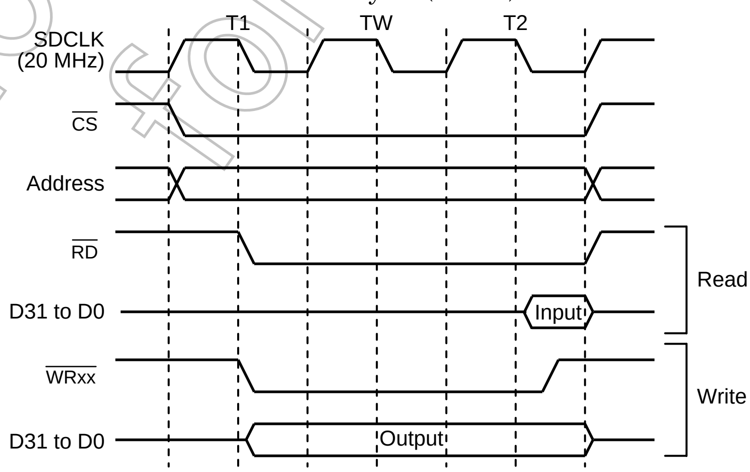


(6) Basic bus timing

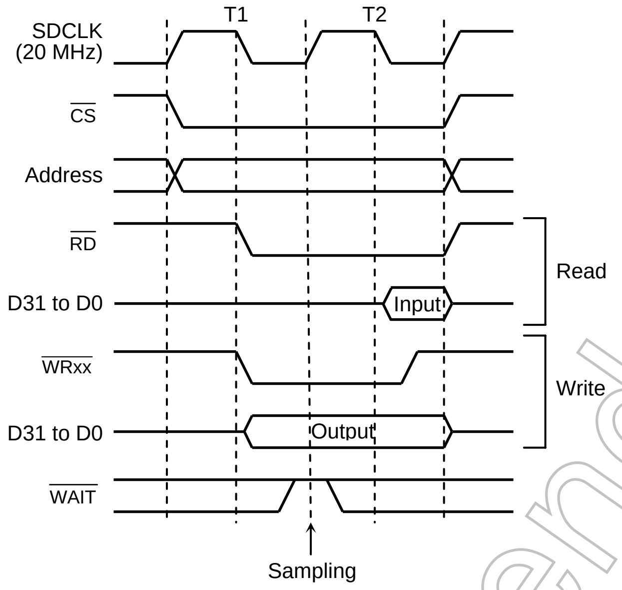
- External read/write bus cycle (0 waits)



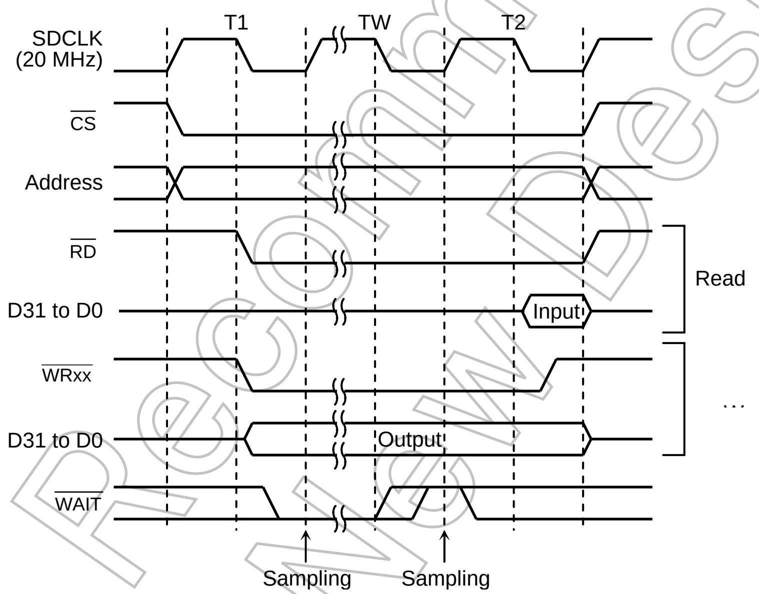
- External read/write bus cycle (1 wait)



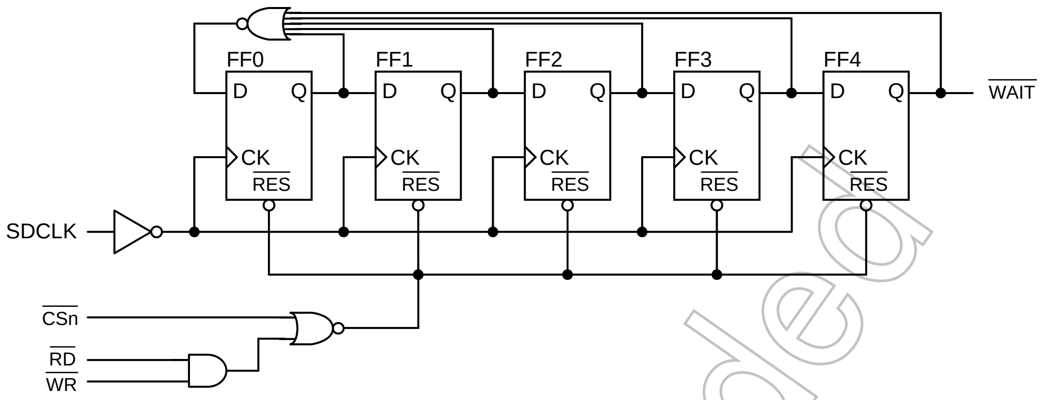
- External read/write bus cycle (0 waits at $\overline{\text{WAIT}}$ pin input mode)

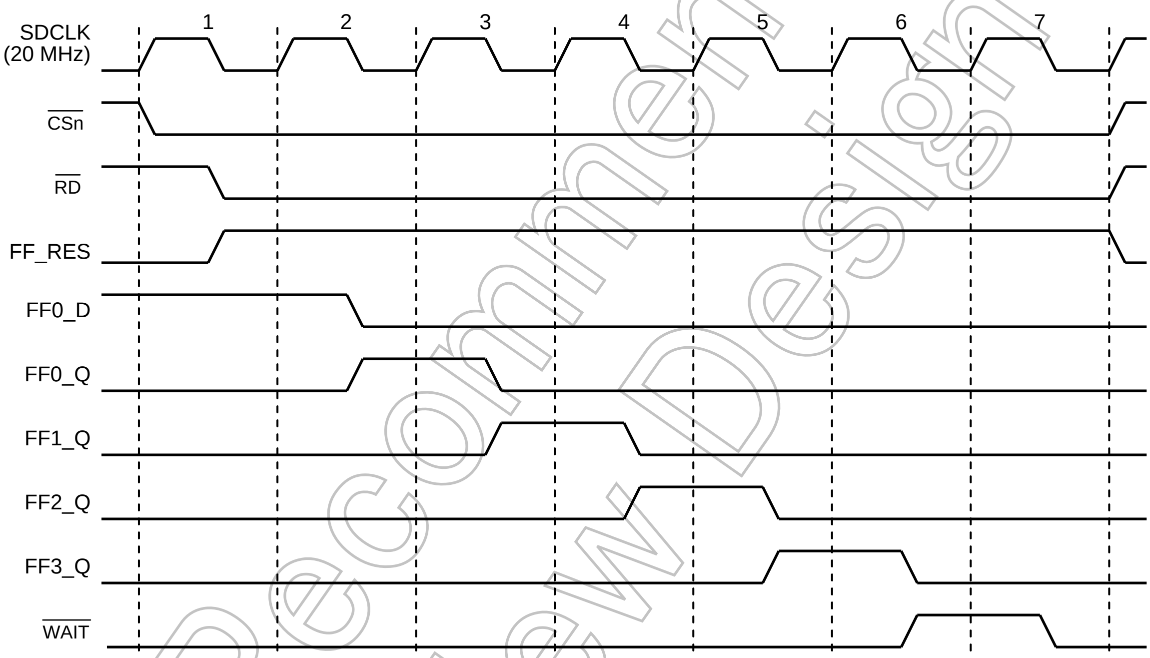


- External read/write bus cycle (n waits at $\overline{\text{WAIT}}$ pin input mode)



- Example of $\overline{\text{WAIT}}$ input cycle (5 waits)





3.6.4 ROM Control (Page mode)

This section describes ROM page mode accessing and how to set registers. ROM page mode is set by the page ROM control register.

(1) Operation and how to set the registers

TMP92C820 supports ROM access of the page mode. The ROM access of the page mode is specified only in the block address area 2.

ROM page mode is set by the page ROM control register (PMEMCR).

Setting OPGE bit of the PMEMCR register to "1" sets the memory access of the block address area to ROM page mode access.

The number of read cycles is set by the OPWR1 and OPWR0 bits of the PMEMCR register.

OPWR1/OPWR0 Bit (PMEMCR register)

OPWR1	OPWR0	Number of Cycle in a Page
0	0	1 state (n-1-1-1 mode) ($n \geq 2$)
0	1	2 state (n-2-2-2 mode) ($n \geq 3$)
1	0	3 state (n-3-3-3 mode) ($n \geq 4$)
1	1	(Reserved)

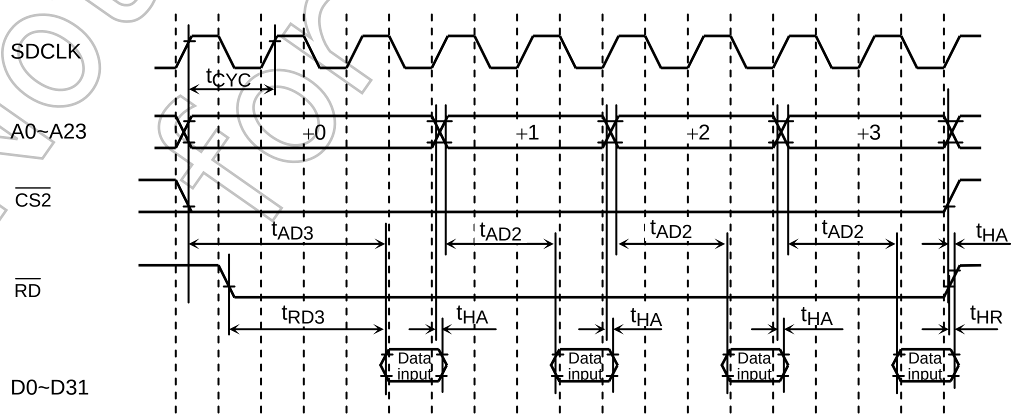
Note: Set the number of waits "n" to the control register (BnCSL) in each block address area.

The page size (the number of bytes) of ROM in the CPU size is set to the PR1 and 0 bit of the PMCME register. When data is read out until a border of the set page, the controller completes the page reading operation. The start data of the next page is read in the normal cycle. The following data is set to page read again.

PR1/PR0 Bit (PMEMCR register)

PR1	PR0	ROM Page Size
0	0	64 bytes
0	1	32 bytes
1	0	16 bytes (Default)
1	1	8 bytes

(2) Signal timing pulse



3.6.5 List of Registers

The memory control registers and the settings are described as follows. For the addresses of the registers, see Section 5 “Table of Special Function Registers (SFRs)”.

(1) Control registers

The control register is a pair of BnCSL and BnCSH. (n is a number of the block address area.) BnCSL has the same configuration regardless of the block address areas. In BnCSH, only B2CSH which is corresponded to the block address area 2 has a different configuration from the others.

BnCSL

	7	6	5	4	3	2	1	0
Bit symbol		BnWW2	BnWW1	BnWW0		BnWR2	BnWR1	BnWR0
Read/Write		W				W		
After reset		0	1	0		0	1	0

BnWW<2:0> Specifies the number of write waits.

- 001 = 2 states (0 waits) access
- 010 = 3 states (1 wait) access
- 101 = 4 states (2 waits) access
- 110 = 5 states (3 waits) access
- 111 = 6 states (4 waits) access
- 011 = $\overline{\text{WAIT}}$ pin input mode
- Others = (Reserved)

BnWR<2:0> Specifies the number of read waits.

- 001 = 2 states (0 waits) access
- 010 = 3 states (1 wait) access
- 101 = 4 states (2 waits) access
- 110 = 5 states (3 waits) access
- 111 = 6 states (4 waits) access
- 011 = $\overline{\text{WAIT}}$ pin input mode
- Others = (Reserved)

B2CSH

	7	6	5	4	3	2	1	0
Bit symbol	B2E	B2M		B2REC	B2OM1	B2OM0	B2BUS1	B2BUS0
Read/Write	W			W				
After reset	1	0		0	0	0	0/1	0/1

B2E: Enable bit

- 0 = No chip select signal output.
- 1 = Chip select signal output (Default).
- Note: After reset release, only the enable bit B2E of B2CS register is valid (“1”).

B2M: Block address area specification

- 0 = Sets the block address area of CS2 to addresses 000000H to FFFFFFH (Default).
- 1 = Sets the block address area of CS2 to programmable.
- Note: After reset release, the block address area 2 is set to addresses 000000H to FFFFFFH.

B2REC: Sets the dummy cycle for data output recovery time.

0 = Not insert a dummy cycle (Default).

1 = Insert a dummy cycle.

Note: When using MMU, LCD of built-in RAM type, this function cannot use.

B2OM<1:0>

00 = SRAM or ROM (Default)

Others = (Reserved)

B2BUS<1:0> Sets the data bus width.

00 = 8 bits (Default)

01 = 16 bits

10 = 32 bits

11 = (Reserved)

Note: The value of B2BUS bit is set according to the state of AM<1:0> pin after reset release.

BnCSH (n = 0, 1, 3)

	7	6	5	4	3	2	1	0
Bit symbol	BnE			BnREC	BnOM1	BnOM0	BnBUS1	BnBUS0
Read/Write	W			W				
After reset	0			0	0	0	0	0

BnE: Enable bit

0 = No chip select signal output (Default).

1 = Chip select signal output.

Note: After reset release, only the enable bit B2E of B2CS register is valid (“1”).

BnREC: Sets the dummy cycle for data output.

0 = Not insert a dummy cycle (Default).

1 = Insert a dummy cycle.

Note: When using MMU, LCD of built-in RAM type, this function cannot use.

BnOM<1:0>

00 = SRAM or ROM (Default)

01 = (Reserved)

10 = (Reserved)

11 = SDRAM

Note: SDRAM is set only by B1CSH.

BnBUS<1:0> Sets the data bus width.

00 = 8 bits (Default)

01 = 16 bits

10 = 32 bits

11 = (Reserved)

BEXCSL

	7	6	5	4	3	2	1	0
Bit symbol		BEXWW2	BEXWW1	BEXWW0		BEXWR2	BEXWR1	BEXWR0
Read/Write		W				W		
After reset		0	1	0		0	1	0

BEXWW<2:0> specifies the number of write waits.

001 = 2 states (0 waits) access
010 = 3 states (1 wait) access
101 = 4 states (2 waits) access
110 = 5 states (3 waits) access
111 = 6 states (4 waits) access
011 = $\overline{\text{WAIT}}$ pin input mode
Others = (Reserved)

BEXWR<2:0> Specifies the number of read waits.

001 = 2 states (0 waits) access
010 = 3 states (1 wait) access
101 = 4 states (2 waits) access
110 = 5 states (3 waits) access
111 = 6 states (4 waits) access
011 = $\overline{\text{WAIT}}$ pin input mode
Others = (Reserved)

BEXCSH

	7	6	5	4	3	2	1	0
Bit symbol					BEXOM1	BEXOM0	BEXBUS1	BEXBUS0
Read/Write					W			
After reset					0	0	0	0

BEXOM<1:0>

00 = SRAM or ROM (Default)
01 = (Reserved)
10 = (Reserved)
11 = (Reserved)

BEXBUS<1:0> Sets the data bus width.

00 = 8 bits (Default)
01 = 16 bits
10 = 32 bits
11 = (Reserved)

(2) Block address register

A start address and an address area of the block address are specified by the memory start address register (MSARn) and the memory address mask register (MAMRn). The memory start address register sets all start address similarly regardless of the block address areas.

The bit to be set by the memory address mask register is depended on the block address area.

MSARn (n = 0 to 3)

	7	6	5	4	3	2	1	0
Bit symbol	MnS23	MnS22	MnS21	MnS20	MnS19	MnS18	MnS17	MnS16
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

MnS<23:16> Sets a start address.

Sets the start address of the block address areas. The bits are corresponding to the address A23 to A16.

MAMR0

	7	6	5	4	3	2	1	0
Bit symbol	M0V20	M0V19	M0V18	M0V17	M0V16	M0V15	M0V14 to M0V9	M0V8
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

M0V<20:8>

Enables or masks comparison of the addresses. M0V20 to M0V8 are corresponding to addresses A20 to A8. The bits of M0V14 to M0V9 are corresponding to address A14 to A9 by 1 bit. If “0” is set, the comparison between the value of the address bus and the start address is enabled. If “1” is set, the comparison is masked.

MAMR1

	7	6	5	4	3	2	1	0
Bit symbol	M1V21	M1V20	M1V19	M1V18	M1V17	M1V16	M1V15 to M1V9	M1V8
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

M1V<21:8>

Enables or masks comparison of the addresses. M1V21 to M1V8 are corresponding to addresses A21 to A8. The bits of M1V15 to M1V9 are corresponding to address A15 to A9 by 1 bit. If “0” is set, the comparison between the value of the address bus and the start address is enabled. If “1” is set, the comparison is masked.

MAMRn (n = 2 to 3)

	7	6	5	4	3	2	1	0
Bit symbol	MnV22	MnV21	MnV20	MnV19	MnV18	MnV17	MnV16	MnV15
Read/Write	R/W							
After reset	1	1	1	1	1	1	1	1

MnV<22:15>

Enables or masks comparison of the addresses. MnV22 to MnV15 are corresponding to addresses A22 to A15. If “0” is set, the comparison between the value of the address bus and the start address is enabled. If “1” is set, the comparison is masked.

After a reset, MASR0 to MASR3 and MAMR0 to MAMR3 are set to “FFH”. B0CSH<B0E>, B1CSH<B1E>, and B3CSH<B3E> are reset to “0”. This disabling the CS0, CS1, and CS3 areas. However, B2CSH<B2M> is reset to “0” and B2CSH<B2E> to “1”, and CS2 is enabled 000000H to FFFFFFH. Also the bus width and number of waits specified in BEXCSH/L are used for accessing address except the specified CS0 to CS3 area.

(3) Page ROM control register (PMEMCR)

The page ROM control register sets page ROM accessing. ROM page accessing is executed only in block address area 2.

PMEMCR

	7	6	5	4	3	2	1	0
Bit symbol				OPGE	OPWR1	OPWR0	PR1	PR0
Read/Write				R/W				
After reset				0	0	0	1	0

OPGE enable bit

0 = No ROM page mode accessing (Default)

1 = ROM page mode accessing

OPWR<1:0> Specifies the number of waits.

00 = 1 state (n-1-1-1 mode) ($n \geq 2$) (Default)

01 = 2 states (n-2-2-2 mode) ($n \geq 3$)

10 = 3 states (n-3-3-3 mode) ($n \geq 4$)

11 = (Reserved)

Note: Set the number of waits "n" to the control register (BnCSL) in each block address area.

PR<1:0> ROM page size

00 = 64 bytes

01 = 32 bytes

10 = 16 bytes (Default)

11 = 8 bytes

Table 3.6.1 Control Register

Register		7	6	5	4	3	2	1	0
B0CSL (0140H)	Bit symbol		B0WW2	B0WW1	B0WW0		B0WR2	B0WR1	B0WR0
	Read/Write		W				W		
	After reset		0	1	0		0	1	0
B0CSH (0141H)	Bit symbol	B0E	–	–	B0REC	B0OM1	B0OM0	B0BUS1	B0BUS0
	Read/Write	W							
	After reset	0	0 (Note)	0 (Note)	0	0	0	0/1	0/1
MAMR0 (0142H)	Bit symbol	M0V20	M0V19	M0V18	M0V17	M0V16	M0V15	M0V14-V9	M0V8
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
MSAR0 (0143H)	Bit symbol	M0S23	M0S22	M0S21	M0S20	M0S19	M0S18	M0S17	M0S16
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
B1CSL (0144H)	Bit symbol		B1WW2	B1WW1	B1WW0		B1WR2	B1WR1	B1WR0
	Read/Write		W				W		
	After reset		0	1	0		0	1	0
B1CSH (0145H)	Bit symbol	B1E	–	–	B1REC	B1OM1	B1OM0	B1BUS1	B1BUS0
	Read/Write	W							
	After reset	0	0 (Note)	0 (Note)	0	0	0	0/1	0/1
MAMR1 (0146H)	Bit symbol	M1V21	M1V20	M1V19	M1V18	M1V17	M1V16	M1V15-V9	M1V8
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
MSAR1 (0147H)	Bit symbol	M1S23	M1S22	M1S21	M1S20	M1S19	M1S18	M1S17	M1S16
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
B2CSL (0148H)	Bit symbol		B2WW2	B2WW1	B2WW0		B2WR2	B2WR1	B2WR0
	Read/Write		W				W		
	After reset		0	1	0		0	1	0
B2CSH (0149H)	Bit symbol	B2E	B2M	–	B2REC	B2OM1	B2OM0	B2BUS1	B2BUS0
	Read/Write	W							
	After reset	1	0	0 (Note)	0	0	0	0/1	0/1
MAMR2 (014AH)	Bit symbol	M2V22	M2V21	M2V20	M2V19	M2V18	M2V17	M2V16	M2V15
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
MSAR2 (014BH)	Bit symbol	M2S23	M2S22	M2S21	M2S20	M2S19	M2S18	M2S17	M2S16
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
B3CSL (014CH)	Bit symbol		B3WW2	B3WW1	B3WW0		B3WR2	B3WR1	B3WR0
	Read/Write		W				W		
	After reset		0	1	0		0	1	0
B3CSH (014DH)	Bit symbol	B3E	–	–	B3REC	B3OM1	B3OM0	B3BUS1	B3BUS0
	Read/Write	W							
	After reset	0	0 (Note)	0 (Note)	0	0	0	0/1	0/1
MAMR3 (014EH)	Bit symbol	M3V22	M3V21	M3V20	M3V19	M3V18	M3V17	M3V16	M3V15
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
MSAR3 (014FH)	Bit symbol	M3S23	M3S22	M3S21	M3S20	M3S19	M3S18	M3S17	M3S16
	Read/Write	R/W							
	After reset	1	1	1	1	1	1	1	1
BEXCSH (0159H)	Bit symbol					BEXOM1	BEXOM0	BEXBUS1	BEXBUS0
	Read/Write					W			
	After reset					0	0	0	0
BEXCSL (0158H)	Bit symbol		BEXWW2	BEXWW1	BEXWW0		BEXWR2	BEXWR1	BEXWR0
	Read/Write		W				W		
	After reset		0	1	0		0	1	0
PMEMCR (0166H)	Bit symbol				OPGE	OPWR1	OPWR0	PR1	PR0
	Read/Write				R/W				
	After reset				0	0	0	1	0

Note: Always write “0”.

3.6.6 Cautions

(1) Note on timing between $\overline{CS}$ and $\overline{RD}$

If the parasitic capacitance of the read signal (Output enable signal) is greater than that of the chip select signal, it is possible that an unintended read cycle occurs due to a delay in the read signal. Such an unintended read cycle may cause a trouble as in the case of (a) in Figure 3.6.1

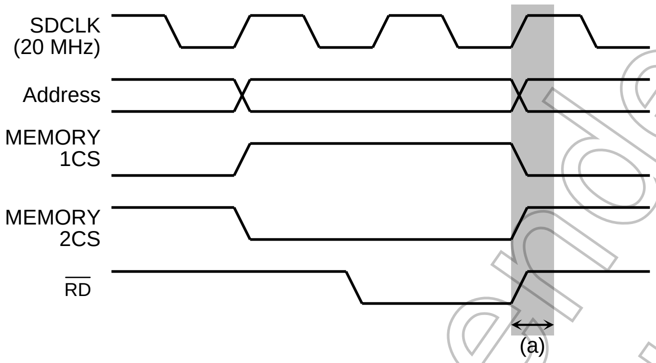


Figure 3.6.1 Read Signal Delay Read Cycle

Example: When using an externally connected flash EEPROM which users JEDEC standard commands, note that the toggle bit may not be read out correctly. If the read signal in the cycle immediately preceding the access to the flash EEPROM does not go high in time, as shown in Figure 3.6.2 an unintended read cycle like the one shown in (b) may occur.

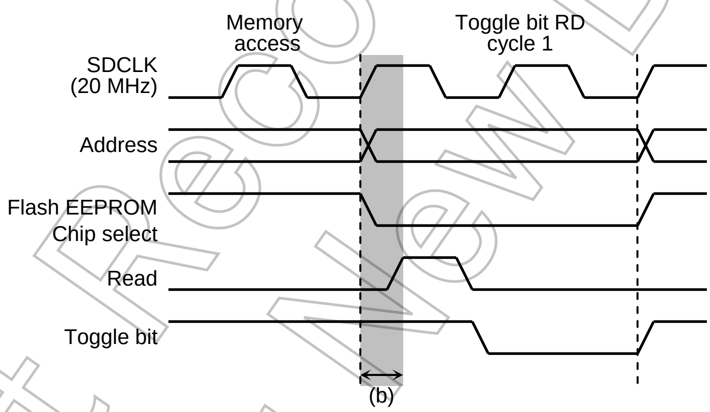


Figure 3.6.2 Flash EEPROM Toggle Bit Read Cycle

When the toggle bit reverse with this unexpected read cycle, TMP92C820 always reads same value of the toggle bit, and cannot read the toggle bit correctly.

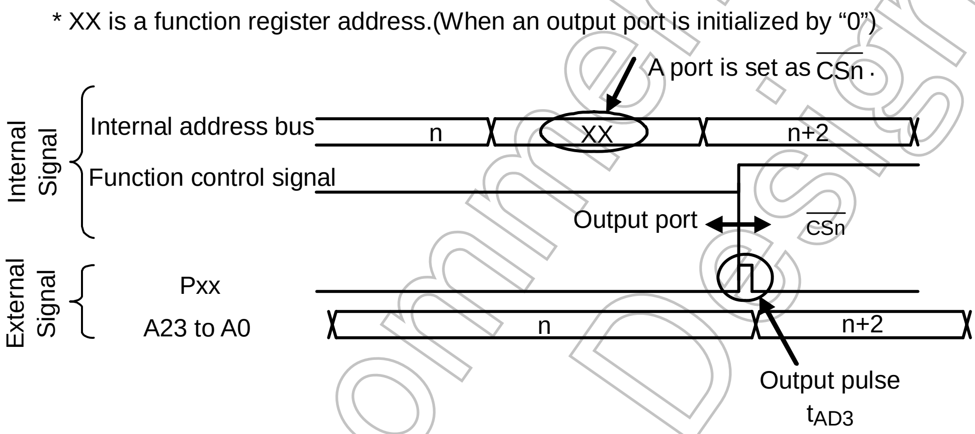
To avoid this phenomena, the data polling control recommended.

(2) The cautions at the time of the functional change of a $\overline{CSn}$.

A chip select signal output has the case of a combination terminal with a general-purpose port function. In this case, an output latch register and a function control register are initialized by the reset action, and an object terminal is initialized by the port output ("1" or "0") by it.

Functional change

Although an object terminal is changed from a port to a chip select signal output by setting up a function control register (PnFC register), the short pulse for several ns may be outputted to the changing timing. Although it does not become especially a problem when using the usual memory, it may become a problem when using a special memory.

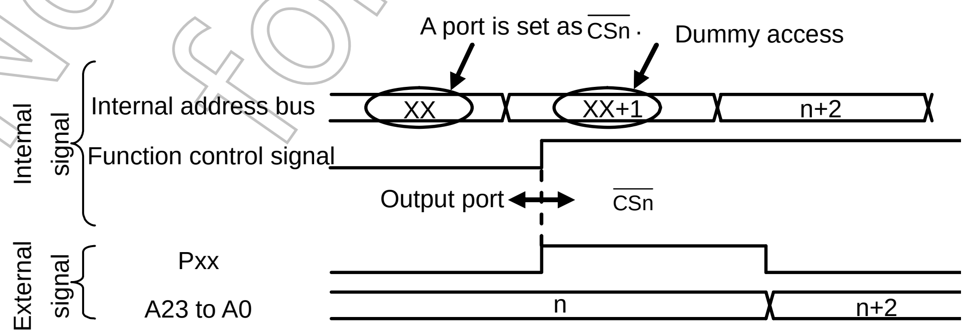


The measure by software

The countermeasures in S/W for avoiding this phenomenon are explained.

Since CS signal decodes the address of the access area and is generated, an unnecessary pulse is outputted by access to the object CS area immediately after setting it as a CSn function. Then, if internal area is accessed also immediately after setting a port as CS function, an unnecessary pulse will not output.

1. The ban on interruption under functional change (DI command)
2. A dummy command is added in order to carry out continuous internal access.
3. (Access to a functional change register is corresponded by 16-bit command. (LDW command))



3.7 8-Bit Timers (TMRA)

The TMP92C820 features 4 built-in 8-bit timers.

These timers are paired into four modules: TMRA01 and TMRA23. Each module consists of two channels and can operate in any of the following four operating modes.

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

Figure 3.7.1 to Figure 3.7.2 Show block diagrams for TMRA01 and TMRA23.

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

The operation mode and timer flip-flops are controlled by five controls SFR (Special function register).

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

The contents of this chapter are as follows.

3.7.1 Block Diagrams
3.7.2 Operation of Each Circuit
3.7.3 SFRs
3.7.4 Operation in Each Mode

(1) 8-bit timer mode
(2) 16-bit timer mode
(3) 8-bit PPG (Programmable pulse generation) output mode
(4) 8-bit PWM output mode
(5) Mode setting

Table 3.7.1 Registers and Pins for Each Module

	Module	TMRA01	TMRA23
External pin	Input pin for external clock	TA0IN (shared with PC0)	No
	Output pin for timer flip-flop	TA1OUT (shared with PC1)	TA3OUT (Shared with PC5)
SFR (Address)	Timer run register	TA01RUN (1100H)	TA23RUN (1108H)
	Timer register	TA0REG (1102H) TA1REG (1103H)	TA2REG (110AH) TA3REG (110BH)
	Timer mode register	TA01MOD (1104H)	TA23MOD (110CH)
	Timer flip-flop control register	TA1FFCR (1105H)	TA3FFCR (110DH)

3.7.1 Block Diagrams

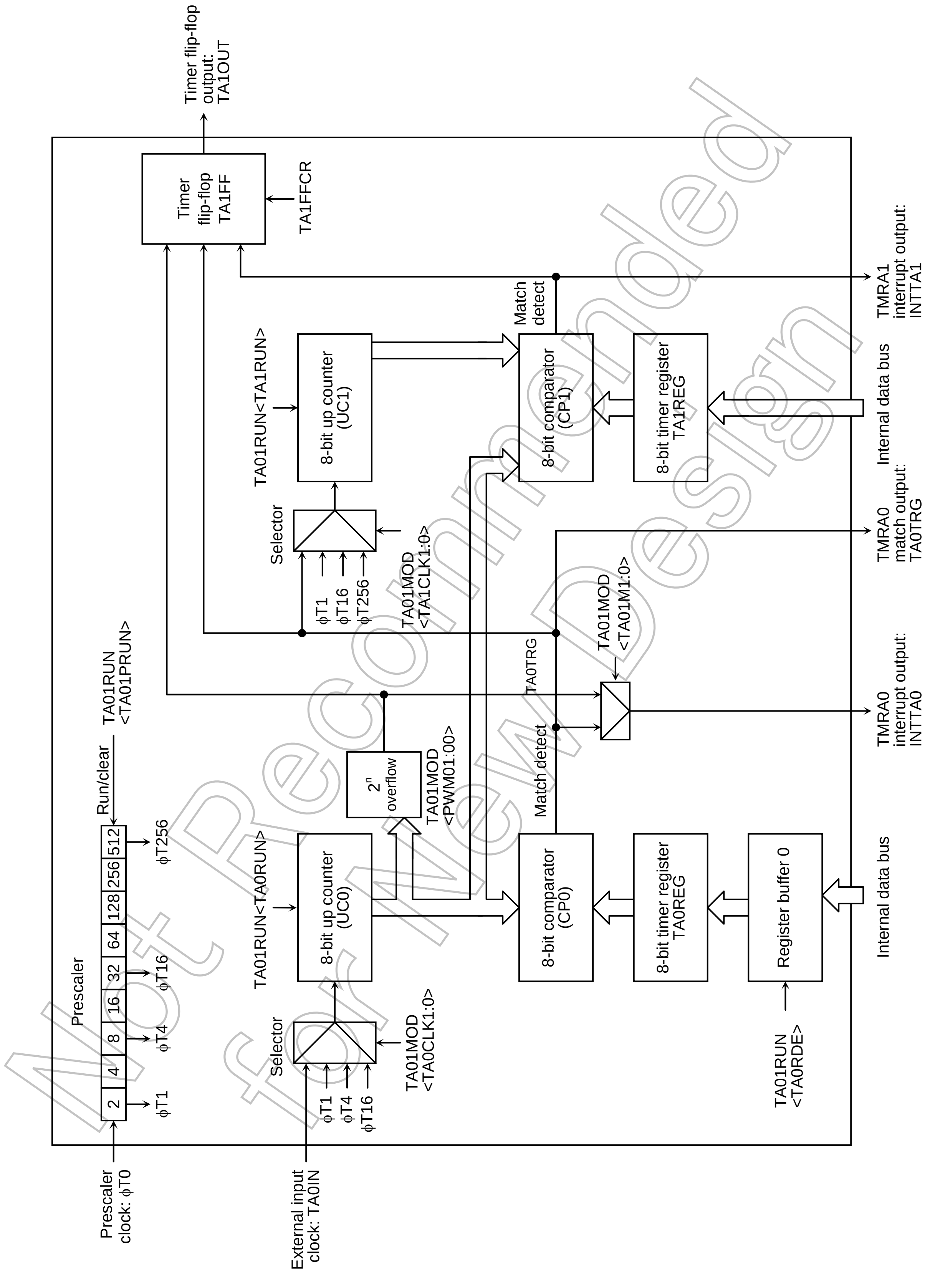


Figure 3.7.1 TMRA01 Block Diagram

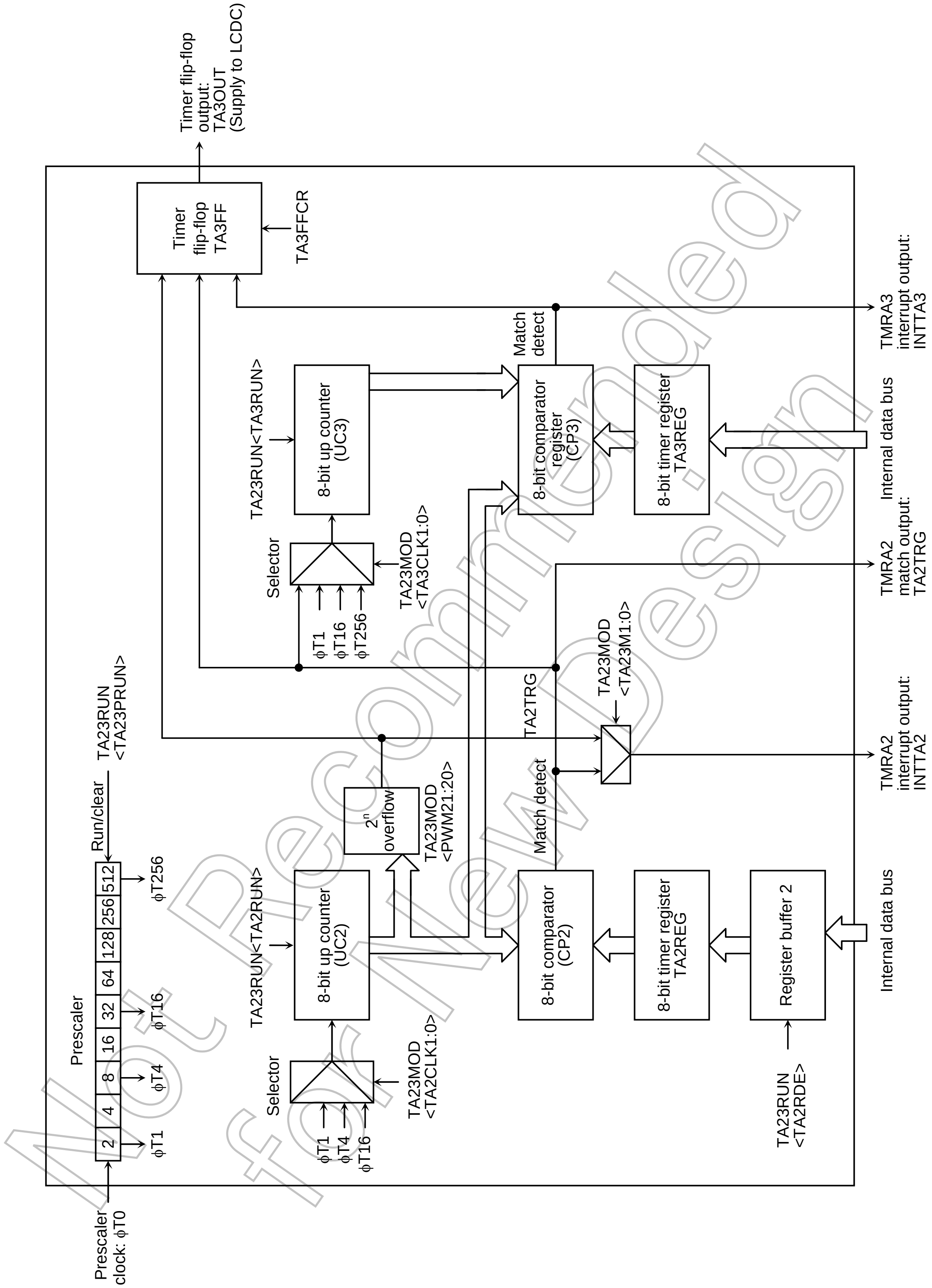


Figure 3.7.2 TMRA23 Block Diagram

3.7.2 Operation of Each Circuit

(1) Prescalers

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

The clock ϕT0 is divided into 8 by the CPU clock f_{SYS} and input to this prescaler.

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

Table 3.7.2 Prescaler Output Clock Resolution

Clock gear selection SYSCR1 <GEAR2:0>	System clock selection SYSCR1 <SYSCK>	–	Timer counter input clock TMRA prescaler TAxMOD<TAxCLK1:0>			
			ϕT1(1/2)	ϕT4(1/8)	ϕT16(1/32)	ϕT256(1/512)
–	1 (fs)	1/8	fs/16	fs/64	fs/256	fs/4096
000 (1/1)	0 (fc)		fc/16	fc/64	fc/256	fc/4096
001 (1/2)			fc/32	fc/128	fc/512	fc/8192
010 (1/4)			fc/64	fc/256	fc/1024	fc/16384
011 (1/8)			fc/128	fc/512	fc/2048	fc/32768
100 (1/16)			fc/256	fc/1024	fc/4096	fc/65536

(2) Up counters (UC0 and UC1)

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

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

The input clock for UC1 depends on the operation mode. In 16-bit timer mode, the overflow output from UC0 is used as the input clock. In any mode other than 16-bit timer mode, the input clock is selectable and can either be one of the internal clocks ϕT1, ϕT16, or ϕT256, or the comparator output (The match detection signal) from TMRA0.

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

(3) Timer registers (TA0REG and TA1REG)

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

The TA0REG are double buffer structure, each of which makes a pair with register buffer.

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

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

A reset initializes <TA0RDE> to "0", disabling the double buffer. To use the double buffer, write data to the timer register, set <TA0RDE> to "1", and write the following data to the register buffer Figure 3.7.3 show the configuration of TA0REG.

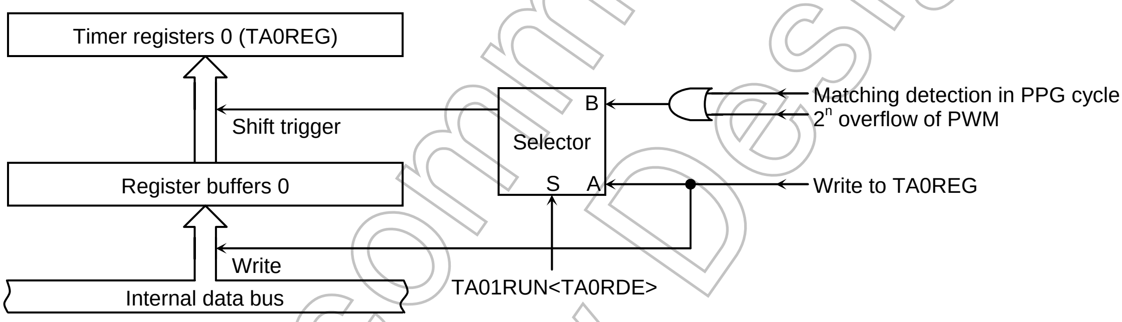


Figure 3.7.3 Configuration of TA0REG

Note: The same memory address is allocated to the timer register and the register buffer. When <TA0RDE> = 0, the same value is written to the register buffer and the timer register; when <TA0RDE> = 1, only the register buffer is written to.

The address of each timer register is as follows.

TA0REG: 001102H   TA1REG: 001103H
TA2REG: 00110AH   TA3REG: 00110BH

All these registers are write-only and cannot be read.

(4) Comparator (CP0)

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

(5) Timer flip-flop (TA1FF)

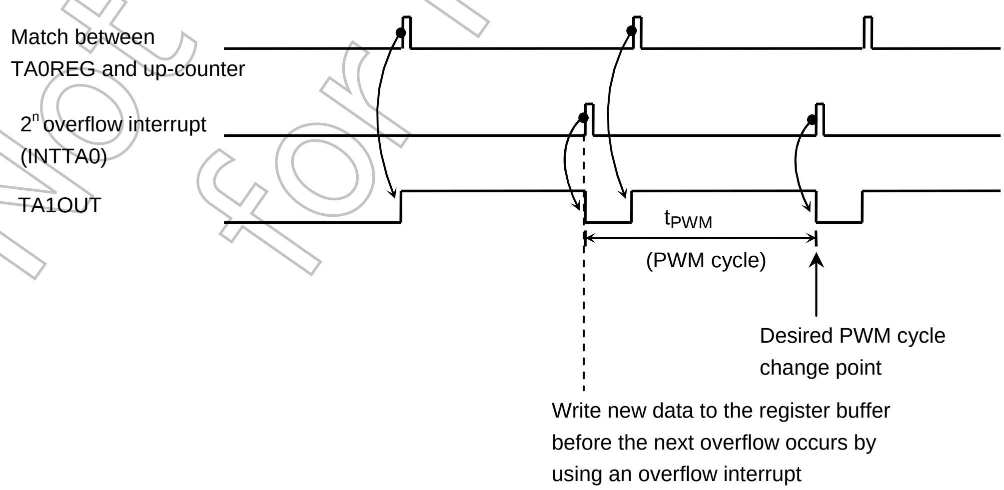
The timer flip-flop (TA1FF) is a flip-flop inverted by the match detects signal (8-bit comparator output) of each interval timer. Whether inversion is enabled or disabled is determined by the setting of the bit TA1FFCR<TA1FFIE> in the timer flip-flops control register.

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

The TA1FF signal is output via the TA1OUT pin (which can also be used as PC1). When this pin is used as the timer output, the timer flip-flop should be set beforehand using the port C function register PCFC.

Note: When the double buffer is enabled for an 8-bit timer in PWM or PPG mode, caution is required as explained below.
If new data is written to the register buffer immediately before an overflow occurs by a match between the timer register value and the up-counter value, the timer flip-flop may output an unexpected value.
For this reason, make sure that in PWM mode new data is written to the register buffer by six cycles ($f_{SYS} \times 6$) before the next overflow occurs by using an overflow interrupt.
When using PPG mode, make sure that new data is written to the register buffer by six cycles before the next cycle compare match occurs by using a cycle compare match interrupt.

Example when using PWM mode



3.7.3 SFRs

TMRA01 Run Register

TA01RUN (1100H)

	7	6	5	4	3	2	1	0
Bit symbol	TA0RDE				I2TA01	TA01PRUN	TA1RUN	TA0RUN
Read/Write	R/W				R/W	R/W		
After reset	0				0	0	0	0
Function	Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	TMRA01 prescaler 0: Stop and clear 1: Run (Count up)	UP counter (UC1)	UP counter (UC0)

Timer run/stop control

0	Stop and clear
1	Run (Count up)

TA0REG double buffer control

0	Disable
1	Enable

Note: The values of bits 4 to 6 of TA01RUN are undefined when read.

TMRA23 Run Register

TA23RUN (1108H)

	7	6	5	4	3	2	1	0
Bit symbol	TA2RDE				I2TA23	TA23PRUN	TA3RUN	TA2RUN
Read/Write	R/W				R/W	R/W		
After reset	0				0	0	0	0
Function	Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	TMRA23 prescaler 0: Stop and clear 1: Run (Count up)	UP counter (UC3)	UP counter (UC4)

Timer run/stop control

0	Stop and clear
1	Run (Count up)

TA2REG double buffer control

0	Disable
1	Enable

Note: The values of bits 4 to 6 of TA23RUN are undefined when read.

Figure 3.7.4 TMRA Registers (1)

TMRA01 Mode Register

TA01MOD (1104H)		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 (Note) 01: φT1 10: φT4 11: φT16	

TMRA0 source clock selection

00	TA0IN (External input)
01	φT1
10	φT4
11	φT16

TMRA1 source clock selection

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

PWM cycle selection

00	Reserved
01	$2^6 \times$ Source clock
10	$2^7 \times$ Source clock
11	$2^8 \times$ Source clock

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)

Note: When set TA0IN pin, must set TA01MOD after set port C.

Figure 3.7.5 TMRA Registers (2)

TMRA23 Mode Register

TA23MOD (110CH)		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
10	φT4
11	φT16

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 × Source clock
10	2^7 × Source clock
11	2^8 × Source clock

TMRA23 operation mode selection

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

Figure 3.7.6 TMRA Registers (3)

TMRA1 Flip-Flop Control Register

TA1FFCR (1105H)

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	TA1FF1 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

Note: The values of bits 4 to 7 of TA1FFCR are undefined when read.

Figure 3.7.7 TMRA Registers (4)

TMRA3 Flip-Flop Control Register

TA3FFCR (110DH)

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		TA3FFcontrol 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

Note: The values of bits 4 to 7 of TA3FFCR are undefined when read.

Figure 3.7.8 TMRA Register

TMRA Register (TA0REG to TA3REG)

Symbol	Address	7	6	5	4	3	2	1	0
TA0REG	1102H	–							
		W							
		Undefined							
TA1REG	1103H	–							
		W							
		Undefined							
TA2REG	110AH	–							
		W							
		Undefined							
TA3REG	110BH	–							
		W							
		Undefined							

Note: Read-modify-write instruction is prohibited for above registers.

Figure 3.7.9 Register for 8-Bit Timers

3.7.4 Operation in Each Mode

(1) 8-bit timer mode

Both timer 0 and timer 1 can be used independently as 8-bit interval timers.

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

   To generate interrupts at constant intervals using timer 1 (INTTA1), first stop TMRA1 then set the operation mode, input clock and a cycle to TA01MOD and TA1REG register, respectively. Then, enable the interrupt INTTA1 and start TMRA1 counting.

Example: To generate an INTTA1 interrupt every 40 μs at f_C = 40 MHz, set each register as follows:

		7	6	5	4	3	2	1	0	
		MSB							LSB	
TA01RUN	←	–	X	X	X	–	–	0	–	Stop TMRA1 and clear it to 0.
TA01MOD	←	0	0	–	–	0	1	–	–	Select 8-bit timer mode and select φT1 (=(16/fc)s at f_C = 40 MHz) as the input clock.
TA1REG	←	0	1	1	0	0	1	0	0	Set TA1REG to 40 μs ÷ φT1 = 100 = 64H
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 3.7.3

Table 3.7.3 Selecting Interrupt Interval and the Input Clock Using 8-Bit Timer

Input Clock	Interrupt Interval (at f_{SYS} = 20 MHz)	Resolution
φT1 (8/f_{SYS})	0.4 μs to 102.4 μs	0.4 μs
φT4 (32/f_{SYS})	1.6 μs to 409.6 μs	1.6 μs
φT16 (128/f_{SYS})	6.4 μs to 1.638 ms	6.4 μs
φT256 (2048/f_{SYS})	102.4 μs to 26.21 ms	102.4 μs

Note: The input clocks for TMRA0 and TMRA1 differ as follows:
TMRA0: Uses TMRA0 input (TA0IN) and can be selected from φT1, φT4, or φT16
TMRA1: Match output of TMRA0 (TA0TRG) and can be selected from φT1, φT16, φT256

2. Generating a 50% duty ratio square wave pulse

The state of the timer flip-flop (TA1FF1) is inverted at constant intervals and its status output via the timer output pin (TA1OUT).

Example: To output a 2.4 μs square wave pulse from the TA1OUT pin at f_C = 40 MHz, use the following procedure to make the appropriate register settings. This example uses timer 1; however, either timer 0 or timer 1 may be used.

		7	6	5	4	3	2	1	0	
TA01RUN	←	–	X	X	X	–	–	0	–	Stop TMRA1 and clear it to 0.
TA01MOD	←	0	0	–	–	0	1	–	–	Select 8-bit timer mode and select φT1 (=(16/fc)s at f_C = 40 MHz) as the input clock.
TA1REG	←	0	0	0	0	0	0	1	1	Set the timer register to 2.4 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 detect signal from timer 1.
PCCR	←	X	–	–	X	–	X	1	–	Set PC1 to function as the TA1OUT pin.
PCFC	←	X	–	–	X	–	X	1	–	(Set PC1 to function as the TA1OUT pin.)
TA01RUN	←	–	X	X	X	–	1	1	–	Start TMRA1 counting.

X: Don't care, –: No change

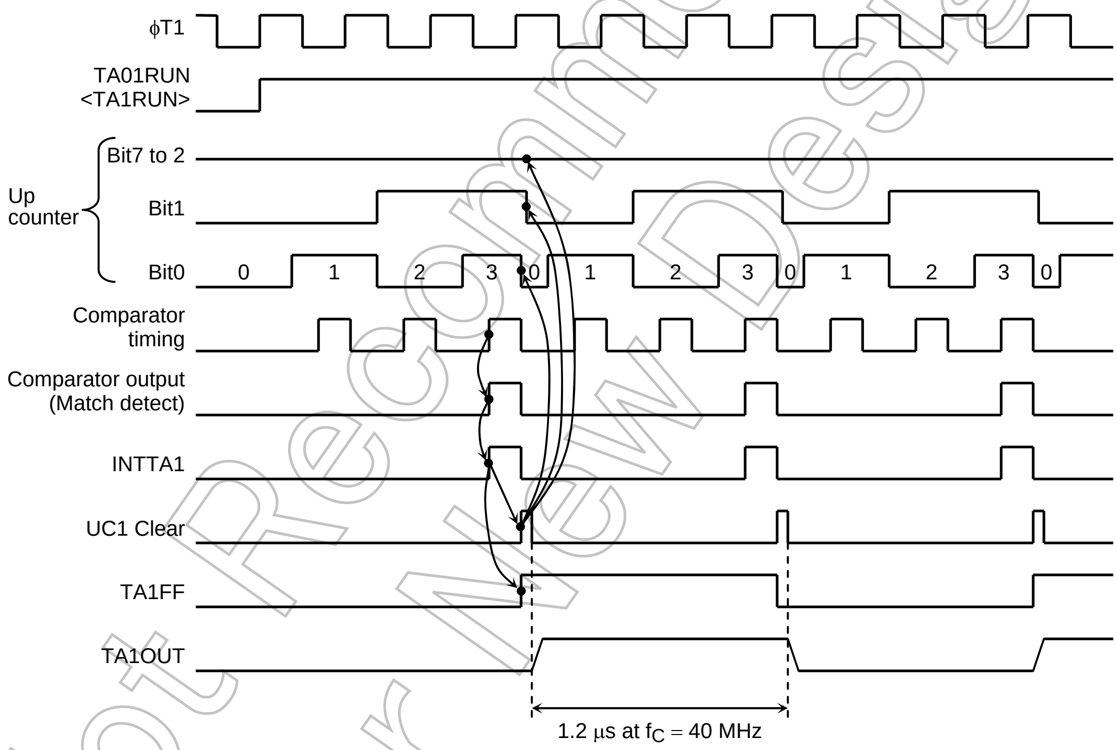


Figure 3.7.10 Square Wave Output Timing Chart (50% duty)

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

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

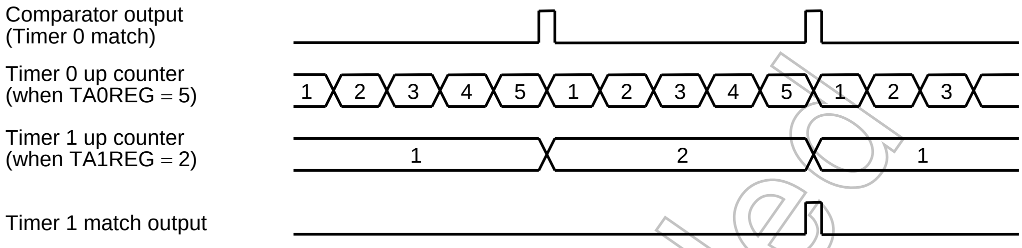


Figure 3.7.11 TMRA1 Count Up on Signal from TMRA0

(2) 16-bit timer mode

A 16-bit interval timer is configured by pairing the two 8-bit timers TMRA0 and TMRA1. To make a 16-bit interval timer in which TMRA0 and TMRA1 are cascaded together, set TA01MOD<TA01M1:0> to 01.

In 16-bit timer mode, the overflow output from TMRA0 is used as the input clock for TMRA1, regardless of the value set in TA01MOD<TA1CLK1:0>. Table 3.7.4 shows the relationship between the timer (Interrupt) cycle and the input clock selection.

To set the timer interrupt interval, set the lower eight bits in timer register TA0REG and the upper eight bits in TA1REG. Be sure to set TA0REG first (as entering data in TA0REG temporarily disables the compare, while entering data in TA1REG starts the compare).

Example: To generate an INTTA1 interrupt every 0.4 s at f_C = 40 MHz, set the timer registers TA0REG and TA1REG as follows:

If φT16 (=(256/fc)s at f_{SYS} = 20 MHz) is used as the input clock for counting, set the following value in the registers: 0.4 s ÷ (256/fc)s = 62500 = F424H; e.g., set TA1REG to F4H and TA0REG to 24H.

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

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

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

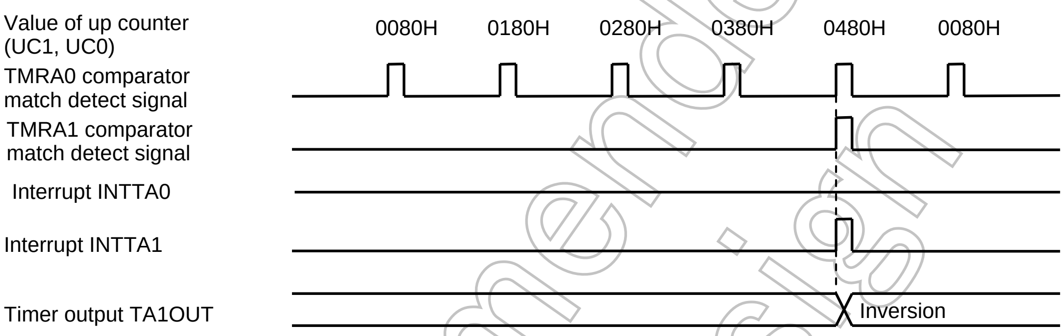


Figure 3.7.12 Timer Output by 16-Bit Timer Mode

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

Square wave pulses can be generated at any frequency and duty ratio by TMRA0. The output pulses may be active low or active high. In this mode TMRA1 cannot be used. TMRA0 outputs pulses on the TA1OUT pin (which can also be used as PC1).

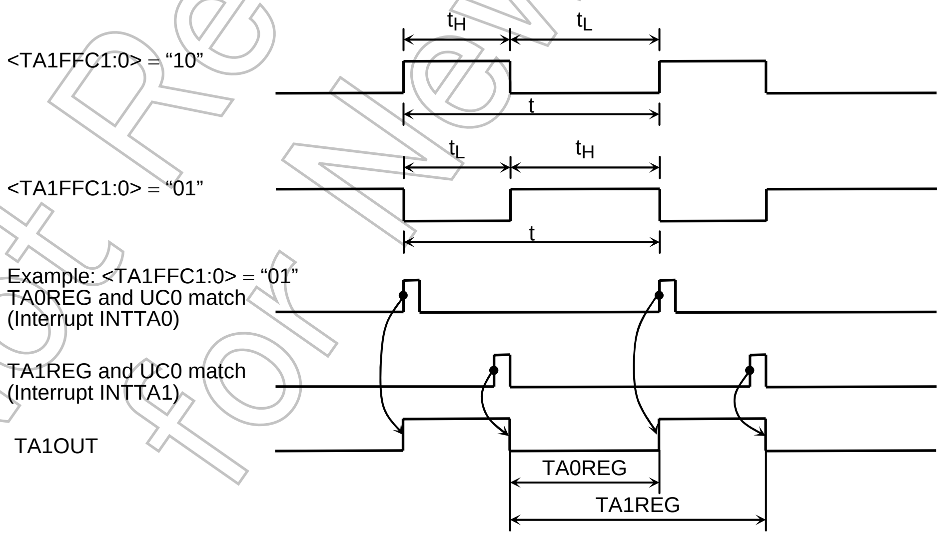


Figure 3.7.13 8-Bit PPG Output Waveforms

In this mode a programmable square wave is generated by inverting the timer output each time the 8-bit up counter (UC0) matches the value in one of the timer registers TA0REG or TA1REG.

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

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

Figure 3.7.14 shows a block diagram representing this mode.

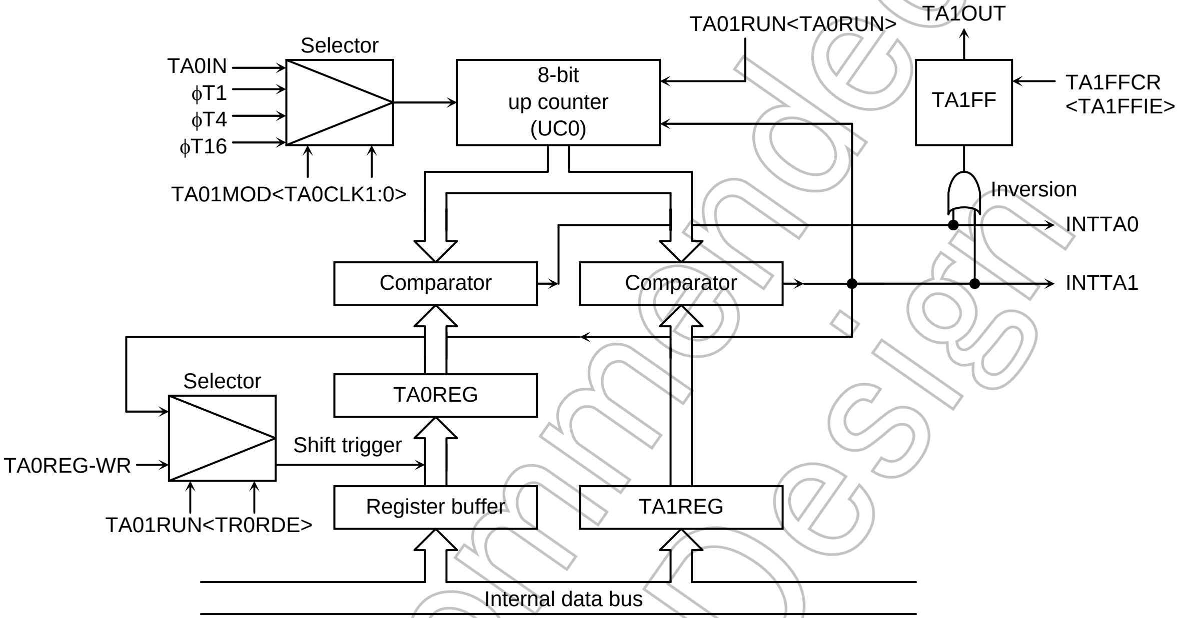


Figure 3.7.14 Block Diagram of 8-Bit PPG Output Mode

If the TA0REG double buffer is enabled in this mode, the value of the register buffer will be shifted into TA0REG each time TA1REG matches UC0.

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

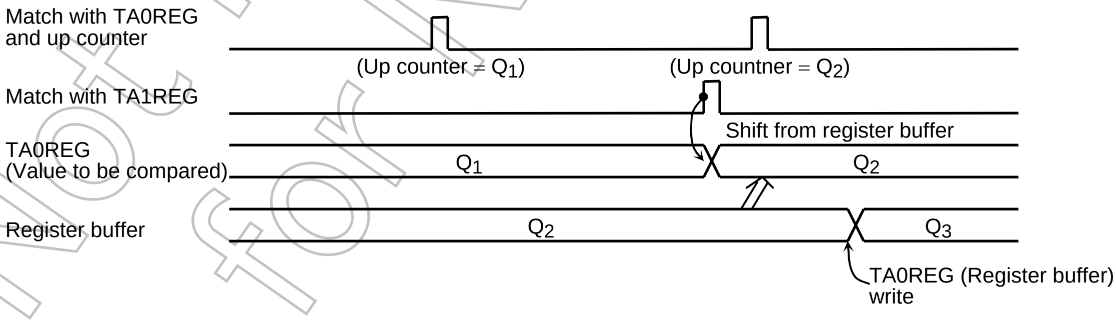
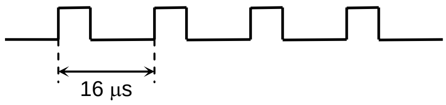


Figure 3.7.15 Operation of Register Buffer

Example: To generate 1/4 duty 62.5 kHz pulses (at f_C = 40 MHz):



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

To obtain a frequency of 62.5 kHz, the pulse cycle t should be: t = 1/62.5 kHz = 16 μs

ϕT1 (=(16/fc)) (at f_C = 40 MHz);

16 μs ÷(16/fc)s = 40

Therefore set TA1REG to 40 (28H)

The duty is to be set to 1/4: t × 1/4 = 16 μs × 1/4 = 4 μs

4 μs ÷ (16/fc)s = 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 TMRA1 and clear it to "0".
TA01MOD	←	1	0	–	–	–	–	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	–	Set TA1FF, enabling both inversion and the double buffer. 10 generates a negative logic pulse.
PCCR	←	X	–	–	X	–	X	1	–	Set PC1 as the TA1OUT pin.
PCFC	←	X	–	–	X	–	X	1	–	Set PC1 as the TA1OUT pin.
TA01RUN	←	1	X	X	X	–	1	1	1	Start TMRA0 and TMRA1 counting.

X: Don't care, –: No change

(4) 8-bit PWM output mode

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

When TMRA0 is used the PWM pulse is output on the TA1OUT pin (which is also used as PC1). TMRA1 can also be used as an 8-bit timer.

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

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

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

Value set in TA0REG $\neq$ 0

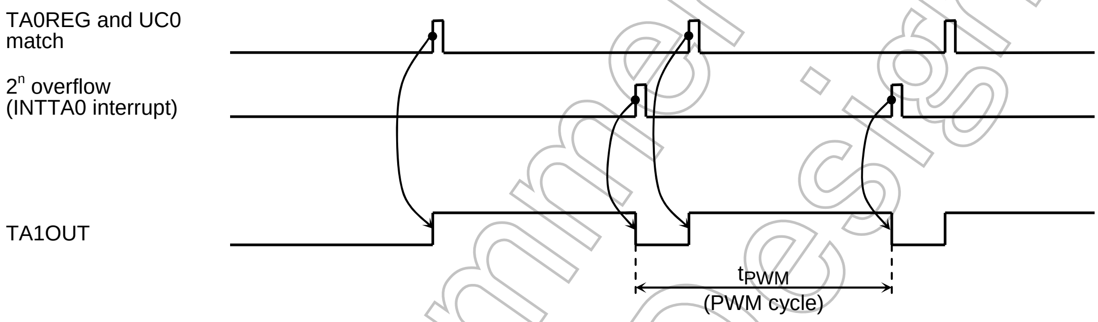


Figure 3.7.16 8-Bit PWM Waveforms

Figure 3.7.17 shows a block diagram representing this mode.

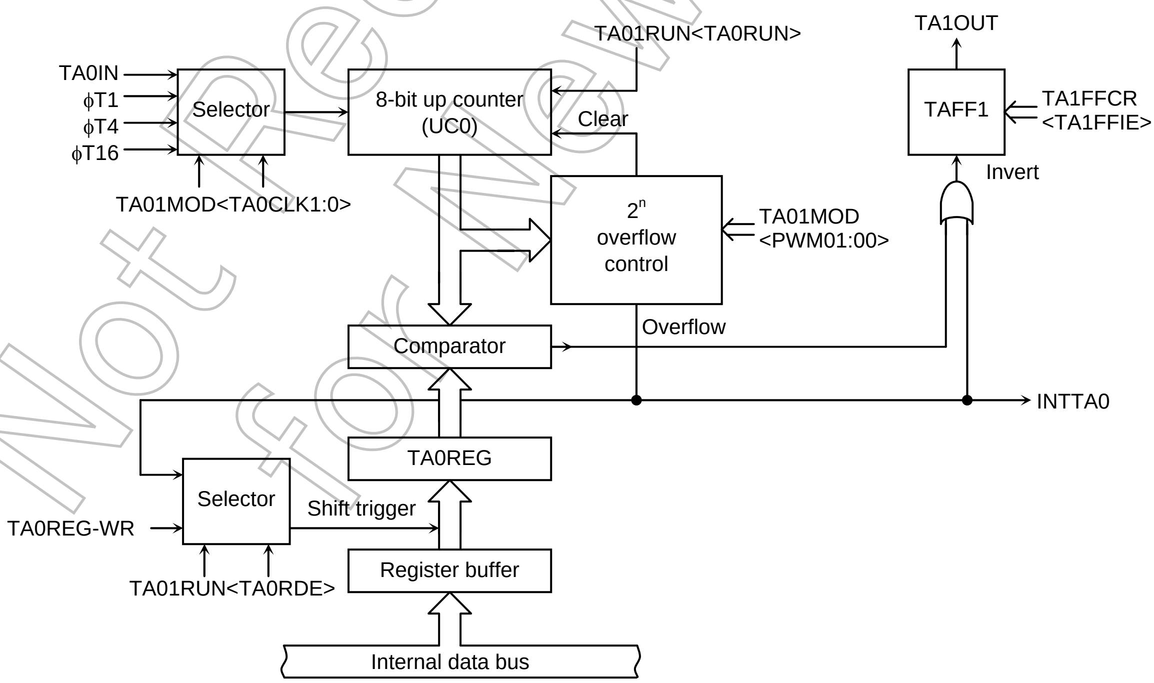


Figure 3.7.17 Block Diagram of 8-Bit PWM Mode

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

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

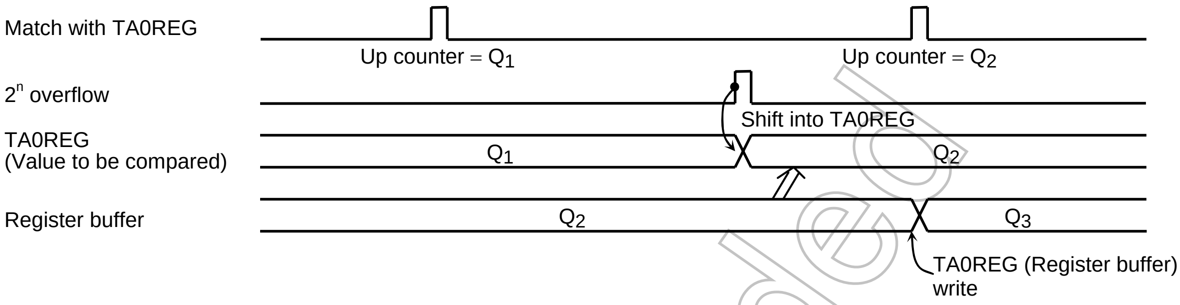
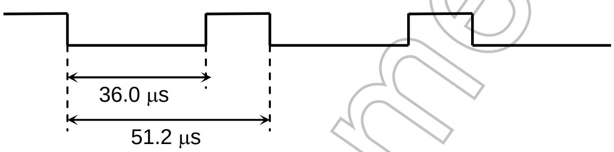


Figure 3.7.18 Register Buffer Operation

Example: To output the following PWM waves on the TA1OUT pin at f_C = 40 MHz:



To achieve a 51.2 µs PWM cycle by setting ϕT1 to 0.4 µs (at f_C = 40 MHz):

51.2 µs ÷ (16/fc)s = 128

2^n = 128

Therefore n should be set to 7.

Since the low-level period is 36.0 µs when ϕT1 = (16/fc) µs,
set the following value for TREG0:

36.0 µs ÷(16/fc)s = 90 = 5AH

		7 (MSB)	6	5	4	3	2	1	0 (LSB)	
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	1	1	0	1	0	Write 5AH.
TA1FFCR	←	X	X	X	X	1	0	1	–	Clear TA1FF to 0, enable the inversion and double buffer.
PCCR	←	X	–	–	X	–	X	1	–	Set PC1 and the TA1OUT pin.
PCFC	←	X	–	–	X	–	X	1	–	(Set PC1 and the TA1OUT pin.)
TA01RUN	←	1	X	X	X	–	1	–	1	Start TMRA0 counting.

X: Don’t care, –: No change

Table 3.7.4 PWM Cycle

Clock gear SYSCR1 <GEAR2:0>	System clock SYSCR0 <SYSCK>	–	PWM cycle TAxxMOD<PWMx1:0>								
			2^6 (x64)			2^7(x128)			2^8(x256)		
			TAxxMOD<TAxCLK1:0>			TAxxMOD<TAxCLK1:0>			TAxxMOD<TAxCLK1:0>		
			φT1(x2)	φT4(x8)	φT16(x32)	φT1(x2)	φT4(x8)	φT16(x32)	φT1(x2)	φT4(x8)	φT16(x32)
–	1(fs)	×8	1024/fs	4096/fs	16384/fs	2048/fs	8192/fs	32768/fs	4096/fs	16384/fs	65536/fs
000(x1)	0(fc)		1024/fc	4096/fc	16384/fc	2048/fc	8192/fc	32768/fc	4096/fc	16384/fc	65536/fc
001(x2)			2048/fc	8192/fc	32768/fc	4096/fc	16384/fc	65536/fc	8192/fc	32768/fc	131072/fc
010(x4)			4096/fc	16384/fc	65536/fc	8192/fc	32768/fc	131072/fc	16384/fc	65536/fc	262144/fc
011(x8)			8192/fc	32768/fc	131072/fc	16384/fc	65536/fc	262144/fc	32768/fc	131072/fc	524288/fc
100(x16)			16384/fc	65536/fc	262144/fc	32768/fc	131072/fc	524288/fc	65536/fc	262144/fc	1048576/fc

(5) Mode setting

Table 3.7.5 shows the SFR settings for each mode.

Table 3.7.5 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

3.8 External Memory Extension Function (MMU)

This is MMU function which can expand program/data area to 136 Mbytes by having 4 local area.

Address pins to external memory are 2 extended address bus pins (EA24, EA25) and 8 extended chip select pins ($\overline{CS2A}$ to $\overline{CS2G}$ and $\overline{CSEXA}$) in addition to 24 address bus pins (A0 to A23) which are common specification of TLCS-900/H1 and 4 chip select pins ($\overline{CS0}$ to $\overline{CS3}$) output from MEMC.

The feature and the recommendation setting method of two types are shown below. In addition, AH in the table is the value which number address 23 to 16 displayed as hex.

Purpose	Item	For Standard Extended Memory	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 to DF: 2 Mbytes × 7 BANK)	
	Setting MEMC	Setup AH = "80 to 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 to BF: 4 Mbytes × 24 BANK)
	Setting MEMC		Setup AH = "80 to FF" to CS2
	Used $\overline{CS}$ pin		$\overline{CS2B}$, $\overline{CS2C}$, $\overline{CS2D}$, $\overline{CS2E}$, $\overline{CS2F}$, $\overline{CS2G}$
Data SDRAM*	Maximum memory size	2 Mbytes: COMMON1 + 14 Mbytes: BANK (16 Mbytes × 1 pcs)	
	Used local area, BANK number	LOCAL1 (AH = 40 to 5F: 2 Mbytes × 7 BANK)	
	Setting MEMC	Setup AH = "40 to 7F" to CS1	
	Used $\overline{CS}$ pin	$\overline{CS1}$	
Data RAM	Maximum memory size	1 Mbyte: COMMON0 + 7 Mbytes: BANK (8 Mbytes × 1 pcs)	
	Used local area, BANK number	LOCAL0 (AH = 10 to 1F: 1 Mbyte × 7 BANK)	
	Setting MEMC	Setup AH = "00 to 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 MEMC	Setup AH = "20 to 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 MEMC	Setup AH = "30 to 3F" to 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 MEMC	Setup AH = "30 to 3F" to CSEX	
	Used $\overline{CS}$ pin	D1BSCP, D2BLP, D3BFR, DLEBCD	
Extended memory 4	Maximum memory size	512 Kbytes	
	Used local area, BANK number	None	
	Setting MEMC	Setup AH = "30 to 3F" to 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 time of variety extension memory correspondence is shown in Figure 3.8.1. And, a physical-address map is shown in Figure 3.8.2.

However, when memory area is less than 16 Mbytes and is not expanded, please refer to section of MEMC. Setting of register in MMU is not necessary.

Since it is being fixed, the address of a local-area cannot be changed. When SDRAM is used, must locate to LOCAL1 area.

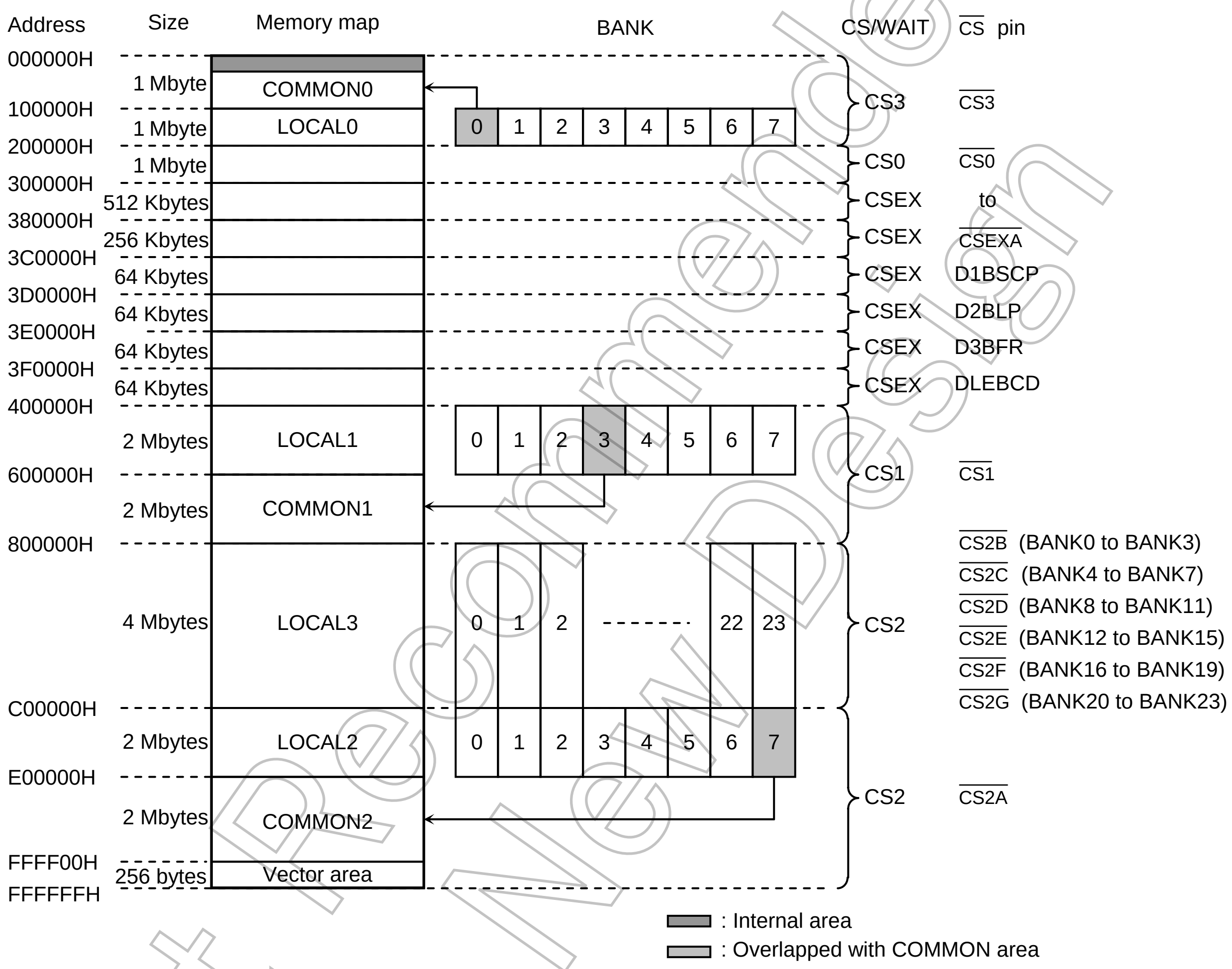


Figure 3.8.1 Logical Address Map

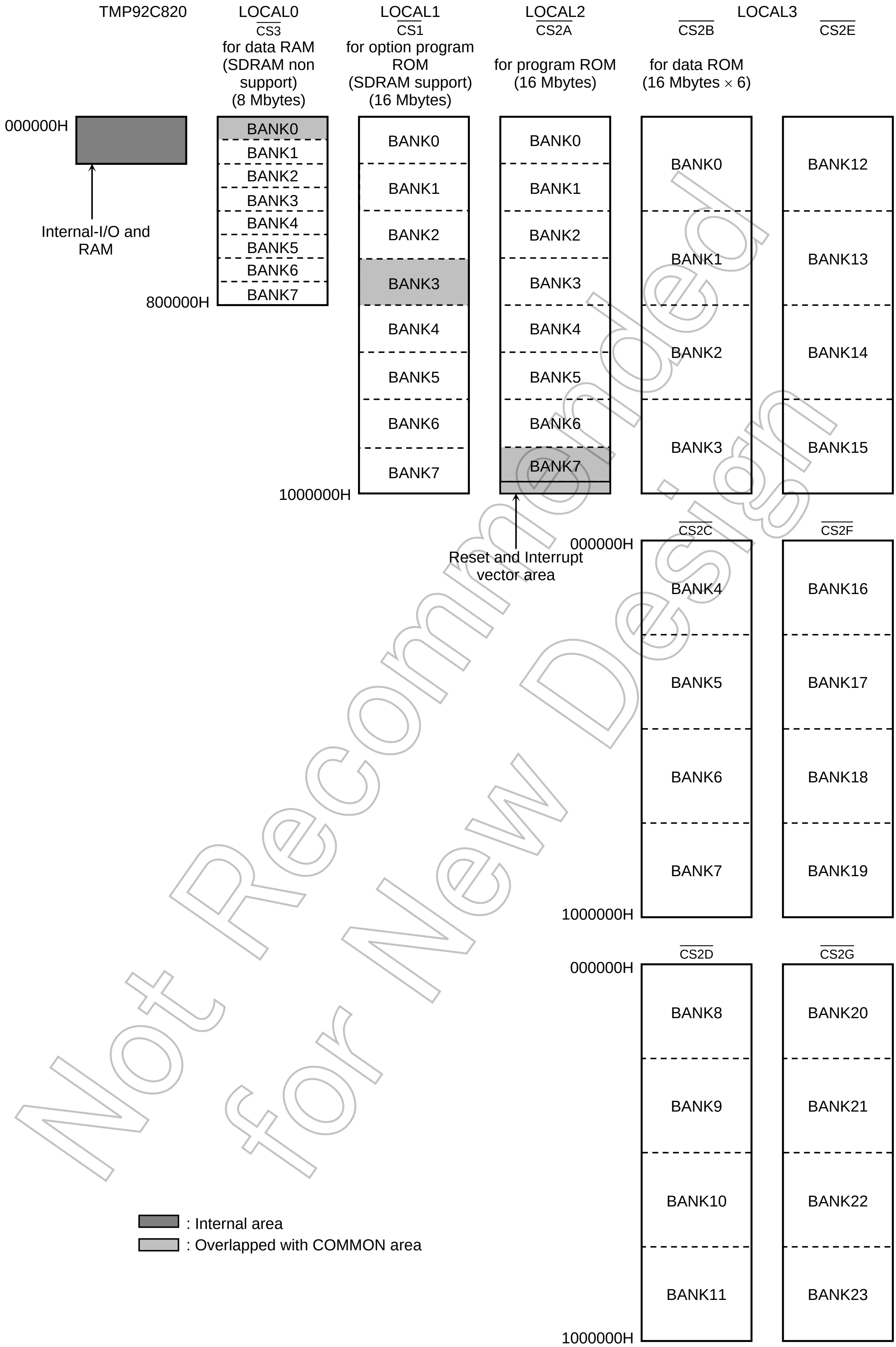


Figure 3.8.2 Physical Address Map

3.8.2 Block Diagram

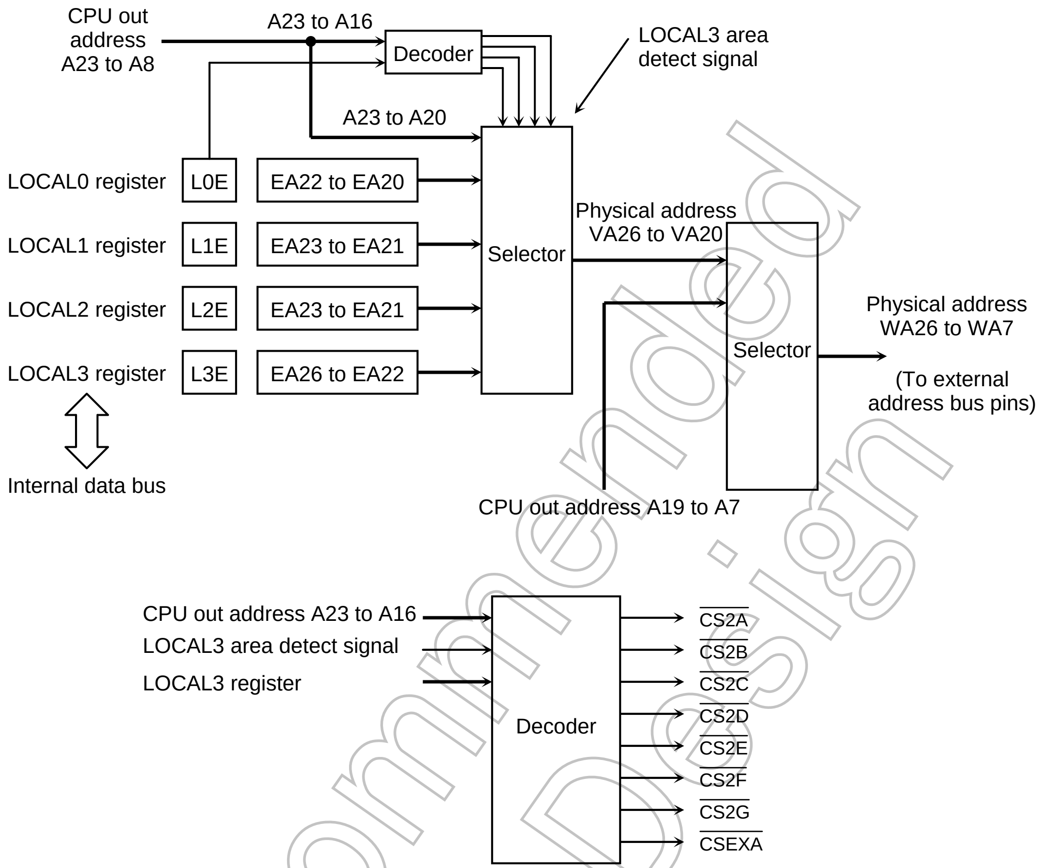


Figure 3.8.3 Block Diagram of MMU

3.8.3 Control Registers

LOCAL0 Register

LOCAL0 (01D0H)		7	6	5	4	3	2	1	0
	Bit symbol	L0E					L0EA22	L0EA21	L0EA20
	Read/Write	R/W					R/W		
	After reset	0					0	0	0
	Function	Use BANK for LOCAL0 0: Not use 1: Use					Setting BANK number for LOCAL0		

LOCAL1 Register

LOCAL1 (01D1H)		7	6	5	4	3	2	1	0
	Bit symbol	L1E					L1EA23	L1EA22	L1EA21
	Read/Write	R/W					R/W		
	After reset	0					0	0	0
	Function	Use BANK for LOCAL1 0: Not use 1: Use					Setting BANK number for LOCAL1		

LOCAL2 Register

LOCAL2 (01D2H)		7	6	5	4	3	2	1	0
	Bit symbol	L2E					L2EA23	L2EA22	L2EA21
	Read/Write	R/W					R/W		
	After reset	0					0	0	0
	Function	Use BANK for LOCAL2 0: Disable 1: Enable					Setting BANK number for LOCAL2		

LOCAL3 Register

LOCAL3 (01D3H)		7	6	5	4	3	2	1	0
	Bit symbol	L3E			L3EA26	L3EA25	L3EA24	L3EA23	L3EA22
	Read/Write	R/W			R/W				
	After reset	0			0	0	0	0	0
	Function	Use BANK for LOCAL3 0: Disable 1: Enable			00000 to 00011 $\overline{CS2B}$ 00100 to 00111 $\overline{CS2C}$ 01000 to 01011 $\overline{CS2D}$ 01100 to 01111 $\overline{CS2E}$ 10000 to 10011 $\overline{CS2F}$ 10100 to 10111 $\overline{CS2G}$ 11000 to 11111: Set prohibition				

Figure 3.8.4 MMU Control Register

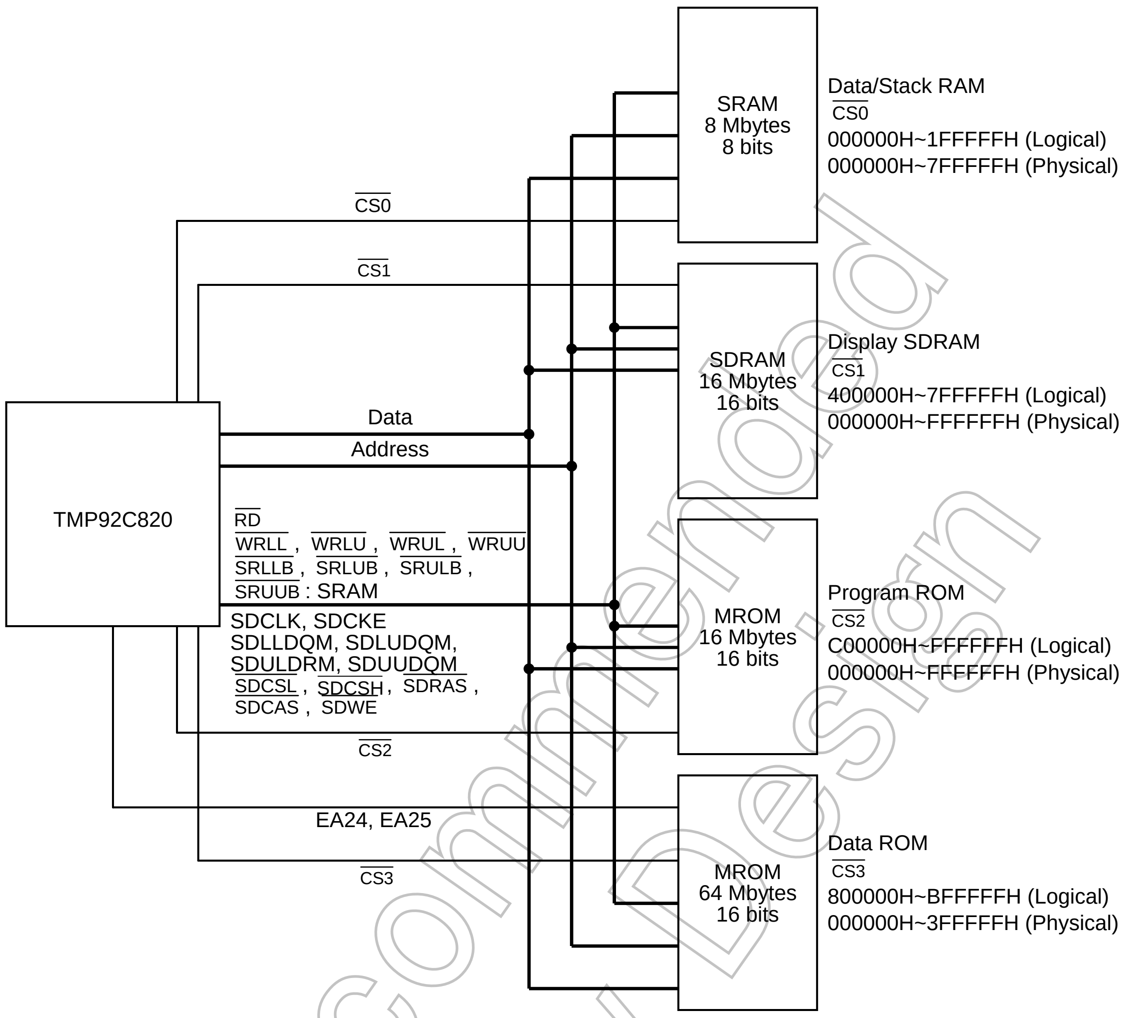
3.8.4 Operational Description

Setup bank value and bank use in bank setting register of each local area of LOCAL register in common area. Moreover, in that case, a combination pin is set up and the MEMC simultaneously sets up mapping. When CPU outputs logical address of the local area, MMU outputs physical address to the outside pin according to value of bank setting register. Access of external memory becomes possible therefore.

Common area located in each local area should be passed surely when changing BANK. For example, when the program jump BANK0 of LOCAL2 to BANK6, please jump from BANK0 to COMMON2 once and afterwards jump to BANK6.

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

Example program is as next page follows.



* In case of 16-bit bus memory, address connection is ⋯: CPU A1 = Memory A0, CPU A2 = Memory A1⋯

* In case of 8-bit bus memory, address connection is ⋯: CPU A0 = Memory A0, CPU A1 = Memory A1⋯

Figure 3.8.5 H/W Setting Example

At, Figure 3.8.5 it shows example of connection TMP92C820 and some memories: Program ROM: MROM, 16 Mbytes, Data ROM: MROM, 64 Mbytes, Data RAM of 8-bit bus: SRAM, 8 Mbytes, Display RAM: SDRAM, 16 Mbytes.

In case of 16-bit bus memory connection, it needs to shift 1-bit address bus from TMP92C820 and 8-bit bus case, direct connection address bus from TMP92C820.

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

Initial condition after reset, because TMP92C820 access from $\overline{\text{CS2}}$ area, $\overline{\text{CS2}}$ area allots to program ROM. It can set free setting except program ROM.

```
; Initial Setting
; CS0
        LD    (MSAR0), 00H       ; Logical address area: 000000H to 1FFFFFH
        LD    (MAMR0), FFH       ; Logical address size: 2 Mbytes
        LD    (B0CSL), 22H       ; Condition: WR 3 states (1 wait), RD 3 states (1 wait)
        LD    (B0CSH), 80H       ; SRAM, 8 bits
; CS1
        LD    (MSAR1), 40H       ; Logical address area: 400000H to 7FFFFFH
        LD    (MAMR1), FFH       ; Logical address size: 4 Mbytes
        LD    (B1CSL), 11H       ; Condition: WR 2 states (0 waits) RD 2 states (0 waits)
        LD    (B1CSH), 8DH       ; Condition: SDRAM, 16 bits
; CS2
        LD    (MSAR2),C0H        ; Logical address area: C00000H to FFFFFFH
        LD    (MAMR2), 7FH       ; Logical address size: 4 Mbytes
        LD    (B2CSL), 11H       ; Condition: WR 2 states (0 waits) RD 2 states (0 waits)
        LD    (B2CSH), 0C1H      ; Condition: ROM, 16 bits
; CS3
        LD    (MSAR3), 80H       ; Logical address area: 800000H to BFFFFFH
        LD    (MAMR3), 7FH       ; Logical address size: 4 Mbytes
        LD    (B3CSL), 66H       ; Condition: WR 5 states (3 waits), RD 5 states (3 waits)
        LD    (B3CSH), 81H       ; Condition: ROM,16 bits
; CSX
        LD    (BEXCSL), 11H      ; Condition: WR 2 states (0 waits), RD 2 states (0 waits)
        LD    (BEXCSH), 01H      ; Condition: 16 bits
; Port
        LD    (P8FC), 3FH        ; /CS0 to /CS3, EA24, EA25: port 8 setting
        LD    (P8FC2), 02H       ; /CS1 → /SDCSL setting
~
        LDW   (P7CR), 1F1FH      ; /WRUU, /WRUL, /WRLU, /WRLL, /RD
        LD    (PJFC), 0FFH       ; PJ<7:0> = SDRAM control
        LD    (SDACR), 083H      ; Add-MUX select type B, SDRAM, auto init enable
~                                  SDRAM setup time

        LD    (SDRCR), 01H       ; Interval refresh
```

Figure 3.8.6 Bank Operation S/W Example 1

Secondly, it shows example of initial setting at Figure 3.8.6.

Because $\overline{\text{CS0}}$ connect to RAM: 8-bit bus, 8 Mbytes, it need to set 8-bit bus. At this example, it set 3 states setting. In the same way $\overline{\text{CS1}}$ set to 16-bit bus and 2 states, $\overline{\text{CS2}}$ set 16-bit bus and 2 states, $\overline{\text{CS3}}$ set 16-bit bus and 5 states.

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

CSEX setting of MEMC is except above CS0 to CS3's setting. This program example isn't used CSEX setting.

Finally pin condition is set. Ports 80 to 85 set to $\overline{\text{CS0}}$, $\overline{\text{CS1}}$, $\overline{\text{CS2}}$, $\overline{\text{CS3}}$, 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)
~
ORG     FFFFFFH                     ; Program ROM: End address at BANK7
                                          (= COMMON2) of LOCAL2

; ***** CS3 *****
ORG     0000000H                    ; Data ROM: Start address at BANK0 of LOCAL3
ORG     0400000H                    ; Data ROM: Start address at BANK1 of LOCAL3
ORG     0800000H                    ; Data ROM: Start address at BANK2 of LOCAL3
ORG     0C00000H                    ; Data ROM: Start address at BANK3 of LOCAL3
ORG     1000000H                    ; Data ROM: Start address at BANK4 of LOCAL3
ORG     1400000H                    ; Data ROM: Start address at BANK5 of LOCAL3
        dw      5555H
~
ORG     1800000H                    ; Data ROM: Start address at BANK6 of LOCAL3
ORG     1C00000H                    ; Data ROM: Start address at BANK7 of LOCAL3
ORG     2000000H                    ; Data ROM: Start address at BANK8 of LOCAL3
        dw      AAAAH
~
ORG     2400000H                    ; Data ROM: Start address at BANK9 of LOCAL3
ORG     2800000H                    ; Data ROM: Start address at BANK10 of LOCAL3
ORG     2C00000H                    ; Data ROM: Start address at BANK11 of LOCAL3
ORG     3000000H                    ; Data ROM: Start address at BANK12 of LOCAL3
ORG     3400000H                    ; Data ROM: Start address at BANK13 of LOCAL3
ORG     3800000H                    ; Data ROM: Start address at BANK14 of LOCAL3
ORG     3C00000H                    ; Data ROM: Start address at BANK15 of LOCAL3
ORG     3FFFFFFH                    ; Data ROM: End address at BANK15 of LOCAL3
```

Figure 3.8.7 Bank Operation S/W Example 2

Here shows example of data access between one BANK and other BANK. Figure 3.8.7 is one software example. A dot line square area shows one memory and each dot line square shows $\overline{CS2}$'s program ROM and $\overline{CS3}$'s data ROM. Program start from E00000H address, firstly, write to BANK register of LOCAL3 area upper 5-bit address of access point.

In case of this example, because most upper address bit of physical address is EA25, most upper address bit of BANK register is meaningless. 4 bits of upper 5 bits address means 16 BANKs. After setting BANK5, accessing 800000H to BFFFFFH address: Logical LOCAL3 address, actually access to physical 1400000H to 1700000H address.

```
; Bank Operation
; *****  CS2  *****
ORG     000000H                 ; Program ROM: Start address at BANK0 of LOCAL2
ORG     200000H                 ; Program ROM: Start address at BANK1 of LOCAL2
        NOP                     ; Operation at BANK1 of LOCAL2
~
        JP      E00100H         ; Jump to BANK7 (= COMMON2) of LOCAL2
ORG     400000H                 ; Program ROM: Start address at BANK2 of LOCAL2
ORG     600000H                 ; Program ROM: Start address at BANK3 of LOCAL2
        NOP                     ; Operation at BANK3 of LOCAL2
~
        JP      E00200H         ; Jump to BANK7 (= COMMON2) of LOCAL2
ORG     800000H                 ; Program ROM: Start address at BANK4 of LOCAL2
ORG     a00000H                 ; Program ROM: Start address at BANK5 of LOCAL2
ORG     c00000H                 ; Program ROM: Start address at BANK6 of LOCAL2

!!!! Program Start !!!!
ORG     E00000H                 ; Program ROM: Start address at BANK7
                                        (= COMMON2) of LOCAL2
                                ; Logical address E00000H to FFFFFFH
                                ; Physical address 0E00000H to 0FFFFFFH
        LD      (LOCAL2), 81H   ; LOCAL2 BANK1 set 20xxxxH
        JP      C00000H         ; Jump to BANK1 (200000H: Physical address) of LOCAL2
~
ORG     E00100H
        LD      (LOCAL2), 83H   ; LOCAL2 BANK3 set 60xxxxH
        JP      C00000H         ; Jump to BANK3 (600000H: Physical address) of LOCAL2
~
ORG     E00200H
        LD      (LOCAL1), 00H   ; Disable Bank!
~                               ; 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
~

ORG     800000H                 ; SDRAM: Start address at BANK4 of LOCAL1
ORG     a00000H                 ; SDRAM: Start address at BANK5 of LOCAL1
ORG     c00000                  ; 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 Bank Operation S/W Example 3

At Figure 3.8.8, it shows example of program jump.

In the same way with before example, two dot line squares show each $\overline{CS2}$'s program ROM and $\overline{CS1}$'s (SDCS) SDRAM. Program start from E00000H common address, firstly, write to BANK register of LOCAL2 area upper 3-bit address of jumping point.

After setting BANK1, jumping C00000H to DFFFFFH address: Logical LOCAL2 address, actually jump to physical 200000H to 3FFFFFH address. When return to common area, it can only jump to E00000H to FFFFFFH without writing to BANK register of LOCAL2 area.

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

Please set similarly when jumping through $\overline{CS}$.

After setting BANK4, jumping 400000H to 5FFFFFH address: Logical local area of $\overline{CS1}$, actually jump to physical 800000H to 9FFFFFH address.

When using LCD display data for SDRAM, we recommend setting display area to common area in SDRAM. Because of, LCD displays DMA occurs at synchronous less. If SDRAM bank is change; you don't need to care only common area. It is a mark paid attention to here, it needs to go by way of common area by all means when moves from a bank to a bank. In other words, it must write to BANK register only in common area and it prohibits writing the BANK registers in BANK area. If it modify the BANK register's data in BANK area, program run away. Please do not set bank function of MMU as display RAM. This is because reading LCDC display data is not controlled by the CPU. Therefore if BANK of display area is changed during LCD displaying, it cannot display. It is recommended to allocate display data to a common area.

3.9 Serial Channels (SIO)

The TMP92C820 includes three serial I/O channels. For each channel either UART mode (Asynchronous transmission) or I/O interface mode (Synchronous transmission) can be selected.
(Channel 2 can be selected only UART mode.)

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

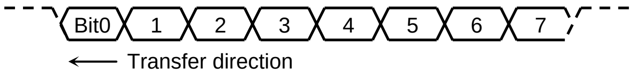
In mode 1 and mode 2 a parity bit can be added. Mode 3 has a wakeup function for making the master controller start slave controllers via a serial link (Multi-controller system).
Figure 3.9.2, Figure 3.9.3, and Figure 3.9.4 are block diagrams for each channel.
Each channel can be used independently.
Each channel operates in the same fashion except for the following points; hence only the operation of channel 0 is explained below.

Table 3.9.1 Differences between Channels 0 to 2

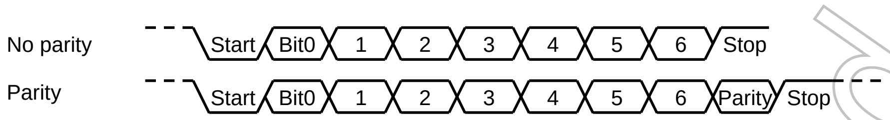
	Channel 0	Channel 1	Channel 2
Pin name	TXD0 (PF0) RXD0 (PF1) $\overline{CTS0}$ /SCLK0 (PF2)	TXD1 (PF3) RXD1 (PF4) $\overline{CTS1}$/SCLK1 (PF5)	TXD2 (P95) RXD2 (P96)
IrDA mode	Yes	No	No

This chapter contains the following sections:
3.9.1 Block Diagrams
3.9.2 Operation for Each Circuit
3.9.3 SFRs
3.9.4 Operation in Each Mode
3.9.5 Support for IrDA

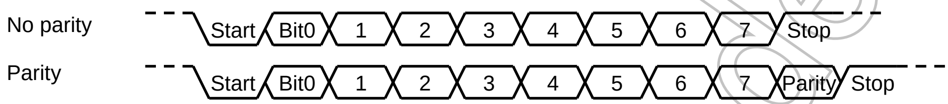
- Mode 0 (I/O interface mode)



- Mode 1 (7-bit UART mode)



- Mode 2 (8-bit UART mode)



- Mode 3 (9-bit UART mode)

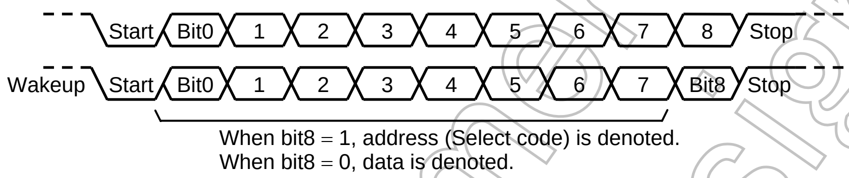


Figure 3.9.1 Data Formats

3.9.1 Block Diagrams

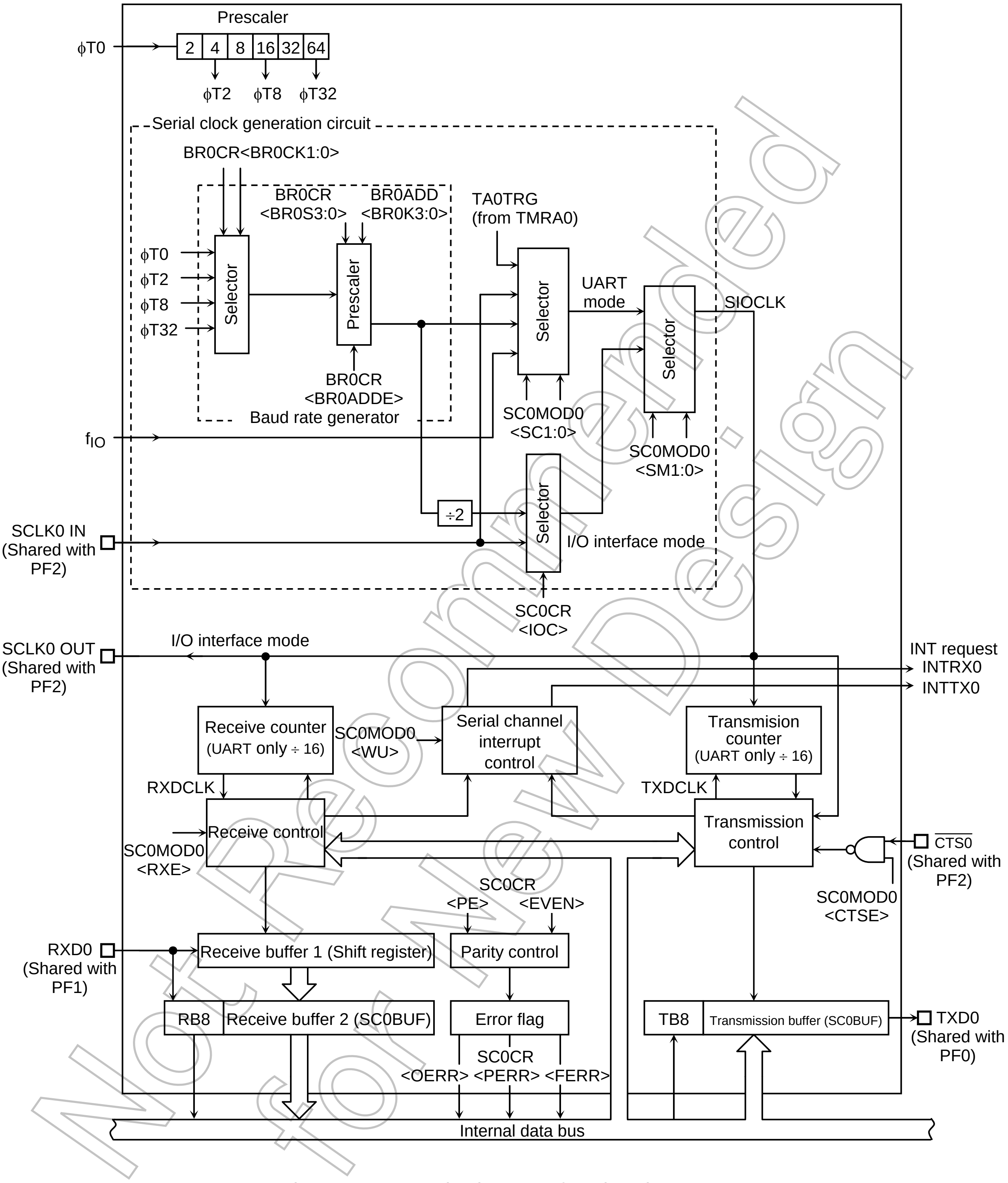


Figure 3.9.2 Block Diagram of Serial Channel 0

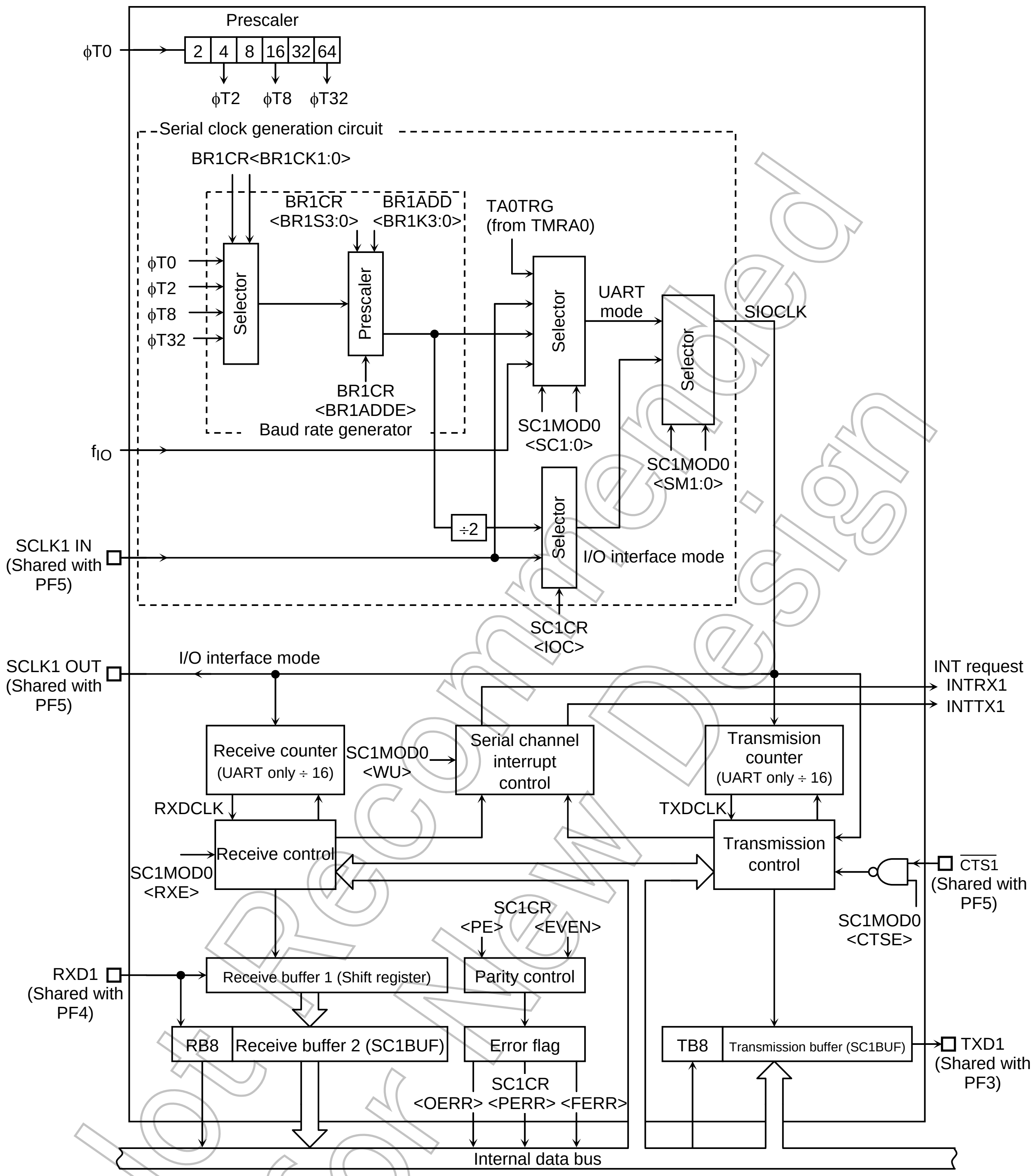


Figure 3.9.3 Block Diagram of Serial Channel 1

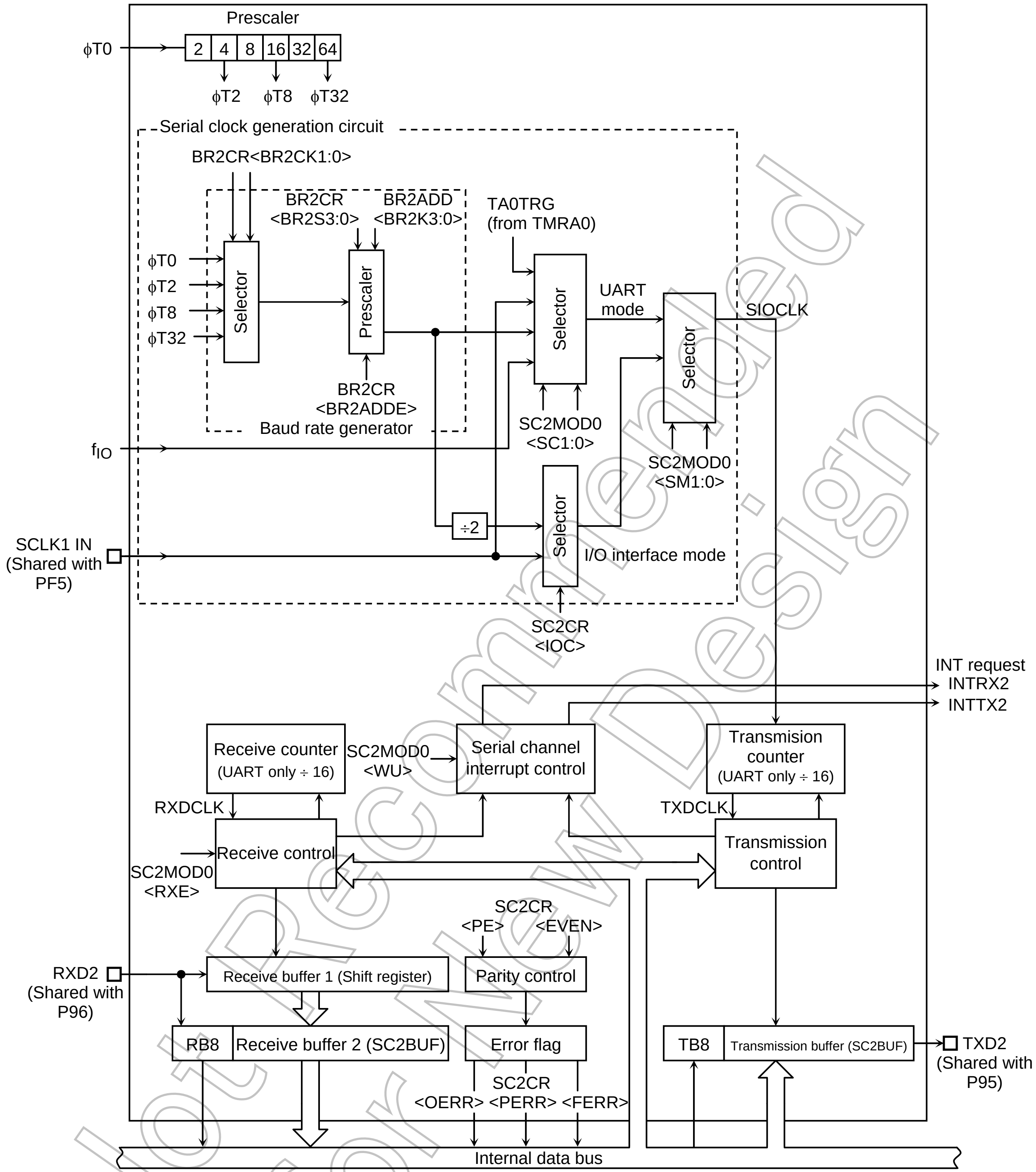


Figure 3.9.4 Block Diagram of Serial Channel 2

3.9.2 Operation for Each Circuit

(1) Prescaler, prescaler clock select

There is a 6-bit prescaler for waking serial clock.

The prescaler can be run by selecting the baud rate generator as the waking serial clock. Table 3.9.2 shows prescaler clock resolution into the baud rate generator.

Table 3.9.2 Prescaler Clock Resolution to Baud Rate Generator

–	Clock gear selection SYSCR1 <GEAR2:0>	–	Baud rate generator input clock SIO prescaler BR0CR<BR0CK1:0>			
			φT0	φT2(1/4)	φT8(1/16)	φT32(1/64)
fc	000(1/1)	1/8	fc/8	fc/32	fc/128	fc/512
	001(1/2)		fc/16	fc/64	fc/256	fc/1024
	010(1/4)		fc/32	fc/128	fc/512	fc/2048
	011(1/8)		fc/64	fc/256	fc/1024	fc/4096
	100(1/16)		fc/128	fc/512	fc/2048	fc/8192

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

(2) Baud rate generator

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

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

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

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

- In UART mode

(1) When BR0CR<BR0ADDE> = 0

The settings BR0ADD<BR0K3:0> are ignored. The baud rate generator divides the selected prescaler clock by N, which is set in BR0CK<BR0S3:0>. (N = 1, 2, 3…16)

(2) When BR0CR<BR0ADDE> = 1

The N + (16 – K)/16 division function is enabled. The baud rate generator divides the selected prescaler clock by N + (16 – K)/16 using the value of N set in BR0CR<BR0S3:0> (N = 2, 3…15) and the value of K set in BR0ADD<BR0K3:0> (K = 1, 2, 3…5)

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

- In I/O interface mode

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

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

- In UART mode

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

- In I/O interface mode

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

- Integer divider (N divider)

For example, when the source clock frequency (f_C) is 39.3216 MHz, the input clock is ϕT2 (f_C/32), the frequency divider N (BR0CR<BR0S3:0>) = 8, and BR0CR<BR0ADDE> = 0, the baud rate in UART mode is as follows:

∗ Clock state [Clock gear: 1/1 (f_C)

$$\text{Baud rate} = \frac{f_C/32}{8} \div 16$$

$$= 39.3216 \times 10^6 \div 16 \div 8 \div 16 = 9600 \text{ (bps)}$$

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

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

Accordingly, when the source clock frequency (f_C) = 31.9488 MHz, the input clock is ϕT2 (f_C/32), the frequency divider N (BR0CR<BR0S3:0>) = 6, K (BR0ADD<BR0K3:0>) = 8, and BR0CR<BR0ADDE> = 1, the baud rate in UART mode is as follows:

∗ Clock state [Clock gear: 1/1 (f_C)

$$\text{Baud rate} = \frac{f_C/32}{6 + \frac{(16-8)}{16}} \div 16$$

$$= 31.9488 \times 10^6 \div 16 \div \left(6 + \frac{8}{16}\right) \div 16 = 9600 \text{ (bps)}$$

Table 3.9.3 show examples of UART mode transfer rates.

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

- In UART mode

  Baud rate = External clock input frequency ÷ 16

  It is necessary to satisfy (External clock input cycle) ≥ 4/f_{SYS}

- In I/O interface mode

  Baud rate = External clock input frequency

  It is necessary to satisfy (External clock input cycle) ≥ 16/f_{SYS}

Table 3.9.3 Selection of Transfer Rate (1)

(When baud rate generator is used and BR0CR<BR0ADDE> = 0) Unit (kbps)

f_{SYS} [MHz]	Input Clock / Frequency Divider	φT0 (f_{SYS}/4)	φT2 (f_{SYS}/16)	φT8 (f_{SYS}/64)	φT32 (f_{SYS}/256)
9.8304	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
↑	10	9.600	2.400	0.600	0.150
12.2880	5	38.400	9.600	2.400	0.600
↑	A	19.200	4.800	1.200	0.300
14.7456	2	115.200	28.800	7.200	1.800
↑	3	76.800	19.200	4.800	1.200
↑	6	38.400	9.600	2.400	0.600
↑	C	19.200	4.800	1.200	0.300
19.6608	1	307.200	76.800	19.200	4.800
↑	2	153.600	38.400	9.600	2.400
↑	4	76.800	19.200	4.800	1.200
↑	8	38.400	9.600	2.400	0.600
↑	10	19.200	4.800	1.200	0.300
22.1184	3	115.200	28.800	7.200	1.800
24.5760	1	384.000	96.000	24.000	6.000
↑	2	192.000	48.000	12.000	3.000
↑	4	96.000	24.000	6.000	1.500
↑	5	76.800	19.200	4.800	1.200
↑	8	48.000	12.000	3.000	0.750
↑	A	38.400	9.600	2.400	0.600
↑	10	24.000	6.000	1.500	0.375

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

In UART mode, TMRA match detect signal (TA0TRG) can be used for serial transfer clock.

Method for calculating the timer output frequency which is needed when outputting trigger of timer

TA0TRG frequency = Baud rate × 16

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

(3) Serial clock generation circuit

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

- In I/O interface mode

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

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

- In UART mode

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

(4) Receiving counter

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

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

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

(5) Receiving control

- In I/O interface mode

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

  In SCLK input mode with the setting SC0CR<IOC> = 1, the RXD0 signal is sampled on the rising or falling edge of the SCLK0 input, according to the SC0CR<SCLKS> setting.

- In UART mode

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

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

(6) The receiving buffers

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

The CPU only reads receiving buffer 2 (SC0BUF). Even before the CPU reads receiving buffer 2 (SC0BUF), the received data can be stored in receiving buffer 1.

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

SC0CR<RB8> is used to store either the parity bit–added in 8-bit UART mode–or the most significant bit (MSB) –in 9-bit UART mode.

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

SIO interrupt mode is selectable by the register SIMC.

(7) Transmission counter

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

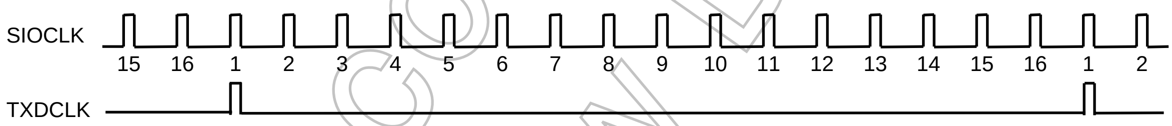


Figure 3.9.5 Generation of the Transmission Clock

(8) Transmission controller

- In I/O interface mode

  In SCLK output mode with the setting SC0CR<IOC> = 0, the data in the transmission buffer is output one bit at a time to the TXD0 pin on the rising or falling edge of the shift clock which is output on the SCLK0 pin, according to the SC0CR<SCLKS> setting.

  In SCLK input mode with the setting SC0CR<IOC> = 1, the data in the transmission buffer is output one bit at a time on the TXD0 pin on the rising or falling edge of the SCLK0 input, according to the SC0CR<SCLKS> setting.

- In UART mode

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

Handshake function

Serial channels 0, 1 each has a $\overline{\text{CTS}}$ pin. Use of this pin allows data can be sent in units of one frame; thus, overrun errors can be avoided. The handshake functions is enabled or disabled by the SC0MOD<CTSE> setting.

When the $\overline{\text{CTS0}}$ pin goes high on completion of the current data send, data transmission is halted until the $\overline{\text{CTS0}}$ pin goes low again. However, the INTTX0 interrupt is generated, it requests the next data send to the CPU. The next data is written in the transmission buffer and data sending is halted.

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

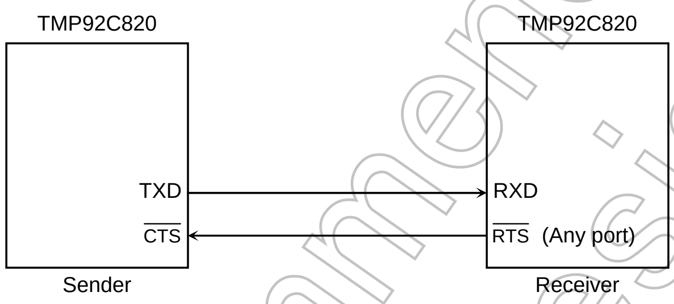
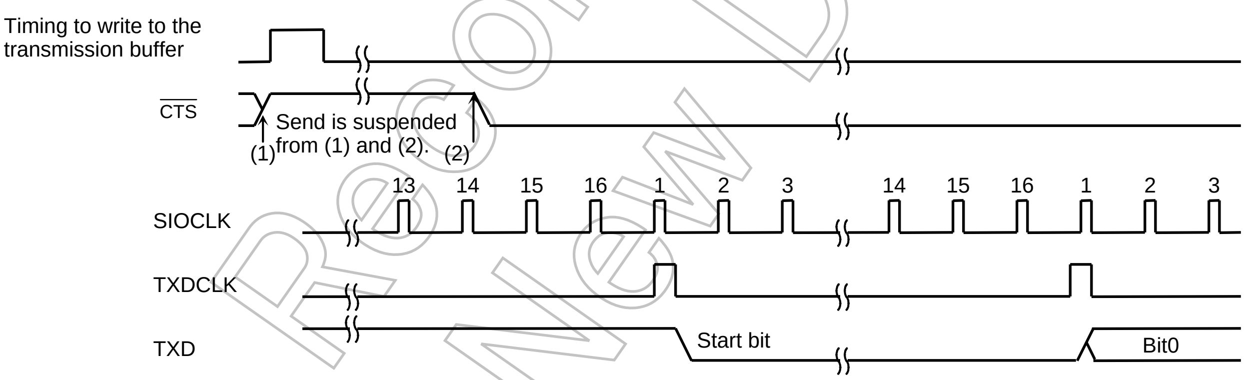


Figure 3.9.6 Handshake Function



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

(10) Parity control circuit

When SC0CR<PE> in the serial channel control register is set to 1, it is possible to transmit and receive data with parity. However, parity can be added only in 7-bit UART mode or 8-bit UART mode. The SC0CR<EVEN> field in the serial channel control register allows either even or odd parity to be selected.

In the case of transmission, parity is automatically generated when data is written to the transmission buffer SC0BUF. The data is transmitted after the parity bit has been stored in SC0BUF<TB7> in 7-bit UART mode or in SC0MOD0<TB8> in 8-bit UART mode. SC0CR<PE> and SC0CR<EVEN> must be set before the transmission data is written to the transmission buffer.

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

(11) Error flags

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

1. Overrun error <OERR>

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

   The below is a recommended flow when the overrun error is generated.

   (INTRX interrupt routine)

   (1) Read receiving buffer
   (2) Read error flag
   (3) If <OERR> = 1
       then
       (a) Set to disable receiving (Write “0” to SC0MOD0<RXE>)
       (b) Wait to terminate current frame
       (c) Read receiving buffer
       (d) Read error flag
       (e) Set to enable receiving (Write “1” to SC0MOD0<RXE>)
       (f) Request to transmit again
   (4) Other

2. Parity error <PERR>

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

3. Framing error <FERR>

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

(12) Timing generation

1. 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

Note1: In 9-bit and 8-bit 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.

Note2: The higher the transfer rate, the later than the middle receive interrupts and errors occur.

Transmitting

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

2. 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 (1202H)

	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	Hand shake 0: CTS disable 1: CTS enable	Receive function 0: Receive disable 1: Receive enable	Wakeup function 0: Disable 1: Enable	Serial transmission mode 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial transmission clock (UART) 00: TA0TRG 01: Baud rate generator 10: Internal clock f_{IO} 11: External clock (SCLK0 input)	

Serial transmission clock source (UART)

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

Note: The clock selection for the I/O interface mode is controlled by the serial control 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 SC0CR<RB8> = 1	

Receiving function

0	Receive disabled
1	Receive enabled

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

0	Disabled (Always transferable)
1	Enabled

Transmission data bit8

Figure 3.9.8 Serial Mode Control Register (Channel 0, SC0MOD0)

SC1MOD0 (120AH)

	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	Hand shake 0: CTS disable 1: CTS enable	Receive function 0: Receive disable 1: Receive enable	Wakeup function 0: Disable 1: Enable	Serial transmission mode 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial transmission clock (UART) 00: TA0TRG 01: Baud rate generator 10: Internal clock f_{IO} 11: External clock (SCLK1 input)	

Serial transmission clock source (for UART)

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

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

Serial transmission mode

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

Wakeup function

	9-bit UART	Other modes
0	Interrupt generated when data is received	Don't care
1	Interrupt generated only when SC1CR<RB8> = 1	

Receiving function

0	Receive disabled
1	Receive enabled

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

0	Disabled (Always transferable)
1	Enabled

Transmission data bit8

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

SC2MOD0 (1212H)

	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 write "0".	Receive function 0: Receive disable 1: Receive enable	Wakeup function 0: Disable 1: Enable	Serial transmission mode 00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		Serial transmission clock (UART) 00: TA0TRG 01: Baud rate generator 10: Internal clock f_{IO} 11: Reserved	

→ Serial transmission clock source (for UART)

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

→ 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 SC2CR<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
(1201H)

	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 (rising edge) 1: SCLK0 (falling edge)	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 receives data on rising edge of SCLK0.
1	Transmits and receives data on falling edge of SCLK0.

Framing error flag
Parity error flag
Overrun error flag
} Cleared to 0 when read

Parity additions 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 (1209H)

	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: SCLK1 (rising edge) 1: SCLK1 (falling edge)	0: Baud rate generator 1: SCLK1 pin input

I/O interface input clock select

0	Baud rate generator
1	SCLK1 pin input

Edge selection for 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 additions 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 (1211H)

	7	6	5	4	3	2	1	0
Bit symbol	RB8	EVEN	PE	OERR	PERR	FERR	–	–
Read/Write	R	R/W		R (Cleared to 0 when read)			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 write "0".	Always write "0".

Bit 2 (FERR) → Framing error flag (Cleared to 0 when read)

Bit 3 (PERR) → Parity error flag (Cleared to 0 when read)

Bit 4 (OERR) → Overrun error flag (Cleared to 0 when read)

Bit 5 (PE) → Parity additions enable

0	Disabled
1	Enabled

Bit 6 (EVEN) → Even parity addition/check

0	Odd parity
1	Even parity

Bit 7 (RB8) → 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
(1203H)

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

+ (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
(1204H)

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

Sets baud rate generator frequency divisor

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

Note1: 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 to “1” in UART mode only when the +(16-K)/16 division function is not used. Do not use in I/O interface mode.

Note2: Set BR0CR <BR0ADDE> to 1 after setting K (K = 1 to 15) to BR0ADD<BR0K3:0> when +(16-K)/16 division function is used. Writes to unused bits in the BR0ADD register do not affect operation, and undefined data is read from these unused bits.

Figure 3.9.14 Baud Rate Generator Control (Channel 0, BR0CR, BR0ADD)

BR1CR (120BH)

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

+ (16 − K)/16 division enable

0	Disabled
1	Enabled

Input clock selection for baud rate generator

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

BR1ADD (120CH)

	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 <BR1S3:0> / BR1ADD <BR1K3:0>	BR1CR<BR1ADDE> = 1		BR1CR<BR1ADDE> = 0
	0000 (N = 16) or 0001 (N = 1)	0010 (N = 2) to 1111 (N = 15)	0001 (N = 1) (UART only) to 1111 (N = 15) 0000 (N = 16)
0000	Disable	Disable	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 only when the +(16-K)/16 division function is not used. Do not use in I/O interface mode.

Note2: Set BR1CR <BR1ADDE> to 1 after setting K (K = 1 to 15) to BR1ADD<BR1K3:0> when the +(16-K)/16 division function is used. Writes to unused bits in the BR1ADD register do not affect operation, and undefined data is read from these unused bits.

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

BR2CR (1213H)

	7	6	5	4	3	2	1	0
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 write "0".	+ (16 − K)/16 division 0: Disable 1: Enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Divided frequency setting			

+ (16 − K)/16 division enable

0	Disabled
1	Enabled

Input clock selection for baud rate generator

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

BR2ADD (1214H)

	7	6	5	4	3	2	1	0
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

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

Note1: 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 only when the +(16-K)/16 division function is not used. Do not use in I/O interface mode.

Note2: Set BR2CR <BR2ADDE> to 1 after setting K (K = 1 to 15) to BR2ADD<BR2K3:0> when the +(16-K)/16 division function is used. Writes to unused bits in the BR2ADD register do not affect operation, and undefined data is read from these unused bits.

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

SC0BUF (1200H)

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 SC0BUF.

Figure 3.9.17 Serial Transmission/Receiving Buffer Registers (Channel 0, SC0BUF)

SC0MOD1 (1205H)

	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 (1208H)

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 (120DH)

	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 (1210H)

	7	6	5	4	3	2	1	0	
SC2BUF (1210H)	TB7	TB6	TB5	TB4	TB3	TB2	TB1	TB0	(Transmission)
	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0	(Receiving)

Note: Prohibit read-modify-write for SC1BUF.

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

SC2MOD1 (1215H)		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 pins available for transmitting data to or receiving data from an external shift register.

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

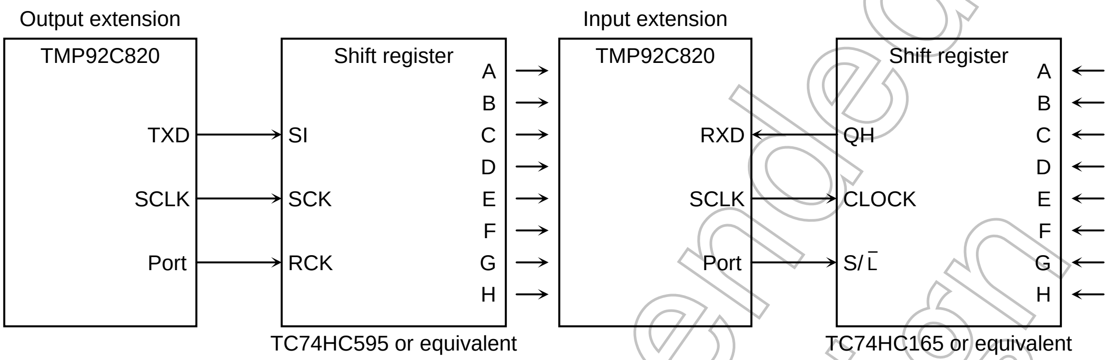


Figure 3.9.23 SCLK Output Mode Connection Example

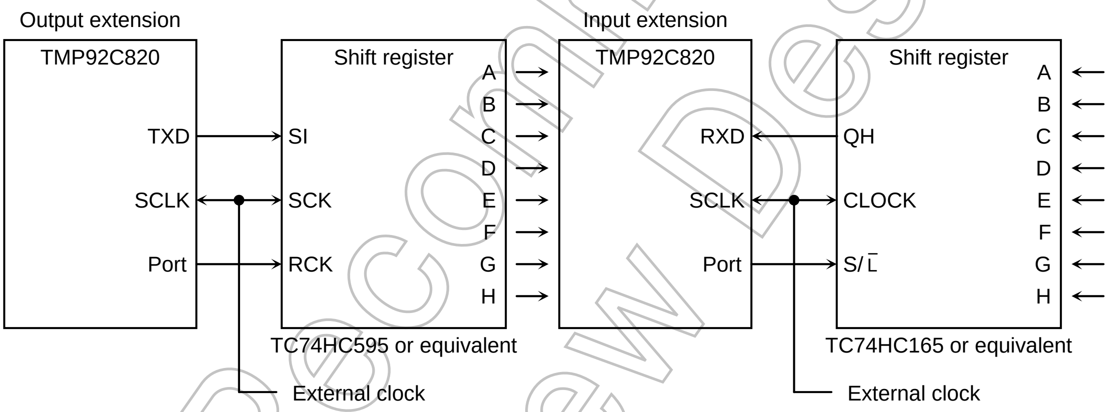


Figure 3.9.24 Example of SCLK Input Mode Connection

1. Transmission

In SCLK output mode 8-bit data and a synchronous clock are output on the TXD0 and SCLK0 pins respectively each time the CPU writes the data to the transmission buffer. When all data is output, INTES0<ITX0C> will be set to generate the INTTX0 interrupt.

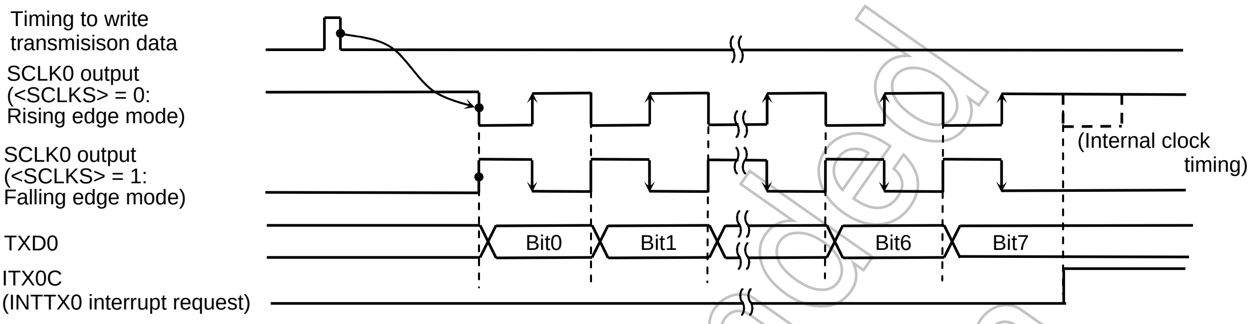


Figure 3.9.25 Transmitting Operation in I/O Interface Mode (SCLK0 output mode) (Channel 0)

In SCLK input mode, 8-bit data is output on the TXD0 pin when the SCLK0 input becomes active after the data has been written to the transmission buffer by the CPU.

When all data is output, INTES0<ITX0C> will be set to generate INTTX0 interrupt.

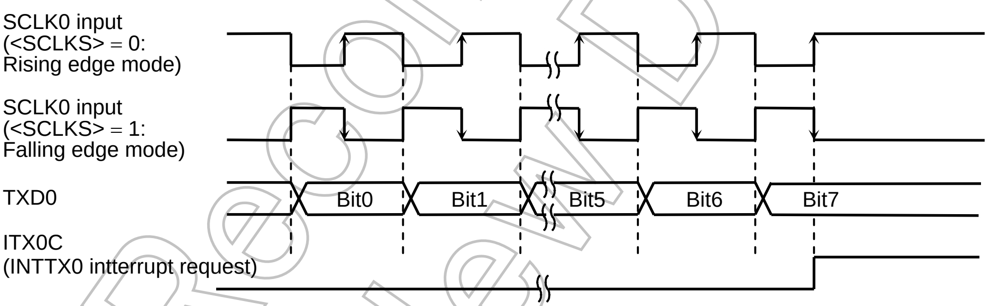


Figure 3.9.26 Transmitting Operation in I/O Interface Mode (SCLK0 input mode) (Channel 0)

2. Receiving

In SCLK output mode the synchronous clock is output on the SCLK0 pin and the data is shifted to receiving buffer 1. This is initiated when the receive interrupt flag INTES0<IRX0C> is cleared as the received data is read. When 8-bit data is received, the data is transferred to receiving buffer 2 (SC0BUF) following the timing shown below and INTES0<IRX0C> is set to 1 again, causing an INTRX0 interrupt to be generated.

Setting SC0MOD0<RXE> to 1 initiates SCLK output.

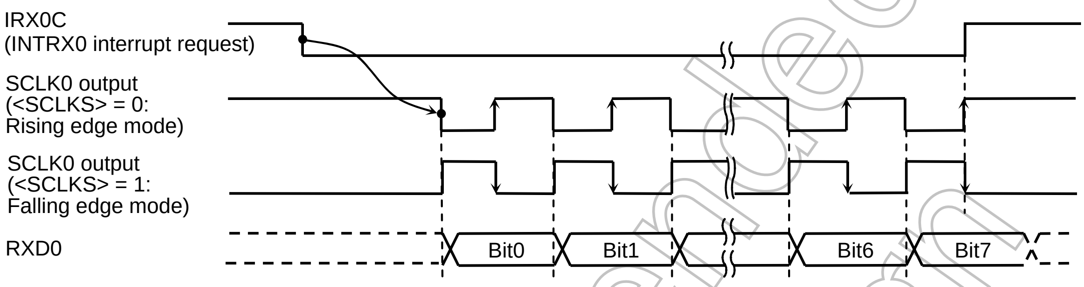


Figure 3.9.27 Receiving Operation in I/O Interface Mode (SCLK0 output mode)

In SCLK input mode the data is shifted to receiving buffer 1 when the SCLK input goes active. The SCLK input goes active when the receive interrupt flag INTES0<IRX0C> is cleared as the received data is read. When 8-bit data is received, the data is shifted to receiving buffer 2 (SC0BUF) following the timing shown below and INTES0<IRX0C> is set to 1 again, causing an INTRX0 interrupt to be generated.

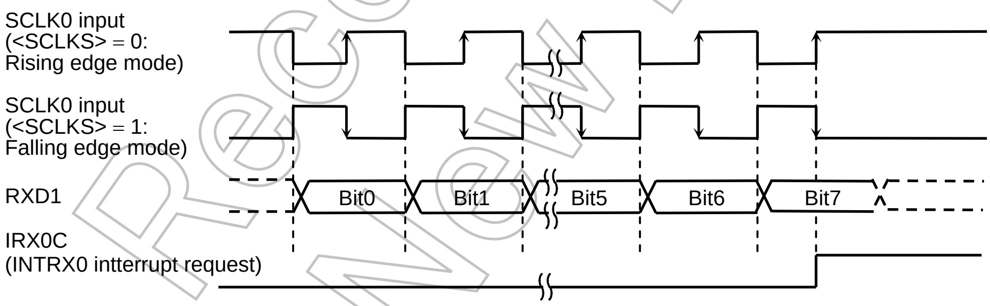


Figure 3.9.28 Receiving Operation in I/O Interface Mode (SCLK0 input mode)

Note: The system must be put in the receive enable state (SC0MOD0<RXE> = 1) before data can be received.

3. Transmission and receiving (Full duplex mode)

When full duplex mode is used, set the receive interrupt level to 0 and set enable the level of transmit interrupt. Ensure that the program which transmits the interrupt reads the receiving buffer before setting the next transmit data.

The following is an example of this:

Example: Channel 0, SCLK output
Baud rate = 9600 bps
f_C = 4.9152 MHz

* Clock state [Clock gear 1/1 (f_C)

Main routine

		7	6	5	4	3	2	1	0	
INTES0	←	X	0	0	1	X	0	0	0	Set the INTTX0 level to 1. Set the INTRX0 level to 0.
PFCR	←	X	X	–	–	–	1	0	1	Set PF0, PF1, and PF2 to function as the TXD0, RXD0,
PFFC	←	X	X	–	X	–	1	X	1	and SCLK0 pins respectively.
SC0MOD0	←	–	–	–	–	0	0	–	–	Enable receiving and select I/O interface mode.
SC0MOD1	←	1	1	X	X	X	X	X	X	Select full duplex mode.
SC0CR	←	0	0	0	0	0	0	0	0	SCLK output, transmit on negative edge, receive on positive edge.
BR0CR	←	0	0	0	1	1	0	0	0	Baud rate = 9600 bps.
SC0MOD0	←	–	–	1	–	–	–	–	–	Enable receiving.
SC0BUF	←	*	*	*	*	*	*	*	*	Set the transmit data and start.

INTTX0 interrupt routine

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

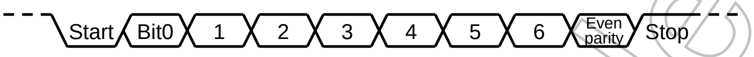
X: Don't care, –: No change

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

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

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

Example: When transmitting data of the following format, the control registers should be set as described below.



← Transmission direction (Transmission rate: 2400 bps at f_{SYS} = 39.3216 MHz)

* Clock state [Clock gear 1/1 (f_C)

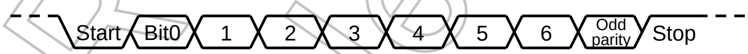
		7	6	5	4	3	2	1	0	
PFCR	←	X	X	–	–	–	–	–	1	Set PF0 to function as the TXD0 pin.
PFFC	←	X	X	–	X	–	–	X	1	
SC0MOD0	←	–	0	–	–	0	1	0	1	Select 7-bit UART mode.
SC0CR	←	–	1	1	–	–	–	0	0	Add even parity.
BR0CR	←	0	0	1	0	1	0	0	0	Set the transfer rate to 2400 bps.
INTES0	←	X	1	0	0	–	–	–	–	Enable the INTTX0 interrupt 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 this mode a parity bit can be added (Use of a parity bit is enabled or disabled by the setting of SC0CR<PE>); whether even parity or odd parity will be used is determined by the SC0CR<EVEN> setting when SC0CR<PE> is set to 1 (Enabled).

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



← Transmission direction (Transmission rate: 9600 bps at f_C = 39.3216 MHz)

* Clock state [Clock gear 1/1 (f_C)

Main settings

		7	6	5	4	3	2	1	0	
PFCR	←	X	X	–	–	–	–	0	–	Set PF1 to function as the RXD0 pin.
SC0MOD0	←	–	0	1	–	1	0	0	1	Enable receiving in 8-bit UART mode.
SC0CR	←	–	0	1	–	–	–	0	0	Add odd parity.
BR0CR	←	0	0	0	1	1	0	0	0	Set the transfer rate to 9600 bps.
INTES0	←	–	–	–	–	X	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

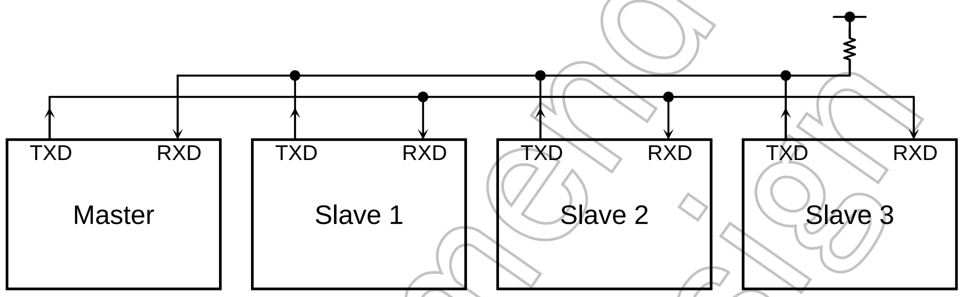
(4) Mode 3 (9-bit UART mode)

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

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

Wakeup function

In 9-bit UART mode, the wakeup function for slave controllers is enabled by setting SC0MOD0<WU> to 1. The interrupt INTRX0 can only be generated when <RB8> = 1.

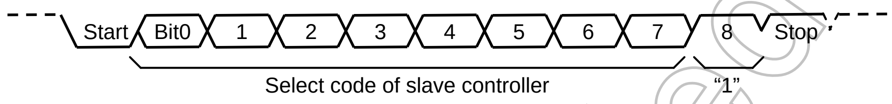


Note: The TXD pin of each slave controller must be in open-drain output mode.

Figure 3.9.29 Serial Link Using Wakeup Function

Protocol

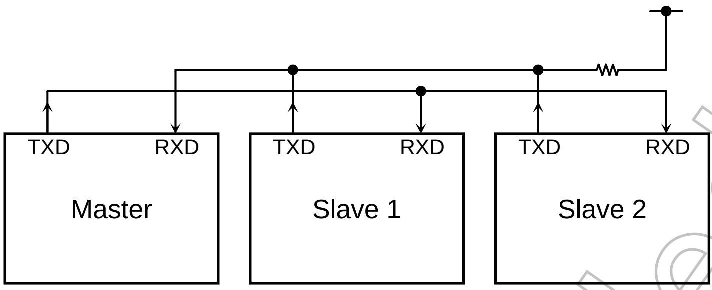
1. Select 9-bit UART mode on the master and slave controllers.
2. Set the SC0MOD0<WU> bit on each slave controller to 1 to enable data receiving.
3. The master controller transmits data one frame at a time. Each frame includes an 8-bit select code which identifies a slave controller. The MSB (Bit8) of the data (<TB8>) is set to 1.



4. Each slave controller receives the above frame. Each controller checks the above select code against its own select code. The controller whose code matches clears its <WU> bit to 0.
5. The master controller transmits data to the specified slave controller (The controller whose SC0MOD0<WU> bit has been cleared to 0). The MSB (Bit8) of the data (<TB8>) is cleared to 0.

6. The other slave controllers (whose <WU> bits remain at 1) ignore the received data because their MSBs (Bit8 or <RB8>) are set to 0, disabling INTRX0 interrupts.
   The slave controller whose <WU> bit = 0 can also transmit to the master controller. In this way it can signal the master controller that the data transmission from the master controller has been completed.

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



Since serial channels 0 and 1 operate in exactly the same way, channel 0 only is used for the purposes of this explanation.

- Setting the master controller

Main

		7	6	5	4	3	2	1	0	
PFCR	←	X	X	–	–	–	–	0	1	Set PF0 and PF1 to function as the TXD0 and RXD0 pins respectively.
PFFC	←	X	X	–	X	–	–	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_{IO} 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

		7	6	5	4	3	2	1	0	
PFCR	←	X	X	–	–	–	–	0	0	Select PF1 and PF0 to function as the RXD and TXD pins respectively (Open-drain output).
PFFC	←	X	X	–	X	–	–	X	1	
INTES0	←	1	1	0	1	1	1	1	0	Enable INTRX0 and INTTX0.
SC0MOD0	←	0	0	1	1	1	1	1	0	Set <WU> to 1 in 9-bit UART transmission mode using f_{SYS} as the transfer clock.

INTRX0 interrupt

Acc ← SC0BUF

if Acc = select code

Then SC0MOD0 ← – – – 0 – – – – Clear <WU> to 0.

3.9.5 Support for IrDA

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

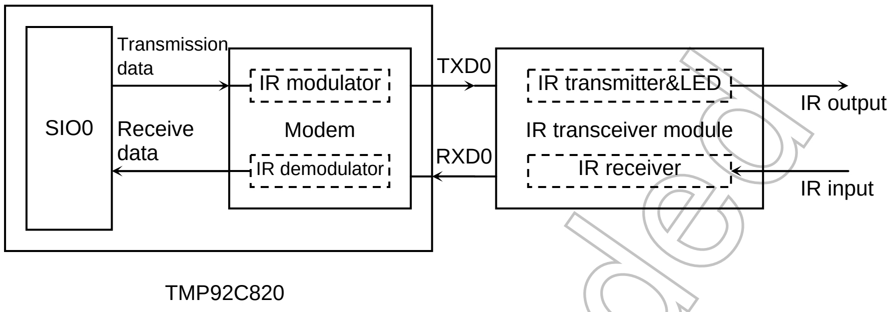


Figure 3.9.30 Block Diagram

(1) Modulation of the transmission data

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

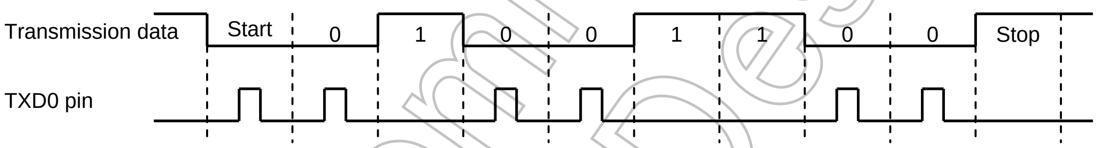


Figure 3.9.31 Transmission Example

(2) Modulation of the receive data

When the receive data is the effective width of pulse "1", the modem outputs "0" to SIO0. Otherwise the modem outputs "1" to SIO0.

The effective pulse width is selected by SIRCR<RXSEL>.

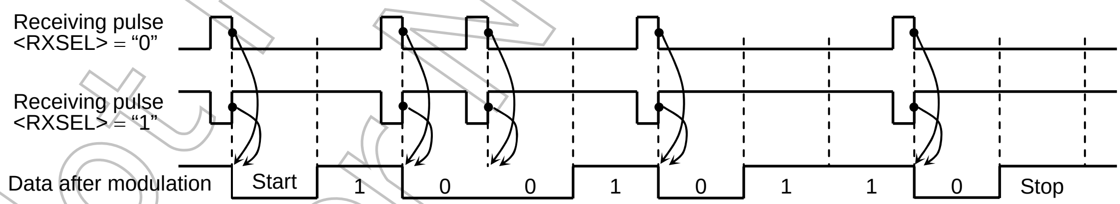


Figure 3.9.32 Receiving Example

(3) Data format

The data format is fixed as follows:

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

(4) SFRs

Figure 3.9.33 shows the control register SIRCR. Set the data SIRCR during SIO0 is 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 16X.
3)	LD (SIRCR), 37H ↓	; TXEN, RXEN enable the transmission and receiving.
4)	Start transmission and receiving for SIO0	; The modem operates as follows: • SIO0 starts transmitting. • IR receiver starts receiving.

(5) Notes

The IrDA 1.0 specification is defined in Table 3.9.4.

1. Making baud rate when using IrDA

   In baud rate during using IrDA, must set "01" to SC0MOD0<SC1:0> in SIO by using baud rate generator. TA0TRG, f_{IO}, SCLK0 input can not using.

2. Output pulse width and baud rate generator during transmission IrDA

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

Table 3.9.4 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 baud rate TX 3/16 or 1.6 μs (1.6 μs is equal to 3/16 pulse width when baud rate is 115.2 kbps).

The TMP92C820 has the function selects the pulse width of transmission either 3/16 or 1/16. But 1/16 pulse width can be selected when the baud rate is equal or less than 38.4 kbps.

As the same reason, + (16 – k)/16 division function in the baud rate generator of SIO0 can not be used to generate 115.2 kbps baud rate. Also when the 38.4 kbps and 1/16 pulse width, + (16 – k)/16 division function can not be used.

Table 3.9.5 Baud Rate and Pulse Width for (16 – k)/16 Division Function

Pulse Width	Baud Rate					
	115.2 kbps	57.6 kbps	38.4 kbps	19.2 kbps	9.6 kbps	2.4 kbps
T × 3/16	×	○	○	○	○	○
T × 1/16	–	–	×	○	○	○

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

SIRCR (1207H)

	7	6	5	4	3	2	1	0
Bit symbol	PLSEL	RXSEL	TXEN	RXEN	SIRWD3	SIRWD2	SIRWD1	SIRWD0
Read/Write	R/W							
After reset	0	0	0	0	0	0	0	0
Function	Select transmit pulse width 0: 3/16 1: 1/16	Receive data 0: "H" pulse 1: "L" pulse	Transmit 0: Disable 1: Enable	Receive 0: Disable 1: Enable	Select receive pulse width Set effective pulse width for equal or more than 2x × (Value + 1) + 100ns Can be set: 1 to 14 Can not be set: 0, 15			

Select receive pulse width

Formula: Effective pulse width ≥ 2x × (Value + 1) + 100ns
$x = 1/f_{SYS}$

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

Receive operation

0	Disabled (Received input is ignored.)
1	Enabled

Transmit operation

0	Disabled (Input from SIO is ignored.)
1	Enabled

Select transmit pulse width

0	3/16
1	1/16

Note: If a pulse width complying with the IrDA1.0 standard (1.6 μs min.) can be guaranteed with a low baud rate, setting this bit to "1" will result in result reduced power dissipation.

Figure 3.9.33 IrDA Control Register

3.10 Serial Bus Interface (SBI)

The TMP92C820 has 1-channel serial bus interface which employs a clocked-synchronous 8-bit SIO mode and an I²C bus mode. It is called SBI0.

The serial bus interface is connected to an external device through P91 (SDA) and P92 (SCL) in the I²C bus mode; and through P90 (SCK), P91 (SO), P92 (SI) in the clocked-synchronous 8-bit SIO mode.

Each pin is specified as follows.

	P9ODE<P92ODE, P91ODE>	P9CR<P92C, P91C, P90C>	P9FC<P92F, P91F, P90F>
I²C Bus Mode	11	11X	11X
Clocked Synchronous 8-Bit SIO Mode	XX	011 010	011 010 (Note)

X: Don't care

Note : When using SI input function and SCK input function, set P9FC<P92F,P90F> to "0" (Function setting).

3.10.1 Configuration

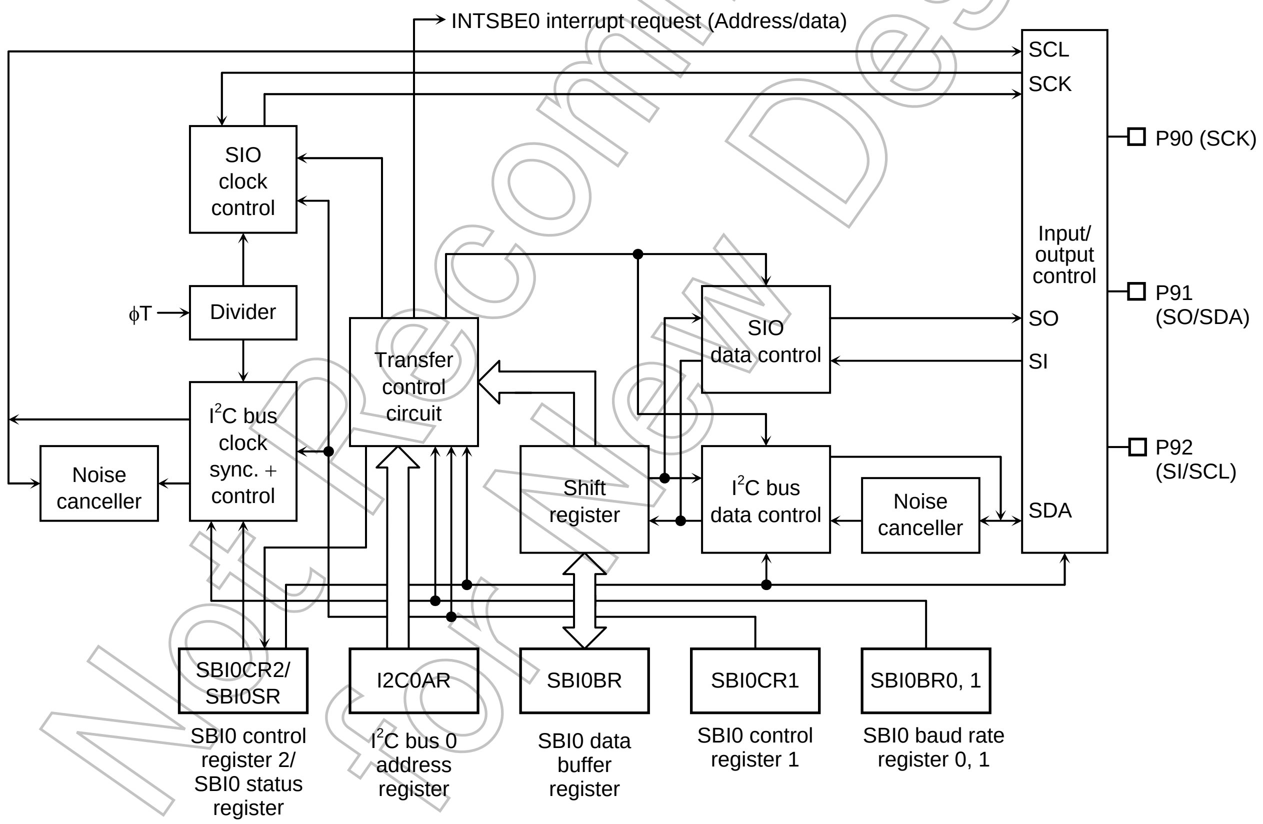


Figure 3.10.1 Serial Bus Interface 0 (SBI0)

3.10.2 Serial Bus Interface (SBI) Control

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

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

The above registers differ depending on a mode to be used. Refer to section 3.10.4 “I^2C Bus Mode Control Register” and 3.10.7 “Clocked-synchronous 8-Bit SIO Mode Control”.

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

The data formats in the I^2C bus mode are shown below.

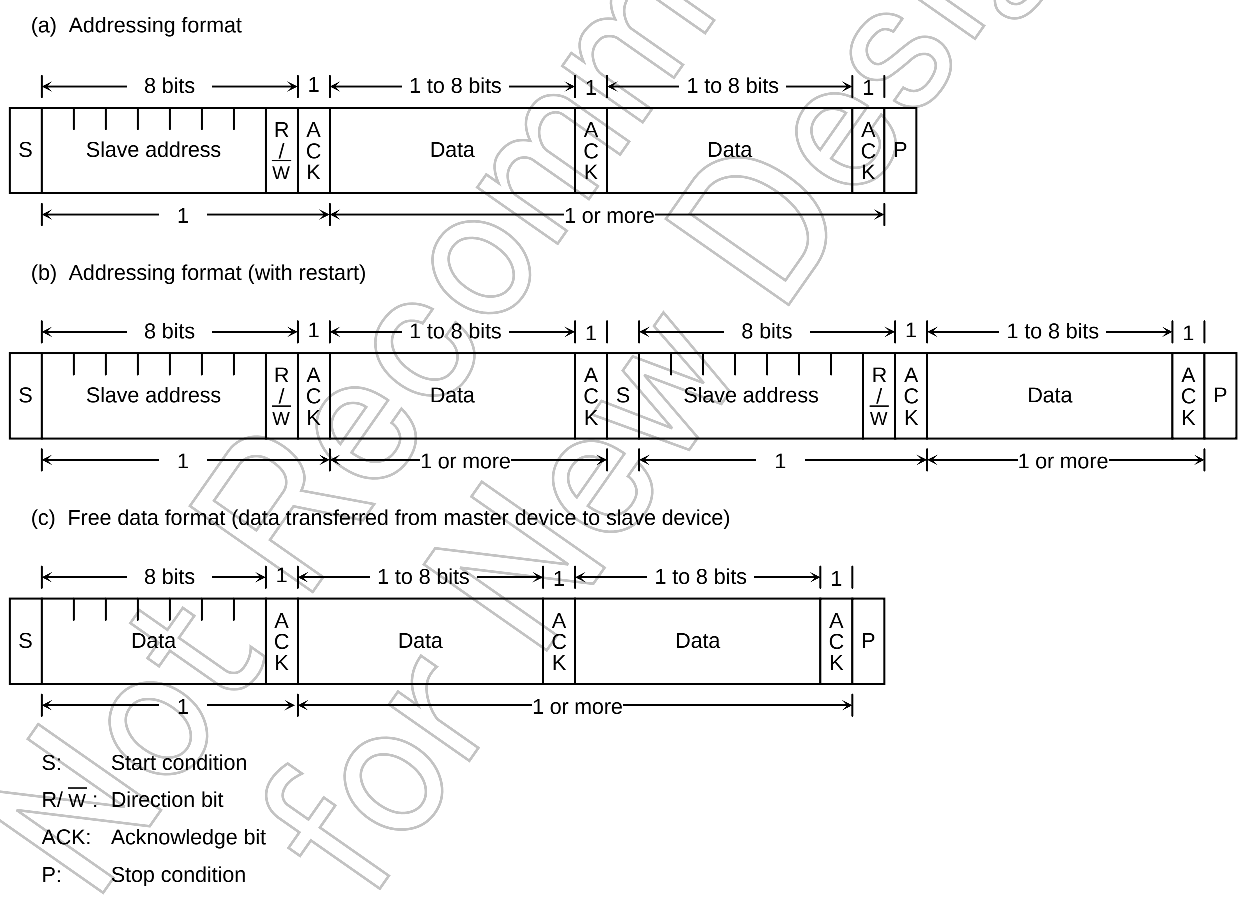


Figure 3.10.2 Data Format in the I^2C Bus Mode

3.10.4 I²C Bus Mode Control Register

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

Serial Bus Interface 0 Control Register 1

SBI0CR1 (1240H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0/ SWRMON
Read/Write	W			R/W		W		R/W
After reset	0	0	0	0		0	0	0/1 (Note 3)
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	– (Note 4)
001	n = 6	– (Note 4)
010	n = 7	– (Note 4)
011	n = 8	75.8 kHz
100	n = 9	38.5 kHz
101	n = 10	19.4 kHz
110	n = 11	9.73 kHz
111	Reserved	(Reserved)

System clock: f_{SYS}
f_{SYS} = 20 MHz
(internal SCL output)

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

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 clocked-synchronous 8-bit SIO mode.

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

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

Note 4: This I²C bus circuit does not support Fast-mode, it supports the Standard mode only. Although the I²C bus circuit itself allows the setting of a baud rate over 100kbps, the compliance with the I²C specification is not guaranteed in that case.

Figure 3.10.3 Registers for the I²C Bus Mode

Serial Bus Interface 0 Control Register 2

SBI0CR2 (1243H) Prohibit read-modify-write		7	6	5	4	3	2	1	0
	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
	Function	Master/ slave selection	Transmitter/ receiver selection	Start/stop condition generation	Cancel INTSBE0 interrupt request	Serial bus interface operating mode selection (Note 2) 00: Port mode 01: SIO mode 10: I²C bus mode 11: (Reserved)		Software reset generate write "10" and "01", then an internal software 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	I²C bus mode
11	(Reserved)

INTSBE0 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 I²C bus mode and clocked-synchronous 8-bit SIO mode after confirming that input signals via port are high level.

Figure 3.10.4 Registers for the I²C Bus Mode

Serial Bus Interface 0 Status Register

SBI0SR (1243H)

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	I²C bus status monitor	INTSBE0 interrupt request monitor	Arbitration lost detection monitor 0: – 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

INTSBE0 interrupt request monitor

0	Interrupt requested
1	Interrupt canceled

I²C 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 I²C Bus Mode

Serial Bus Interface 0 Baud Rate Register 0

SBI0BR0 (1244H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	–	I2SBI0						
Read/Write	W	R/W						
After reset	0	0						
Function	Always write "0".	IDLE2 0: Stop 1: Run						

Operation during IDLE2 mode

0	Stop
1	Operation

Serial Bus Interface 0 Baud Rate Register 1

SBI0BR1 (1245H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	P4EN	–						
Read/Write	W	W						
After reset	0	0						
Function	Internal clock 0: Stop 1: Operate	Always write "0".						

Baud rate clock control

0	Stop
1	Operate

Serial Bus Interface 0 Data Buffer Register

SBI0DBR (1241H)

Prohibit read-modify-write

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

Note 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.

I^2C Bus 0 Address Register

I2C0AR (1242H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
Read/Write	W							
After reset	0	0	0	0	0	0	0	0
Function	Slave address selection for when device is operating as slave device							Address recognition mode specification

Address recognition mode specification

0	Slave address recognition
1	Non slave address recognition

Figure 3.10.6 Registers for the I^2C Bus Mode

3.10.5 Control in I^2C Bus Mode

(1) Acknowledge mode specification

Set the SBI0CR1<ACK> to “1” for operation in the acknowledge mode. The TMP92C820 generates an additional clock pulse for an acknowledge signal when operating in master mode. In the transmitter mode during the clock pulse cycle, the SDA pin is released in order to receive the acknowledge signal from the receiver. In the receiver mode during the clock pulse cycle, the SDA pin is set to the low in order to generate the acknowledge signal.

Clear the <ACK> to “0” for operation in the non-acknowledge mode. The TMP92C820 does not generate a clock pulse for the acknowledge signal when operating in the master mode.

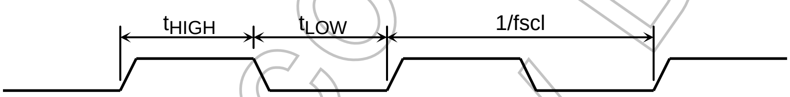
(2) Number of transfer bits

Since the SBI0CR1<BC2:0> is cleared to “000” on start up, a slave address and direction bit transmissions are executed in 8 bits. Other than these, the <BC2:0> retains a specified value.

(3) Serial clock

1. Clock source

The SBI0CR1<SCK2:0> is used to specify the maximum transfer frequency for output on the SCL pin in the master mode. Set the baud rates, which have been calculated according to the formula below, to meet the specifications of the I^2C bus, such as the smallest pulse width of t_{LOW}.



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

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

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

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

Note: f_{SBI} is the clock f_{SYS}.

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

Figure 3.10.7 Clock Source

2. Clock synchronization

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

This device has a clock synchronization function which allows normal data transfer even when more than one master exists on the bus.

The following example explains the clock synchronization procedures used when there are two masters present on the bus.

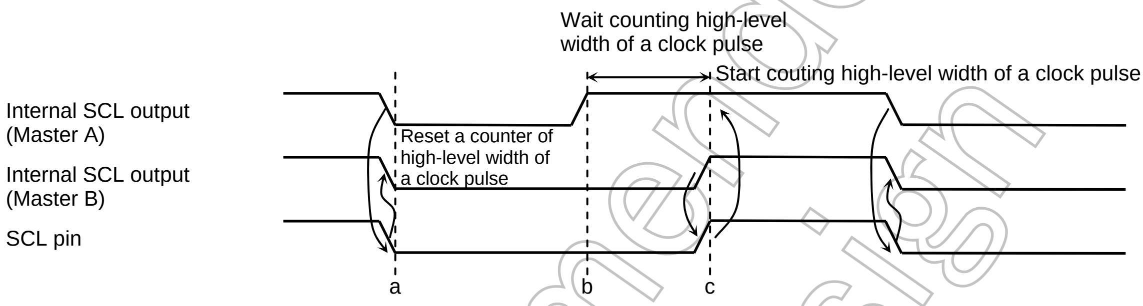


Figure 3.10.8 Clock Synchronization

When master A pulls the internal SCL output to the low level at point "a", the bus's SCL pin goes to the low level. After detecting this, master B resets a counter of high-level width of an own clock pulse and sets the internal SCL output the low level.

Master A finishes counting low-level width of an own clock pulse at point "b" and sets the internal SCL output to the high level. Since master B is holding the bus's SCL pin the low level, master A waits for counting high-level width of an own clock pulse. After master B has finished counting low-level width of an own clock pulse at point "c" and master A detects the SCL pin of the bus at the high level, and starts counting high level of an own clock pulse.

The clock pulse on the bus is determined by the master device with the shortest high-level width and the master device with the longest low-level width from among those master devices connected to the bus.

(4) Slave address and address recognition mode specification

When this device is to be used as a slave device, set the slave address <SA6:0> and <ALS> in I2C0AR.

Clear the <ALS> to "0" for the address recognition mode.

(5) Master/slave selection

To operate this device as a master device set the SBI0CR2<MST> to "1".

To operate it as a slave device clear the SBI0CR2<MST> to "0". The <MST> is cleared to "0" in hardware when a stop condition is detected on the bus or when arbitration is lost.

(6) Transmitter/receiver selection

To operate this device as a transmitter set the SBI0CR2<TRX> to "1". To operate it as a receiver clear the SBI0CR2<TRX> to "0".

When data with an addressing format is transferred in the slave mode, when a slave address with the same value that an I2C0AR or a GENERAL CALL is received (All 8-bit data are "0" after a start condition), the <TRX> is set to "1" in hardware if the direction bit (R/$\overline{\text{W}}$) sent from the master device is "1", and is cleared to "0" in hardware if the bit is "0".

In the master mode, when an acknowledge signal is returned from the slave device, the <TRX> is cleared to "0" in hardware if the value of the transmitted direction bit is "1", and is set to "1" in hardware if the value of the bit is "0". If an acknowledge signal is not returned, the current state is maintained.

The <TRX> is cleared to "0" in hardware when a stop condition is detected on the I^{2}C bus or when arbitration is lost.

(7) Start/stop condition generation

When the SBI0SR<BB> = "0", slave address and direction bit which are set to SBI0DBR is output on the bus after generating a start condition by writing "1111" to the SBI0CR2<MST, TRX, BB, PIN>. It is necessary to set transmitted data to the data buffer register (SBI0DBR) and set "1" to the <ACK> beforehand.

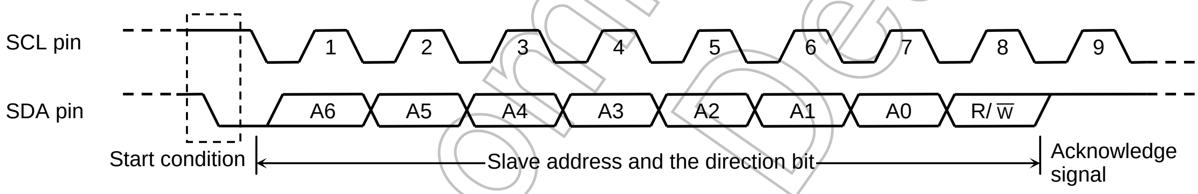


Figure 3.10.9 Start Condition Generation and Slave Address Generation

When the SBI0SR<BB> = "1", the sequence for generating a stop condition can be initiated by writing "111" to the SBI0CR2<MST, TRX, PIN> and writing "0" to the SBI0CR2<BB>. Do not modify the contents of the SBI0CR2<MST, TRX, BB, PIN> until a stop condition has been generated on the bus.

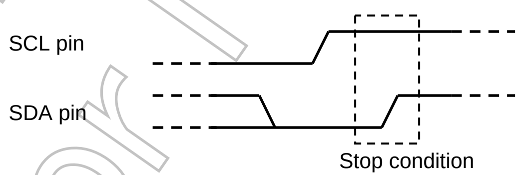


Figure 3.10.10 Stop Condition Generation

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

Stop condition generation in master mode have limit. Therefore, please refer to 3.10.6 (4) "Stop condition generation".

(8) Interrupt service requests and interrupt cancellation

When a serial bus interface interrupt request 0 by transfer of the slave address or the data (INTSBE0) is generated, the SBI0SR<PIN> is cleared to “0”. The SCL pin is pulled down to the low-level while the <PIN> = “0”.

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

The time from the <PIN> being set to “1” until the release of the SCL pin is t_{LOW}.

In the address recognition mode (e.g., when <ALS> = “0”), the <PIN> is cleared to “0” when the slave address matches the value set in I2C0AR or when a GENERAL CALL is received (All 8-bit data are “0” after a start condition). Although the SBI0CR2<PIN> can be set to “1” by a program, writing “0” to the SBI0CR2<PIN> does not clear it to “0”.

(9) Serial bus interface operation mode selection

The SBI0CR2<SBIM1:0> is used to specify the serial bus interface operation mode.

Set the SBI0CR2<SBIM1:0> to “10” when the device is to be used in I^2C bus mode after confirming pin condition of serial bus interface to “H”.

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

(10) Arbitration lost detection monitor

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

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

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

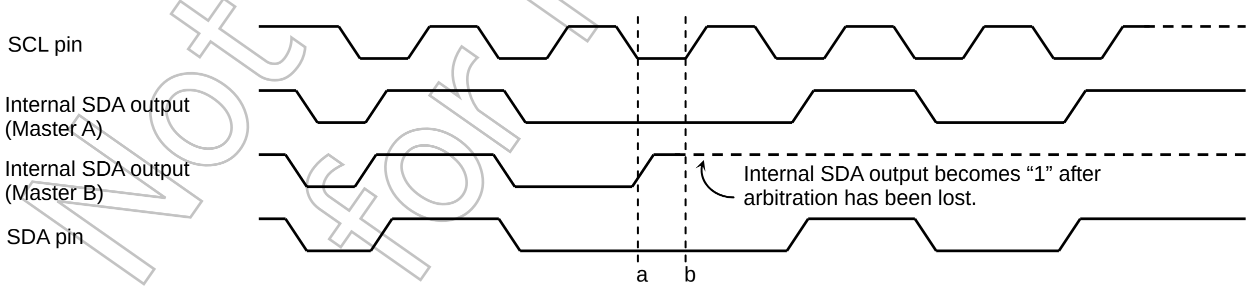


Figure 3.10.11 Arbitration Lost

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

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

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

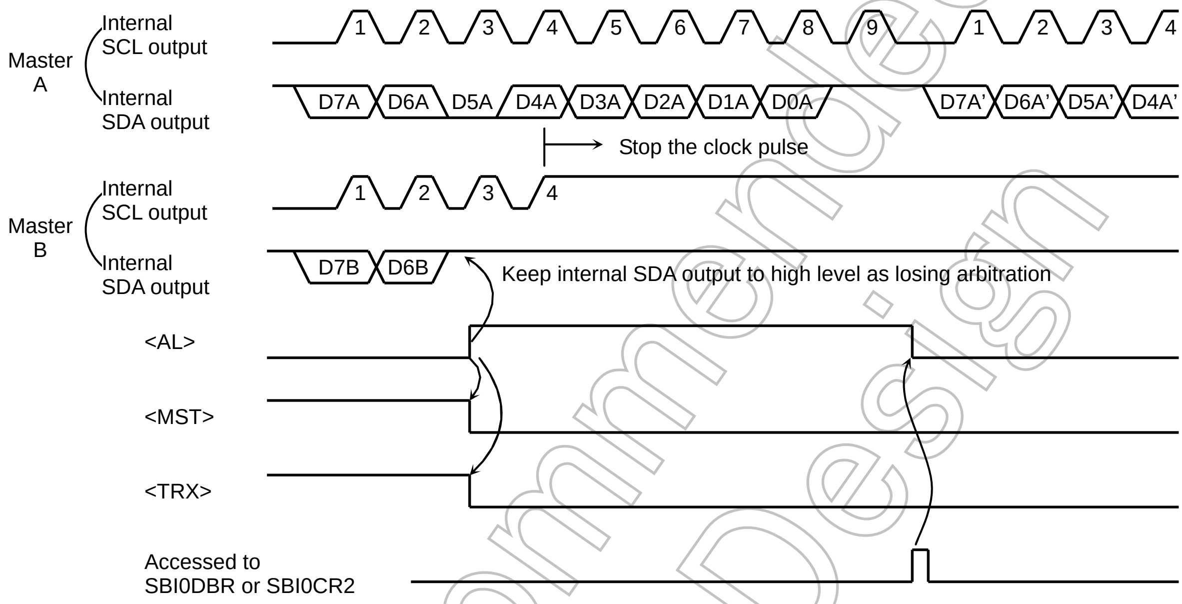


Figure 3.10.12 Example of a Master Device B (D7A = D7B, D6A = D6B)

(11) Slave address match detection monitor

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

(12) GENERAL CALL detection monitor

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

(13) Last received bit monitor

The value on the SDA pin detected on the rising edge of the SCL pin is stored in the SBI0SR<LRB>.

In the acknowledge mode, immediately after an INTSBE0 interrupt request has been generated, an acknowledge signal is read by reading the contents of the SBI0SR<LRB>.

(14) Software reset function

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

An internal reset signal pulse can be generated by setting SBI0CR2<SWRST1:0> to “10” and “01”. This initializes the SBI circuit internally.

All command (except SBI0CR2<SBIM1:0>) registers and status registers are initialized as well.

The SBI0CR1<SWRMON> is automatically set to “1” after the SBI circuit has been initialized.

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

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

When the start condition has been generated in the master mode, the slave address and the direction bit are set in this register.

(16) I^{2}C bus address register (I2C0AR)

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

The slave address output from the master device is recognized by setting I2C0AR<ALS> is set to “0”. The data format is the addressing format. When the slave address in not recognized at the <ALS> is set to “1”, the data format is the free data format.

(17) Baud rate register (SBI0BR1)

Write “1” to the SBI0BR1<P4EN> before operation commences.

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

The setting of SBI0BR0<I2SBI0> determines whether the device is operating or is stopped in IDLE2 mode.

Therefore, setting <I2SBI0> is necessary before the HALT instruction is executed.

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

(1) Device initialization

Set the SBI0BR1<P4EN> and the SBI0CR1<ACK, SCK2:0>. Set the SBI0BR1<P4EN> to “1” and clear bits 7 to 5 and 3 of the SBI0CR1 to “0”.

Set a slave address in I2C0AR<SA6:0> and the I2C0AR<ALS> (<ALS> = “0” when an addressing format.)

For specifying the default setting to a slave receiver mode, clear “000” to the <MST, TRX, BB>, set “1” to the <PIN>, set “10” to the <SBIM1:0> and set “00” to the <SWRST1:0>.

(2) Start condition generation and slave address generation

1. Master mode

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

Check a bus free status (when <BB> = “0”).

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

When the <BB> is ”0”, the start condition is generated by writing “1111” to the SBI0CR2<MST, TRX, BB, PIN>. Subsequently to the start condition, 9 clocks are output from the SCL pin. While 8 clocks are output, the slave address and the direction bit which are set to the SBI0DBR. At the 9th clock pulse the SDA pin is released and the acknowledge signal is received from the slave device.

An INTSBE0 interrupt request occurs on the falling edge of the 9th clock pulse. The <PIN> is cleared to “0”. In the master mode the SCL pin is pulled down to the low level while the <PIN> is “0”. When an INTSBE0 interrupt request occurs, the value of <TRX> is changed according to the direction bit setting only if the slave device returns an acknowledge signal.

2. Slave mode

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

After the start condition has been received from the master device, while 8 clocks are output from the SCL pin, the slave address and the direction bit which are output from the master device are received.

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

An INTSBE0 interrupt request occurs on the falling edge of the 9th clock pulse. The <PIN> is cleared to “0”. In the slave mode the SCL pin is pulled down to the low-level while the <PIN> = “0”. When an interrupt request occurs, the value of <TRX> is changed according to the direction bit setting only if the slave device returns an acknowledge signal.

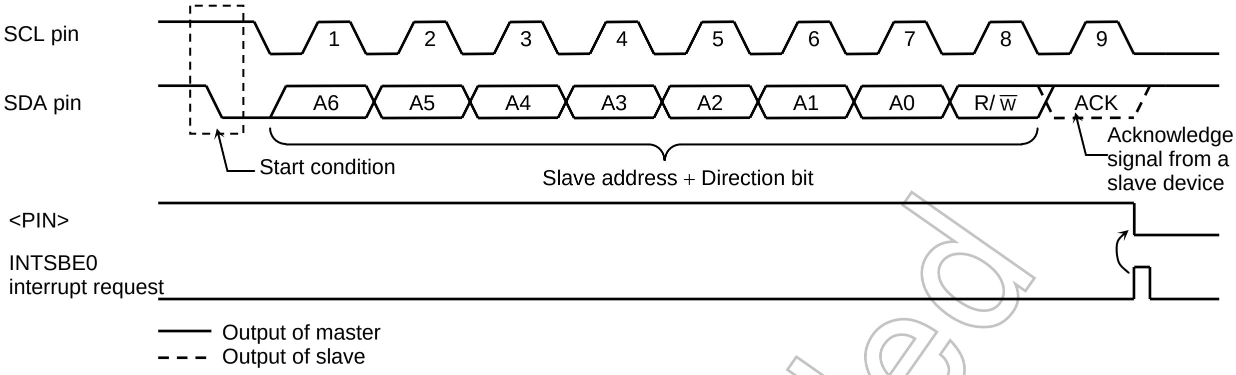


Figure 3.10.13 Start Condition Generation and Slave Address Transfer

(3) 1-word data transfer

Check the <MST> setting using an INTSBE0 interrupt process after the transfer of each word of data is completed and determine whether the device is in the master mode or the slave mode.

1. When the <MST> is "1" (Master mode)

   Check the <TRX> setting and determine whether the device is in the transmitter mode or the receiver mode.

   When the <TRX> is "1" (Transmitter mode)

   Check the <LRB> setting. When the <LRB> = "1", there is no receiver requesting data. Implement the process for generating a stop condition (See section 3.10.6 (4).) and terminate data transfer.

   When the <LRB> = "0", the receiver is requesting new data. When the next transmitted data is 8 bits, write the transmitted data to the SBI0DBR. When the next transmitted data is other than 8 bits, set the <BC2:0>, set the <ACK> to "1" and write the transmitted data to the SBI0DBR. After the data has been written, the <PIN> is set to "1", a serial clock pulse is generated to trigger transfer of the next word of data via the SCL pin, and the word is transmitted. After the data has been transmitted, an INTSBE0 interrupt request is generated. The <PIN> is set to "0" and the SCL pin is pulled down to the low level. If the length of the data to be transferred is greater than one word, repeat the latter steps of the procedure, starting from the check of the <LRB> setting.

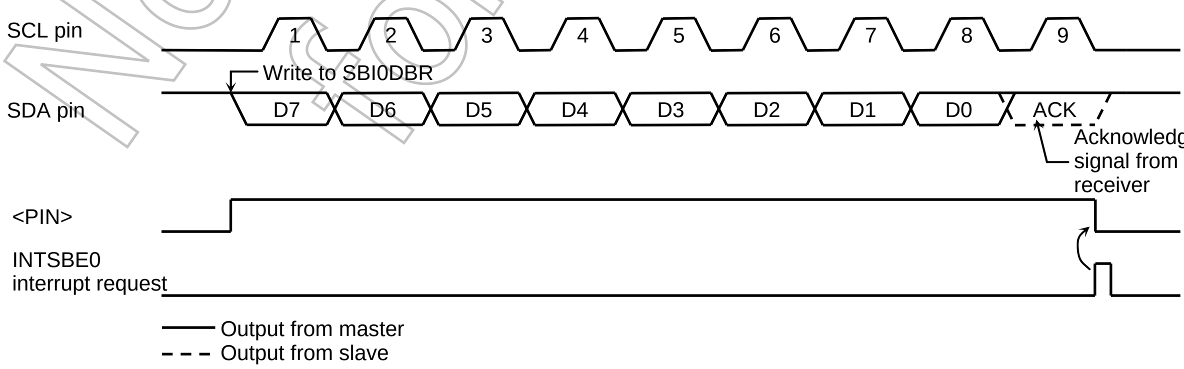


Figure 3.10.14 Example in which <BC2:0> = "000" and <ACK> = "1" in Transmitter Mode

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

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

An INTSBE0 interrupt request is generated and the <PIN> is set to "0". Then this device pulls down the SCL pin to the low level. This device outputs a clock pulse for 1 word of data transfer and the acknowledge signal each time that received data is read from SBI0DBR.

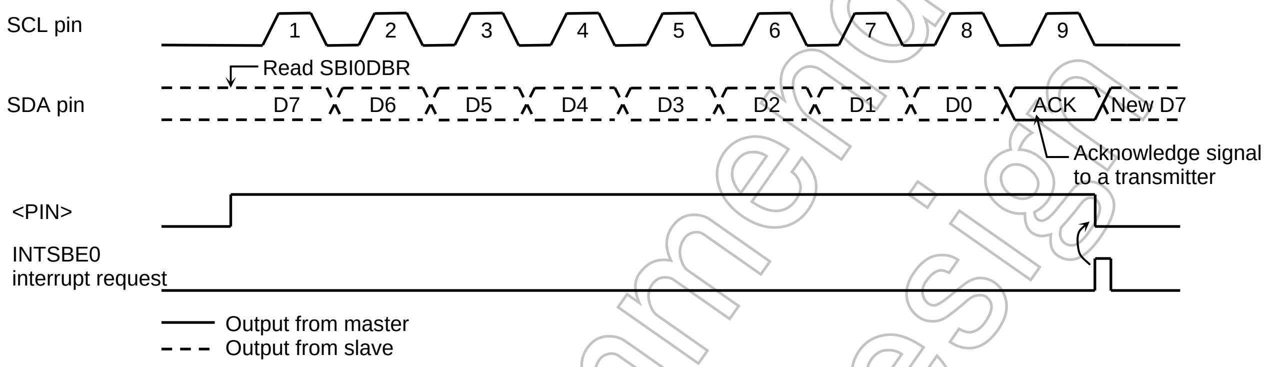


Figure 3.10.15 Example of when <BC2:0> = "000", <ACK> = "1" in Receiver Mode

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

After 1-bit data is received and an interrupt request has occurred, this device generates a stop condition (See section 3.10.6 (4).) and terminates data transfer.

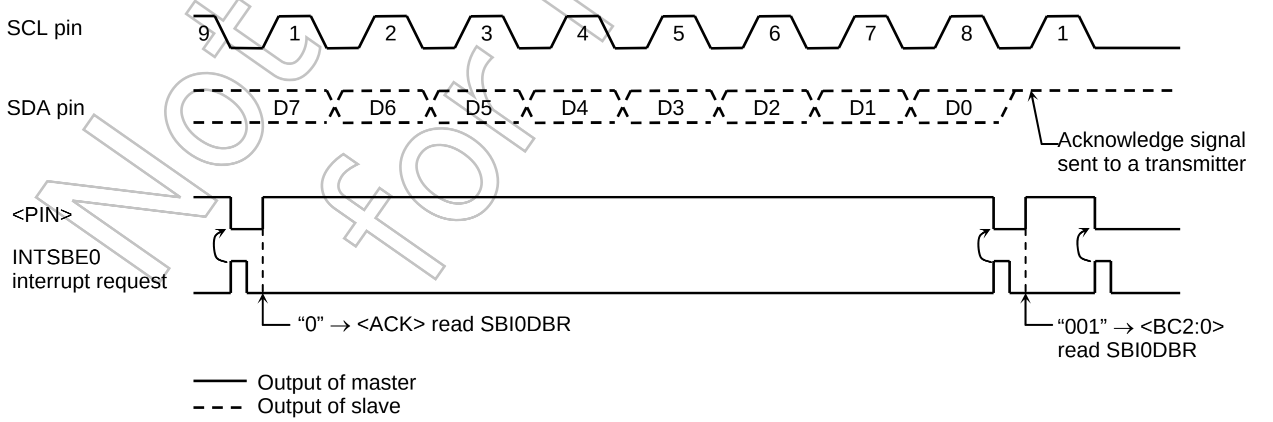


Figure 3.10.16 Termination of Data Transfer in Master Receiver Mode

2. When the <MST> is “0” (Slave mode)

In the slave mode, this device operates either in normal slave mode or in slave mode after losing arbitration.

In the slave mode, an INTSBE0 interrupt request occurs when this device receives a slave address or a GENERAL CALL from the master device, or when a GENERAL CALL is received and data transfer is complete, or after matching a received slave address. In the master mode, this device operates in a slave mode if it is losing arbitration. An INTSBE0 interrupt request occurs when word data transfer terminates after losing arbitration. When an INTSBE0 interrupt request occurs, the <PIN> is cleared to “0”, and the SCL pin is pulled down to the low level. Either reading data to or writing data from the SBI0DBR, or setting the <PIN> to “1” releases the SCL pin after taking t_{LOW} time.

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

Table 3.10.1 Operation in the Slave Mode

<TRX>	<AL>	<AAS>	<AD0>	Conditions	Process
1	1	1	0	This device loses arbitration when transmitting a slave address and receives a slave address of which the value of the direction bit sent from another master is “1”.	Set the number of bits in 1 word to the <BC2:0> and write the transmitted data to the SBI0DBR.
	0	1	0	In the slave receiver mode, this device receives a slave address of which the value of the direction bit sent from the master is “1”.	
		0	0	In the slave transmitter mode, 1-word data is transmitted.	Check the <LRB>. If the <LRB> is set to “1”, set the <PIN> to “1” since the receiver does not request the next data. Then, clear the <TRX> to “0” to release the bus. If the <LRB> is cleared to “0”, set the number of bits in a word to the <BC2:0> and write transmitted data to the SBI0DBR since the receiver requests next data.
0	1	1	1/0	This device loses arbitration when transmitting a slave address and receives a GENERAL CALL or slave address of 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	This device loses arbitration when transmitting a slave address or data and terminates transferring word data.	
	0	1	1/0	In the slave receiver mode, this device receives a GENERAL CALL or slave address of which the value of the direction bit sent from the master is “0”.	
		0	1/0	In the slave receiver mode, the device terminates receiving 1-word data.	Set the number of bits in a word to the <BC2:0> and read received data from the SBI0DBR.

(4) Stop condition generation

When the SBI0SR<BB> is "1", the sequence of generating a stop condition is started by setting "111" to the SBI0CR2<MST, TRX, PIN> and "0" to the SBI0CR2<BB>. Do not modify the contents of the SBI0CR2<MST, TRX, PIN, BB> until a stop condition is generated on a bus.

When a SCL pin of bus is pulled down by other devices, this device generates a stop condition after they release a SCL pin and the SDA becomes "1".

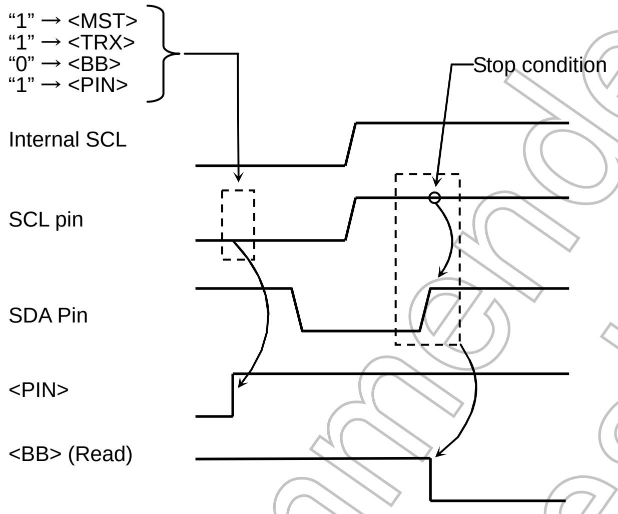


Figure 3.10.17 Stop Condition Generation (Single master)

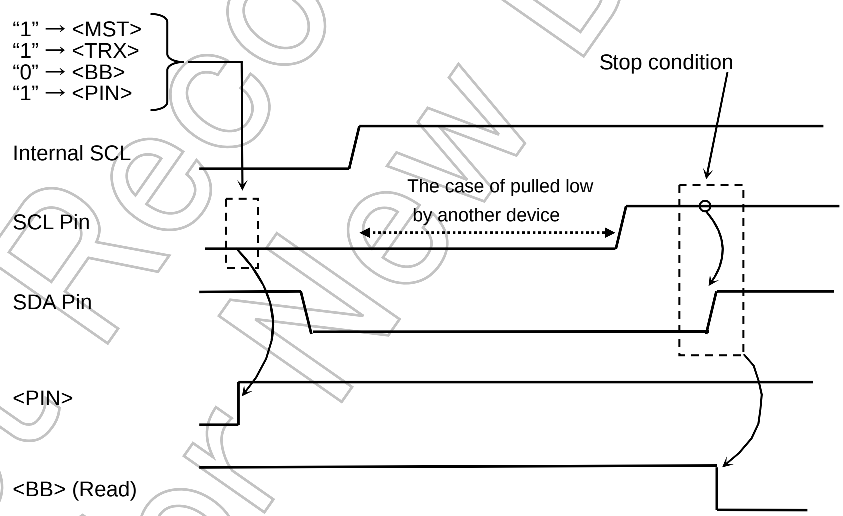


Figure 3.10.18 Stop Condition Generation (Multi master)

(5) Restart

Restart is used during data transfer between a master device and a slave device to change the data transfer direction. The following description explains how to restart when this device is in the master mode.

Clear the SBI0CR2<MST, TRX, BB> to "000" and set the SBI0CR2<PIN> to "1" to release the bus. The SDA line remains the high level and the SCL pin is released. Since a stop condition is not generated on the bus, other devices assume the bus to be in a busy state. Check the SBI0SR<BB> until it becomes "0" to check that the SCL pin of this device is released. Check the <LRB> until it becomes 1 to check that the SCL line on a bus is not pulled down to the low level by other devices. After confirming that the bus stays in a free state, generate a start condition with procedure described in 3.10.6 (2).

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

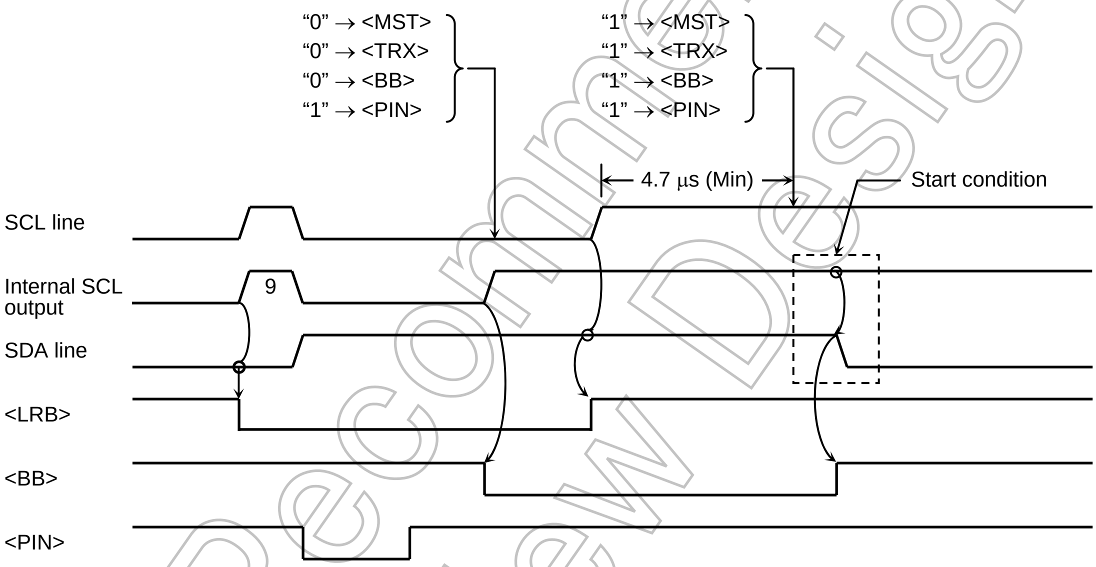


Figure 3.10.19 Timing Diagram when Restarting

3.10.7 Clocked-synchronous 8-Bit SIO Mode Control

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

Serial Bus Interface 0 Control Register 1

SBI0CR1 (1240H)

Prohibit read-modify-write

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

Serial clock selection <SCK2:0> at write

000	n = 4	1.25 MHz
001	n = 5	625 kHz
010	n = 6	313 kHz
011	n = 7	156 kHz
100	n = 8	78.1 kHz
101	n = 9	39.1 kHz
110	n = 10	19.5 kHz
111	–	External clock: SCK0

System clock: f_{SYS}
f_{SYS} = 20 MHz
(output to SCK pin)

$$fscl = \frac{f_{SYS}}{2^n} \text{ [Hz]}$$

Transfer mode selection

00	8-bit transmit mode
01	(Reserved)
10	8-bit transmit/receive mode
11	8-bit receive mode

Continue/abort transfer

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

Indicate transfer start/stop

0	Stop
1	Start

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

Serial Bus Interface 0 Data Buffer Register

SBI0DBR (1241H)

Prohibit read-modify-write

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

Figure 3.10.20 Register for the SIO Mode

Serial Bus Interface 0 Control Register 2

SBI0CR2 (1243H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol					SBIM1	SBIM0	–	–
Read/Write					W		W	W
After reset					0	0	0	0
Function					Serial bus interface operation mode selection 00: Port mode 01: SIO mode 10: I^2C bus mode 11: (Reserved)		(Note 2)	(Note 2)

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 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)

Serial Bus Interface 0 Status Register

SBI0SR (1243H)

	7	6	5	4	3	2	1	0
Bit symbol					SIOF	SEF		
Read/Write					R			
After reset					0	0		
Function					Serial transfer operation status monitor	Shift operation status monitor		

Serial transfer operating status monitor

0	Transfer terminated
1	Transfer in progress

Shift operation status monitor

0	Shift operation terminated
1	Shift operation in progress

Serial Bus Interface 0 Baud Rate Register 0

SBI0BR0 (1244H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	–	–						
Read/Write	W	R/W						
After reset	0	0						
Function	Always write “0”.	Always write “0”.						

Note: Clocked-synchronous mode cannot operate in IDLE2 mode.

Serial Bus Interface 0 Baud Rate Register 1

SBI0BR1 (1245H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	P4EN	–						
Read/Write	W	W						
After reset	0	0						
Function	Internal clock 0: Stop 1: Operate	Always write “0”.						

Baud rate clock control

0	Stop
1	Operate

Figure 3.10.21 Registers for the SIO Mode

(1) Serial clock

1. Clock source

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

Internal clock

In an internal clock mode, any of seven frequencies can be selected. The serial clock is output to the outside on the SCK pin.

When the device is writing (in the transmit mode) or reading (in the receive mode) data cannot follow the serial clock rate, an automatic wait function is executed to stop the serial clock automatically and holds the next shift operation until reading or writing is complete.

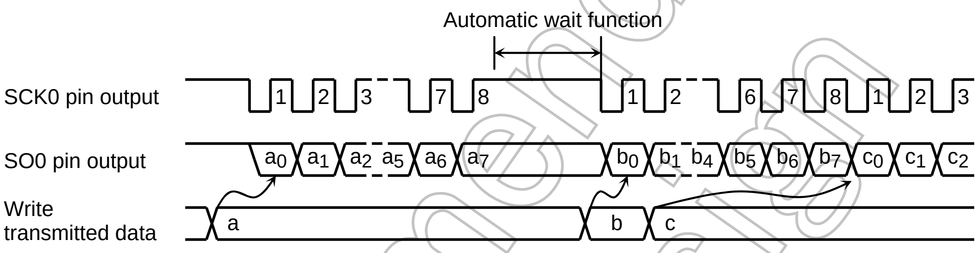


Figure 3.10.22 Automatic Wait Function

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

An external clock input via the SCK pin is used as the serial clock. In order to ensure the integrity of shift operations, both the high and low-level serial clock pulse widths shown below must be maintained. The maximum data transfer frequency is 1.25 MHz (when f_{SYS} = 20 MHz).

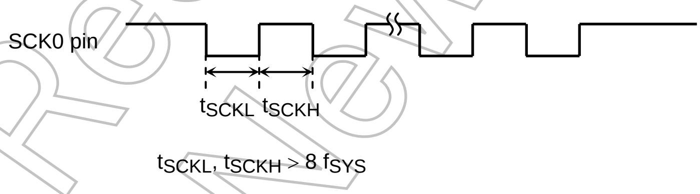


Figure 3.10.23 Maximum Data Transfer Frequency when External Clock Input

2. Shift edge

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

(a) Leading edge shift

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

(b) Trailing edge shift

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

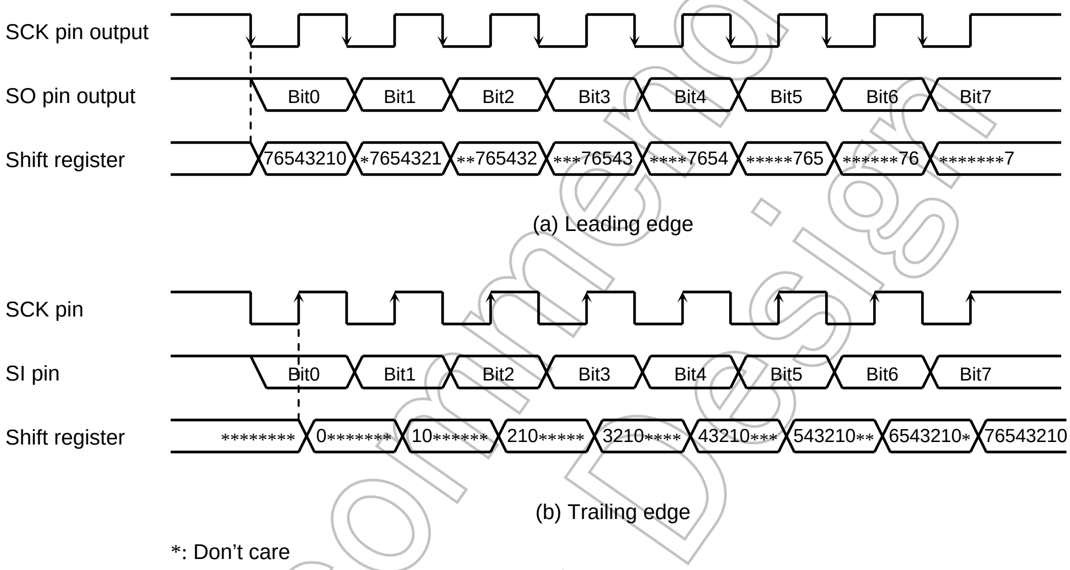


Figure 3.10.24 Shift Edge

(2) Transfer modes

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

1. 8-bit transmit mode

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

After the transmit data has been written, set the SBI0CR1<SIOS> to "1" to start data transfer. The transmitted data is transferred from the SBI0DBR to the shift register and output, starting with the least significant bit (LSB), via the SO pin and synchronized with the serial clock. When the transmission data has been transferred to the shift register, the SBI0DBR becomes empty. The INTSBE0 (Buffer empty) interrupt request is generated to request new data.

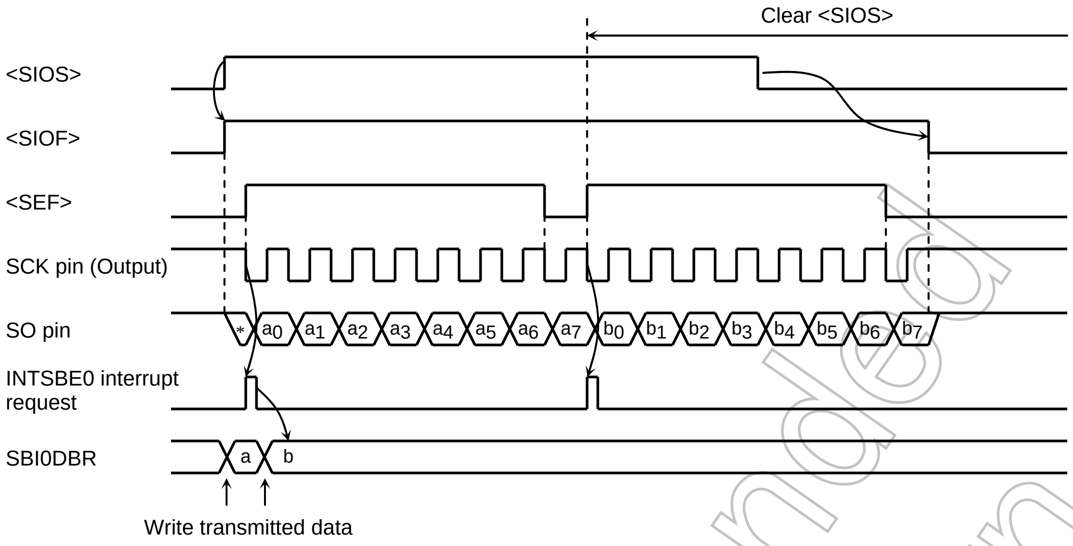
When the internal clock is used, the serial clock will stop and the automatic wait function will be initiated if new data is not loaded to the data buffer register after the specified 8-bit data is transmitted. When new transmission data is written, the automatic wait function is canceled.

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

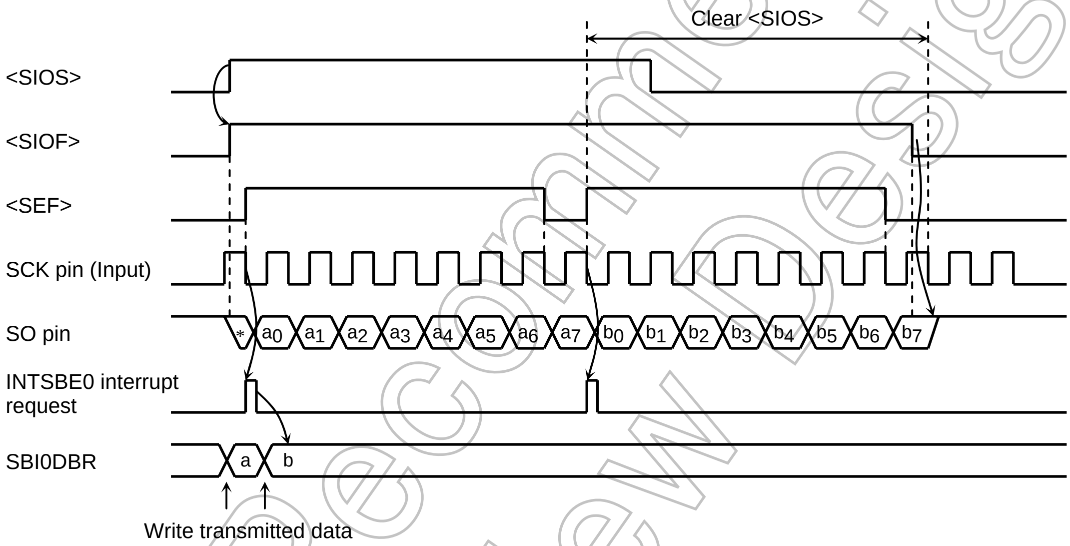
When the transmit is started, after the SBI0SR<SIOF> goes "1" output from the SO pin holds final bit of the last data until falling edge of the SCK.

Data transmission ends when the <SIOS> is cleared to "0" by the INTSBE0 interrupt service program or when the <SIOINH> is set to "1". When the <SIOS> is cleared to "0", the transmitted mode ends when all data is output. In order to confirm whether data is being transmitted properly by the program, the <SIOF> (Bit3 of the SBI0SR) to be sensed. The SBI0SR<SIOF> is cleared to "0" when transmission has been completed. When the <SIOINH> is set to "1", transmitting data stops. The <SIOF> turns "0".

When the external clock is used, it is also necessary to clear the <SIOS> to "0" before new data is shifted; otherwise, dummy data is transmitted and operation ends.



(a) Internal clock



(b) External clock

Figure 3.10.25 Transfer Mode

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

```
STEST1:  BIT   2, (SBI0SR)              ;  If <SEF> = 1 then loop
         JR    NZ, STEST1
STEST2:  BIT   0, (P9)                  ;  If SCK0 = 0 then loop
         JR    Z, STEST2
         LD    (SBI0CR1), 00000111B     ;  <SIOS> ← 0
```

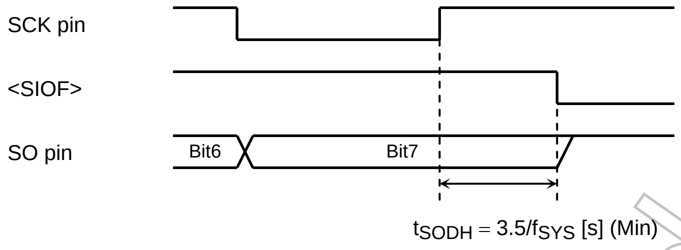


Figure 3.10.26 Transmitted Data Hold Time at End of Transmission

2. 8-bit receive mode

Set the control register to receive mode and set the SBI0CR1<SIOS> to "1" for switching to receive mode. Data is received into the shift register via the SI pin and synchronized with the serial clock, starting from the least significant bit (LSB). When the 8-bit data is received, the data is transferred from the shift register to the SBI0DBR. The INTSBE0 (Buffer full) interrupt request is generated to request that the received data be read. The data is then read from the SBI0DBR by the interrupt service program.

When the internal clock is used, the serial clock will stop and the automatic wait function will be in effect until the received data is read from the SBI0DBR.

When the external clock is used, since shift operation is synchronized with an external clock pulse, the received data should be read from the SBI0DBR before the next serial clock pulse is input. If the received data is not read, further data to be received is canceled. The maximum transfer speed when an external clock is used is determined by the delay time between the time when an interrupt request is generated and the time when the received data is read.

Receiving of data ends when the <SIOS> is cleared to "0" by the INTSBE0 interrupt service program or when the <SIOINH> is set to "1". If <SIOS> is cleared to "0", received data is transferred to the SBI0DBR in complete blocks. The received mode ends when the transfer is complete. In order to confirm whether data is being received properly by the program, the SBI0SR<SIOF> to be sensed. The <SIOF> is cleared to "0" when receiving is complete. When it is confirmed that receiving has been completed, the last data is read. When the <SIOINH> is set to "1", data receiving stops. The <SIOF> is cleared to "0". (The received data becomes invalid, therefore no need to read it.)

Note: When the transfer mode is changed, the contents of the SBI0DBR will be lost. If the mode must be changed, conclude data receiving by clearing the <SIOS> to "0", read the last data, then change the mode.

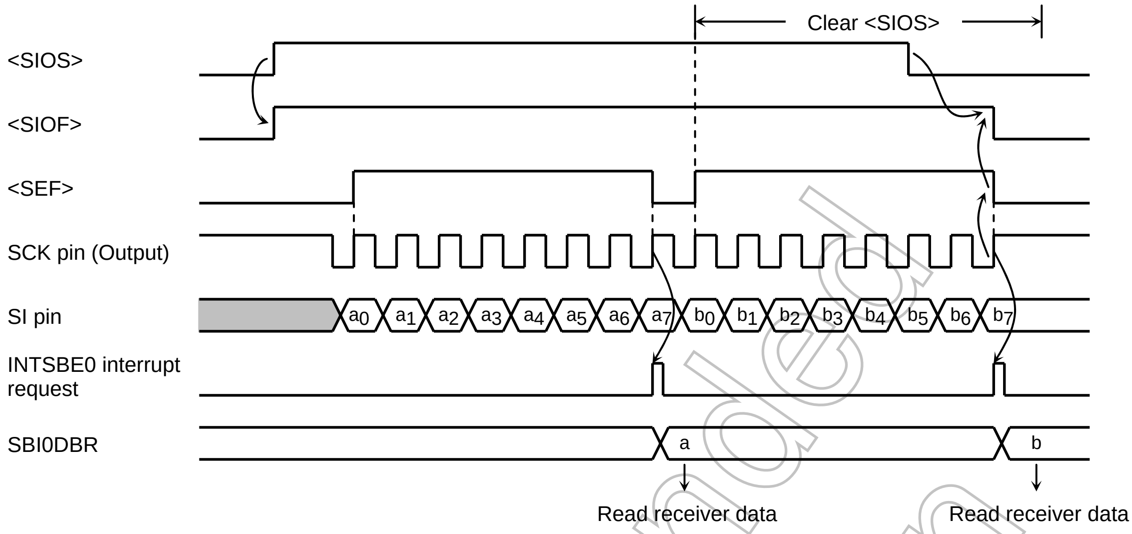


Figure 3.10.27 Receiver Mode (Example: Internal clock)

3. 8-bit transmit/receive mode

Set a control register to a transmit/receive mode and write data to the SBI0DBR. After the data is written, set the SBI0CR<SIOS> to "1" to start transmitting/receiving. When data is transmitted, the data is output from the SO pin, starting from the least significant bit (LSB) and synchronized with the leading edge of the serial clock signal. When data is received, the data is input via the SI pin on the trailing edge of the serial clock signal. 8-bit data is transferred from the shift register to the SBI0DBR and the INTSBE0 interrupt request is generated. The interrupt service program reads the received data from the data buffer register and writes the data which is to be transmitted. The SBI0DBR is used for both transmitting and receiving. Transmitted data should always be written after received data is read.

When the internal clock is used, the automatic wait function will be in effect until the received data is read and the next data is written.

When the external clock is used, since the shift operation is synchronized with the external clock, the received data is read and transmitted data is written before a new shift operation is executed. The maximum transfer speed when the external clock is used is determined by the delay time between the time when an interrupt request is generated and the time at which received data is read and transmitted data is written.

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

Transmitting/receiving data ends when the <SIOS> is cleared to "0" by the INTSBE0 interrupt service program or when the SBI0CR1<SIOINH> is set to "1". When the <SIOS> is cleared to "0", received data is transferred to the SBI0DBR in complete blocks. The transmit/receive mode ends when the transfer is complete. In order to confirm whether data is being transmitted/received properly by the program, set the SBI0SR to be sensed. The <SIOF> is set to "0" when transmitting/receiving is completed. When the <SIOINH> is set to "1", data transmitting/receiving stops. The <SIOF> is then cleared to "0".

Note: When the transfer mode is changed, the contents of the SBI0DBR will be lost. If the mode must be changed, conclude data transmitting/receiving by clearing the <SIOS> to "0", read the last data, then change the transfer mode.

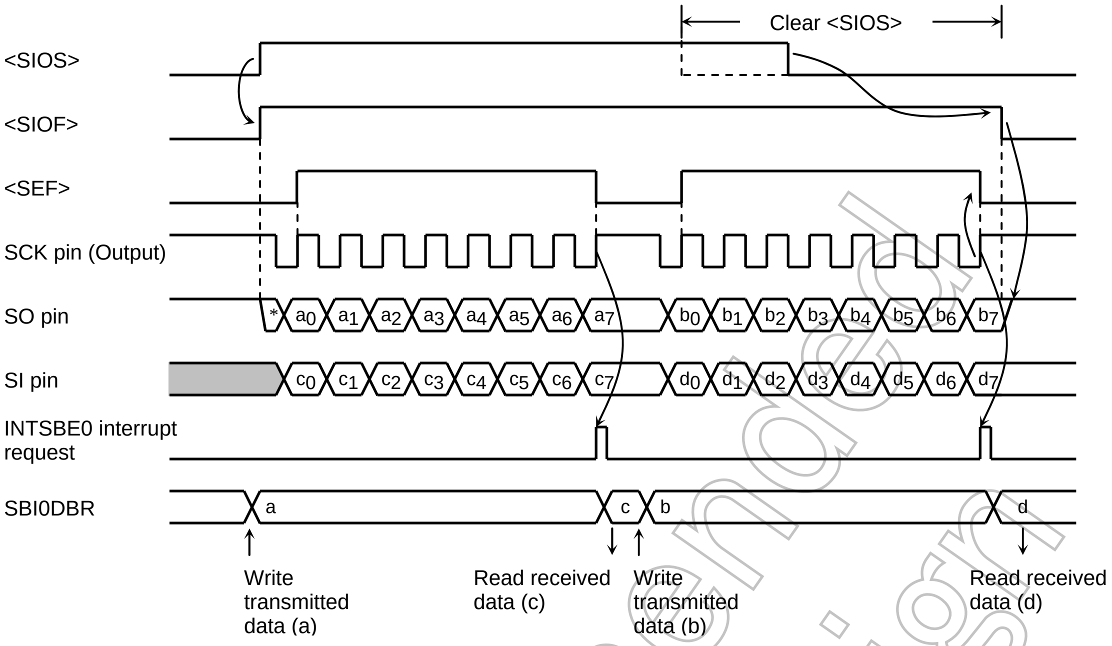


Figure 3.10.28 Transmit/Received Mode (Example: Internal clock)

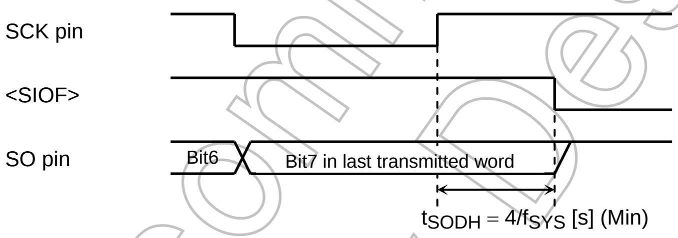


Figure 3.10.29 Transmitted Data Hold Time at End of Transmit/Receive

3.11 Analog/Digital Converter

The TMP92C820 incorporates a 10-bit successive approximation-type analog/digital converter (AD converter) with 5-channel analog input.

Figure 3.11.1 is a block diagram of the AD converter. The 5-channel analog input pins (AN0 to AN4) are shared with the input-only port (Port G) so they can be used as an input port.

Note: When IDLE2, IDLE1 or STOP mode is selected, 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.

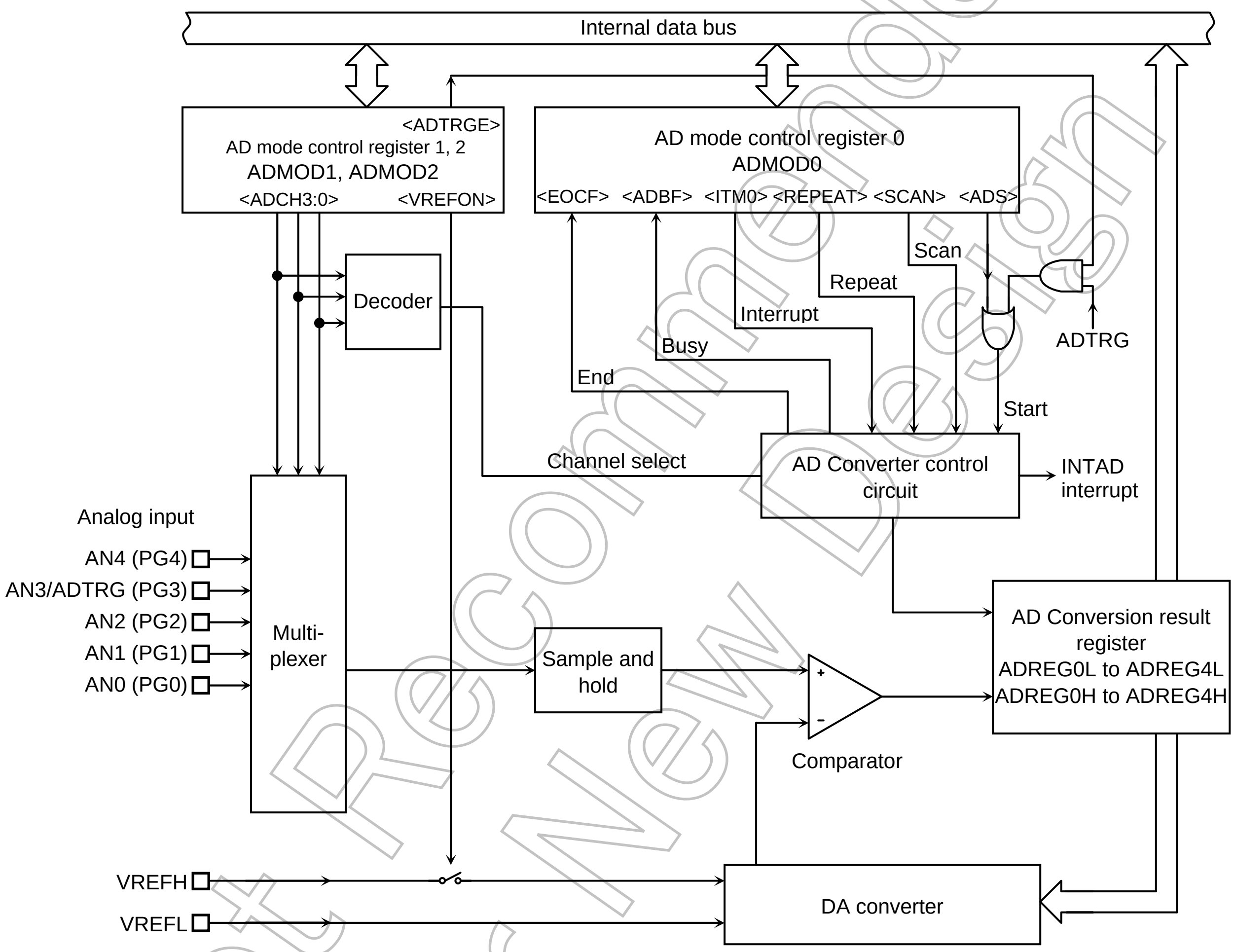


Figure 3.11.1 Block Diagram of AD Converter

3.11.1 Analog/Digital Converter Registers

The AD converter is controlled by the three AD mode control registers: ADMOD0, ADMOD1 and ADMOD2. The five AD conversion data result registers (ADREG0H/L to ADREG4H/L) store the results of AD conversion. Figure 3.11.2 shows the registers related to the AD converter.

AD Mode Control Register 0

ADMOD0 (12B8H)

	7	6	5	4	3	2	1	0
Bit symbol	EOCF	ADBF	–	–	ITM0	REPEAT	SCAN	ADS
Read/Write	R		R/W					
After reset	0	0	0	0	0	0	0	0
Function	AD conversion end flag 0: Conversion in progress 1: Conversion complete	AD conversion busy flag 0: Conversion stopped 1: Conversion in progress	Always write "0".	Always write "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 (12B9H)		7	6	5	4	3	2	1	0
	Bit symbol	VREFON	I2AD	–	–	–	ADCH2	ADCH1	ADCH0
	Read/Write	R/W	R/W	R/W					
	After reset	0	0	0	0	0	0	0	0
	Function	VREF application 0: OFF 1: ON	IDLE2 0: Stop 1: Operate	Always write "0".	Always write "0".	Always write "0".	Analog input channel selection		

Analog input channel selection

<ADCH2:0> \ <SCAN>	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 (Note)	AN4	AN0 → AN1 → AN2 → AN3 → AN4

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.

AD Mode Control Register 2

ADMOD2 (12BAH)		7	6	5	4	3	2	1	0
	Bit symbol								ADTRGE
	Read/Write								R/W
	After reset								0
	Function								AD external trigger start control 0: Disable 1: Enable

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

0	Disabled
1	Enabled

Note: As pin AN3 also function as the $\overline{\text{ADTRG}}$ input pin, do not set <ADCH2:0> = "011, 100" when using $\overline{\text{ADTRG}}$ with <ADTRGE> set to "1".

Figure 3.11.3 AD Converter Related Register

AD Conversion Result Register 0 Low

		7	6	5	4	3	2	1	0
ADREG0L (12A0H)	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 Result Register 0 High

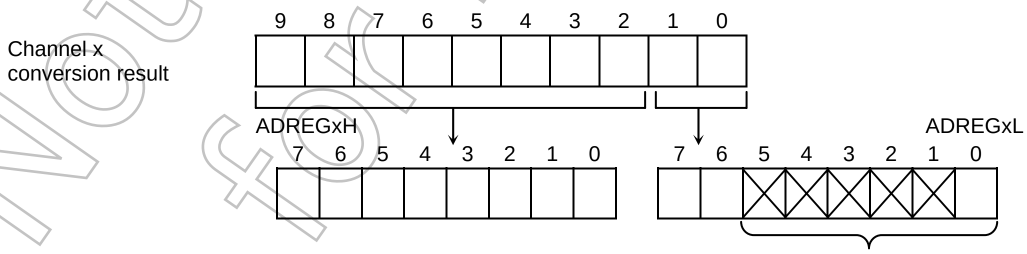
		7	6	5	4	3	2	1	0
ADREG0H (12A1H)	Bit symbol	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
	Read/Write	R							
	After reset	Undefined							
	Function	Stores upper eight bits AD conversion result.							

AD Conversion Result Register 1 Low

		7	6	5	4	3	2	1	0
ADREG1L (12A2H)	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 Result Register 1 High

		7	6	5	4	3	2	1	0
ADREG1H (12A3H)	Bit symbol	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
	Read/Write	R							
	After reset	Undefined							
	Function	Stores upper eight 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 Register 2 Low

ADREG2L (12A4H)		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 Result Register 2 High

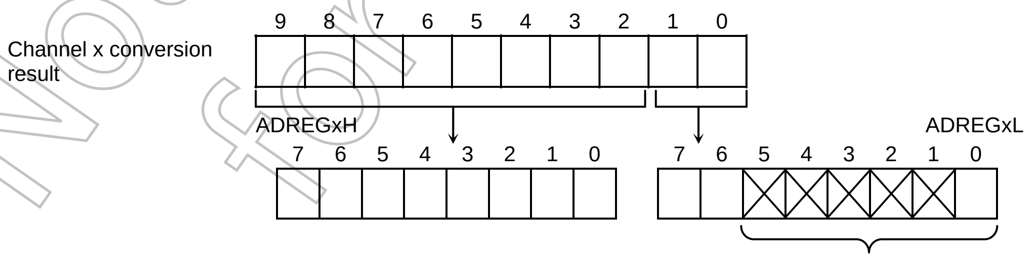
ADREG2H (12A5H)		7	6	5	4	3	2	1	0
	Bit symbol	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
	Read/Write	R							
	After reset	Undefined							
	Function	Stores upper eight bits of AD conversion result.							

AD Conversion Result Register 3 Low

ADREG3L (12A6H)		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 Register 3 High

ADREG3H (12A7H)		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 eight 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 3.11.5 AD Converter Related Registers

AD Conversion Result Register 4 Low

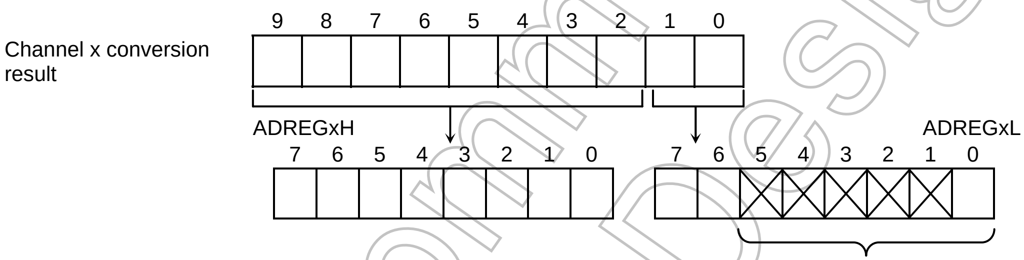
ADREG4L (12A8H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR41	ADR40						ADR4RF
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 Register 4 High

ADREG4H (12A9H)

	7	6	5	4	3	2	1	0
Bit symbol	ADR49	ADR48	ADR47	ADR46	ADR45	ADR44	ADR43	ADR42
Read/Write	R							
After reset	Undefined							
Function	Stores upper eight 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 3.11.6 AD Converter Related Registers

3.11.2 Description of Operation

(1) Analog reference voltage

A high-level analog reference voltage is applied to the VREFH pin; a low-level analog reference voltage is applied to the VREFL pin. To perform AD conversion, the reference voltage, the difference between VREFH and VREFL, is divided by 1024 using string resistance. The result of the division is then compared with the analog input voltage.

To turn off the switch between VREFH and VREFL, write a 0 to ADMOD1 <VREFON> in AD mode control register 1. To start AD conversion in the OFF state, first write a 1 to ADMOD1<VREFON>, wait 3 μs until the internal reference voltage stabilizes (This is not related to fc.), then set ADMOD0<ADS> to 1.

(2) Analog input channel selection

The analog input channel selection varies depends on the operation mode of the AD converter.

- In analog input channel fixed mode (ADMOD0<SCAN> = 0)
  Setting ADMOD1<ADCH2:0> selects one of the input pins AN0 to AN4 as the input channel.
- In analog input channel scan mode (ADMOD0<SCAN> = 1)
  Setting ADMOD1<ADCH2:0> selects one of the five scan modes.

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

On a reset, ADMOD0<SCAN> is set to 0 and ADMOD1<ADCH2:0> is initialized to 000. Thus pin AN0 is selected as the fixed input channel. Pins not used as analog input channels can be used as standard input port pins.

Table 3.11.1 Analog Input Channel Selection

<ADCH2:0>	Channel Fixed <SCAN> = "0"	Channel Scan <SCAN> = "1"
000	AN0	AN0
001	AN1	AN0 → AN1
010	AN2	AN0 → AN1 → AN2
011	AN3	AN0 → AN1 → AN2 → AN3
100	AN4	AN0 → AN1 → AN2 → AN3 → AN4

(3) Starting AD conversion

To start AD conversion, write a 1 to ADMOD0<ADS> in AD mode control register "0" or ADMOD2<ADTRGE> in AD mode control register 2, and input falling edge on $\overline{\text{ADTRG}}$ pin. When AD conversion starts, the AD conversion busy flag ADMOD0<ADBF> will be set to 1, indicating that AD conversion is in progress. During A/D conversion, a falling edge input on the $\overline{\text{ADTRG}}$ pin will be ignored.

(4) AD conversion modes and the AD conversion end interrupt

The 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 AD mode control register 0 determine the AD mode setting.

Completion of AD conversion triggers an INTAD AD conversion end interrupt request. Also, ADMOD0<EOCF> will be set to 1 to indicate that AD conversion has been completed.

a. Channel fixed single conversion mode

Setting ADMOD0<REPEAT> and ADMOD0<SCAN> to 00 selects conversion channel fixed single conversion mode.

In this mode data on one specified channel is converted once only. When the conversion has been completed, the ADMOD0<EOCF> flag is set to 1, ADMOD0<ADBF> is cleared to 0, and an INTAD interrupt request is generated.

b. Channel scan single conversion mode

Setting ADMOD0<REPEAT> and ADMOD0<SCAN> to 01 selects conversion channel scan single conversion mode.

In this mode data on the specified scan channels is converted once only. When scan conversion has been completed, ADMOD0<EOCF> is set to 1, ADMOD0<ADBF> is cleared to 0, and an INTAD interrupt request is generated.

c. Channel fixed repeat conversion mode

Setting ADMOD0<REPEAT> and ADMOD0<SCAN> to 10 selects conversion channel fixed repeat conversion mode.

In this mode data on one specified channel is converted repeatedly. When conversion has been completed, ADMOD0<EOCF> is set to 1 and ADMOD0<ADBF> is not cleared to 0 but held at 1. INTAD interrupt request generation timing is determined by the setting of ADMOD0<ITM0>.

Setting <ITM0> to 0 generates an interrupt request every time an AD conversion is completed. Setting <ITM0> to 1 generates an interrupt request on completion of every fourth conversion.

d. Channel scan repeat conversion mode

Setting ADMOD0<REPEAT> and ADMOD0<SCAN> to 11 selects conversion channel scan repeat conversion mode.

In this mode data on the specified scan channels is converted repeatedly. When each scan conversion has been completed, ADMOD0<EOCF> is set to 1 and an INTAD interrupt request is generated. ADMOD0<ADBF> is not cleared to 0 but held at 1.

To stop conversion in a repeat conversion mode (e.g., in cases c. and d.), write a 0 to ADMOD0<REPEAT>. After the current conversion has been completed, the repeat conversion mode terminates and ADMOD0<ADBF> is cleared to 0.

Switching to a halt state (IDLE2 mode with ADMOD1<I2AD> cleared to 0, IDLE1 mode or STOP mode) immediately stops operation of the AD converter even when AD conversion is still in progress. In repeat conversion modes (e.g., in cases c. and d.), when the halt is released, conversion restarts from the beginning. In single conversion modes (e.g., in cases a. and b.), conversion does not restart when the halt is released (The converter remains stopped).

Table 3.11.2 shows the relationship between the AD conversion modes and 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 4th conversion	1		
Channel Scan Repeat Conversion Mode	After completion of every scan conversion	X	1	1

X: Don't care

(5) AD conversion time

132 state (6.6 μs at f_{SYS} = 20 MHz) are required for the AD conversion of one channel.

(6) Storing and reading the results of AD conversion

The AD conversion data upper and lower registers (ADREG0H/L to ADREG4H/L) store the results of AD conversion. (ADREG0H/L to ADREG4H/L are read-only registers.)

In channel fixed repeat conversion mode, the conversion results are stored successively in registers ADREG0H/L to ADREG3H/L. In other modes the AN0, AN1, AN2, AN3, AN4 conversion results are stored in ADREG0H/L, ADREG1H/L, ADREG2H/L, ADREG3H/L and ADREG4H/L respectively.

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

Table 3.11.3 Correspondence between Analog Input Channels and AD Conversion Result Registers

Analog Input Channel (Port G)	AD Conversion Result Register	
	Conversion Modes other than at Right	Channel Fixed Repeat Conversion Mode (<ITM0>=1)
AN0	ADREG0H/L	ADREG0H/L ← (from ADREG3H/L)
AN1	ADREG1H/L	↓ ADREG1H/L
AN2	ADREG2H/L	↓ ADREG2H/L
AN3	ADREG3H/L	↓ ADREG3H/L (→ back to ADREG0H/L)
AN4	ADREG4H/L	

<ADRxRF>, bit0 of the AD conversion data lower register, is used as the AD conversion data storage flag. The storage flag indicates whether the AD conversion result register has been read or not. When a conversion result is stored in the AD conversion result register, the flag is set to 1. When either of the AD conversion result registers (ADREGxH or ADREGxL) is read, the flag is cleared to 0.

Reading the AD conversion result also clears the AD conversion end flag ADMOD0<EOCF> to 0.

Setting example:

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

Main routine:

		7	6	5	4	3	2	1	0	
INTE0AD	←	1	1	0	0	–	–	–	–	Enable INTAD and set it to interrupt level 4.
ADMOD1	←	1	1	0	0	0	0	1	1	Set pin AN3 to be the analog input channel.
ADMOD0	←	–	–	0	0	0	0	0	1	Start conversion in channel fixed single conversion mode.

Interrupt routine processing example:

WA	← ADREG3	Read value of ADREG3L and ADREG3H into 16-bit general-purpose register WA.
WA	> > 6	Shift contents read into WA 6 times to right and zero-fill upper bits.
(0800H)	← WA	Write contents of WA to memory address 0800H.

2. This example repeatedly converts the analog input voltages on the three pins AN0, AN1 and AN2, using channel scan repeat conversion mode.

INTE0AD	←	1	0	0	0	–	–	–	–	Disable INTAD.
ADMOD1	←	1	1	0	0	0	0	1	0	Set pins AN0 to AN2 to be the analog input channels.
ADMOD0	←	–	–	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 TMP92C820 contains a watchdog timer of runaway detecting.

The watchdog timer (WDT) is used to return the CPU to the normal state when it detects that the CPU has started to malfunction (Runaway) due to causes such as noise. When the watchdog timer detects a malfunction, it generates a non-maskable interrupt INTWD to notify the CPU of the malfunction.

Connecting the watchdog timer output to the reset pin internally forces a reset. (The level of external $\overline{\text{RESET}}$ pin is not changed.)

3.12.1 Configuration

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

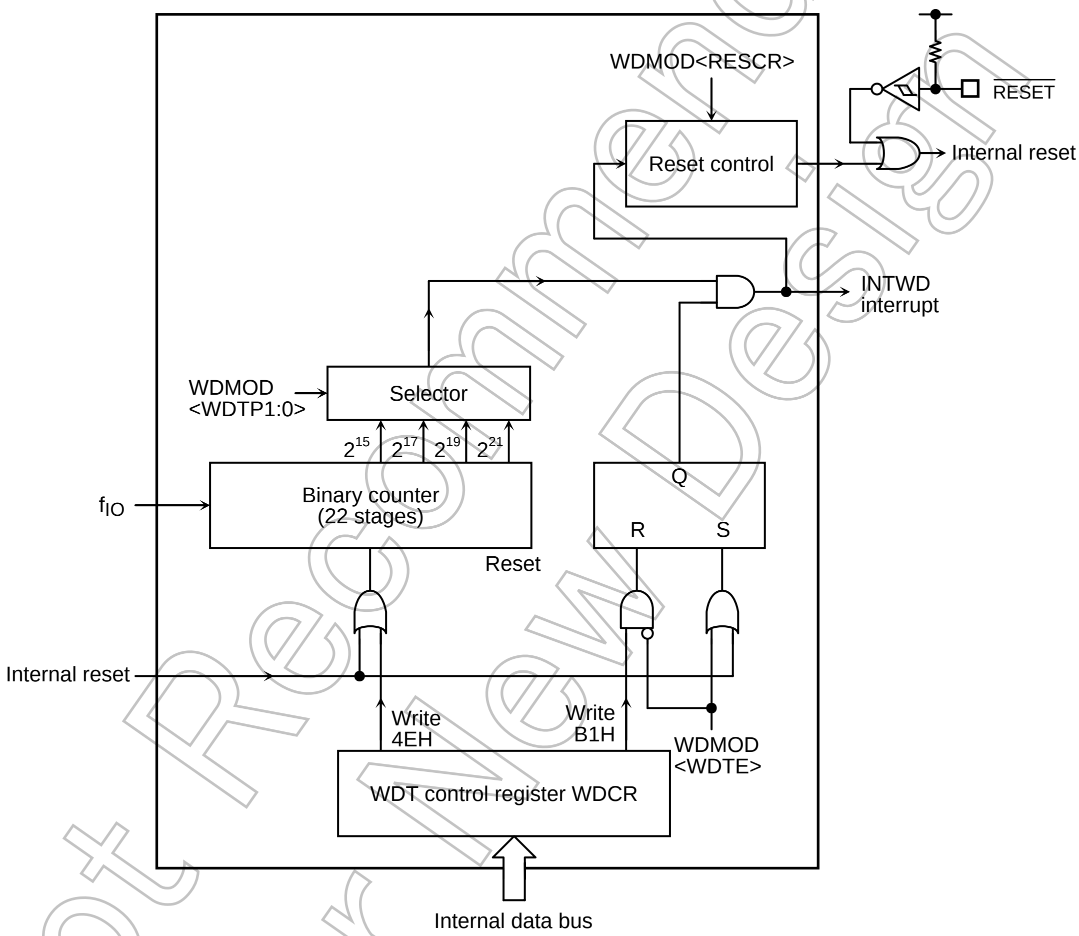


Figure 3.12.1 Block Diagram of Watchdog Timer

Note: Care must be exercised in the overall design of the apparatus since the watchdog timer may fail to function correctly due to external noise, etc.

3.12.2 Operation

The watchdog timer generates an INTWD interrupt when the detection time set in the WDMOD<WDTP1:0> has elapsed. The watchdog timer must be cleared to zero in software before an INTWD interrupt will be generated. If the CPU malfunctions (e.g., if runaway occurs) due to causes such as noise, but does not execute the instruction used to clear the binary counter, the binary counter will overflow and an INTWD interrupt will be generated. The CPU will detect malfunction (runaway) due to the INTWD interrupt, and in this case it is possible to return the CPU to normal operation by means of an anti-malfunction program.

The watchdog timer begins operating immediately on release of the watchdog timer reset.

The watchdog timer is reset and halted in IDLE1 or $\overline{\text{STOP}}$ mode. The watchdog timer counter continues counting during bus release (when $\overline{\text{BUSAK}}$ goes low).

When the device is in IDLE2 mode, the operation of the WDT depends on the WDMOD<I2WDT> setting. Ensure that WDMOD<I2WDT> is set before the device enters IDLE2 mode.

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

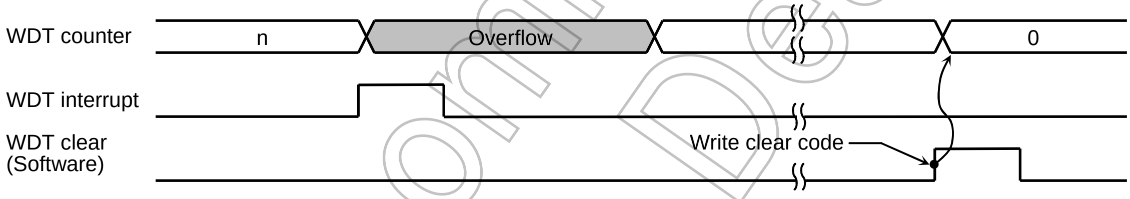


Figure 3.12.2 Normal Mode

The runaway detection result can also be connected to the reset pin internally. In this case, the reset time will be between 44 and 58 system clocks (35.2 to 46.4μs at f_{OSCH} = 40 MHz) as shown in Figure 3.12.3. After a reset, the f_{IO} clock (1 cycle = 1 state) is $f_{FPH}/4$, where f_{FPH} is generated by dividing the high-speed oscillator clock (f_{OSCH}) by sixteen through the clock gear function

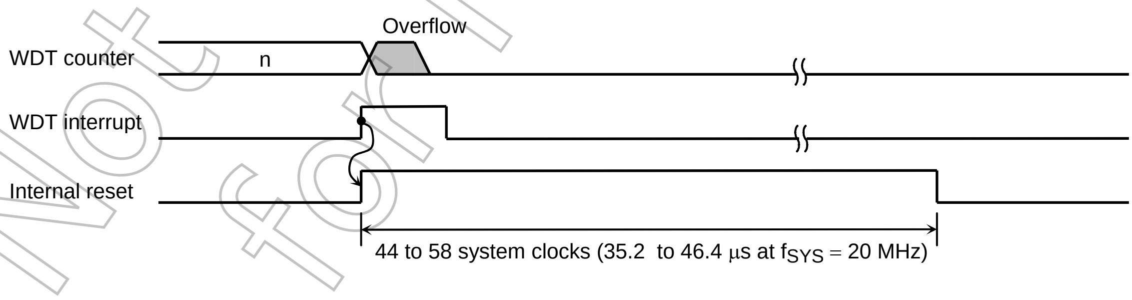


Figure 3.12.3 Reset Mode

3.12.3 Control Registers

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

(1) Watchdog timer mode register (WDMOD)

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

This 2-bit register is used for setting the watchdog timer interrupt time used when detecting runaway.

On a reset this register is initialized to WDMOD<WDTP1:0> = 00.

The detection times for WDT is $2^{15}/f_{IO}$ [s]. (The number of system clocks is approximately 65,536.)

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

At reset, the WDMOD<WDTE> is initialized to 1, enabling the watchdog timer.

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

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

c. Watchdog timer out reset connection <RESCR>

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

(2) Watchdog timer control register (WDCR)

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

- Disable control

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

WDCR	←	0 1 0 0 1 1 1 0	Write the clear code (4EH).
WDMOD	←	0 – – X 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 to resume, write the clear code (4EH) to the WDCR register.

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

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

Note2: If the watchdog timer setting is changed, change setting after setting to disable condition once.

WDMOD (1300H)

	7	6	5	4	3	2	1	0
Bit symbol	WDTE	WDTP1	WDTP0		–	I2WDT	RESCR	–
Read/Write	R/W				R/W			
After reset	1	0	0		0	0	0	0
Function	WDT control 1: Enable	Select detecting time 00: $2^{15}/f_{IO}$ 01: $2^{17}/f_{IO}$ 10: $2^{19}/f_{IO}$ 11: $2^{21}/f_{IO}$			Always write "0".	IDLE2 0: Stop 1: Operate	1: Internally connects WDT 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

00	$2^{15}/f_{IO}$ (Approximately 3.28 ms at f_{OSCH} = 40 MHz)
01	$2^{17}/f_{IO}$ (Approximately 13.1 ms at f_{OSCH} = 40 MHz)
10	$2^{19}/f_{IO}$ (Approximately 52.4 ms at f_{OSCH} = 40 MHz)
11	$2^{21}/f_{IO}$ (Approximately 210 ms at f_{OSCH} = 40 MHz)

Watchdog timer enable/disable control

0	Disabled
1	Enabled

Figure 3.12.4 Watchdog Timer Mode Register

WDCR (1301H)

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							

WDT disable/clear control

B1H	Disable code
4EH	Clear code
Others	–

Figure 3.12.5 Watchdog Timer Control Register

3.13 Real Time Clock (RTC)

3.13.1 Function Description for RTC

(1) Clock function (Hour, minute, second)

(2) Calendar function (Month and day, day of the week, and leap year)

(3) 24 or 12-hour (AM/PM) clock function

(4) ±30 second adjustment function (by software)

(5) $\overline{\text{ALARM}}$ function (Alarm output)

(6) Alarm interrupt generate

(7) Divided power supply

3.13.2 Block Diagram

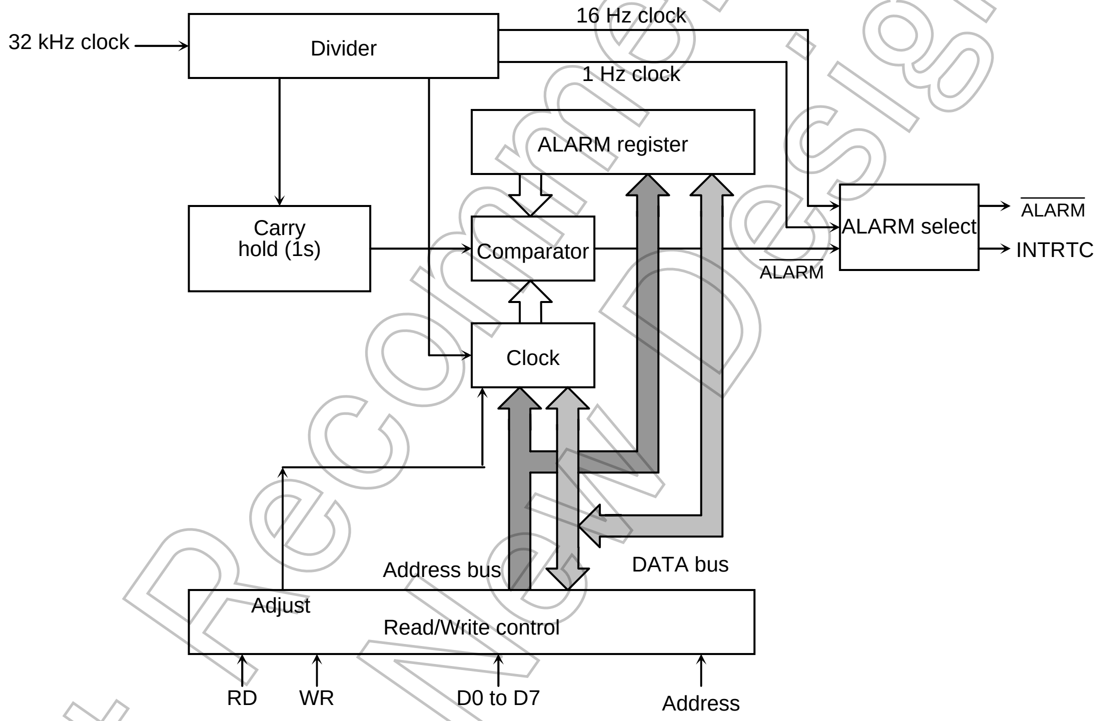


Figure 3.13.1 RTC Block Diagram

Note 1: Western calendar year column:

This product uses only the final two digits of the year. Therefore, the year following 99 is 00 years. In use, please take into account the first two digits when handling years in the western calendar.

Note 2: Leap year:

A leap year is divisible by 4, but the exception is any leap year which is divisible by 100; this is not considered a leap year. However, any year which is divisible by 400, is a leap year. This product does not take into account the above exceptions . Since this product accounts only for leap years divisible by 4, please adjust the system for any problems.

3.13.3 Detailed Explanation of Control Register

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

(1) Second column register (for PAGE0 only)

		7	6	5	4	3	2	1	0
SECR (1320H)	Bit symbol		SE6	SE5	SE4	SE3	SE2	SE1	SE0
	Read/Write		R/W						
	After reset		Undefined						
	Function	"0" is read.	40 sec. column	20 sec. column	10 sec. column	8 sec. column	4 sec. column	2 sec. column	1 sec. column

0	0	0	0	0	0	0	0 sec
0	0	0	0	0	0	1	1 sec
0	0	0	0	0	1	0	2 sec
0	0	0	0	0	1	1	3 sec
0	0	0	0	1	0	0	4 sec
0	0	0	0	1	0	1	5 sec
0	0	0	0	1	1	0	6 sec
0	0	0	0	1	1	1	7 sec
0	0	0	1	0	0	0	8 sec
0	0	0	1	0	0	1	9 sec
0	0	1	0	0	0	0	10 sec
				:			
0	0	1	1	0	0	1	19 sec
0	1	0	0	0	0	0	20 sec
				:			
0	1	0	1	0	0	1	29 sec
0	1	1	0	0	0	0	30 sec
				:			
0	1	1	1	0	0	1	39 sec
1	0	0	0	0	0	0	40 sec
				:			
1	0	0	1	0	0	1	49 sec
1	0	1	0	0	0	0	50 sec
				:			
1	0	1	1	0	0	1	59 sec

Note: Do not set data other than as shown above.

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

MINR (1321H)		7	6	5	4	3	2	1	0
Bit symbol			MI6	MI5	MI4	MI3	MI2	MI1	MI0
Read/Write			R/W						
After reset			Undefined						
Function		"0" is read.	40 min column	20 min column	10 min column	8 min column	4 min column	2 min column	1 min column

0	0	0	0	0	0	0	0 min
0	0	0	0	0	0	1	1 min
0	0	0	0	0	1	0	2 min
0	0	0	0	0	1	1	3 min
0	0	0	0	1	0	0	4 min
0	0	0	0	1	0	1	5 min
0	0	0	0	1	1	0	6 min
0	0	0	0	1	1	1	7 min
0	0	0	1	0	0	0	8 min
0	0	0	1	0	0	1	9 min
0	0	1	0	0	0	0	10 min
			:				
0	0	1	1	0	0	1	19 min
0	1	0	0	0	0	0	20 min
			:				
0	1	0	1	0	0	1	29 min
0	1	1	0	0	0	0	30 min
			:				
0	1	1	1	0	0	1	39 min
1	0	0	0	0	0	0	40 min
			:				
1	0	0	1	0	0	1	49 min
1	0	1	0	0	0	0	50 min
			:				
1	0	1	1	0	0	1	59 min

Note: Do not set data other than as shown above.

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

1. In 24-hour clock mode (MONTHR<MO0> = “1”)

		7	6	5	4	3	2	1	0
HOURR (1322H)	Bit symbol			HO5	HO4	HO3	HO2	HO1	HO0
	Read/Write			R/W					
	After reset			Undefined					
	Function	“0” is read.		20 hours column	10 hours column	8 hours column	4 hours column	2 hours 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

Note: Do not set data other than as shown above.

2. In 12-hour clock mode (MONTHR<MO0> =“0”)

		7	6	5	4	3	2	1	0
HOURR (1322H)	Bit symbol			HO5	HO4	HO3	HO2	HO1	HO0
	Read/Write			R/W					
	After reset			Undefined					
	Function	“0” is read.		PM/AM	10 hours column	8 hours column	4 hours column	2 hours 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

Note: Do not set data other than as shown above.

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

DAYR (1323H)		7	6	5	4	3	2	1	0
	Bit symbol						WE2	WE1	WE0
	Read/Write						R/W		
	After reset						Undefined		
	Function	"0" is read.					W2	W1	W0

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

Note: Do not set data other than as shown above.

(5) Day column register (PAGE0/1)

DATER (1324H)		7	6	5	4	3	2	1	0
	Bit symbol			DA5	DA4	DA3	DA2	DA1	DA0
	Read/Write			R/W					
	After reset			Undefined					
	Function	"0" is read.		Day 20	Day 10	Day 8	Day 4	Day 2	Day 1

0	0	0	0	0	0	0
0	0	0	0	0	1	1st day
0	0	0	0	1	0	2nd day
0	0	0	0	1	1	3rd day
0	0	0	1	0	0	4th day
:						
0	0	1	0	0	1	9th day
0	1	0	0	0	0	10th day
0	1	0	0	0	1	11th day
:						
0	1	1	0	0	1	19th day
1	0	0	0	0	0	20th day
:						
1	0	1	0	0	1	29th day
1	1	0	0	0	0	30th day
1	1	0	0	0	1	31st day

Note1: Do not set data other than as shown above.

Note2: Do not set for non-existent days (e.g.: 30th Feb).

(6) Month column register (for PAGE0 only)

MONTHR (1325H)		7	6	5	4	3	2	1	0
	Bit symbol				MO4	MO4	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

Note: Do not set data other than as shown above.

(7) Select 24-hour clock or 12-hour clock (for PAGE1 only)

MONTHR (1325H)		7	6	5	4	3	2	1	0
	Bit symbol								MO0
	Read/Write								R/W
	After reset								Undefined
	Function	"0" is read.							1: 24-hour 0: 12-hour

(8) Year column register (for PAGE0 only)

YEARR (1326H)		7	6	5	4	3	2	1	0
	Bit symbol	YE7	YE6	YE5	YE4	YE3	YE2	YE1	YE0
	Read/Write	R/W							
	After reset	Undefined							
	Function	80 years	40 years	20 years	10 years	8 years	4 years	2 years	1 year

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

Note: Do not set data other than as shown above.

(9) Leap year register (for PAGE1 only)

YEARR (1326H)		7	6	5	4	3	2	1	0
	Bit symbol							LEAP1	LEAP0
	Read/Write							R/W	
	After reset							Undefined	
	Function	"0" is read.						00: Leap year 01: One year after leap year 10: Two years after leap year 11: Three years after leap year	

0	0	Current year is a leap year
0	1	Current year is the year following a leap year
1	0	Current year is two years after a leap year
1	1	Current year is three years after a leap year

(10) Setting PAGE register (for PAGE0/1)

PAGER (1327H)

Read-modify-write instruction is prohibited.

	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	INTRTC 0: Disable 1: Enable	"0" is read.		0: Don't care 1: Adjust	Clock 0: Disable 1: Enable	ALARM 0: Disable 1: Enable	"0" is read.	PAGE selection

Note: Please keep the setting order below of <ENATMR>, <ENAAML> and <INTENA>. Set different times for Clock/Alarm setting and interrupt setting.

(Example) Clock setting/Alarm setting

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

ld    (pager), 8ch    :    Interrupt enable
```

PAGE	0	Select Page0
	1	Select Page1

ADJUST	0	Don't care
	1	Adjust sec. counter. When this bit is set to "1" the sec. counter becomes "0" when the value of the sec. counter is 0 – 29. When the value of the sec. counter is 30-59, the min. counter is carried and sec. counter becomes "0". Output Adjust signal during 1 cycle of f_{SYS}. After being adjusted once, Adjust is released automatically. (PAGE0 only)

(11) Setting reset register (for PAGE0/1)

RESTR (1328H)

Read-modify write-instruction is prohibited.

	7	6	5	4	3	2	1	0
Bit symbol	DIS1Hz	DIS16Hz	RSTTMR	RSTALM	–	–	–	–
Read/Write	W							
After reset	Undefined							
Function	1Hz 0: Enable 1: Disable	16Hz 0: Enable 1: Disable	1:Clock reset	1: Alarm reset	Always write "0"			

RSTALM	0	Unused
	1	Reset alarm register

RSTTMR	0	Unused
	1	Reset counter

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

3.13.4 Operational description

(1) Reading clock data

1. Using 1Hz interrupt

1Hz interrupt and the count up of internal data synchronize. Therefore, data can read correctly if reading data after 1Hz interrupt occurred.

2. Using two times reading

There is a possibility of incorrect clock data reading when the internal counter carries over. To ensure correct data reading, please read twice, as follows:

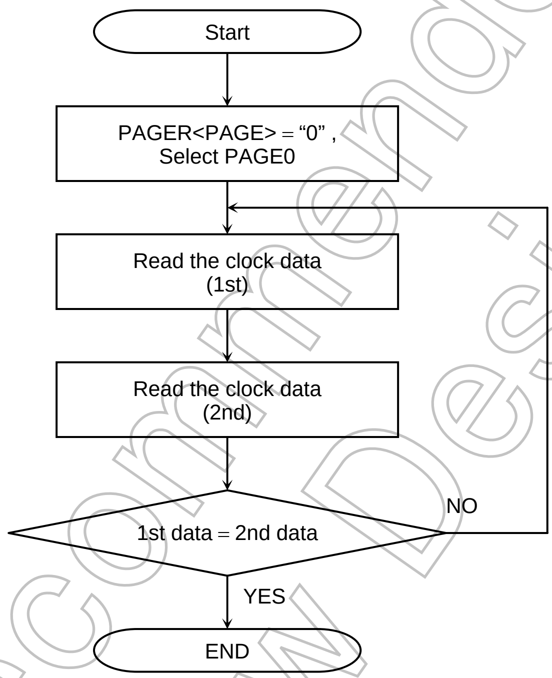


Figure 3.13.2 Flowchart of clock data read

(2) Writing clock data

When a carry over occurs during a write operation, the data cannot be written correctly. Please use the following method to ensure data is written correctly.

1. Using 1Hz interrupt

1Hz interrupt and the count up of internal data synchronize. Therefore, data can write correctly if writing data after 1Hz interrupt occurred.

2. Resetting a counter

There are 15-stage counter inside the RTC, which generate a 1Hz clock from 32,768 KHz. The data is written after reset this counter.

However, if clearing the counter, it is counted up only first writing at half of the setting time, first writing only. Therefore, if setting the clock counter correctly, after clearing the counter, set the 1Hz-interrupt to enable. And set the time after the first interrupt (occurs at 0.5Hz) is occurred.

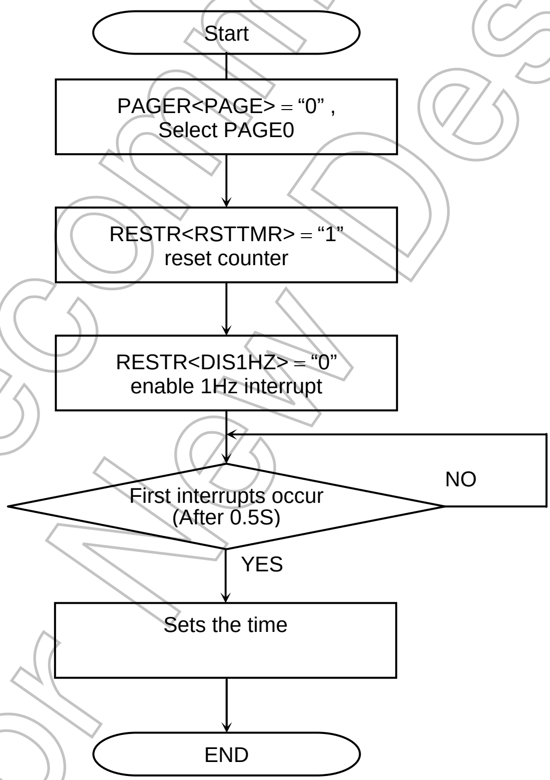


Figure 3.13.3 Flowchart of data write

2. Disabling the clock

A clock carry over is prohibited when "0" is written to PAGER<ENATMR> in order to prevent malfunction caused by the Carry hold circuit. While the clock is prohibited, the Carry hold circuit holds a one sec. carry signal from a divider. When the clock becomes enabled, the carry signal is output to the clock, the time is revised and operation continues. However, the clock is delayed when clock-disabled state continues for one second or more. Note that at this time system power is down while the clock is disabled. . In this case the clock is stopped and clock is delayed.

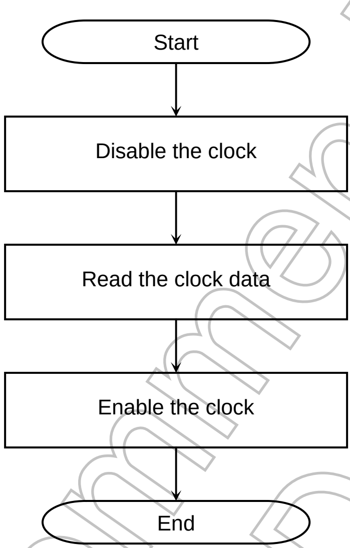


Figure 3.13.4 Flowchart of Clock disable

3.13.5 Explanation of the interrupt signal and alarm signal

The alarm function used by setting the PAGE1 register and outputting either of the following three signals from $\overline{\text{ALARM}}$ pin by writing “1” to PAGER<PAGE>. INTRTC outputs a 1-shot pulse when the falling edge is detected. RTC is not initialized by RESET. Therefore, when the clock or alarm function is used, clear interrupt request flag in INTC (interrupt controller).

(1) When the alarm register and the clock correspond, output “0”.

(2) 1Hz Output clock .

(3) 16Hz Output clock.

(1) When the alarm register and the clock correspond, output “0”

When PAGER<ENAALM>= “1”, and the value of PAGE0 clock corresponds with PAGE1 alarm register, output “0” to $\overline{\text{ALARM}}$ pin and generate INTRTC.

The methods for using the alarm are as follows:

Initialization of alarm is done by writing “1” to RESTR<RSTALM>. All alarm settings become Don’t care. In this case, the alarm always corresponds with value of the clock, and if PAGER<ENAALM> is “1”, INTRTC interrupt request is generated.

Setting alarm min., alarm hour, alarm date and alarm day is done by writing data to the relevant PAGE1 register.

When all setting contents correspond, RTC generates an INTRTC interrupt if PAGER<INTENA><ENAALM> is “1”. However, contents which have not been set up (don't care state) are always considered to correspond.

Contents which have already been set up, cannot be returned independently to the Don't care state. In this case, the alarm must be initialized and alarm register reset.

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

```
   LD     (PAGER), 09H        ;   Alarm disable, setting PAGE1
   LD     (RESTR), D0H        ;   Alarm initialize
   LD     (DAYR), 01H         ;   W0
   LD     (DATAR),01H         ;   1 day
   LD     (HOURR), 12H        ;   Setting 12 o’clock
   LD     (MINR), 00H         ;   Setting 00 min
                              ;   Set up time 31 μs (Note)
   LD     (PAGER), 0CH        ;   Alarm enable
 ( LD     (PAGER), 8CH        ;   Interrupt enable )
```

When the CPU is operating at high frequency oscillation, it may take a maximum of one clock at 32 kHz (about 30us) for the time register setting to become valid. In the above example, it is necessary to set 31us of set up time between setting the time register and enabling the alarm register.

Note: This set up time is unnecessary when you use only internal interruption.

(2) With 1Hz output clock

RTC outputs a clock of 1Hz to $\overline{\text{ALARM}}$ pin by setting up PAGER<ENAALM>= “0”, RESTR<DIS1HZ>= “0”, <DIS16HZ>= “1”. RTC also generates an INTRC interrupt on the falling edge of the clock.

(3) With 16Hz output clock

RTC outputs a clock of 16Hz to $\overline{\text{ALARM}}$ pin by setting up PAGER<ENAALM>= “0”, RESTR<DIS1HZ>= “1”, <DIS16HZ>= “0”. RTC also generates INTRC an interrupt on the falling edge of the clock.

3.14 LCD Controller (LCDC)

The TMP92C820 incorporates two types liquid crystal display driving circuit for controlling LCD driver LSI. One circuit handles a RAM built-in type LCD driver that can store display data in the LCD driver itself, and the other circuit handles a shift-register type LCD driver that must serially transfer the display data to LCD driver for each display picture.

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

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

After finish data transmission, LCDC cancels the bus release request and CPU will re-start.

As the DISPLAY RAM, SDRAM burst mode can be used in TMP92C820.

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

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

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

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

		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
Transfer data bus width		32 bits or 16 bits	8 bits fixed
Internal RAM		Not allow to use	Allow to use
Transfer rate (at f_{SYS} = 20 [MHz])		50 ns/1 word at SDRAM/BURST 100 ns/1 word at SRAM	–
External pins	LCD data bus: LD7 to LD0 pin	Data bus: Connect to data input pin of column driver.	Not used
	Data bus: D7 to D0 pin	Not used	Data bus: Connect to data input pin of LCD driver.
	Bus state: R/W pin	Not used	Bus state: Connect with /WR pin of column/row driver.
	Address bus: A0 pin	Not used	Address 0: Connect with D/I pin of column driver. When A0 = 1 data bus value means display data, when A0 = 0 data bus means instruction data.
	Shift clock pulse: D1BSCP pin	Shift clock pulses: Connect with SCP pin of column driver. LCD driver latches data bus value by falling edge of this pin.	Chip enable for column driver 1: Connect with $\overline{CE}$ pin of column driver 1.
	Latch pulse: D2BLP pin	Latch pulses output: Connect with LP/EIO1 pin of column/row driver. Display data is latched in 1st shift register in LCD driver by rising edge of this pin. And shift to next shift register by LP and SCP = "H".	Chip enable for column driver 2: Connect with $\overline{CE}$ pin of column driver 2.
	Frame: D3BFR pin	LCD frame output: Connect with FR pin of column/row driver.	Chip enable for column driver 3: Connect with $\overline{CE}$ pin of column driver 3.
	Cascade pulse: DLEBCD pin	Cascade pulses output: Connect with DIO1 pin of row driver. These pin outputs 1 shot pulse by every D3BFR pin changes.	Chip enable for row driver: Connect with $\overline{LE}$ pin of row driver.
	Display OFF: $\overline{DOFF}$ pin	Display OFF output: Connect with $\overline{DSPOF}$ terminal of column/row driver. "L" means display off and "H" means display ON.	

3.14.2 Block Diagram

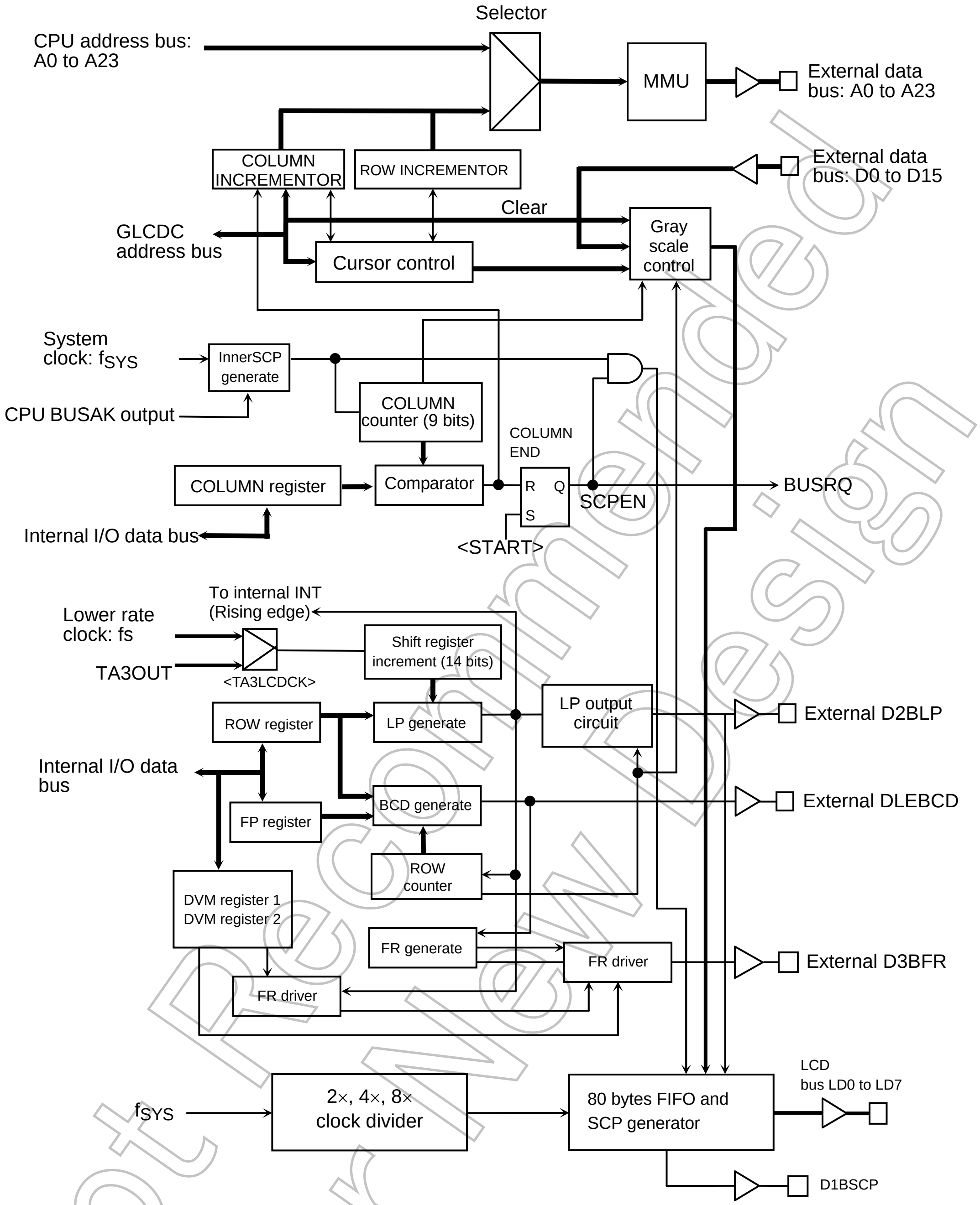


Figure 3.14.1 LCDC Block Diagram

3.14.3 SFRs

LCDMODE Register

LCDMODE (0200H)

	7	6	5	4	3	2	1	0
Bit symbol	BAE	AAE	SCPW1	SCPW0	TA3LCDCK	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	B-area 0: Disable 1: Enable	A-area 0: Disable 1: Enable	00: Base SCP 01: 2 clocks 10: 4 clocks 11: 8 clocks		Select low-frequency clock 0: 32 kHz 1: TA3OUT	Byte-number/ Common 0: 512 bytes 1: 1024 bytes * (Note 4)	Display RAM 0: SRAM 1: SDRAM	Mode selection 0: RAM 1: SR

Note 1: <BULK> is effective when <RAMTYPE> is set to "1". <BULK> shows how to generate address for next common.

Note 2: The SDRAM accessing way of LCDC is only "Burst 1CLK access".

Note 3: Base SCPW<1:0> is introduced in section. 3.14.4.6.

Note 4: Refer to Table 3.14.1.

Table 3.14.2 SDRAM BULK and Column Address

LCDMODE<BULK>	0	1
SDRAMC SDACR<SMUXW>	Type A	Type B
Bulk of 1 page	512 bytes	1024 bytes

Divide FRM Register

LCDDVM (0201H)

	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	0	0	0	0	0	0	0
Function	Setting DVM bit7 to 0							

LCD Size Setting Register

LCDSIZE (0202H)

	7	6	5	4	3	2	1	0
Bit symbol	COM3	COM2	COM1	COM0	SEG3	SEG2	SEG1	SEG0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Function	Setting the LCD common number for SR mode 0000: 128 0101: 400 0001: 160 0110: 480 0010: 200 0011: 240 0100: 320 Other: Reserved				Setting the LCD segment number for SR mode 0000: 128 0101: 480 0001: 160 0110: 560 0010: 240 0111: 640 0011: 320 0100: 400 Other: Reserved			

LCD Control Register

LCDCTL (0203H)		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	$\overline{DOFF}$ port 0: OFF 1: ON	LD bus output control 0: Normal 1: All display data = 0	Divided FR mode 0: Disable 1: Enable	Always write "0".	Setting bit9 for f_{FP} [9:0]	Type selection of LCD driver with built-in RAM 0: Sequential access type 1: Random access type	Setting bit8 for f_{FP} [9:0]	Start control in SR mode 0: Stop 1: Start

→ LCDC start/stop bit

0	LCDC stop
1	LCDC start

→ RAM internal LCD driver TYPE selection

0	Sequential access type
1	Random access type

→ Flame frequency division mode

0	Disable
1	Enable

→ LD bus output control

0	Normal
1	All "0"

Note: This bit is forced setting it to "0" (light OFF) by writing "1" that data transfer to LCDD. Usually, writing "0".

→ Pin of LCD driver: $\overline{DOFF}$

0	Driver OFF
1	Driver ON

Note: This bit decide state of $\overline{DOFF}$ pin.
Case of "0": output "0"
Case of "1": output "1"

Figure 3.14.2 LCDC Control Register 1

LCD f_{FP} Register

LCDFFP (0204H)		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	0	0	0	0	0	0	0
	Function	Setting bit7 to 0 for f_{FP}							

LCD Gray Level Setting Register

LCDGL (0205H)		7	6	5	4	3	2	1	0
	Bit symbol							GRAY1	GRAY0
	Read/Write							R/W	
	After reset							0	0
	Function							00: Monochrome 01: 4 levels 10: 8 levels 11: 16 levels	

Figure 3.14.3 LCDC Control Register 2

Table 3.14.3 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 (0211H)	LSARAM (0210H)	–	LEARAH (0213H)	LEARAM (0212H)	–
After reset	40H	00H	–	40H	00H	–
B-area	LSARBH (0215H)	LSARBM (0214H)	–	LEARBH (0217H)	LEARBM (0216H)	–
After reset	40H	00H	–	40H	00H	–
C-area	LSARCH (021AH)	LSARCM (0219H)	LSARCL (0218H)	–	–	–
After reset	40H	00H	00H	–	–	–

Note: All registers are available for R (Read)/W (Write).

LCD Cursor Setting Register

LCDCM (0206H)

	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 (XT1: 32 kHz) 00: Don't blink 01: 2 Hz 10: 1 Hz 11: 0.5 Hz	

Note 1: Cursor brink interval make using low clock (fs). This function doesn't depend on LCDMODE. Therefore if you use blink function, you set low clock condition.

Note 2: Also case of using timer out "TA3OUT" to LCDCK, cursor brink internal depend on fs.

LCD Cursor Width Setting Register

LCDCW (0207H)

	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 (X size) 00000: 1 dot (Min) 11111: 32 dots (Max)				

LCD Cursor Height Setting Register

LCDCH (0208H)

	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 (Y size) 00000: 1 dot (Min) 11111: 32 dots (Max)				

Figure 3.14.4 LCDC Control Register 3

Hot Point of LCD Cursor X Bit Setting Register

LCDCP (0209H)

	7	6	5	4	3	2	1	0
Bit symbol					APB3	APB2	APB1	APB0
Read/Write					R/W			
After reset					0	0	0	0
Function					Setting bit3 to bit0 for cursor hot point (for 1-dot correction)			

In case of monochrome 0000: Position pixel 0

(Except BURST mode) 1111: Position pixel 15

Figure 3.14.5 LCDC Control Register 4

LCD Cursor Absolute Position Setting Register

LCDCPL (020AH)

	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 absolute position							

LCD Cursor Absolute Position Setting Register

LCDCPM (020BH)

	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 absolute position							

LCD Cursor Absolute Position Setting Register

LCDCPH (020CH)

	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 absolute position							

Figure 3.14.6 LCDC Control Register 5

LCDC1L, LCDC1H, LCDC2L, LCDC2H, LCDC3L, LCDC3H, LCDR1L and 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							

Figure 3.14.7 LCDC Control Register 6

These registers do not exist on TMP92C820. These are image for instruction registers and display registers of external RAM built-in sequential access type LCD driver.

Address as Figure 3.14.4 is assigned to these registers, and the following chip enable pin becomes active when accesses corresponding address.

And, the area of these address is external area, so $\overline{RD}$, $\overline{WR}$ terminal becomes active by external access.

Figure 3.14.5 shows the address map in the case of controlling RAM built-in random access type LCD driver. The explanation part of MMU circuit also explains this. This setup is performed by LCDCTL<MMULCD>.

Table 3.14.4 Memory Mapping for Built-in RAM Sequential Access Type

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

Table 3.14.5 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	DLEBCD

Note 1: We call built-in RAM sequential access type LCD driver that use register to access to display RAM without address. (Example: 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. (Example: T6C23,T6K01 etc., mar/2000)

3.14.4 Shift Register Type LCD Driver Control Mode (SR mode)

3.14.4.1 Operation

Set the mode of operation, start address of source data save memory, grayscale level and LCD size to control registers before setting start register.

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

Note: SR mode LCDC, during data reading (during DMA operation), CPU is stopped by internal BUSREQ signal. When using SR mode LCDC, programmer need to care the CPU stop time. For detail, see the Table 3.14.4.

3.14.4.2 Grayscale Mode Indication

Monochrome, 4, 8 and 16 grayscale mode can be selected by setting LCDGL<GRAY1:0>.

And when SDRAM mode, you can select the size of SDRAM by setting (LCDMODE)<BULK>.

TMP92C820 realize grayscale display by thinning out the frame. Grayscale control palette is defined by 16 bit register (LGnL/H) shown in Table 3.14.6. Palette is selected according to the grayscale level (Monochrome, 4, 8, 16 gray) for use. (cf. Table 3.14.7). ON/OFF for data of each level (e.g., each density) can modify by 16-bit register (LGnL/H). However each register of palette has a initial value, it is possible to adjust finely which matches to LCD driver you use and the characteristic of LCD panel.

Table 3.14.6 Grayscale Control Palette Default Setting

Level Code	Density	Data Setting Register (Address/After reset)	Bit 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
F	16/16	LGFH/L (023FH to E/FFFFH)	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
E	14/16	LGEH/L (023DH to C/FDFDH)	●	○	●	●	●	●	●	●	●	○	●	●	●	●	●	●
D	1316	LGDH/L (023BH to A/FDDDH)	●	○	●	●	●	○	●	●	●	○	●	●	●	●	●	●
C	12/16	LGCH/L (0239H to 8/DDDDH)	●	○	●	●	●	○	●	●	●	○	●	●	●	○	●	●
B	11/16	LGBH/L (0237H to 6/DDD5H)	●	○	●	○	●	○	●	●	●	○	●	●	●	○	●	●
A	10/16	LGAH/L (0235H to 4/D5D5H)	●	○	●	○	●	○	●	●	●	○	●	○	●	○	●	●
9	9/16	LG9H/L (0233H to 2/D555H)	●	○	●	○	●	○	●	○	●	○	●	○	●	○	●	●
8	8/16	LG8H/L (0231H to 0/AAAAH)	○	●	○	●	○	●	○	●	○	●	○	●	○	●	○	●
7	7/16	LG7H/L (022FH to E/8AAAH)	○	●	○	●	○	●	○	●	○	●	○	●	○	○	○	●
6	6/16	LG6H/L (022DH to C/8A8AH)	○	●	○	●	○	○	○	●	○	●	○	●	○	○	○	●
5	5/16	LG5H/L (022BH to A/888AH)	○	●	○	●	○	○	○	●	○	○	○	●	○	○	○	●
4	4/16	LG4H/L (0229H to 8/8888H)	○	○	○	●	○	○	○	●	○	○	○	●	○	○	○	●
3	3/16	LG3H/L (0227H to 6/8880H)	○	○	○	○	○	○	○	●	○	○	○	●	○	○	○	●
2	2/16	LG2H/L (0225H to 4/8080H)	○	○	○	○	○	○	○	●	○	○	○	○	○	○	○	●
1	1/16	LG1H/L (0223H to 2/8000H)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	●
0	0/16	LG0H/L (0221H to 0/0000H)	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○

●: Display ON, ○: Display OFF

Table 3.14.7 Grayscale 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 levels	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
8 gray levels	●	×	●	×	●	×	●	×	●	×	●	×	●	×	×	●
4 gray levels	●	×	×	×	●	×	×	×	●	×	×	×	×	×	×	●
Monochrome	●	×	×	×	×	×	×	×	×	×	×	×	×	×	×	●

×: Don't care, ●: Effective

3.14.4.3 Memory Mapping

The LCDC can display the LCD panel image which is divided horizontally into 3 parts; upper, middle and lower. Each area calls A, B and C area that has some characteristics showing below.

Start/End address of each area in the physical memory space can be defined in the LCD Start/End address registers (See Table 3.14.3). (C area can be defined only start address.)

A and B areas are programmable visibility and they are set enable or not in LCDMODE register. When A and B area are disable, the C area take over all panel space. When the size of A or B area is greater than LCD panel, the area of the panel is all C area because the displaying priority is A > B > C.

If the A area set to enable while the panel area is defined as all C area (that is A and B area are disable), C area is shifted to under the LCD panel and A area is inserted from the top of the LCD panel. Similarly if the B area set to enable while the panel area is defined as all C area, B area is inserted from the bottom of the C area overlapping.

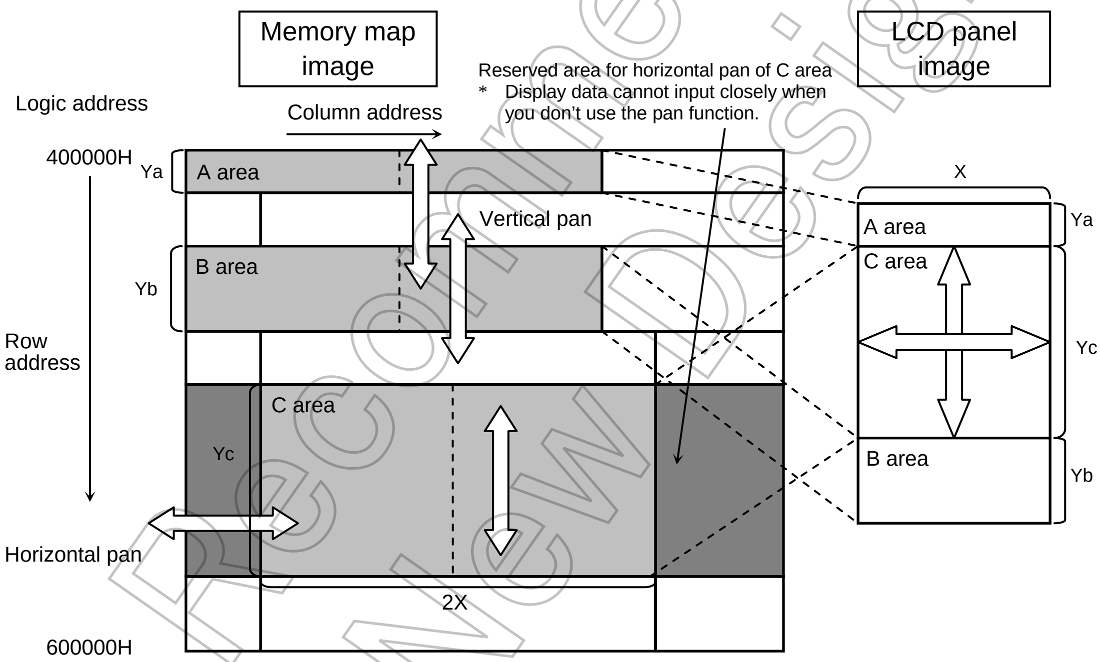


Figure 3.14.8 Memory Mapping from Physical Memory to LCD Panel

- Display memory mapping and panning function

LCDC can change the panel window if only you change each start address of A, B and C area. A and B area can be vertical panned by changing row address. While C area can be vertical and horizontal panned by changing row and column address.

An important thing is that display data from one line to the next line, cannot be input continuously even if you don't use the panning function. One row address of display RAM corresponds to 1st line of display panel. Now display data of 2nd line cannot be set within the 1st row address of display RAM even if the necessary data for the size you want to display do not fill the capacity of 1st row address of display RAM. Adding the one line to display panel is equal to adding one address to row address of display RAM.

And another important thing is, this limitation is also for SRAM as display RAM without address multiplex. When you use SDRAM as display RAM, you can select the size for display RAM capacity of one line (Number of column address: select 512 byte = 64 Mbytes 1024 byte = 128 Mbytes) bit. But in case of using SRAM, display RAM capacity of one line is fixed to 512 bytes.

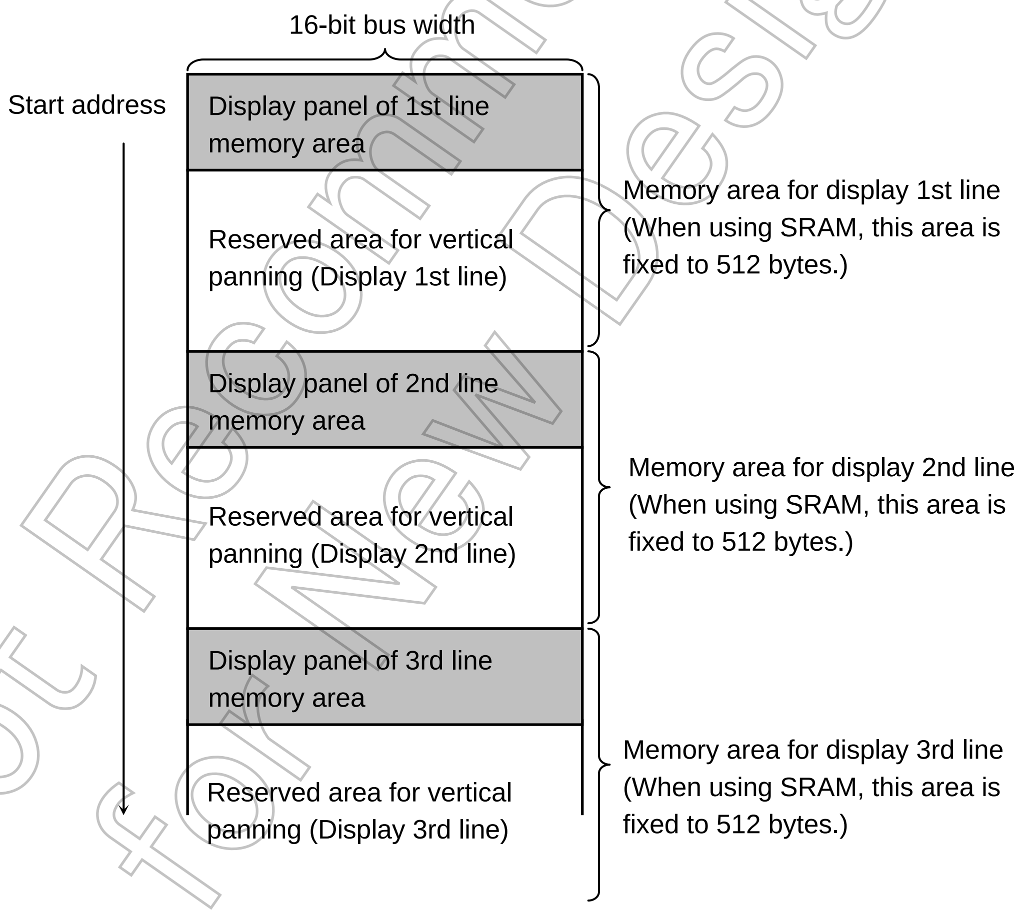


Figure 3.14.9 Memory Mapping Image for SRAM as Display RAM

TMP92C820 can select four display scale; monochrome, 4 gray, 8 gray and 16 gray levels. With the intrinsic property of gray levels, a pixel is decoded in each gray level from different memory size.

A pixel is equal to a bit in memory for monochrome, while a pixel is equal to 2 bits in memory for 4 gray levels, 3 bits for 8 gray levels and 4 bits for 16 gray levels. Therefore when the 4 gray mode, column address in the memory needs twice data capacity as large as dots that is displayed in the LCD panel actually showing Figure 3.14.8. Place for display data setting has some differences for each grayscale or sort of memory.

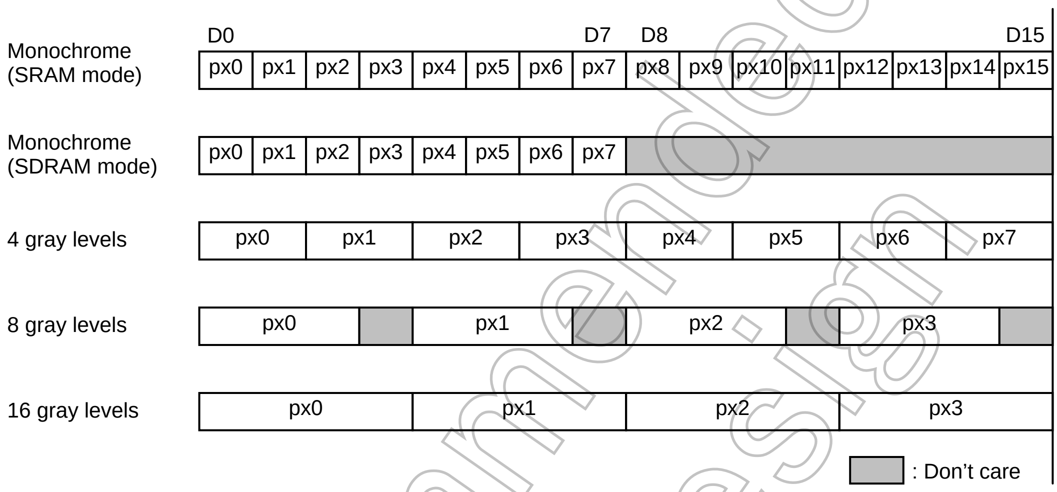


Figure 3.14.10 Memory Codes for Each Gray Level in a Read Cycle (16 bits)

And "px" in above Figure 3.14.10 corresponds to the image of LCD panel as below (Figure 3.14.11). But TMP92C820 outputs data of px0 from PE7 (LD7), and data of px7 from PE0 (LD0). Therefore PE0 (LD0) should be connected to the MSB of LCD driver (e.g., DI7) according to LCD driver you use. Please note that the way TMP92C820 outputs the data differs from LCD controller built-in TLCS-900/L1 series of TOSHIBA (e.g., TMP91C815, TMP91C016, and TMP91C025 etc.).

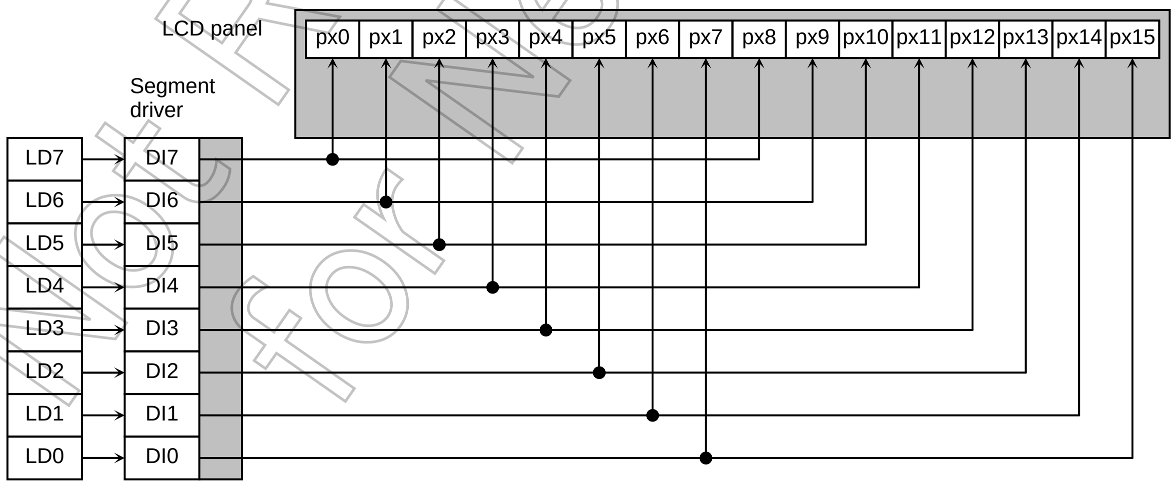


Figure 3.14.11 Connection between LD Bus of TMP92C820 and Data Bus of LCDD

3.14.4.4 Hardware Cursor

TMP92C820 has a cursor that is blinking interval, color and size can be specified, and maximum size is 32 × 32.

A programmer can control the cursor attributes easily by filling those cursor registers, for example color (White/black), blinking interval time, size and precise pixel location. Its space location is specified by left-up hot point. (See Figure 3.14.12)

The precise location of the hot point is determined by memory address (LCDCPH, LCDCPM, LCDCPL) and bit correction number (LCDCP). For example, however 1 pixel for displaying needs 2 bits of setting data under 4 gray mode, you can correct the location of hot point every 1 bit by setting pixel number which you want to move in the register (LCDCP).

Cursor image is showed under the setting A, B, C area are enable, 4 gray mode, start address = 410004H and correction bit (LCDCP) = 3H in the following figure.

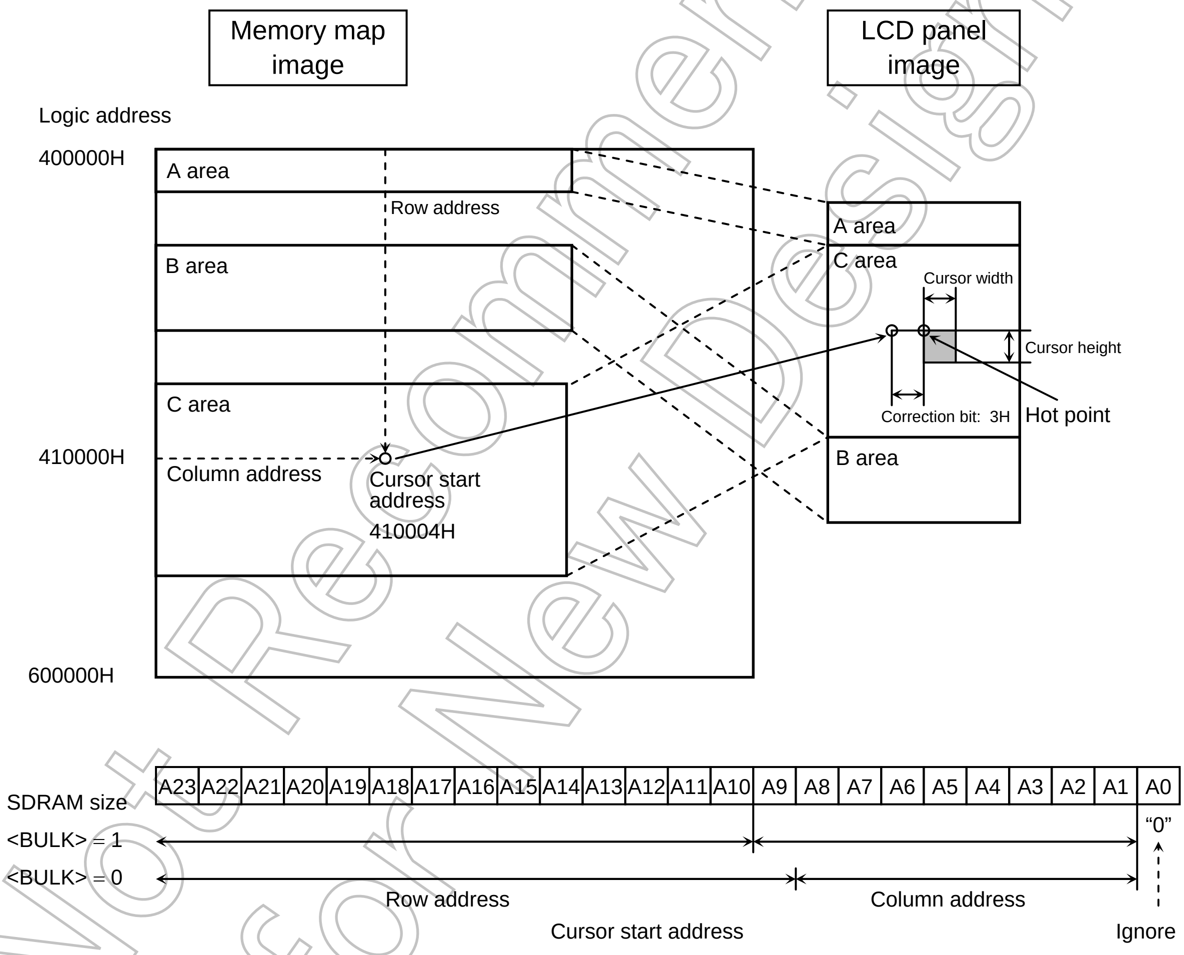


Figure 3.14.12 Cursor Hot Point Position and Size

Note: If panning function is set to enable during hardware cursor displaying, the cursor moves with the data in the memory. Because TMP92C820 sets the hardware cursor in the memory address.

LCD Cursor Setting Register

LCDCM (0206H)

	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 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 32 kHz.

Note 2: The function of cursor blink depends on the low-frequency oscillator even if you use timer out “TA3OUT” as LCDCK.

LCD Cursor Width Setting Register

LCDCW (0207H)

	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 (X size) 00000: 1 dot (Min) 11111: 32 dots (Max)				

LCD Cursor Height Setting Register

LCDCH (0208H)

	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 (Y size) 00000: 1 dot (Min) 11111: 32 dots (Max)				

LCD Cursor Start Address Setting Register

LCDCPL (020AH)

	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 (020BH)

	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 (020CH)

	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

		7	6	5	4	3	2	1	0
LCDCP (0209H)	Bit symbol					APB3	APB2	APB1	APB0
	Read/Write					R/W			
	After reset					0	0	0	0
	Function					Setting bit3 to bit0 of pixel for correction of hot point (for 1-dot correction)			

In case of monochrome (SRAM mode)	0000: 0 pixels correct	
	1111: 0 pixels correct	
In case of monochrome (SDRAM mode) and 4 gray levels	x000: 0 pixels correct	x100: 4 pixels correct
	x001: 1 pixel correct	x101: 5 pixels correct
	x010: 2 pixels correct	x110: 6 pixels correct
	x011: 3 pixels correct	x111: 7 pixels correct
In case of 8 and 16 gray levels	xx00: 0 pixels correct	xx10: 2 pixels correct
	xx01: 1 pixel correct	xx11: 3 pixels correct

X: Don't care

Here, it is possible to correct the cursor per 1 bit from the start address set before. Pixel number should be adjusted in response to the gray mode setting showing above.

For example, when 4 gray levels and 16-bit bus mode, correction should be less than 7 because the smallest pixel is 8 pixels that can set by start address setting. Similarly correction pixel should be less than 15 at monochrome mode, 3 at 8 or 16 gray modes.

Example: When monochrome mode, correction value is (LCDCP) = 011H, and cursor size = (8×8)

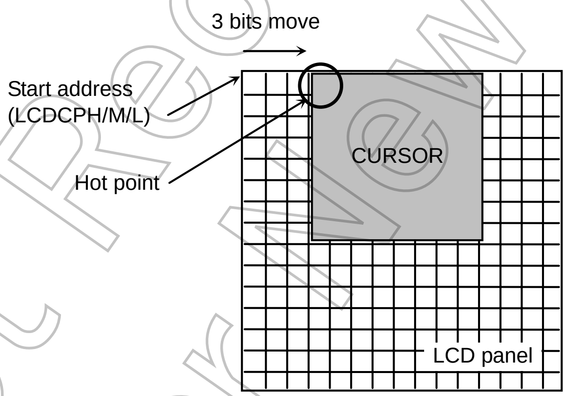


Figure 3.14.13 The Location Hot Point by Setting of Pixel

3.14.4.5 Frame Signal Settlement

TMP92C820 defines so-called frame period (refresh interval for LCD panel) by the value set in f_{FP} [9:0]. DLEBCD pin outputs pulse every frame period. D3BFR pin usually outputs the signal inverts polarity every frame period.

And TMP92C820 has a special function that can set the timing of inverting frame polarity irrelevant to above frame frequency for the purpose of preventing the patches of display.

LCD Control Register

LCDCTL (0203H)

	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: OFF 1: 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]	Type setting of LCD driver with built-in RAM 0: Sequen-tial access type 1: Random access type	Setting bit8 for f_{FP} [9:0]	Start control in SR mode 0: Stop 1: Start

LCD f_{FP} Register

LCDFFP (0204H)

	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	0	0	0	0	0	0	0
Function	Setting bit7 to 0 for f_{FP}							

Divide FRM Register

LCDDVM (0201H)

	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	0	0	0	0	0	0	0
Function	Setting DVM bit7 to 0							

(1) Settlement of frame frequency function

Basic frame period; DLEBCD signal, is made according to the resister f_{FP} [9:0] setting mentioned before. However this f_{FP} [9:0] setting is generally equal to common number, frame period can be corrected by increasing f_{FP} [9:0] with ease. This function cannot correct frame frequency higher than that of Table 3.14.8. If it is necessary to set frame frequency higher or detailed, please refer to (3) Timer out LCDCK.

The equation can calculate frame period.

Frame period = LCDCK/(D × f_{FP}) [Hz]

D: Constant for each common (Table 3.14.8)

f_{FP}: Setting of f_{FP} [9:0] register

LCDCK: Source clock of LCD
(Low clock is usually selected)

Please select the value of f_{FP} [9:0] as the frame period you want to set in the Table 3.14.8

Note: Please make the value set to f_{FP} [9:0] into the following range.
COM (Common number) ≤ f_{FP} ≤ 1024

Example 1: In the case where frame period is set to 72.10 Hz by 240 coms.
f_{FP} = 240 (COM) + 63 = 303 = 12FH (by Table 3.14.8)
Therefore, LCDCTL<FP8> = 1_hex and LCDFFP<FP7:0> = 2FH are setup.

(2) Frame invert adjustment function

This mode can prevent the deterioration of display (e.g., patches of display).
*Note 1:

If N is set in (LCDDVM) register while this function is set to enable in register (LCDCTL)(<FRMON> "1"), D3BFR pin outputs the signal inverted polarity every (D2BLP × N) timing.

If this function isn't necessary, D3BFR pin outputs the signal inverted polarity every frequency of DLEBCD pin after setting this function disable ((LCDCTL) <FRMON> = "0").

And it is no change wave and timing for DLEBCD pin by LCDDVM setting.

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

Example 2: Here indicates the method that frame period is set 70 [Hz] by selecting TA3OUT for source clock of LCD. (fc = 6 [MHz], 128 COM)

The next equation calculates frame period.

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

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

t_{LP} = D/XT   D: The value is 3 at 128 COM

Therefore if you set the frame period at 70 [Hz] under 128 COM,

XT = 128 × 3 × 70

= 26880 [Hz]

XT should be above value.

In order to make XT = 26880 [Hz] under fc = 6 [MHz] with ϕT1 of timer3,

1/XT = T3 × 2 × 8 × 2/ fc [s]   T3: the value of timer register (TA3REG)

in short,

XT = fc/(T3 × 2 × 8 × 2) [Hz]

However T3 = (TA3REG) is 6.98 after calculate, it's impossible to set the value under a decimal point. So if (TA3REG) is set 06H, XT = 31250 [Hz].

And because of D = 3,

Frame period = 31250/(128 × 3)

= 81.38 [Hz]

Further if f_{FP} is 148 (COM + 20) with correction,

Frame period = 31250/(148 × 3)

= 70.38 [Hz]

Reference: To maintain quality for display, please refer to following value for each grayscale.
(You have to use settlement of frame frequency function, frame invert adjustment function and timer out LCDCK.)
Monochrome: Frame period = 70 [Hz]
4/8/16 gray levels: Frame period = 140 [Hz]

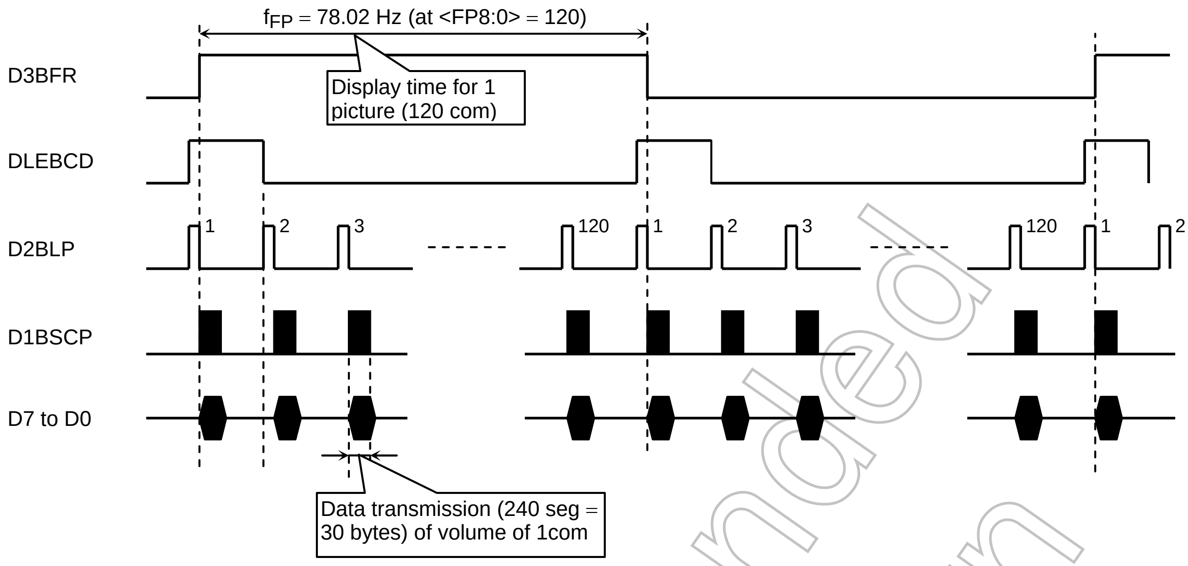


Figure 3.14.14 Timing Diagram for SR Mode

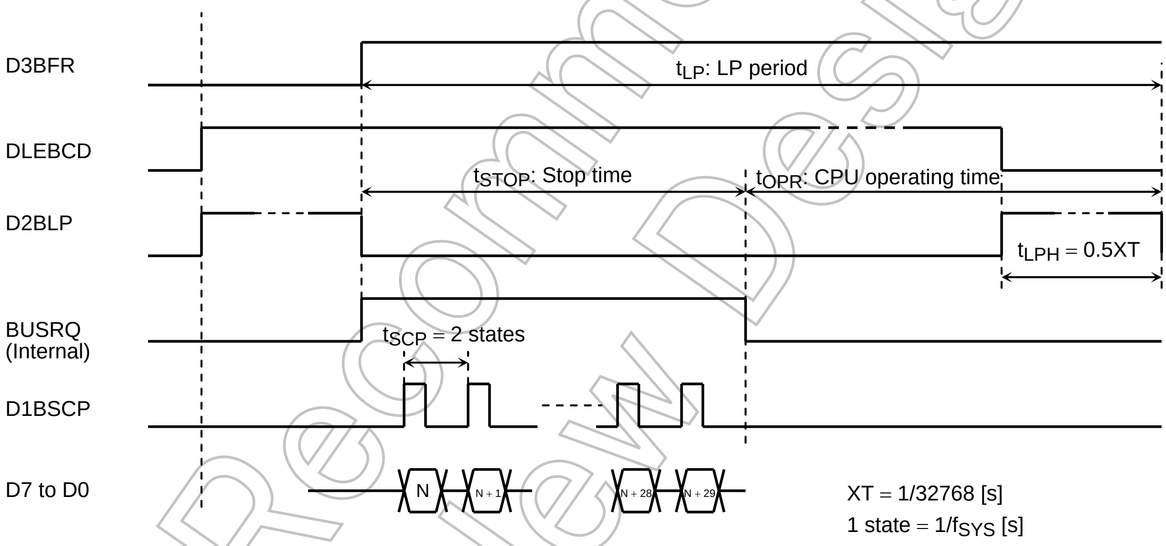


Figure 3.14.15 Timing Diagram for SR Mode (Detail)

D3BFR waveform (in case of 240 rows + 63 (f_{FP}) and LCDDVM<FMN7:0> = 0BH)

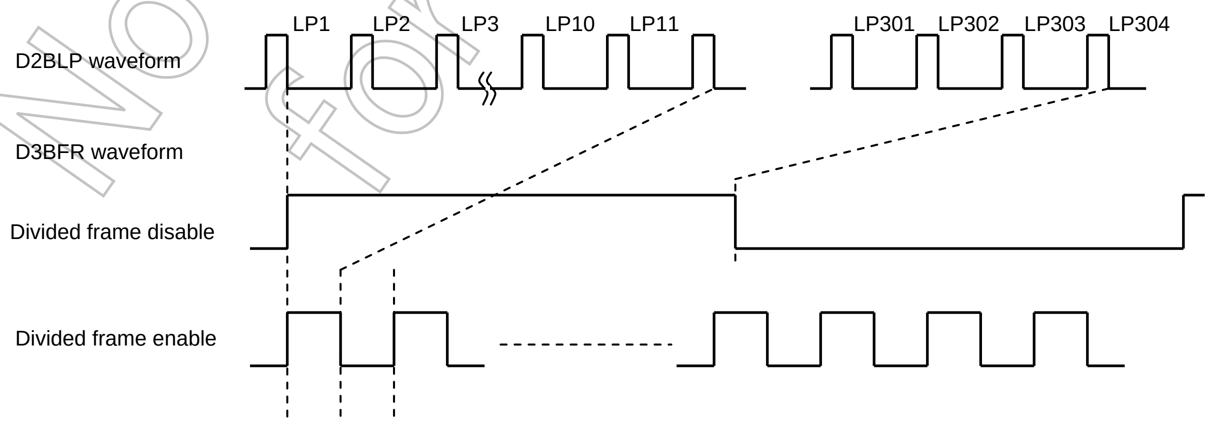


Figure 3.14.16 D2BLP and D3BFR Waveform

Table 3.14.8 f_{FP} Table for Each Common Number (1/2)

D	3	2.5	2	1.5	1.5	1	1
COM	128	160	200	240	320	400	480
COM + 0	85.33	81.92	81.92	91.02	68.27	81.92	68.27
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 3.14.8 f_{FP} Table for Each Common Number (2/2)

D	3	2.5	2	1.5	1.5	1	1
COM	128	160	200	240	320	400	480
COM + 52	60.68	61.83	65.02	74.81	58.72	72.50	61.59
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: The above time distance are value which used fs = 32.768 [kHz].

Table 3.14.9 Performance Listing for Each Segment and Common Number

(1) SDRAM (Burst) 16 bits, 8/16 gray levels

Segment	Common	128	160	200	240	320	400	480
	D	3	2.5	2	1.5	1.5	1	1
	t_{LP} [μs]	91.6	76.3	61	45.8	45.8	30.5	30.5
128	t_{STOP} [μs]	1.2	1.2	1.2	1.2	1.2	1.2	1.2
	RATE [%]	1.3	1.6	2.0	2.6	2.6	3.9	3.9
160	t_{STOP} [μs]	1.4	1.4	1.4	1.4	1.4	1.4	1.4
	RATE [%]	1.5	1.8	2.3	3.1	3.1	4.6	4.6
240	t_{STOP} [μs]	1.9	1.9	1.9	1.9	1.9	1.9	1.9
	RATE [%]	2.1	2.5	3.1	4.1	4.1	6.2	6.2
320	t_{STOP} [μs]	2.4	2.4	2.4	2.4	2.4	2.4	2.4
	RATE [%]	2.6	3.1	3.9	5.2	5.2	7.9	7.9
400	t_{STOP} [μs]	2.9	2.9	2.9	2.9	2.9	2.9	2.9
	RATE [%]	3.2	3.8	4.8	6.3	6.3	9.5	9.5
480	t_{STOP} [μs]	3.4	3.4	3.4	3.4	3.4	3.4	3.4
	RATE [%]	3.7	4.5	5.6	7.4	7.4	11.1	11.1
560	t_{STOP} [μs]	3.9	3.9	3.9	3.9	3.9	3.9	3.9
	RATE [%]	4.3	5.1	6.4	8.5	8.5	12.8	12.8
640	t_{STOP} [μs]	4.4	4.4	4.4	4.4	4.4	4.4	4.4
	RATE [%]	4.8	5.8	7.2	9.6	9.6	14.4	14.4

(2) SDRAM (Burst) 16 bits, 4 gray levels

Segment	Common	128	160	200	240	320	400	480
	D	3	2.5	2	1.5	1.5	1	1
	t_{LP} [μs]	91.6	76.3	61	45.8	45.8	30.5	30.5
128	t_{STOP} [μs]	1.2	1.2	1.2	1.2	1.2	1.2	1.2
	RATE [%]	1.3	1.6	2.0	2.6	2.6	3.9	3.9
160	t_{STOP} [μs]	1.4	1.4	1.4	1.4	1.4	1.4	1.4
	RATE [%]	1.5	1.8	2.3	3.1	3.1	4.6	4.6
240	t_{STOP} [μs]	1.9	1.9	1.9	1.9	1.9	1.9	1.9
	RATE [%]	2.1	2.5	3.1	4.1	4.1	6.2	6.2
320	t_{STOP} [μs]	2.4	2.4	2.4	2.4	2.4	2.4	2.4
	RATE [%]	2.6	3.1	3.9	5.2	5.2	7.9	7.9
400	t_{STOP} [μs]	2.9	2.9	2.9	2.9	2.9	2.9	2.9
	RATE [%]	3.2	3.8	4.8	6.3	6.3	9.5	9.5
480	t_{STOP} [μs]	3.4	3.4	3.4	3.4	3.4	3.4	3.4
	RATE [%]	3.7	4.5	5.6	7.4	7.4	11.1	11.1
560	t_{STOP} [μs]	3.9	3.9	3.9	3.9	3.9	3.9	3.9
	RATE [%]	4.3	5.1	6.4	8.5	8.5	12.8	12.8
640	t_{STOP} [μs]	4.4	4.4	4.4	4.4	4.4	4.4	4.4
	RATE [%]	4.8	5.8	7.2	9.6	9.6	14.4	14.4

(3) SDRAM (Burst) 16 bits, monochrome

Segment	Common	128	160	200	240	320	400	480
	D	3	2.5	2	1.5	1.5	1	1
	t_{LP} [μs]	91.6	76.3	61	45.8	45.8	30.5	30.5
128	t_{STOP} [μs]	0.8	0.8	0.8	0.8	0.8	0.8	0.8
	RATE [%]	0.9	1.0	1.3	1.7	1.7	2.6	2.6
160	t_{STOP} [μs]	0.9	0.9	0.9	0.9	0.9	0.9	0.9
	RATE [%]	1.0	1.2	1.5	2.0	2.0	3.0	3.0
240	t_{STOP} [μs]	1.2	1.2	1.2	1.2	1.2	1.2	1.2
	RATE [%]	1.3	1.5	1.9	2.5	2.5	3.8	3.8
320	t_{STOP} [μs]	1.4	1.4	1.4	1.4	1.4	1.4	1.4
	RATE [%]	1.5	1.8	2.3	3.1	3.1	4.6	4.6
400	t_{STOP} [μs]	1.7	1.7	1.7	1.7	1.7	1.7	1.7
	RATE [%]	1.8	2.2	2.7	3.6	3.6	5.4	5.4
480	t_{STOP} [μs]	1.9	1.9	1.9	1.9	1.9	1.9	1.9
	RATE [%]	2.1	2.5	3.1	4.1	4.1	6.2	6.2
560	t_{STOP} [μs]	2.2	2.2	2.2	2.2	2.2	2.2	2.2
	RATE [%]	2.3	2.8	3.5	4.7	4.7	7.0	7.0
640	t_{STOP} [μs]	2.4	2.4	2.4	2.4	2.4	2.4	2.4
	RATE [%]	2.6	3.1	3.9	5.2	5.2	7.9	7.9

(4) SRAM (2 states) 16 bits, 8/16 gray levels (Note 2)

Segment	Common	128	160	200	240	320	400	480
	D	3	2.5	2	1.5	1.5	1	1
	t_{LP} [μs]	91.6	76.3	61	45.8	45.8	30.5	30.5
128	t_{STOP} [μs]	3.4	3.4	3.4	3.4	3.4	3.4	3.4
	RATE [%]	3.7	4.4	5.5	7.3	7.3	11.0	11.0
160	t_{STOP} [μs]	4.2	4.2	4.2	4.2	4.2	4.2	4.2
	RATE [%]	4.5	5.4	6.8	9.1	9.1	13.6	13.6
240	t_{STOP} [μs]	6.2	6.2	6.2	6.2	6.2	6.2	6.2
	RATE [%]	6.7	8.1	10.1	13.4	13.4	20.2	20.2
320	t_{STOP} [μs]	8.2	8.2	8.2	8.2	8.2	8.2	8.2
	RATE [%]	8.9	10.7	13.4	17.8	17.8	26.7	26.7
400	t_{STOP} [μs]	10.2	10.2	10.2	10.2	10.2	10.2	10.2
	RATE [%]	11.1	13.3	16.6	22.2	22.2	33.3	33.3
480	t_{STOP} [μs]	12.2	12.2	12.2	12.2	12.2	12.2	12.2
	RATE [%]	13.3	15.9	19.9	26.5	26.5	39.8	39.8
560	t_{STOP} [μs]	14.2	14.2	14.2	14.2	14.2	14.2	14.2
	RATE [%]	15.4	18.5	23.2	30.9	30.9	46.4	46.4
640	t_{STOP} [μs]	16.2	16.2	16.2	16.2	16.2	16.2	16.2
	RATE [%]	17.6	21.2	26.5	35.3	35.3	53.0	53.0

(5) SRAM (2 states) 16 bits, 4 gray levels (Note 2)

Segment	Common	128	160	200	240	320	400	480
	D	3	2.5	2	1.5	1.5	1	1
	t_{LP} [μs]	91.6	76.3	61	45.8	45.8	30.5	30.5
128	t_{STOP} [μs]	1.8	1.8	1.8	1.8	1.8	1.8	1.8
	RATE [%]	1.9	2.3	2.9	3.8	3.8	5.7	5.7
160	t_{STOP} [μs]	2.2	2.2	2.2	2.2	2.2	2.2	2.2
	RATE [%]	2.3	2.8	3.5	4.7	4.7	7.0	7.0
240	t_{STOP} [μs]	3.2	3.2	3.2	3.2	3.2	3.2	3.2
	RATE [%]	3.4	4.1	5.2	6.9	6.9	10.3	10.3
320	t_{STOP} [μs]	4.2	4.2	4.2	4.2	4.2	4.2	4.2
	RATE [%]	4.5	5.4	6.8	9.1	9.1	13.6	13.6
400	t_{STOP} [μs]	5.2	5.2	5.2	5.2	5.2	5.2	5.2
	RATE [%]	5.6	6.7	8.4	11.2	11.2	16.9	16.9
480	t_{STOP} [μs]	6.2	6.2	6.2	6.2	6.2	6.2	6.2
	RATE [%]	6.7	8.1	10.1	13.4	13.4	20.2	20.2
560	t_{STOP} [μs]	7.2	7.2	7.2	7.2	7.2	7.2	7.2
	RATE [%]	7.8	9.4	11.7	15.6	15.6	23.4	23.4
640	t_{STOP} [μs]	8.2	8.2	8.2	8.2	8.2	8.2	8.2
	RATE [%]	8.9	10.7	13.4	17.8	17.8	26.7	26.7

(6) SRAM (2 states) 16 bits, monochrome (Note 2)

Segment	Common	128	160	200	240	320	400	480
	D	3	2.5	2	1.5	1.5	1	1
	t_{LP} [μs]	91.6	76.3	61	45.8	45.8	30.5	30.5
128	t_{STOP} [μs]	1.0	1.0	1.0	1.0	1.0	1.0	1.0
	RATE [%]	1.0	1.2	1.6	2.1	2.1	3.1	3.1
160	t_{STOP} [μs]	1.2	1.2	1.2	1.2	1.2	1.2	1.2
	RATE [%]	1.3	1.5	1.9	2.5	2.5	3.8	3.8
240	t_{STOP} [μs]	1.7	1.7	1.7	1.7	1.7	1.7	1.7
	RATE [%]	1.8	2.2	2.7	3.6	3.6	5.4	5.4
320	t_{STOP} [μs]	2.2	2.2	2.2	2.2	2.2	2.2	2.2
	RATE [%]	2.3	2.8	3.5	4.7	4.7	7.0	7.0
400	t_{STOP} [μs]	2.7	2.7	2.7	2.7	2.7	2.7	2.7
	RATE [%]	2.9	3.5	4.3	5.8	5.8	8.7	8.7
480	t_{STOP} [μs]	3.2	3.2	3.2	3.2	3.2	3.2	3.2
	RATE [%]	3.4	4.1	5.2	6.9	6.9	10.3	10.3
560	t_{STOP} [μs]	3.7	3.7	3.7	3.7	3.7	3.7	3.7
	RATE [%]	4.0	4.8	6.0	8.0	8.0	12.0	12.0
640	t_{STOP} [μs]	4.2	4.2	4.2	4.2	4.2	4.2	4.2
	RATE [%]	4.5	5.4	6.8	9.1	9.1	13.6	13.6

Note 1: These tables are calculated at following condition.

1) f_{SYS} = 20 [MHz]
2) fs = 32.768 [kHz]
3) Overhead state number are 8 states for SDRAM and 3 states for SRAM.

Note 2: For SRAM tables ((4) to (6)), t_{STOP} is calculated at 2-state accessing.

Table 3.14.10 Possible Panel Size of Panning

64-Mbit SDRAM/BURST

Horizontal	SEG	128	160	240	320	400	480	560	640	
Monochrome		16.0	12.8	8.5	6.2	5.1	4.3	3.7	3.2	Panels
4 gray levels		16.0	12.8	8.5	6.4	5.1	4.3	3.7	3.2	Panels
8 gray levels		8.0	6.4	4.3	3.2	2.6	2.1	1.8	1.6	Panels
16 gray levels		8.0	6.4	4.3	3.2	2.6	2.1	1.8	1.6	Panels

Vertical	COM	128	160	200	240	320	400	480	
		32.0	25.6	20.5	17.1	12.8	10.2	8.5	Panels

128-Mbit SDRAM/BURST

Horizontal	SEG	128	160	240	320	400	480	560	640	
Monochrome		32.0	25.6	17.1	12.8	10.2	8.5	7.3	6.4	Panels
4 gray levels		32.0	25.6	17.1	12.8	10.2	8.5	7.3	6.4	Panels
8 gray levels		16.0	12.8	8.5	6.4	5.1	4.3	3.7	3.2	Panels
16 gray levels		16.0	12.8	8.5	6.4	5.1	4.3	3.7	3.2	Panels

Vertical	COM	128	160	200	240	320	400	480	
		32.0	25.6	20.5	17.1	12.8	10.2	8.5	Panels

SRAM

Horizontal	SEG	128	160	240	320	400	480	560	640	
Monochrome		32.0	25.6	17.1	12.8	10.2	8.5	7.3	6.4	Panels
4 gray levels		16.0	12.8	8.5	6.4	5.1	4.3	3.7	3.2	Panels
8 gray levels		8.0	6.4	4.3	3.2	2.6	2.1	1.8	1.6	Panels
16 gray levels		8.0	6.4	4.3	3.2	2.6	2.1	1.8	1.6	Panels

Vertical	COM	128	160	200	240	320	400	480	
		32.0	25.6	20.5	17.1	12.8	10.2	8.5	Panels

Note 1: The value of the Table 3.14.8 is at f_{FPH} = 36 [MHz].

Note 2: CPU stop time; t_{STOP} (in the Figure 3.14.17) is the time which CPU reads the memory of transferring with 0 waits.

Note 3: The following equation can calculate t_{LP} listed below. (fs = 32.768 [kHz])

$$t_{LP} = D/32768 \text{ [s]}$$

Example: If the row is 240 and D = 1.5 by the above table

$$t_{LP} = 1.5/32768 = 45.8 \text{ [}\mu\text{s]}$$

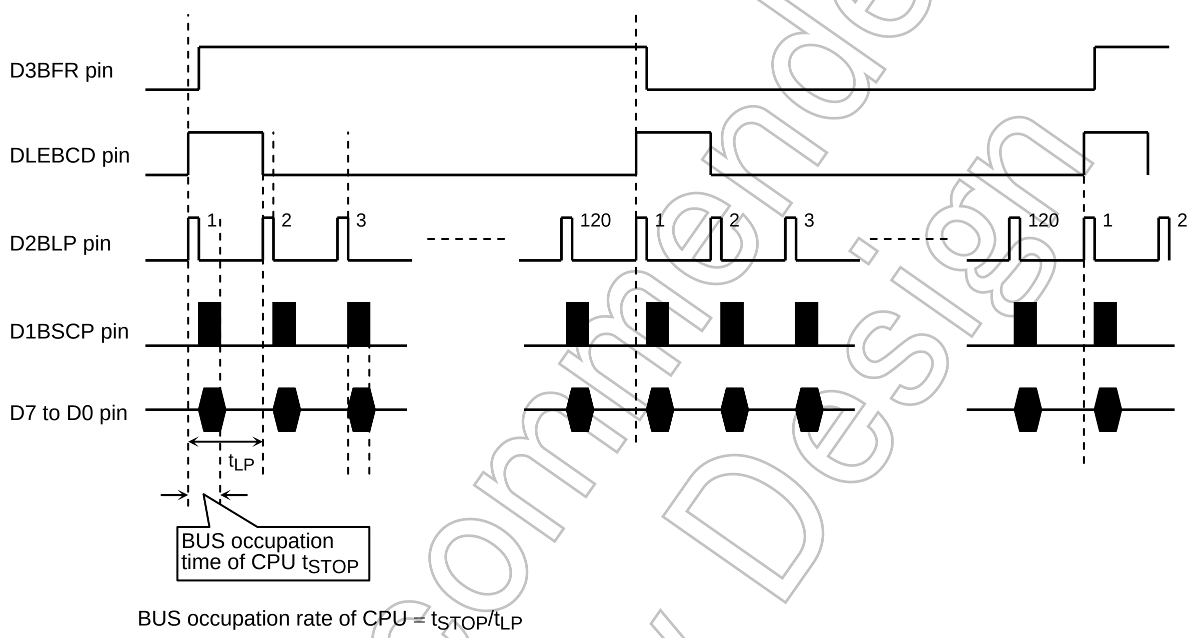


Figure 3.14.17 Stop Time and BUS Occupation Rate of CPU

3.14.4.6 Timing Charts of Interpreting Memory Codes

TMP92C820 supports different memory accessing. They are SRAM with waits, SDRAM burst modes, and the size of SDRAM is 64 or 128 Mbits. The access signals for the LCD panel are shown in Figure 3.14.18. TMP92C820 include 80 bytes FIFO. Therefore, If CPU operate high speed, TMP92C820 can to use low-speed LCD driver. To catch low speed LCD drivers, 4 types of SCP rates (f_{SYS}, $f_{SYS}/2$, $f_{SYS}/4$, and $f_{SYS}/8$) can be selected. The output data (LD7:0) will be issued from the built-in FIFO at the rising edge of D1BSCP when the FIFO is no empty. The work of the FIFO is illustrated in Figure 3.14.19, where the buffer size 80 bytes. The FIFO latches BaseLD<7:0> signal at the falling edge of BaseSCP that is shown in Figure 3.14.20 and Figure 3.14.21 for SRAM and SDRAM modes respectively. The FIFO is always reset to the empty state by the rising edge of D2BLP. In base SCP mode (e.g., for SCPW1:0 = 00), D1BCP is equal to BaseSCP, LD<7:0> equal to BaseLD<7:0> and no FIFO used. Generally, the data input rate of FIFO should be greater than the output one.

To make FIFO work correctly, the following condition have to be satisfied by setting SFR properly.

$$(SegNum/8 + 1) \times tcw + 24 \times t_{FPH} < t_{LP} - t_{LPH}$$

Here, SegNum is the segment number, and t_{CW} D1BSCP clock cycle width.

Referring Figure 3.14.22, we can know this relation means that the last LD<7:0> data must be generated before the rising edge of D2BLP. For example, in case of f_{FPH} = 36 MHz, XT = 32 kHz, 4 gray levels, 240 commons, 640 segments, and SDRAM burst mode, the following table can be obtained, which tells user that 8 clock mode is impossible and base, 2, 4 clock modes can be used.

Table 3.14.11 f_{FPH} = 36 [MHz], XT = 32 [kHz], 4 Grayscale, 240 Common, 640 Segment, SDRAM Burst Mode

SCPW	D1BSCP Rate (MHz)	tcw (ns)	$(SegNum/8 + 1) \times tcw + 24 \times t_{FPH}$ (ns)	$t_{LP} - t_{LPH}$ (ns)	Judgment
Base	9	111.2	9674.4	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: In case of SDRAM burst mode and 8/16 gray, the speed of base setting is equal to that of 2 CLK.

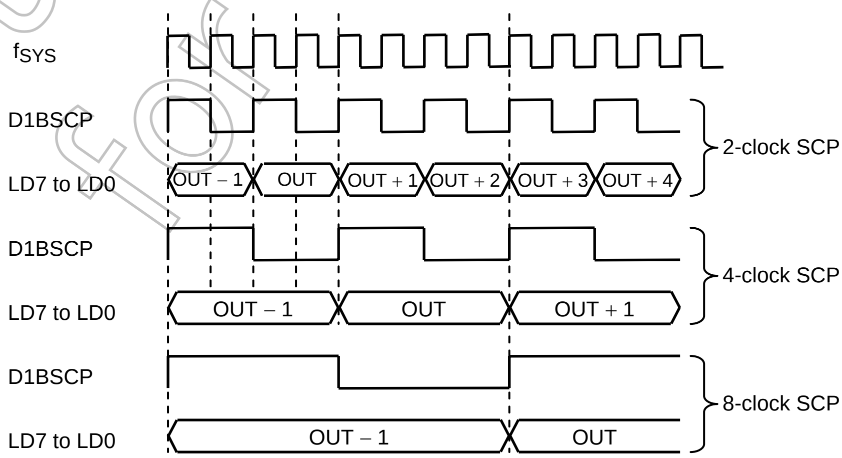
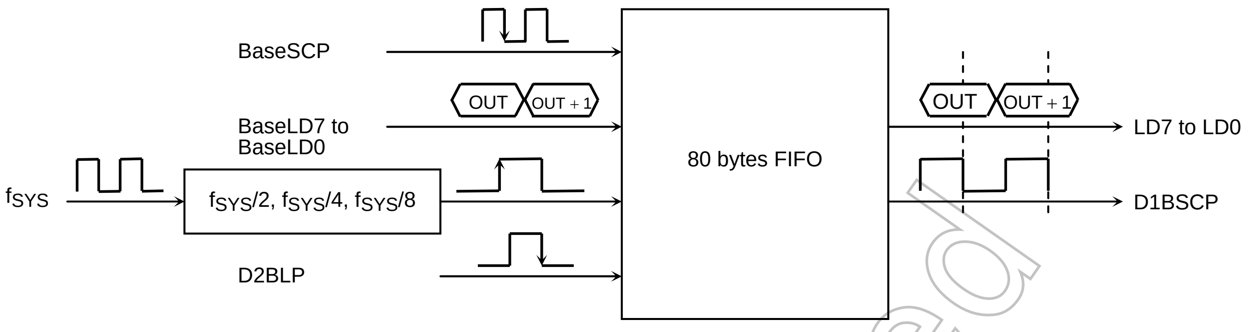


Figure 3.14.18 Timing Diagram for The LCD Driver Access Signals



Note: D1BCP = BaseSCP and LD<7:0> = BaseLD<7:0> in BaseSCP mode (e.g., for SCPW<1:0> = 00)

Figure 3.14.19 Timing Diagram for FIFO

SRAM 0 WAIT Mode

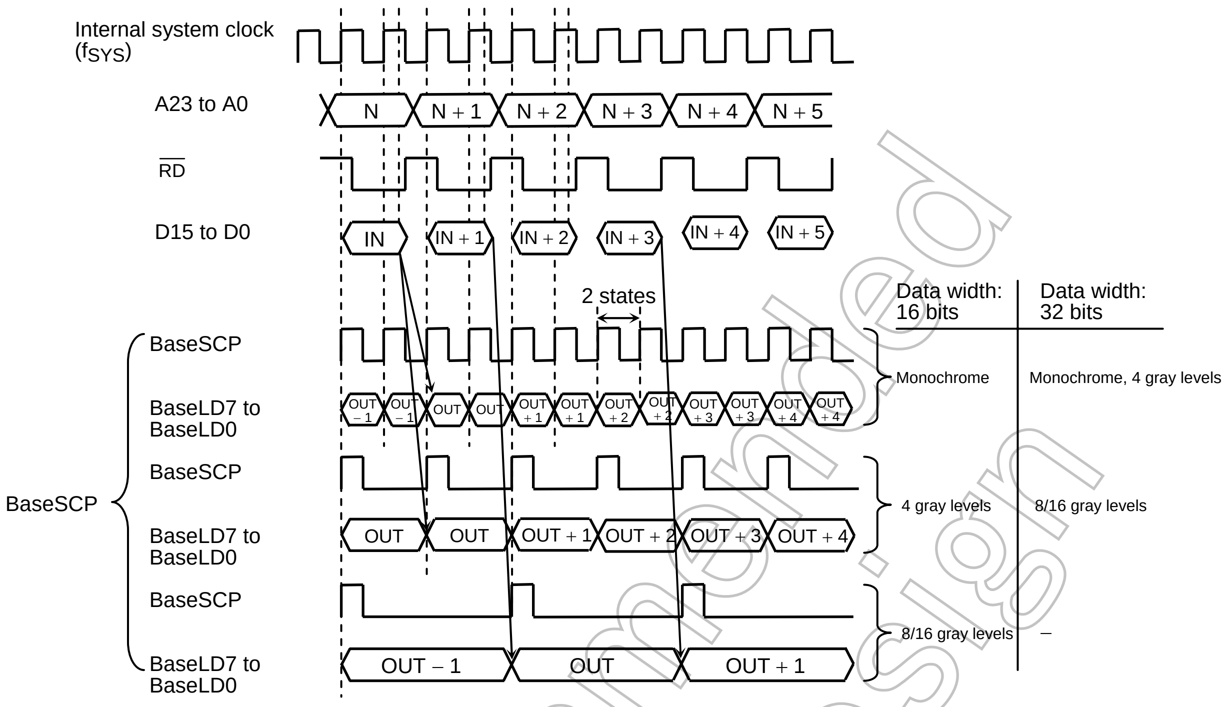


Figure 3.14.20 Timing Diagram for SRAM Mode with BaseSCP

SDRAM Burst Mode

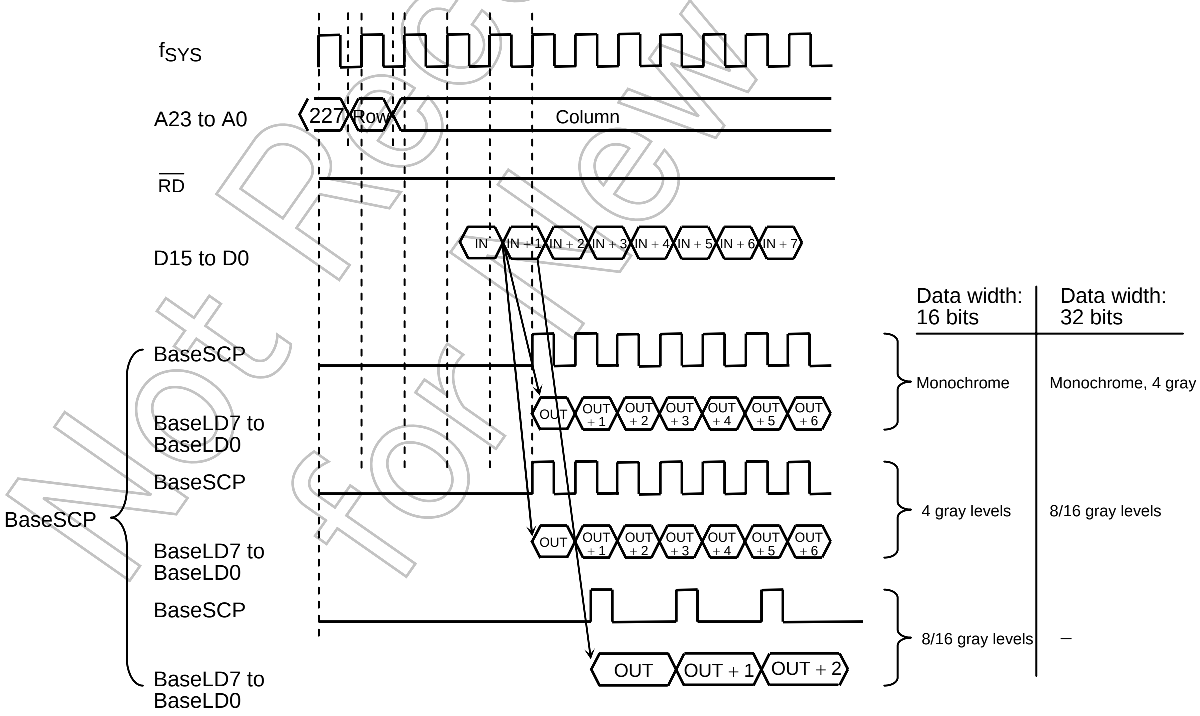
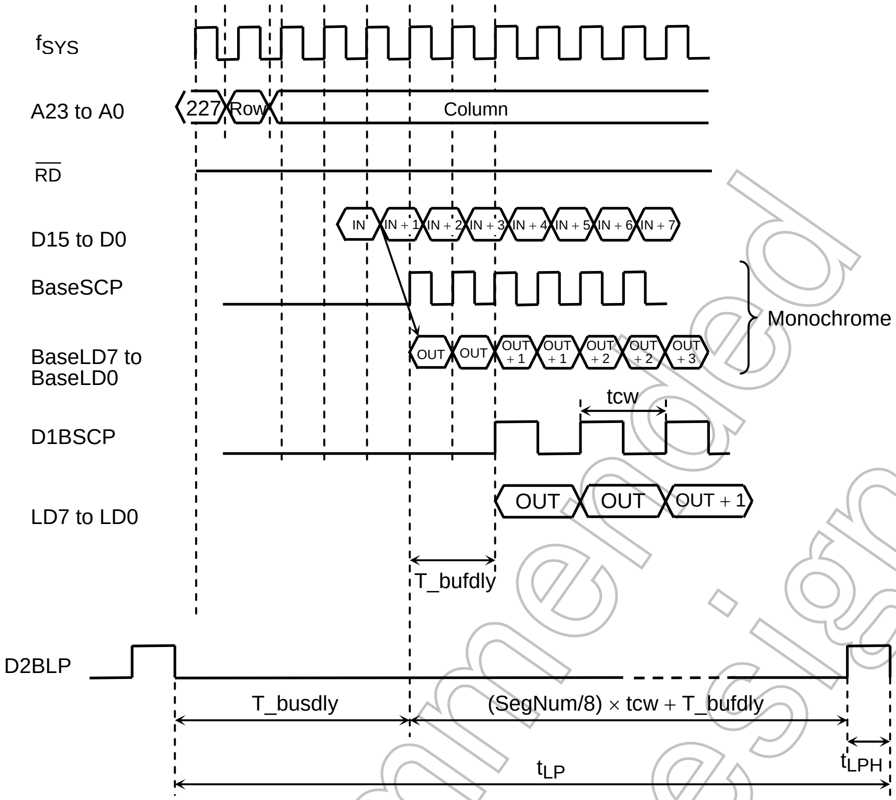


Figure 3.14.21 Timing Diagram for SDRAM Burst Mode with BaseSCP

SDRAM BURST1 Clock Mode

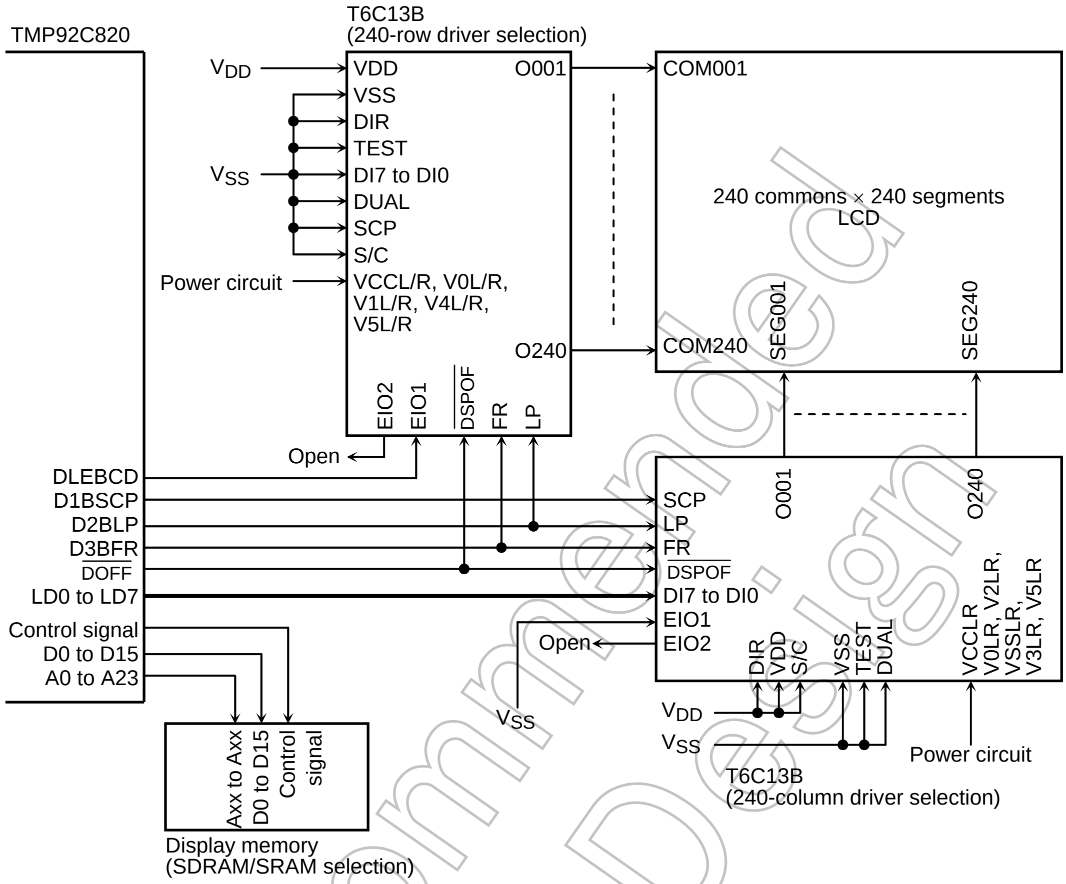


Note 1: $4t_{FPH} \le$ T_bufdly $\le$ tc + $2t_{FPH}$

Note 2: T_busdly is about 11 times as long as f_{SYS} period (22 t_{FPH}).

Figure 3.14.22 Timing Diagram for Maximum FIFO Delay Time

3.14.4.7 Examples to Use



Note 1: Display memory support only 16-bit bus.

Note 2: Other circuit is necessary for LCD drive power supply for LCD driver display.

Figure 3.14.23 Interface Example for Shift Register Type LCD Driver

Note: Because the connection between the line of display RAM data and output bus: LD<0:7> is just the mirror invertion, please care of connection. The data LSB of display RAM is output from LD7. In the above figure, LD0 should be connected to DI7 of LCDD driver, and LD1 to DI6. For detail information, please refer to Figure 3.14.11.

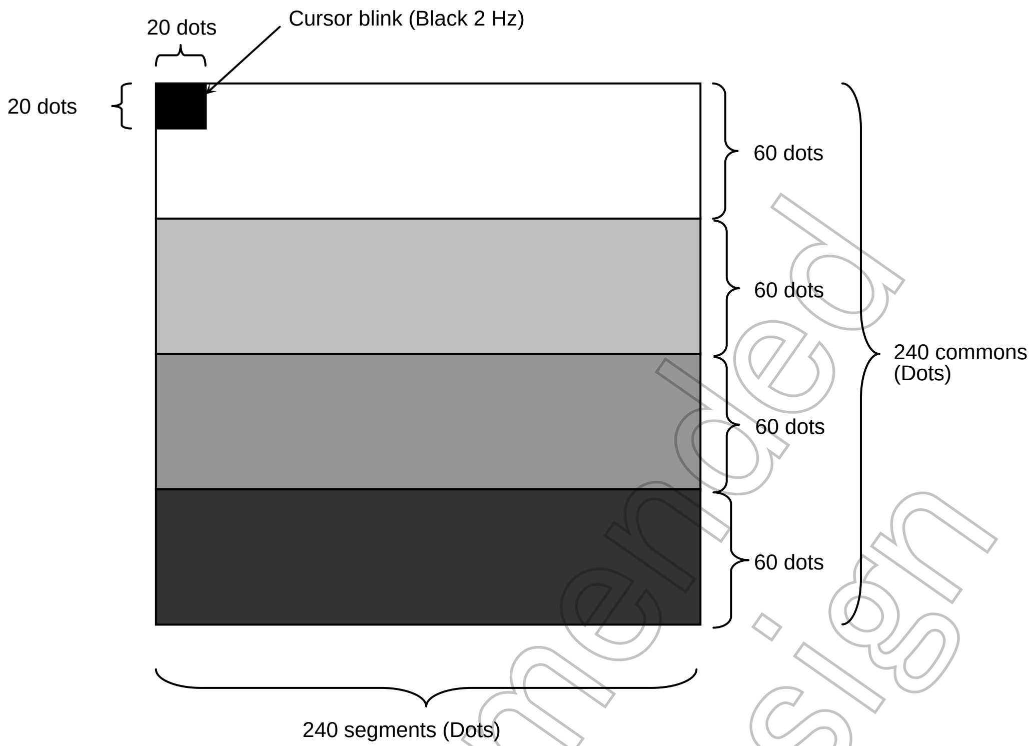


Figure 3.14.24 Display Reference below Sample Program

3.14.4.8 Sample Program

- Setting example: In case of use 240 segments × 240 commons, 4-level grayscale display, 64-Mbit SDRAM.

This sample program operate correctly, LCD panel shows Figure 3.14.18 display.

```
;***** SDRAM SET *****
        ld  (sdacr), 2bH           ; Add-MUX enable, 64-Mbit select
        ld  (sdrcr), 01H           ; Interval refresh

;***** GLCDC SET *****
        ld (lcdmode), 17H          ; A/B area OFF, SDRAM 64 Mbits, SR type
                                   ; SCP width 2clocks
        ld  (lcddvm), 11           ; 11-count DVM set
        ld  (lcdsize), 32H         ; COM = 240, SEG = 240
        ld  (lcdctl), 20H          ; Divide frame ON, display OFF
        ld  (lcdffp), 240          ; Frame frequency correction (91 Hz)
        ld  (lcdgl), 01H           ; 4-level grayscale
        ld  (lcdcm), 0c1H          ; Cursor ON, Black, 2 Hz blink
        ld  (lcdcw), 19            ; Width = 20 dots
        ld  (lcdch), 19            ; Height = 20 dots
        ld  (lcdcp), 00H           ; Pixel = 0
        ld  (lcdcpl), 00H          ; Cursor address
        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-level data (0000000000000000B)
loop1:  ld  (xix), wa              ;
        inc  2, xix                ;
        cp  xix, 407800H           ; 400000H to 4077FFH: 60 rows (Dots)
        jr  nz, loop1              ;
;***** 2/4 data write 60 ROW *****
        ld  xix, 407800H           ;
        ld  wa, 05555H             ; Write data 1/4-level data (0101010101010101B)
loop2:  ld  (xix), wa              ;
        inc  2, xix                ;
        cp  xix, 40F000H           ; 407800H to 40EFFFH: 60 rows (Dots)
        jr  nz, loop2              ;
;***** 3/4 data write 60 ROW *****
        ld  xix, 40F000H           ;
        ld  wa, 0aaaaH             ; Write data 2/4-level data (1010101010101010B)
loop3:  ld  (xix), wa              ;
        inc  2, xix                ;
        cp  xix, 416800H           ; 40F000H to 4167FFH: 60 rows (Dots)
        jr  nz, loop3              ;
```

```
;***** 4/4 data write 60 ROW *****
             ld  xix, 416800H            ;
             ld  wa, 0ffffH              ; Write data 3/4-level data (1111111111111111B)
loop4:       ld  (xix), wa               ;
             inc  2, xix                 ;
             cp  xix, 41e000H            ; 416800H to 41DFFFH: 60 rows (Dots)
             jr  nz, loop4               ;
;***** 4-level gray palette pattern set *****
             ld  (lg0l), 00H             ; 0/4 grayscale palette 0000B
             ld  (lg1l), 05H             ; 2/4 grayscale palette 0101B
             ld  (lg2l), 0eH             ; 3/4 grayscale palette 1110B
             ld  (lg3l), 0fH             ; 4/4 grayscale palette 1111B
;***** DMA, DISPLAY-ON start *****
             ld  (lcdctl), 0a1H          ; Display ON, divide ON
```

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

3.14.5.1 Operation

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

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

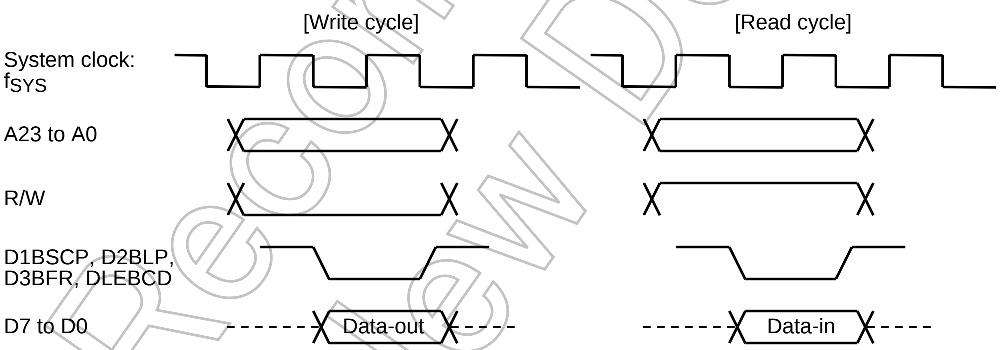
It corresponds to LCD driver which has every 1 byte of instruction register and display data register in LCD driver at the time of <MMULCD> = “0”. Please make the transmission place address at this time into either of FE0H to FE7F. (SEQUENTIAL ACCESS TYPE: See Table 3.14.4.)

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

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

Note: This operation mode cannot use cursor function.

Figure 3.14.25 shows access timing example in <MMULCD> = “0”. Also, Figure 3.14.26 shows example of connection.

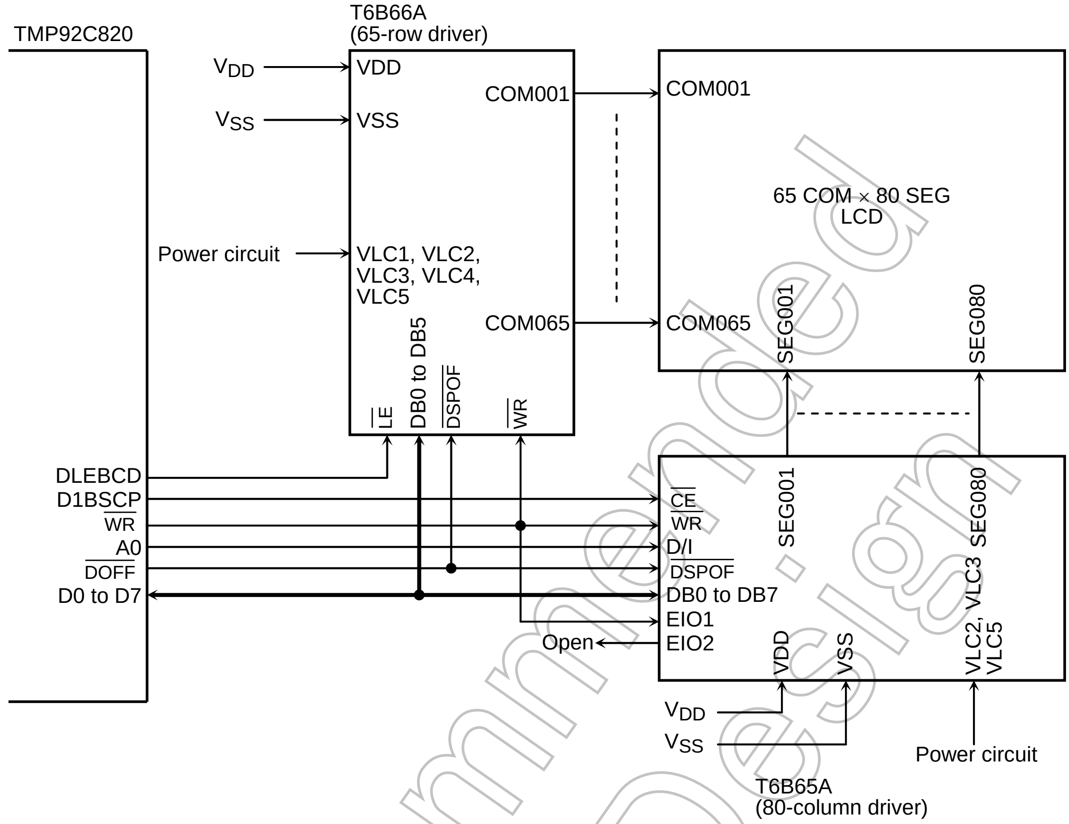


Note 1: This waveform is the case of 3-state access.

Note 2: Note the different rising timing for D1BSCP etc.

Figure 3.14.25 Example of Access Timing for RAM Built-in Type LCD Driver (Wait = 0)

3.14.5.2 Examples to Use



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

Figure 3.14.26 Interface Example for RAM Built-in Type Sequential Access Type LCD Driver

3.14.5.3 Sample Program

- Setting example: In case of use 80 segment × 65 commons LCD driver.

Assign external column driver to LCDC1 and row driver to LCDR1. This example used LD instruction in setting of instruction and used burst function of micro DMA by soft start in setting of display data.

In case of store 650 bytes transfer data to LCD driver.

```
; Setting external terminal
        LD  (PDFC), 19H             ; CE for LCDC1: D1BSCP,
                                    ; LE for LCDR1: DLEBCD,
                                    ; Setting for DOFF
; Setting for LCDC
        LD  (LCDMODE), 00H          ; Select RAM mode
        LD  (LCDCTL), 00H           ; Sequential access mode
; Setting for mode of LCDC1/LCDR1
        LD  (LCDC1L), XX            ; Setting instruction for LCDC1
        LD  (LCDR1L), XX            ; Setting instruction for LCDR1
; 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, 400000H            ; Source address = 400000H
        LDC  DMAS0, XWA             ;
        LD  XWA, 1FE1H              ; Destination address = 1FE1H (LCDC0H)
        LDC  DMAD0, XWA             ;
        LD  (INTETC01), 06H         ; INTTC0 level = 6
        EI  6                       ; Interrupt level = 6
        LD  (DMAB), 01H             ; Burst mode
        LD  (DMAR), 01H             ; Soft start
```

3.15 Melody/Alarm Generator (MLD)

The TMP92C820 contains a melody function and alarm function, both of which are output from the MLDALM pin. Five kinds of fixed cycle interrupt are generated using a 15-bit counter for use as the alarm generator.

The features are as follows.

1) Melody generator

The Melody function generates signals of any frequency (4 Hz to 5461 Hz) based on a low-speed clock (32.768 kHz), and outputs the signals from the MLDALM pin.

The melody tone can easily be heard by connecting an external loudspeaker.

2) Alarm generator

The alarm function generates eight kinds of alarm waveform having a modulation frequency (4096 Hz) determined by the low-speed clock (32.768 kHz). This waveform can be inverted by setting a value to a register.

The alarm tone can easily be heard by connecting an external loudspeaker.

Five kinds of fixed cycle interrupts are generated (1 Hz, 2 Hz, 64 Hz, 512 Hz, and 8192 Hz) by using a counter which is used for the alarm generator.

This section is constituted as follows.

3.15.1 Block Diagram

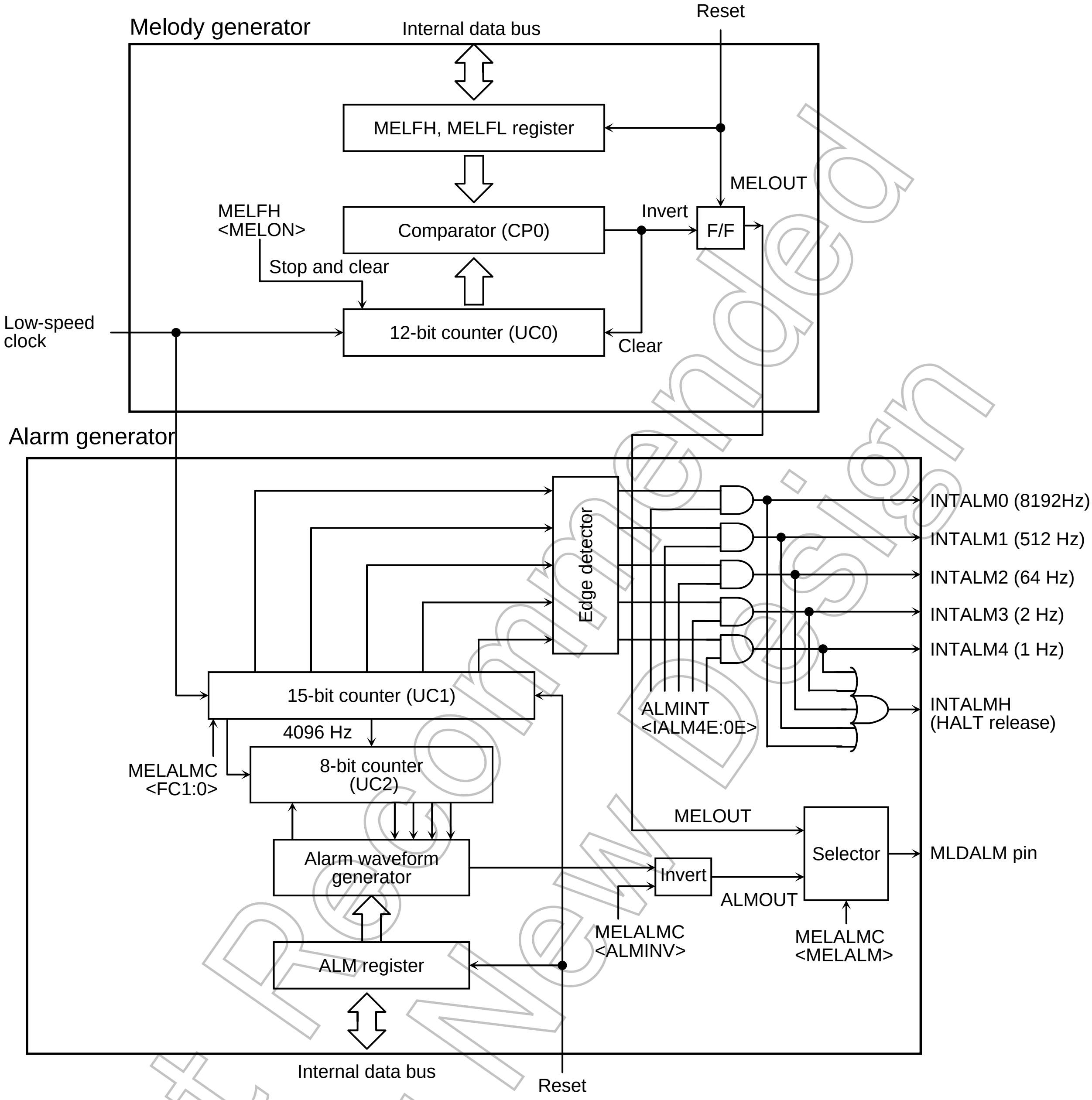


Figure 3.15.1 MLD Block Diagram

3.15.2 Control Registers

ALM Register

ALM (1330H)

	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	0	0	0	0	0	0	0
Function	Setting alarm pattern							

MELALMC Register

MELALMC (1331H)

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

Note 1: MELALMC<FC1> is always read “0”.

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

MELFL Register

MELFL (1332H)

	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	0	0	0	0	0	0	0
Function	Setting melody frequency (Lower 8 bits)							

MELFH Register

MELFH (1333H)

	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	0	0	0
Function	Control melody counter 0: Stop and clear 1: Start				Setting melody frequency (Upper 4 bits)			

ALMINT Register

ALMINT (1334H)

	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	0	0	0	0
Function			Always write “0”	1: Interrupt enable for INTALM4 to INTALM0				

3.15.3 Operational Description

3.15.3.1 Melody Generator

The Melody function generates signals of any frequency (4 Hz to 5461 Hz) based on a low-speed clock (32.768 kHz) and outputs the signals from the MLDALM pin.

The melody tone can easily be heard by connecting an external loud speaker.

(Operation)

MELALMC<MELALM> must first be set as 1 in order to select the melody waveform to be output from MLDALM. The melody output frequency must then be set to 12-bit registers MELFH and MELFL.

The following are examples of settings and calculations of melody output frequency.

(Formula for calculating of melody waveform frequency)

at fs = 32.768 [kHz]

Melody output waveform	f_{MLD} [Hz] = 32768/(2 × N + 4)
Setting value for melody	N = (16384/ f_{MLD}) − 2

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

(Example program)

When outputting an "A" musical note (440 Hz)

```
LD    (MELALMC), – – X X X X X 1 B   ; Select melody waveform
LD    (MELFL), 23H                   ; N = 16384/440 − 2 = 35.2 = 023H
LD    (MELFH), 80H                   ; Start to generate waveform
```

(Reference: 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 modulation frequency of 4096 Hz determined by the low-speed clock (32.768 kHz). This waveform is reversible by setting a value to a register.

The alarm tone can easily be heard by connecting an external loud speaker .

Five kinds of fixed cycle (interrupts can be generated 1 Hz, 2 Hz, 64 Hz, 512 Hz, 8 192 Hz) by using a counter which is used for the alarm generator.

(Operation)

MELALMC<MELALM> must first be set as 0 in order to select the alarm waveform to be output from MLDALMC. The “10” must be set on the MELALMC <FC1:0> register, and clear internal counter.

Finally the alarm pattern must then be set on the 8-bit register of ALM. If it is inverted output-data, set <ALMINV> as invert.

The following are examples of program, setting value of alarm pattern and waveform of each setting value.

(Setting Value of Alarm Pattern)

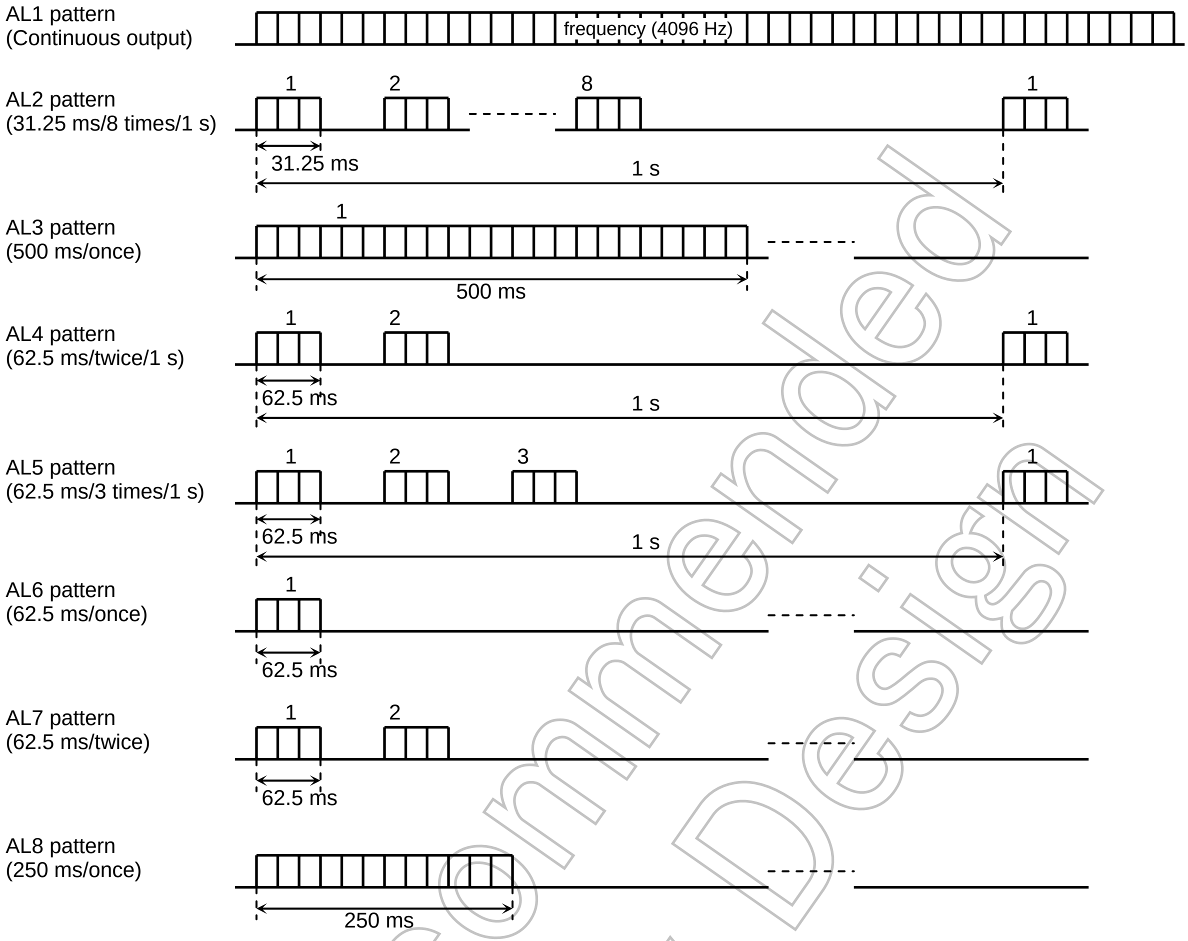
Setting Value for ALM Register	Alarm Waveform
00H	“0” fixed
01H	AL1 pattern
02H	AL2 pattern
04H	AL3 pattern
08H	AL4 pattern
10H	AL5 pattern
20H	AL6 pattern
40H	AL7 pattern
80H	AL8 pattern
Other	Undefined (do not set)

(Example program)

When outputting AL2 pattern (31.25 ms/8 times/1 s)

```
LD    (MELALMC), C0H    ;  Set output alarm waveform
                        ;  Free-run counter start
LD    (ALM), 02H        ;  Set AL2 pattern, start
```

Example: Waveform of alarm pattern for each setting value: Not inverted



3.16 SDRAM Controller (SDRAMC)

TMP92C820 includes SDRAM controller which supports SDRAM access by CPU/LCDC. The features are as follows.

(1) Support SDRAM

16-M/64-M/128-Mbit SDRAM (× 16 bits × 2/4 banks)

64-M/128-Mbit SDRAM (× 32 bits × 4 banks)

(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 (200 ns at f_{SYS} =20 MHz)

  Write cycle: 3 states (150 ns at f_{SYS} =20 MHz)

  Access data width: 8 bits/16 bits/32 bits

- LCDC burst access (Read only)

  Read cycle: 1 state (50 ns at f_{SYS} =20 MHz)

  Over head: 4 states (200 ns at f_{SYS} =20 MHz)

  Access data width: 16 bits/32 bits

(5) Refresh cycle auto generate

- Auto refresh is generated while another area is being accessed.
- Refresh interval is programmable.
- Self refresh is supported

Notes:

- Display data for LCDC must be set from the head of each page.
- Program is not operated on SDRAM.
- Condition of SDRAM's area is set by CS1 setting of Memory Controller.

3.16.1 Control Registers

Figure 3.16.1 shows the SDRAMC control registers. Setting these registers controls the operation of SDRAMC.

SDRAM Access Control Register

SDACR (0250H)

	7	6	5	4	3	2	1	0
Bit symbol	SDINI		SDBUS1	SDBUS0		SMUXW1	SMUXW0	SMAC
Read/Write	R/W		R/W			R/W		R/W
After reset	0		0	0		0	0	0
Function	Auto initialize 0: Disable 1: Enable		Selecting structure of data bus 00: 16 bits × 1 01: 16 bits × 2 10: 32 bits × 1			Selecting address multiplex type 00: Type A 01: Type B 10: Type C 11: Reserved		SDRAM controller 0: Disable 1: Enable

SDRAM Refresh Control Register

SDRCR (0251H)

	7	6	5	4	3	2	1	0
Bit symbol	SFRC	SRS2	SRS1	SRS0	SASFRC			SRC
Read/Write	R/W	R/W			R/W			R/W
After reset	0	0	0	0	0			0
Function	Self refresh 0: Disable 1: Enable	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			Interval refresh 0: Disable 1: Enable

Figure 3.16.1 SDRAMC Control Registers

3.16.2 Operation Description

(1) Memory access control

Access control block is enabled when SDACR<SMAC> = 1. And then SDRAM control signals (SDCSL, SDCSH, SDRAS, SDCAS, SDWE, SDLLDQM, SDLUDQM, SDULDQM, SDUUDQM, SDCLK and SDCKE) are operating during the time CPU or LCDC accesses CS1 area.

1. Address multiplex function

In the access cycle, address multiplex outputs row/column address through A1 to A15 pin. And multiplex width is decided by setting SDACR<SMUXW0:1> of use memory size. The relation between multiplex width and memory sizeRow/Column address is below.

Table 3.16.1 Address Multiplex

92C820 Pin Name	Address of SDRAM Accessing Cycle			
	Column Address	Row Address		
		Type A SDACR <SMUXW> = "00"	Type B SDACR <SMUXW> = "01"	Type C SDACR <SMUXW> = "10"
A0	A0	A0	A0	A0
A1	A1	A9	A10	A11
A2	A2	A10	A11	A12
A3	A3	A11	A12	A13
A4	A4	A12	A13	A14
A5	A5	A13	A14	A15
A6	A6	A14	A15	A16
A7	A7	A15	A16	A17
A8	A8	A16	A17	A18
A9	A9	A17	A18	A19
A10	A10	A18	A19	A20
A11	A11	A19	A20	A21
A12	A12	A20	A21	A22
A13	A13	A21	A22	A23
A14	A14	A22	A23	A14
A15	A15	A23	A15	A15

2. Burst length

SDRAM access by CPU is performed by the 1-word burst mode. And SDRAM access by LCDC is performed by the full-page burst mode.

SDRAM access cycle is shown in Figure 3.16.2 to Figure 3.16.3.

SDRAM accessing cycle number is depending on B1CSL register setting. For read cycle, setting of 4 states is necessary (B1CSL<B1WRn>). For write cycle, setting of 3 states is necessary (B1CSL<B1WWn>).

In the burst read cycle by LCDC, a mode setup and a pre-charge cycle are automatically inserted in a read cycle front and back.

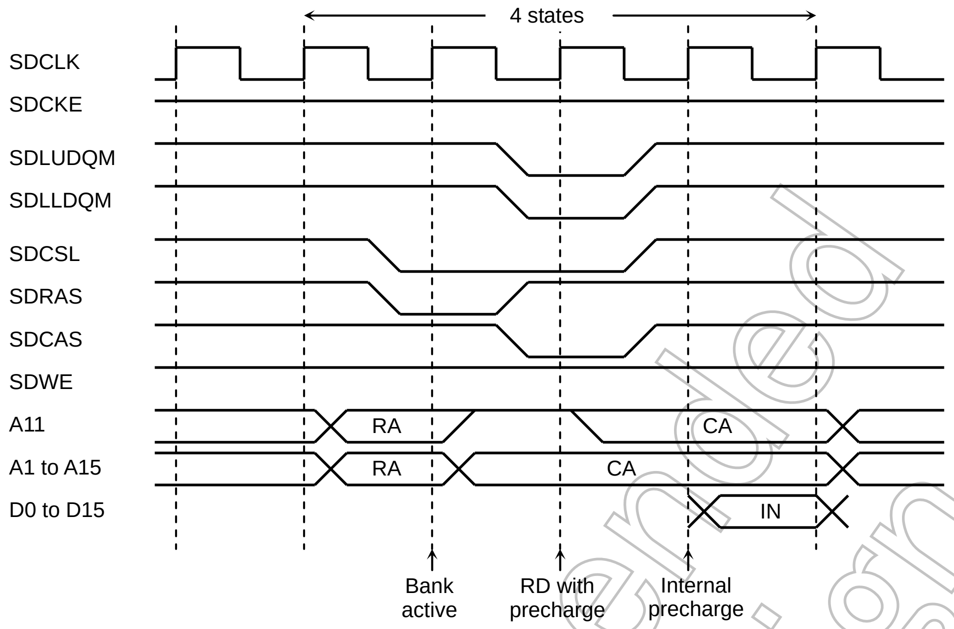


Figure 3.16.2 Timing of CPU Read Cycle

(Structure of data bus: 16 bits × 1, Operand size: 2 bytes, Address: 2 n + 0)

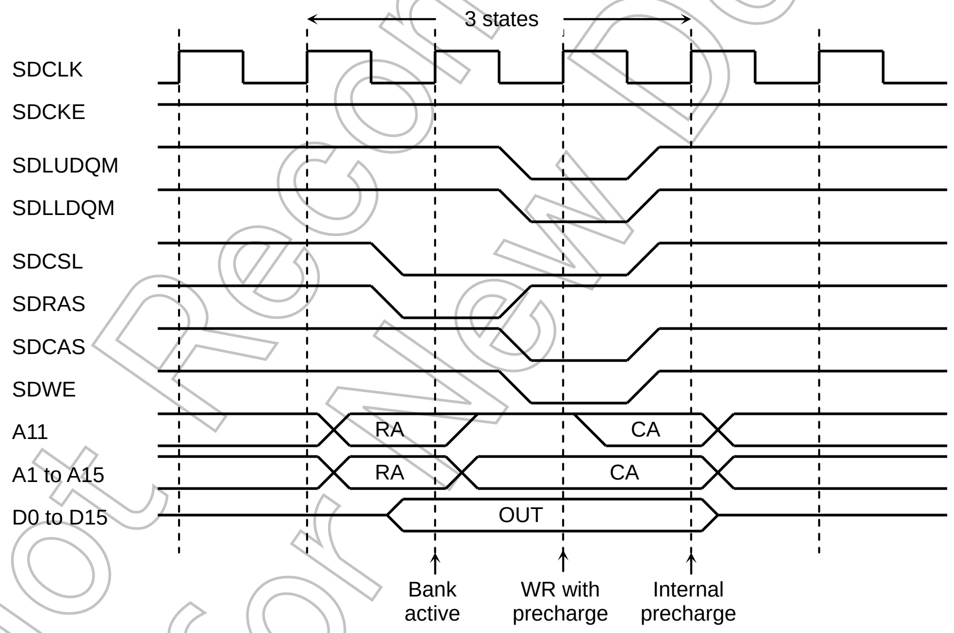


Figure 3.16.3 Timing of CPU Write Cycle

(Structure of data bus: 16 bits × 1, Operand size: 2 bytes, Address: 2 n + 0)

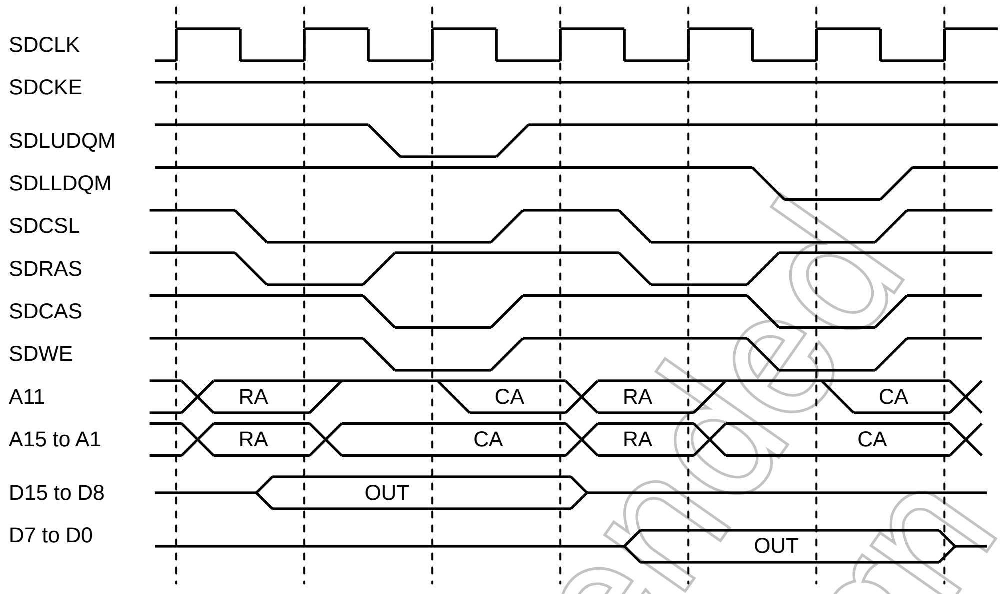


Figure 3.16.4 Timing of CPU Write Cycle

(Structure of data bus: 16 bits × 1, Operand size: 2 bytes, Address: 2 n + 1)

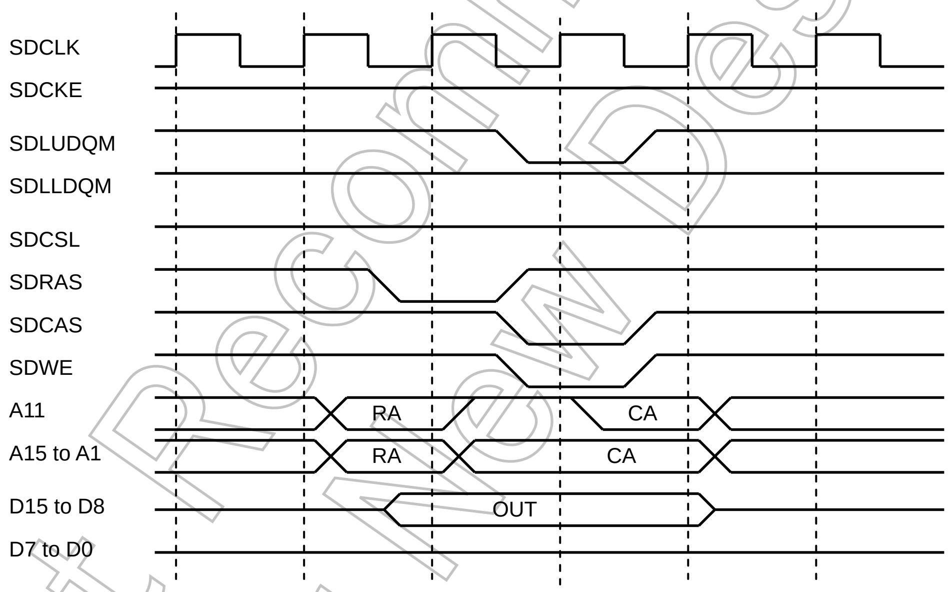


Figure 3.16.5 Timing of CPU Write Cycle

(Structure of data bus: 16 bits × 1, Operand size: 1 byte, Address: 2 n + 1)

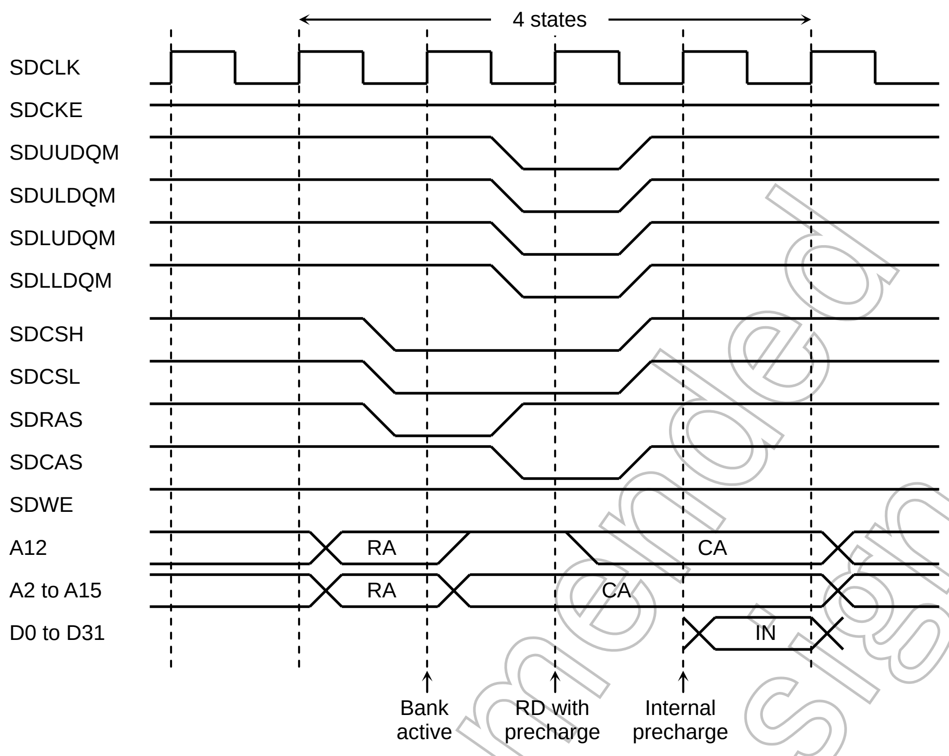


Figure 3.16.6 Timing of CPU Read Cycle

(Structure of data bus: 16 bits × 2 = 32 bits, Operand size: 4 bytes, Address: 4 n + 0)

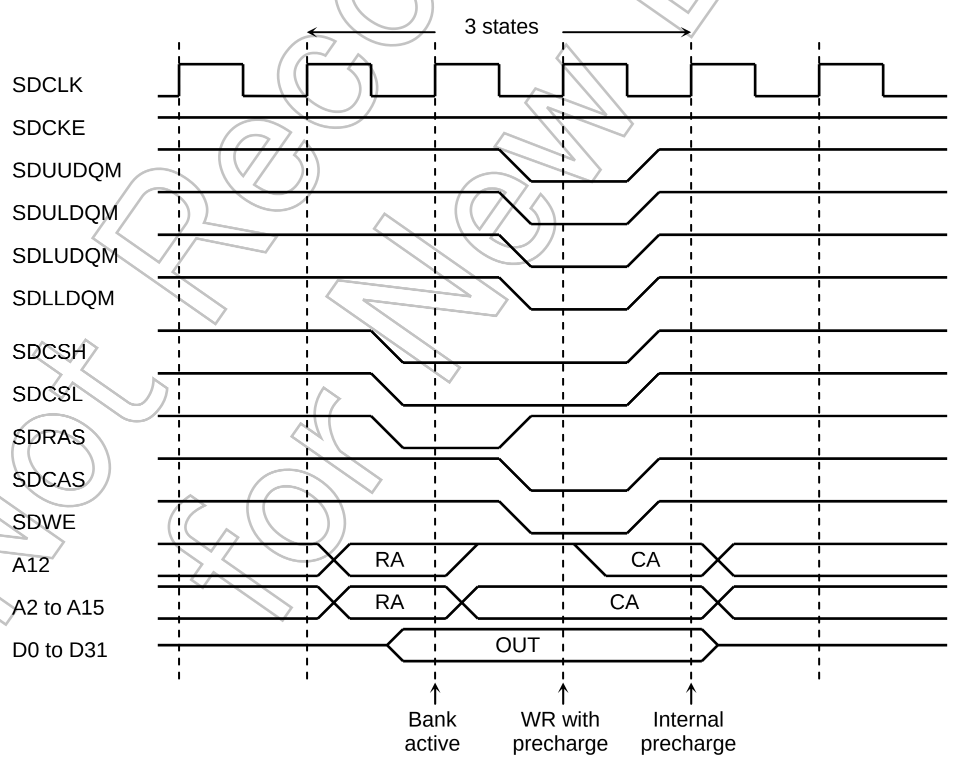


Figure 3.16.7 Timing of CPU Write Cycle

(Structure of data bus: 16 bits × 2 = 32 bits, Operand size: 4 bytes, Address: 4 n + 0)

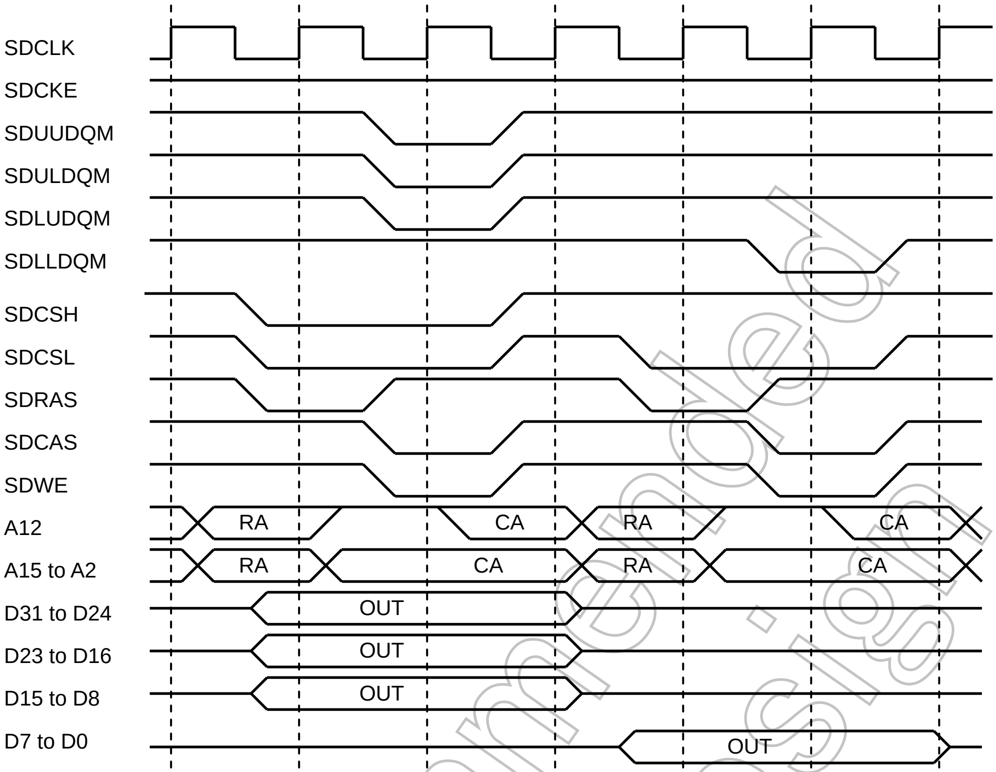


Figure 3.16.8 Timing of CPU Write Cycle

(Structure of data bus: 16 bits × 2 = 32 bits, Operand size: 4 bytes, Address: 4 n + 1)

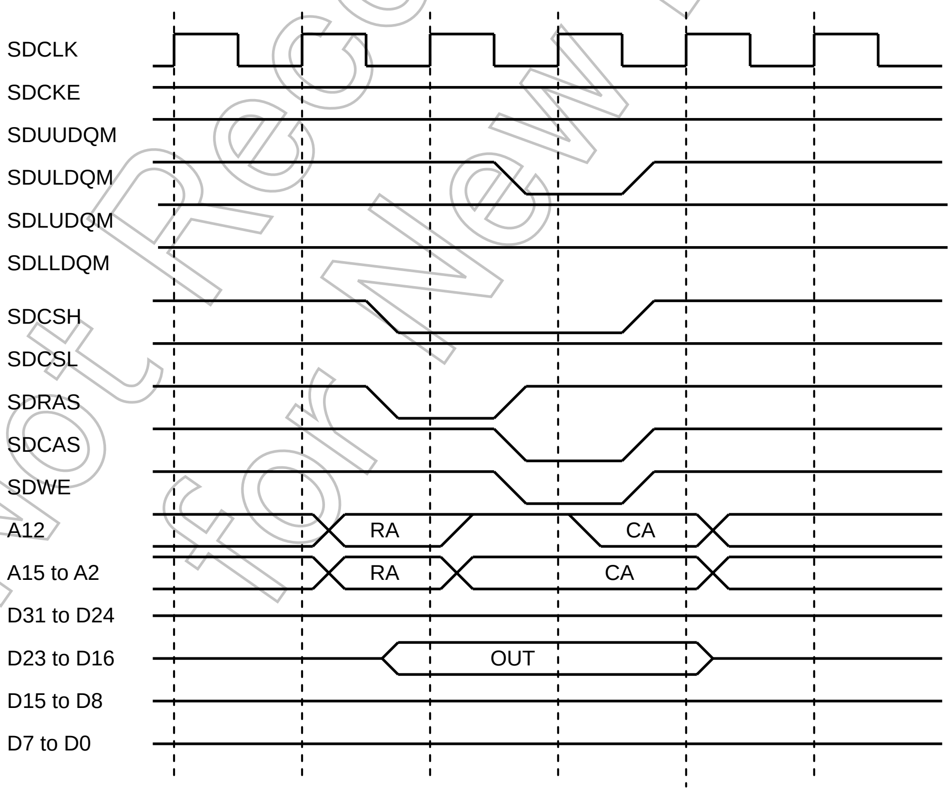


Figure 3.16.9 Timing of CPU Write Cycle

(Structure of data bus: 16 bits × 2 = 32 bits, Operand size: 8 bytes, Address: 4 n + 3)

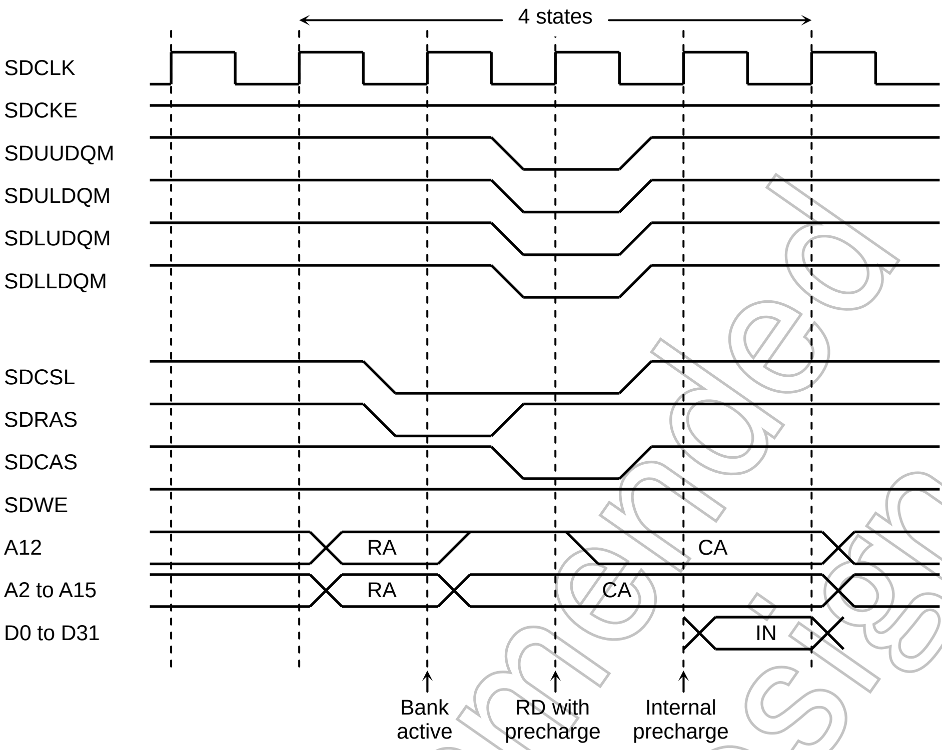


Figure 3.16.10 Timing of CPU Read Cycle

(Structure of data bus: 32 bits × 1, Operand size: 4 bytes, Address: 4 n + 0)

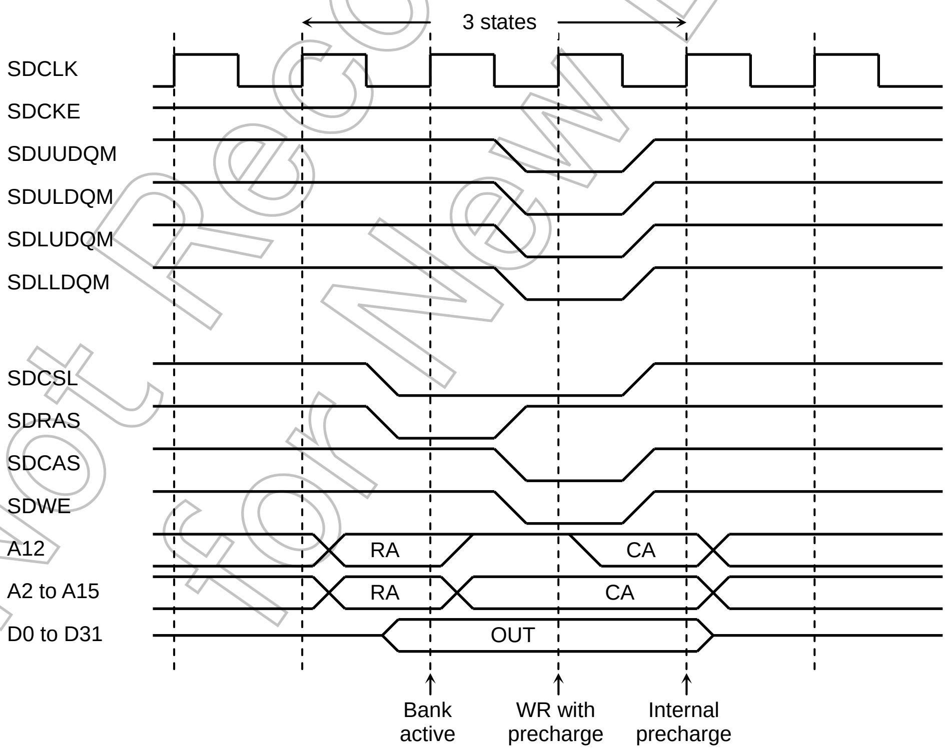


Figure 3.16.11 Timing of CPU Write Cycle

(Structure of data bus: 32 bits × 1, Operand size: 4 bytes, Address: 4 n + 0)

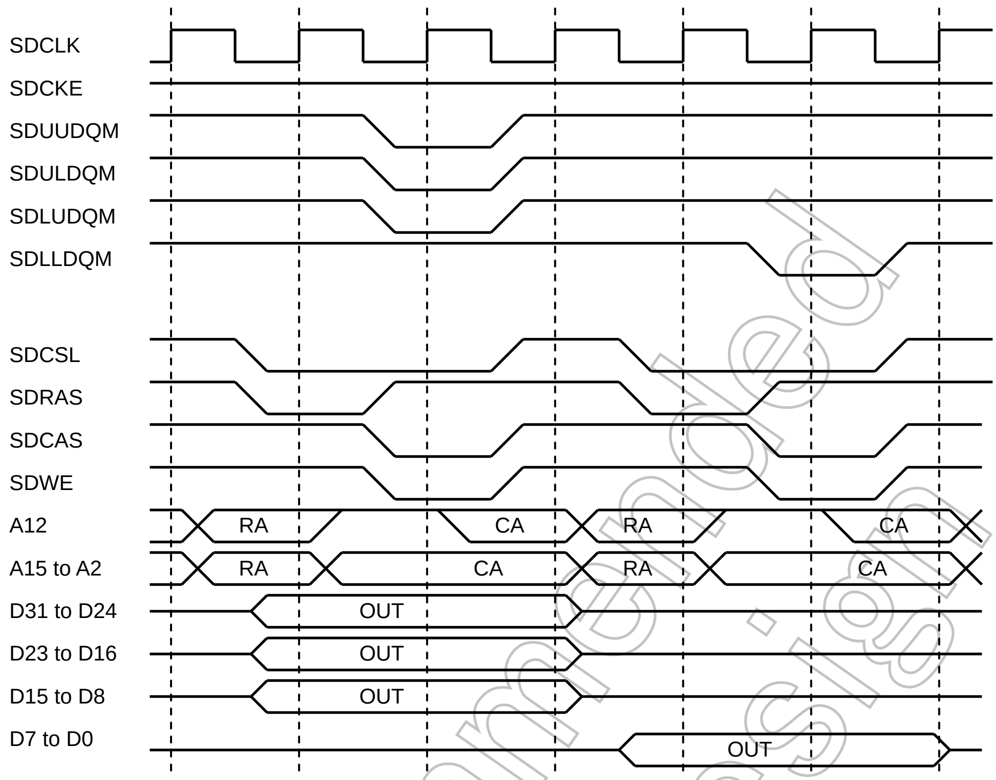


Figure 3.16.12 Timing of CPU Write Cycle

(Structure of data bus: 32 bits × 1, Operand size: 4 bytes, Address: 4 n + 1)

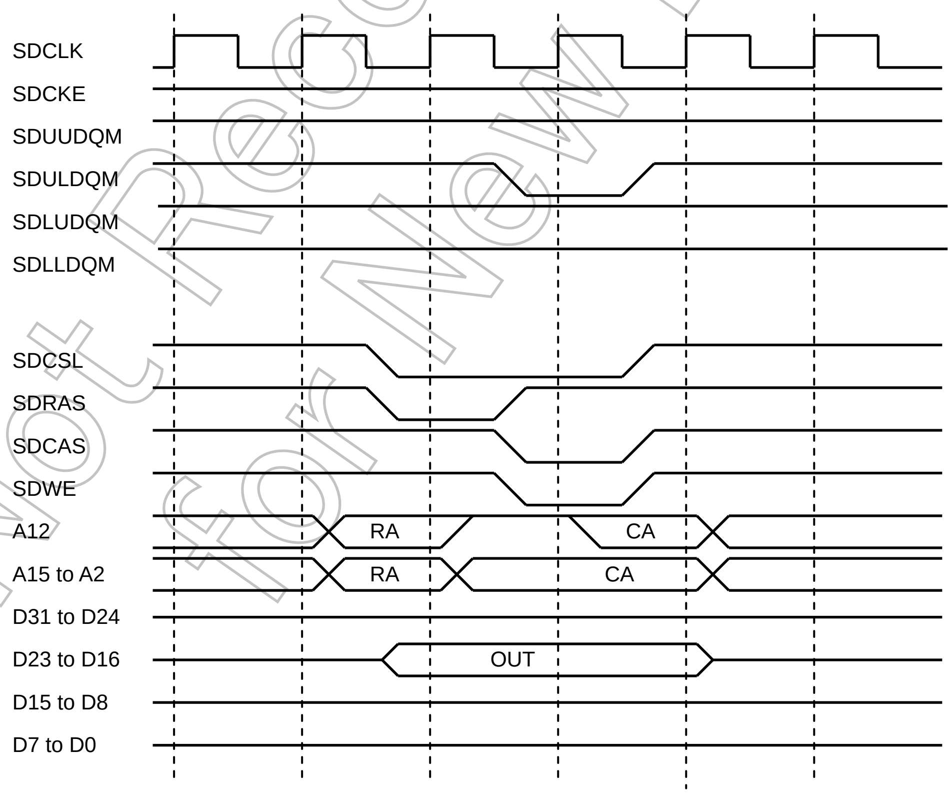


Figure 3.16.13 Timing of CPU Write Cycle

(Structure of data bus: 32 bits × 1, size: 8 bytes, Address: 4 n + 3)

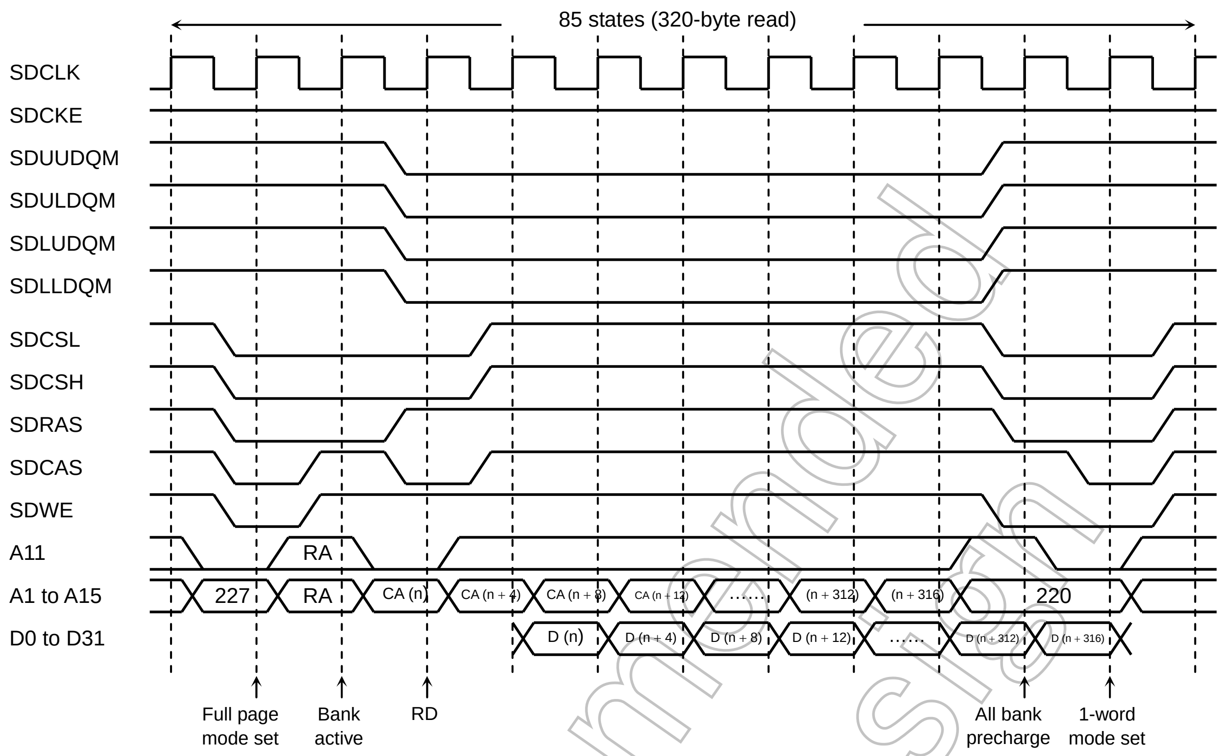


Figure 3.16.14 Timing of LCDC Burst Read Cycle

(2) Refresh control

TMP92C820 can generate automatically an auto-refresh cycle for data maintenance of SDRAM. Auto-refresh cycle is generated by setting SDRCR<SRC> to "1". Interval of auto refresh can be set by SDRCR<SRS0:2> from the 78 states to the 312 states (3.9 μs to 15.6 μs at 20 MHz).

The generating timing of an auto-refresh cycle becomes into accessing cycles other than SDRAM area (CS1). The auto-refresh cycle is shown in Figure 3.16.15 moreover, the interval of auto refresh is shown in Table 3.16.2.

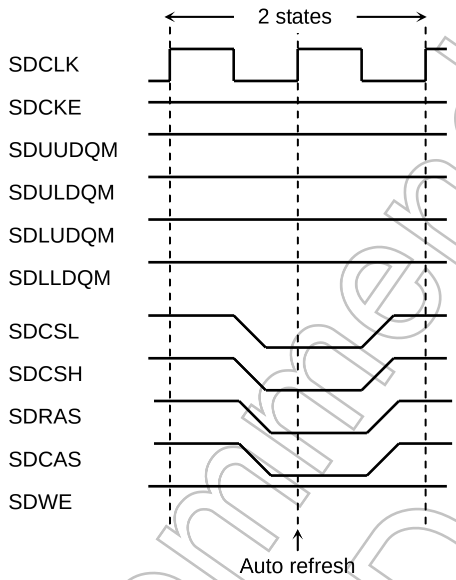


Figure 3.16.15 Timing of Auto-refresh Cycle

Table 3.16.2 Refresh Cycle Insertion Interval

(Unit: μs)

SDRCR<SRS0:2>			Insertion Interval (State)	f_{SYS} Frequency (System clock)					
SRS2	SRS1	SRS0		5 MHz	10 MHz	12.5 MHz	15 MHz	17.5 MHz	20 MHz
0	0	0	78	15.6	7.8	6.2	5.2	4.5	3.9
0	0	1	97	19.4	9.7	7.8	6.5	5.5	4.9
0	1	0	124	24.8	12.4	9.9	8.3	7.1	6.2
0	1	1	156	31.2	15.6	12.5	10.4	8.9	7.8
1	0	0	195	39.0	19.5	15.6	13.0	11.1	9.8
1	0	1	210	42.0	21.0	16.8	14.0	12.0	10.5
1	1	0	247	49.4	24.7	19.8	16.5	14.1	12.4
1	1	1	312	62.4	31.2	25.0	20.8	17.8	15.6

It does not generate an auto-refresh cycle during the burst access to SDRAM by LCDC. The demand of auto-refresh cycle is held in this period. When it returns to CPU access cycle, an auto-refresh cycle is generated.

Furthermore, TMP92C820 can to generate a self-refresh cycle. The timing of a self-refresh cycle is shown in Figure 3.16.16.

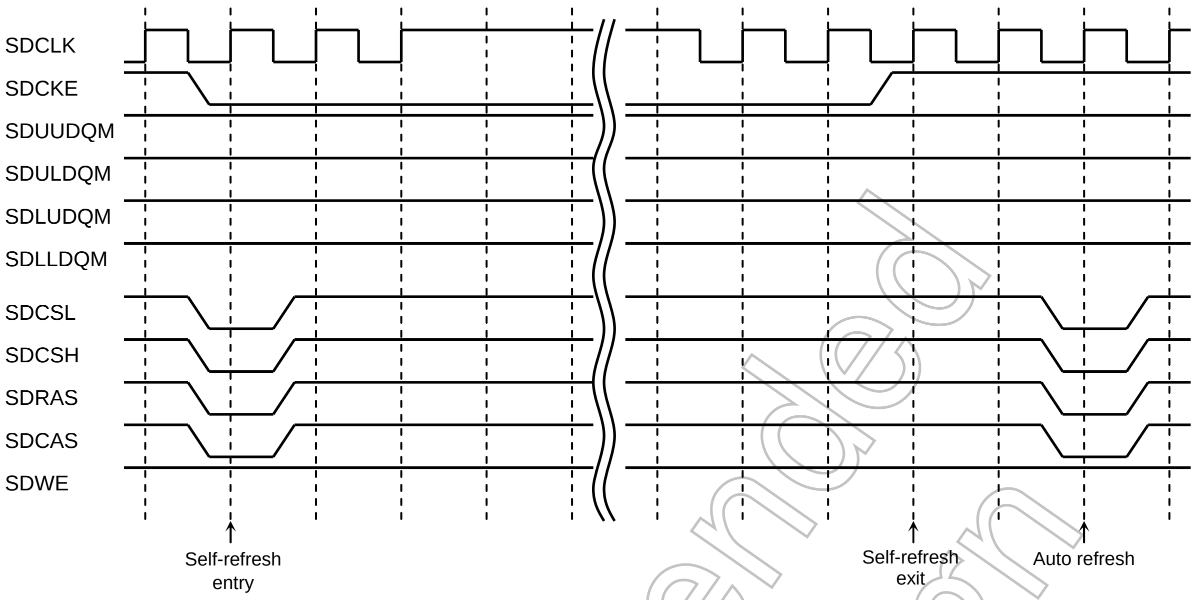


Figure 3.16.16 Timing of Self-refresh Cycle

Note 1: When IDLE2 mode, continue with output clock. Therefore, If want to stop SDCLK, switch PF6 to output port before execution HALT instruction.

Note 2: Pin condition in IDLE1/STOP mode depends on SYSCR2<DRVE> setting. However, when self-refresh mode, pin don't depend on SYSCR2<DRVE>, and output low level.

If SDRCR<SFRC> is set to "1", the self-refresh cycle shown in Figure 3.16.16 will occur. The self-refresh mode is used when using the standby mode (STOP, IDLE1), which an internal clock stops. In the case of standby mode using self refresh, please set SDRCR<SFRC> to "1", before HALT instruction (STOP, IDLE1).

Release of a self-refresh cycle is automatically performed by release in the standby mode. It inserts automatically one auto refresh after self refresh is released, and returns to the auto refresh mode.

Note: When standby mode is cancelled by a reset, the I/O registers are initialized, therefore, auto refresh is not performed.

Please do not place the command which accesses SDRAM, just before setting SDRCR<SFRC> to "1". After setting SDRCR<SFRC> to "1", at least 4 times of "NOP (s)" are required before halt command execution.

Example:

```
SET 7, (SDRCR)
NOP  \
NOP   |
NOP   |  * at least 4 times NOP(s).
NOP  /
HALT
```

(3) SDRAM initialize

TMP92C820 can generate the following SDRAM initialize routine after injection power-supply to SDRAM. The cycle is shown in Figure 3.16.17.

1. Precharge of all banks
2. The initial configuration to a mode register
3. The auto-refresh cycle of 8 cycles

The above cycle is generated by setting SDACR<SDINI> to "1".

While performing this cycle, operation (an instruction fetch, command execution) of CPU is stopped.

In addition, before performing an initialization cycle, a port needs to be set as SDRAM control signal and an address signal (A1 to A12). After the initialization cycle is finished, SDACR<SDINI> is cleared to "0" automatically.

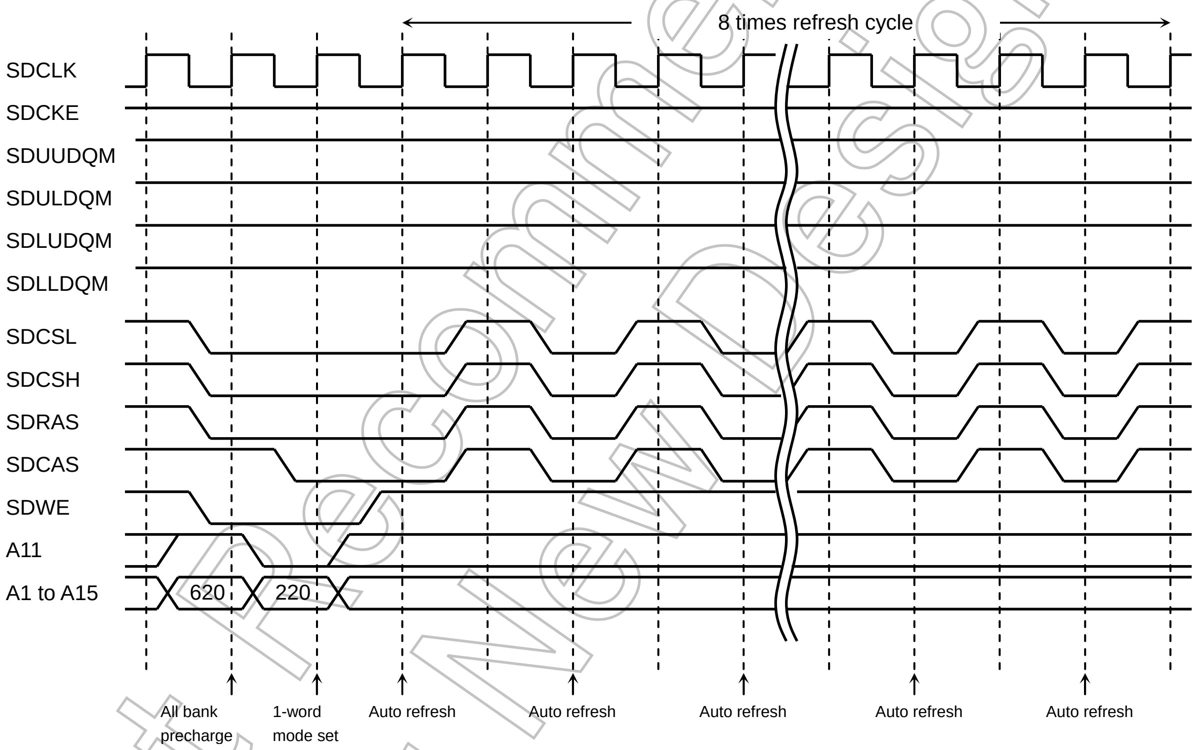


Figure 3.16.17 Timing of Initialization Cycle

(4) Connection example

The example of connection with SDRAM is shown in Figure 3.16.18 to Figure 3.16.20.

Table 3.16.3 Connection with SDRAM

TMP92C820 Pin Name	SDRAM Pin Name								
	Data Bus Width 16 Bits			Data Bus Width 32 Bits					
	16 Mbits	64 Mbits	128 Mbits	16 Mbits × 16 Bits × 2		64 Mbits × 16 Bits × 2		64 Mbits × 32 Bits	128 Mbits × 32 Bits
A0	–	–	–	–	–	–	–	–	–
A1	A0 (A9)	A0 (A9)	A0 (A10)	–	–	–	–	–	–
A2	A1 (A10)	A1 (A10)	A1 (A11)	A0 (A10)	A0 (A10)	A0 (A10)	A0 (A10)	A0 (A10)	A0 (A10)
A3	A2 (A11)	A2 (A11)	A2 (A12)	A1 (A11)	A1 (A11)	A1 (A11)	A1 (A11)	A1 (A11)	A1 (A11)
A4	A3 (A12)	A3 (A12)	A3 (A13)	A2 (A12)	A2 (A12)	A2 (A12)	A2 (A12)	A2 (A12)	A2 (A12)
A5	A4 (A13)	A4 (A13)	A4 (A14)	A3 (A13)	A3 (A13)	A3 (A13)	A3 (A13)	A3 (A13)	A3 (A13)
A6	A5 (A14)	A5 (A14)	A5 (A15)	A4 (A14)	A4 (A14)	A4 (A14)	A4 (A14)	A4 (A14)	A4 (A14)
A7	A6 (A15)	A6 (A15)	A6 (A16)	A5 (A15)	A5 (A15)	A5 (A15)	A5 (A15)	A5 (A15)	A5 (A15)
A8	A7 (A16)	A7 (A16)	A7 (A17)	A6 (A16)	A6 (A16)	A6 (A16)	A6 (A16)	A6 (A16)	A6 (A16)
A9	A8 (A17)	A8 (A17)	A8 (A18)	A7 (A17)	A7 (A17)	A7 (A17)	A7 (A17)	A7 (A17)	A7 (A17)
A10	A9 (A18)	A9 (A18)	A9 (A19)	A8 (A18)	A8 (A18)	A8 (A18)	A8 (A18)	A8 (A18)	A8 (A18)
A11	A10 (A19)	A10 (A19)	A10 (A20)	A9 (A19)	A9 (A19)	A9 (A19)	A9 (A19)	A9 (A19)	A9 (A19)
A12	BS (A20)	A11 (A20)	A11 (A21)	A10 (A20)	A10 (A20)	A10 (A20)	A10 (A20)	A10 (A20)	A10 (A20)
A13	–	BS0 (A21)	BS0 (A22)	BS (A21)	BS (A21)	A11 (A21)	A11 (A21)	BS0 (A21)	A11 (A21)
A14	–	BS1 (A22)	BS1 (A23)	–	–	BS0 (A22)	BS0 (A22)	BS1 (A22)	BS0 (A22)
A15	–	–	–	–	–	BS1 (A23)	BS1 (A23)	–	BS1 (A23)
SDCSH	–	–	–	CS	–	CS	–	–	–
SDCSL	CS	CS	CS	–	CS	–	CS	CS	CS
SDUUDQM	–	–	–	UDQM		UDQM		DQM3	DQM3
SDULDQM	–	–	–	LDQM		LDQM		DQM2	DQM2
SDLUDQM	UDQM	UDQM	UDQM		UDQM		UDQM	DQM1	DQM1
SDLLDQM	LDQM	LDQM	LDQM		LDQM		LDQM	DQM0	DQM0
SDRAS	RAS	RAS	RAS	RAS	RAS	RAS	RAS	RAS	RAS
SDCAS	CAS	CAS	CAS	CAS	CAS	CAS	CAS	CAS	CAS
SDWE	WE	WE	WE	WE	WE	WE	WE	WE	WE
SDCKE	CKE	CKE	CKE	CKE	CKE	CKE	CKE	CKE	CKE
SDCLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK	CLK
SDACR <SDBUS>	00: 16 bits × 1	00: 16 bits × 1	00: 16 bits × 1	01: 16 bits × 2		01: 16 bits × 2		10: 32 bits × 1	10: 32 bits × 1
SDACR <SMUXW>	00: Type A	00: Type A	01: Type B	00: Type A		00: Type A		00: Type A	00: Type A

(An): Row address

(shaded) : Command address pin of SDRAM

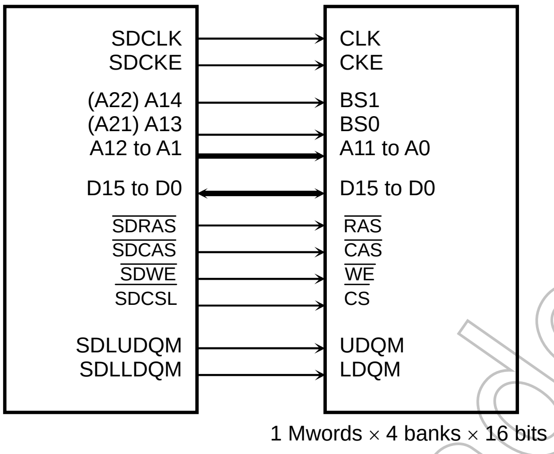


Figure 3.16.18 Connection with SDRAM (1 Mwords × 16 bits)

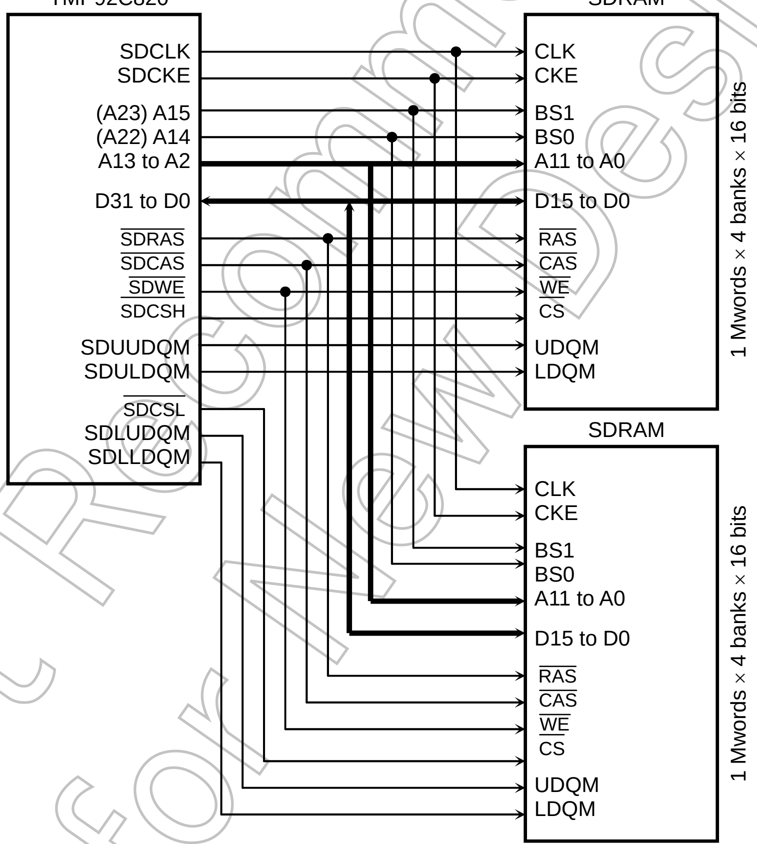


Figure 3.16.19 Connection with SDRAM (1 Mwords × 16 bits × 2)

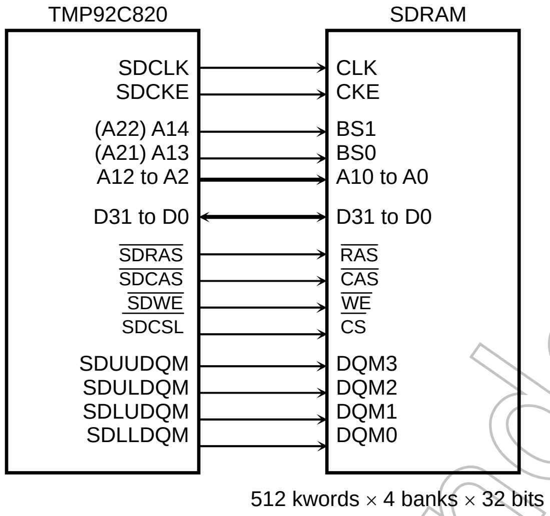


Figure 3.16.20 Connection with SDRAM (512 kwords × 32 bits)

(5) Limitation point for SDRAM

There are some points to notice when using SDRAMC. Please refer to the section under below and take care.

1) WAIT access

When using SDRAM, it is added some limitation of access to all other memories.

Under the WAIT pin input setting of Memory Controller, it is prohibited inserting the time over (14 × refresh interval time; in Auto Refresh function controlled by SDRAM controller).

2) Execution of SDRAM command before HALT instruction (SR(Self refresh)-Entry , Initialize , Mode-set)

It requires execution time (a few states) to execute the command that SDRAMC has (SR- Entry, Initialize).

Therefore when executing HALT instruction after the SDRAM command, please insert over 10 bytes NOP or other 10 bytes instructions before HALT instruction.

3) AR (Auto Refresh) interval time

When using SDRAM, system clock frequency must be set suitable speed for SDRAM's specification that is minimum operating clock and minimum Refresh interval time.

When using SDRAM under slow mode or down the Clock Gear, please design the system with special care for Auto Refresh interval time.

And please set Auto Refresh interval time after adding 10 states to distributed Auto Refresh interval time, because it might not meet the A.C specification of SDRAM by stopping Auto Refresh.

(Example of calculation)

Condition:

f_{SYS} = 20MHz, SDRAM specification of distributed Auto Refresh interval time = 4096 times/64 ms

64ms/ 4096 times = 15.625μs/1 time = 312.5state/1 time

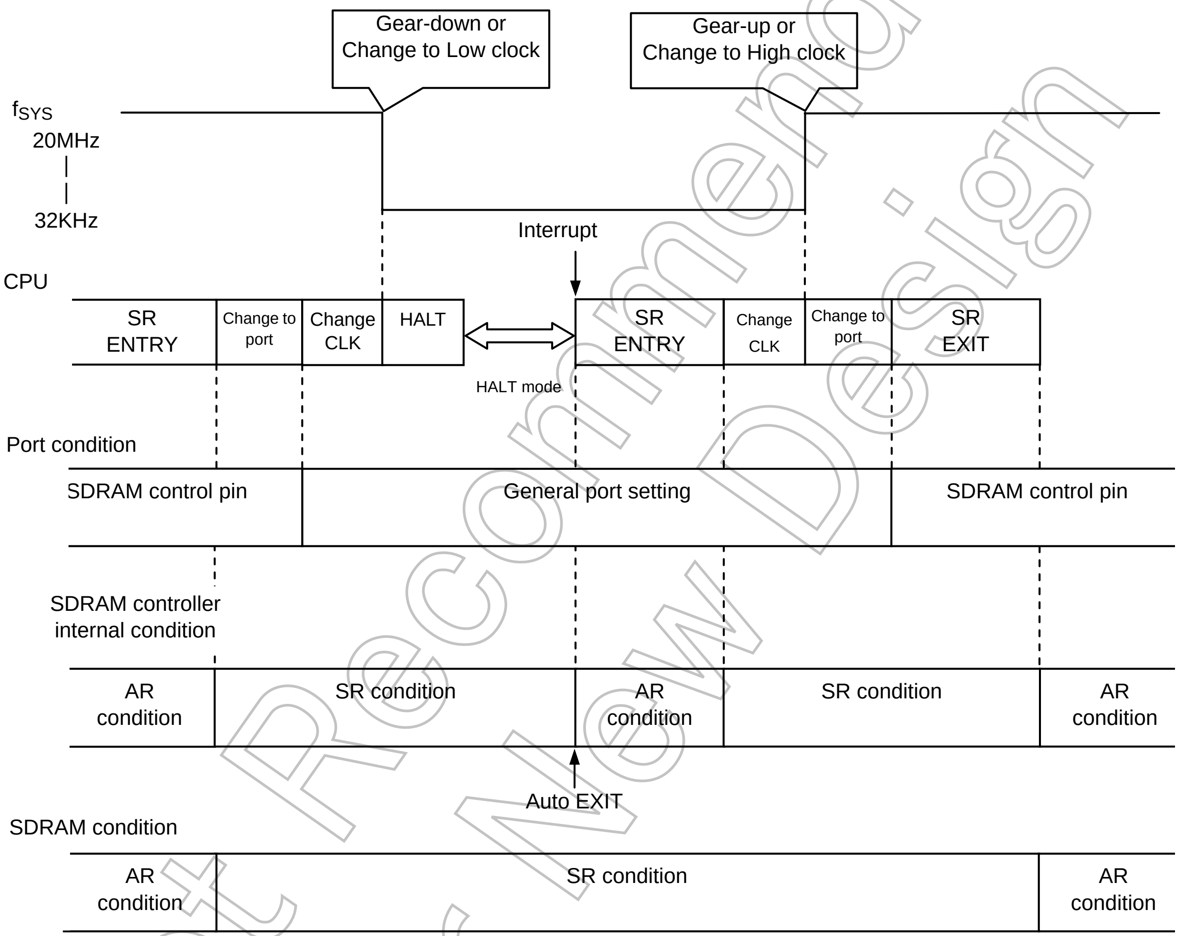
312.5 – 10 = 302.5 state/less than 1 time is needed → 247 state is needed

4) Auto Exit problem when exiting from Self Refresh Mode of SDRAM

When using Self Refresh function together with stand-by function of CPU or changing clock, it might not be suit specification of SDRAM. Because automatic releasing Self Refresh function (Auto Exit function) operates by CPU releasing HALT mode.

Following figure shows example for avoid this problem by S/W.

(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 TMP92C820 incorporates one multifunctional 16-bit timer/event counter (TMRB0) which have the following operation modes:

- 16-bit interval timer mode
- 16-bit event counter mode
- 16-bit programmable pulse generation (PPG) mode

Timer/event counter consists of a 16-bit up counter, two 16-bit timer registers (one of them with a double-buffer structure), a 16-bit capture registers, two comparators, a capture input 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) output mode

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 PC6)
SFR (Address)	Timer run register	TB0RUN (1180H)
	Timer mode register	TB0MOD (1182H)
	Timer flip-flop control register	TB0FFCR (1183H)
	Timer register	TB0RG0L (1188H) TB0RG0H (1189H) TB0RG1L (118AH) TB0RG1H (118BH)
	Capture register	TB0CP0L (118CH) TB0CP0H (118DH) TB0CP1L (118EH) TB0CP1H (118FH)

3.17.1 Block Diagram

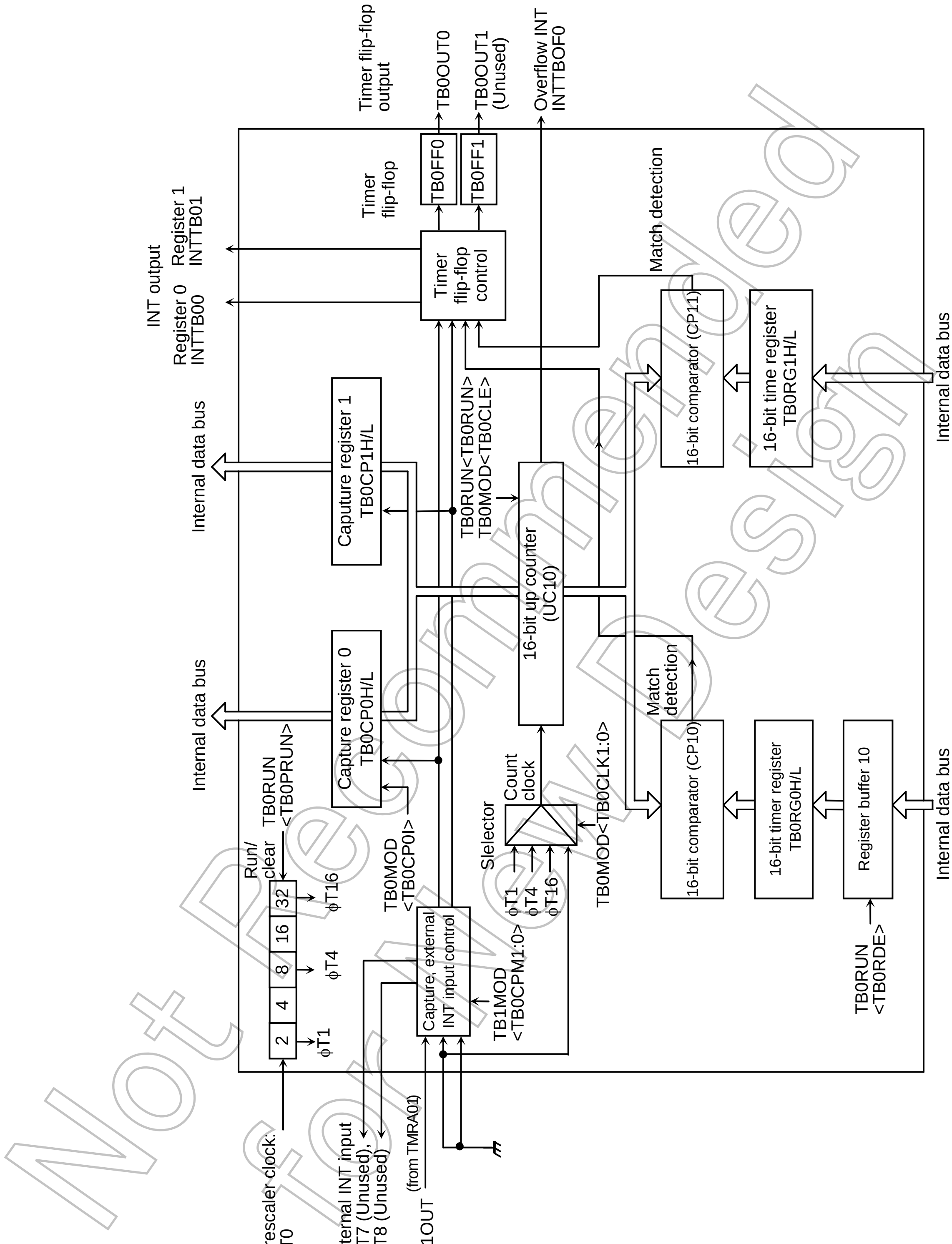


Figure 3.17.1 Block Diagram of TMRB0

3.17.2 Operation

(1) Prescaler

The 5-bit prescaler generates the source clock for timer 0. The prescaler clock (φT0) is divided clock (Divided by 8) from the f_{FPH}.

This prescaler can be started or stopped using TB0RUN<TB0PRUN>. Counting starts when <TB0PRUN> is set to 1; the prescaler is cleared to 0 and stops operation when <TB0PRUN> is cleared to 0.

Table 3.17.2 Prescaler Clock Resolution

Clock gear selection SYSCR1 <GEAR2:0>	System clock selection SYSCR1 <SYSCK>	–	Timer counter input clock TMRB prescaler TB0MOD<TB0CLK1:0>		
			φT1(1/2)	φT4(1/8)	φT16(1/32)
–	1 (fs)	1/8	fs/16	fs/64	fs/256
000 (1/1)	0 (fc)		fc/16	fc/64	fc/256
001 (1/2)			fc/32	fc/128	fc/512
010 (1/4)			fc/64	fc/256	fc/1024
011 (1/8)			fc/128	fc/512	fc/2048
100 (1/16)			fc/256	fc/1024	fc/4096

(2) Up counter (UC10)

UC10 is a 16-bit binary counter which counts up pulses input from the clock specified by TB0MOD<TB0CLK1:0>.

Any one of the prescaler internal clocks φT1, φT4 and φT16 or an external clock input via the TB0IN0 pin can be selected as the input clock. Counting or stopping and clearing of the counter is controlled by TB0RUN<TB0RUN>.

When clearing is enabled, the up counter UC10 will be cleared to 0 each time its value matches the value in the timer register TB0RG1H/L. If clearing is disabled, the counter operates as a free-running counter.

Clearing can be enabled or disabled using TB0MOD<TB0CLE>.

A timer overflow interrupt (INTTBOF0) is generated when UC10 overflow occurs.

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

These two 16-bit registers are used to set the interval time. When the value in the up counter UC10 matches the value set in this timer register, the comparator match detect signal will go active.

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

The TB0RG0H/L timer register has a double-buffer structure, which is paired with a register buffer. The value set in TB0RUN<TB0RDE> determines whether the double-buffer structure is enabled or disabled: It is disabled when <TB0RDE> = 0, and enabled when <TB0RDE> = 1.

When the double buffer is enabled, data is transferred from the register buffer to the timer register when the values in the up counter (UC10) and the timer register TB0RG1H/L match.

After a reset, TB0RG0H/L and TB0RG1H/L are undefined. If the 16-bit timer is to be used after a reset, data should be written to it beforehand.

On a reset <TB0RDE> is initialized to 0, disabling the double buffer. To use the double buffer, write data to the timer register, set <TB0RDE> to 1, then write data to the register buffer as shown below.

TB0RG0H/L and the register buffer both have the same memory addresses (001188H and 001189H) allocated to them. If <TB0RDE> = 0, the value is written to both the timer register and the register buffer. If <TB0RDE> = 1, the value is written to the register buffer only.

The addresses of the timer registers are as follows:

TMRB0

TB0RG0H/L		TB0RG1H/L	
Upper 8 bits (TB0RG0H)	Lower 8 bits (TB0RG0L)	Upper 8 bits (TB0RG1H)	Lower 8 bits (TB0RG1L)
1189H	1188H	118BH	118AH

Note: The timer registers are write-only registers and thus cannot be read.

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

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

All 16 bits of data in the capture registers should be read both Upper and Lower. For example, using a 2-byte data load instruction or two 1-byte data load instructions. The least significant byte is read first, followed by the most significant byte. The addresses of the capture registers are as follows:

TMRB0

TB0CP0H/L		TB0CP1H/L	
Upper 8 bits (TB0CP0H)	Lower 8 bits (TB0CP0L)	Upper 8 bits (TB0CP1H)	Lower 8 bits (TB0CP1L)
118DH	118CH	118FH	118EH

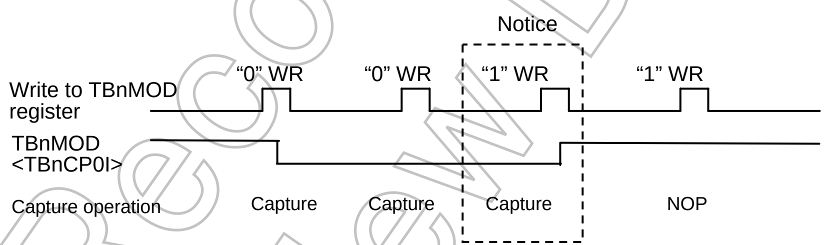
Note: The capture registers are read-only registers and thus cannot be written to.

(5) Capture input control

This circuit controls the timing to latch the value of the up counter UC10 into TB0CP0H/L, TB0CP1H/L.

The value in the up counter can be loaded into a capture register by software. Whenever 0 is written to TB0MOD<TB0CP0I>, the current value in the up counter is loaded into capture register TB0CP0H/L. It is necessary to keep the prescaler in Run Mode (i.e., TB0RUN<TB0PRUN> must be held at a value of 1).

Note: As described above, whenever 0 is programmed to TB0MOD<TB0CP0I>, the current value in the up counter is loaded into capture register TB0CP0H/L. However, note that the current value in the up counter is also loaded into capture register TB0CP0H/L when 1 is programmed to TB0MOD<TB0CP0I> while this bit is holding



0.

(6) Comparators (CP10 and CP11)

CP10 and CP11 are 16-bit comparators which compare the value in the up counter UC10 with the value set in TB0RG0 or TB0RG1 respectively, in order to detect a match. If a match is detected, the comparator generates an interrupt (INTTB00 or INTTB01 respectively).

(7) Timer flip-flops (TB0FF0)

These flip-flops are inverted by the match detect signals from the comparators and the latch signals to the capture registers. Inversion can be enabled and disabled for each element using TB0FFCR<TB0C0T1, TB0E1T1, TB0E0T1>. After a reset the value of TB0FF0 is undefined. If "00" is written to TB0FFCR<TB0FF0C1:0>, TB0FF0 will be inverted. If "01" is written to the capture registers, the value of TB0FF0 will be set to "1". If "10" is written to the capture registers, the value of TB0FF0 will be cleared to "0".

The values of TB0FF0 can be output via the timer output pin TB0OUT0 (which is shared with PC6). Timer output should be specified using the port C function register.

3.17.3 SFRs

TMRB0 Run Register

TB0RUN (1180H)

	7	6	5	4	3	2	1	0
Bit symbol	TB0RDE	–			I2TB0	TB0PRUN		TB0RUN
Read/Write	R/W	R/W			R/W	R/W		R/W
After reset	0	0			0	0		0
Function	Double buffer 0: Disable 1: Enable	Always write "0".			IDLE2 0: Stop 1: Operate	TMRB0 Prescaler 0: Stop and clear 1: Run (Count up)		Up counter (UC10) 0: Stop and clear 1: Run (Count up)

Count operation

0	Stop and clear
1	Count

Note: 1, 4, and 5 of TB0RUN are read as undefined values.

Figure 3.17.2 The Registers for TMRB

TMRB0 Mode Register

TB0MOD (1182H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	–	–	TB0CP0I	TB0CPM1	TB0CPM0	TB0CLE	TB0CLK1	TB0CLK0
Read/Write	R/W		W*	R/W				
After reset	0	0	1	0	0	0	0	0
Function	Always write “0”.		Execute software capture 0: Software capture 1: Undefined	Capture timing 00: Disable 01: Reserved 10: Reserved 11: TA1OUT↑ TA1OUT↓		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 (UC10) clear control

0	Disable
1	Enable clearing on match with TB0RG1H/L.

Capture/interrupt timing

	Capture control
00	Disable
01	Reserved
10	Reserved
11	Capture to TB0CP0H/L at rising edge of TA1OUT Capture to TB0CP1H/L at falling edge of TA1OUT

Software capture

0	The value in the up counter is captured to TB0CP0H/L.
1	Undefined (Note)

Note: Whenever programming “0” to TB0MOD<TB0CP0I> bit, present value of up counter is received to capture register TB0CP0H/L. But, program “1” to TB0MOD<TB0CP0I> in condition of programmed “0” to TB0MOD<TB0CP0I> bit, present value of up counter is received to capture register TB0CP0H/L. Therefore you must to regard.

Figure 3.17.3 The Registers for TMRB

TMRB0 Flip-Flop Control Register

TB0FFCR (1183H)

Prohibit read-modify-write

	7	6	5	4	3	2	1	0
Bit symbol	–	–	TB0C1T1	TB0C0T1	TB0E1T1	TB0E0T1	TB0FFC1	TB0FFC0
Read/Write	W		R/W				W*	
After reset	1	1	0	0	0	0	1	1
Function	Always write "11".		TB0FF0 inversion trigger 0: Disable trigger 1: Enable trigger				Control TB0FF0 00: Invert 01: Set 10: Clear 11: Don't care * Always read as 11.	
			Invert when the UC10 value is loaded into TB0CP1H/L.	Invert when the UC10 value is loaded into TB0CP0H/L.	Invert when the UC10 value matches the value in TB0RG1H/L.	Invert when the UC10 value matches the value in TB0RG0H/L.		

Timer flip-flop control (TB0FF0)

00	Invert
01	Set to 1
10	Clear to 0
11	Don't care

Inverted when the UC10 value matches the value in TB0RG0H/L.

0	Disable trigger
1	Enable trigger

Inverted when the UC10 value matches the value in TB0RG1H/L.

0	Disable trigger
1	Enable trigger

Inverted when the UC10 value is loaded in to TB0CP0H/L.

0	Disable trigger
1	Enable trigger

Inverted when the UC10 value is loaded in to TB0CP1H/L.

0	Disable trigger
1	Enable trigger

Figure 3.17.4 The Registers for TMRB

TMRB0 Register

		7	6	5	4	3	2	1	0
TB0RG0L (1188H)	bit Symbol	–							
	Read/Write	W							
	After reset	Undefined							
TB0RG0H (1189H)	bit Symbol	–							
	Read/Write	W							
	After reset	Undefined							
TB0RG1L (118AH)	bit Symbol	–							
	Read/Write	W							
	After reset	Undefined							
TB0RG1H (118BH)	bit Symbol	–							
	Read/Write	W							
	After reset	Undefined							
TB0CP0L (118CH)	bit Symbol	–							
	Read/Write	W							
	After reset	Undefined							
TB0CP0H (118DH)	bit Symbol	–							
	Read/Write	W							
	After reset	Undefined							
TB0CP1L (118EH)	bit Symbol	–							
	Read/Write	W							
	After reset	Undefined							
TB0CP1H (118FH)	bit Symbol	–							
	Read/Write	W							
	After reset	Undefined							

Note: All registers are prohibited to execute read-modify-write instruction.

Figure 3.17.5 The Registers 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 intervals. The interval time is set in the timer register TB0RG1H/L.

		7	6	5	4	3	2	1	0	
TB0RUN	←	0	0	X	X	–	0	X	0	Stop TMRB0.
INTTB01	←	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 disable the capture function.
		(** = 01, 10, 11)								
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) output mode

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

The PPG mode is obtained by inversion of the timer flip-flop TB0FF0 that is enabled by the match of the up counter UC10 with timer register TB0RG0H/L or TB0RG1H/L and is output to TB0OUT0. In this mode the following conditions must be satisfied.

(Value set in TB0RG0H/L) < (Value set in TB0RG1H/L)

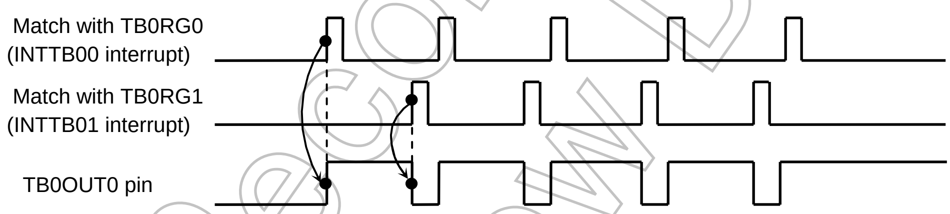


Figure 3.17.6 Programmable Pulse Generation (PPG) Output Waveforms

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

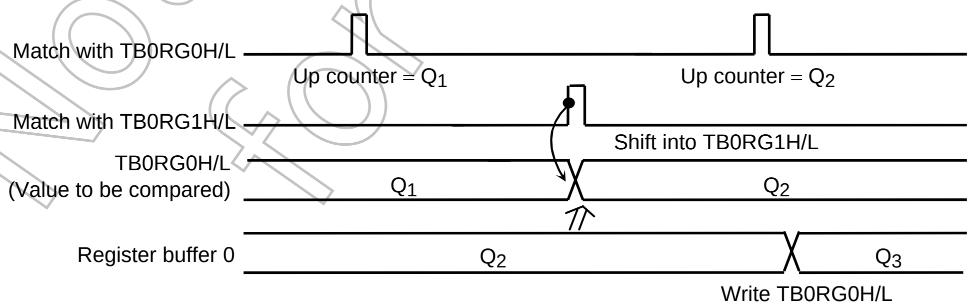


Figure 3.17.7 Operation of Register Buffer

The following block diagram illustrates this mode.

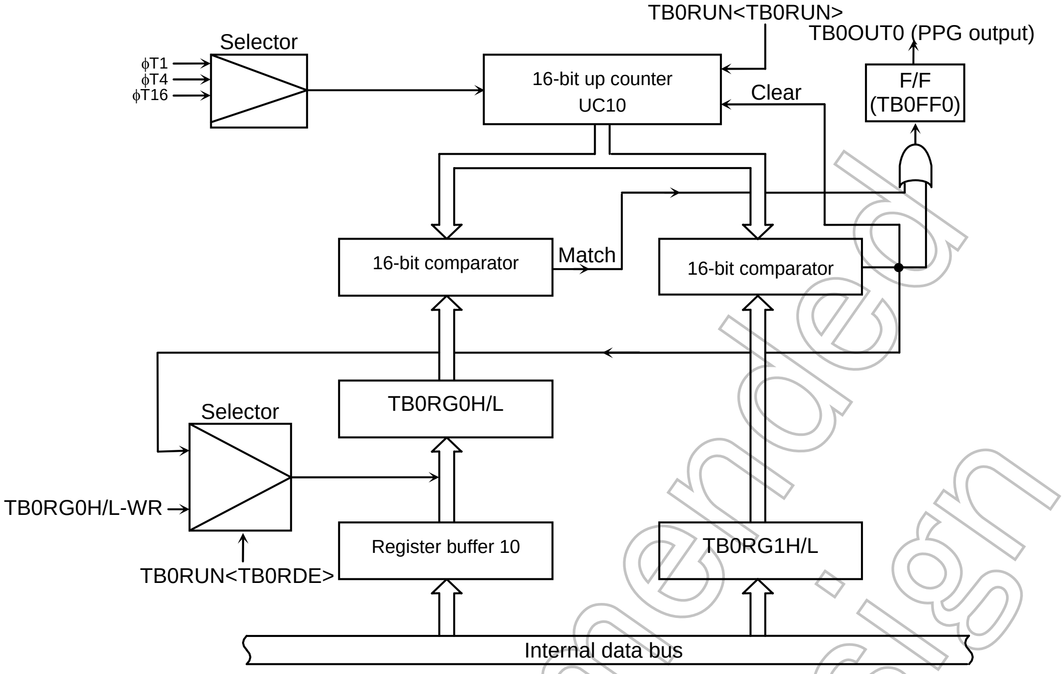


Figure 3.17.8 Block Diagram of 16-Bit Mode

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

		7	6	5	4	3	2	1	0	
TB0RUN	←	0	0	X	X	–	0	X	0	Disable the TB0RG0H/L double buffer and stop TMRB0.
TB0RG0H/L	←	*	*	*	*	*	*	*	*	Set the duty ratio (16 bits).
		*	*	*	*	*	*	*	*	
TB0RG1H/L	←	*	*	*	*	*	*	*	*	Set the frequency (16 bits).
		*	*	*	*	*	*	*	*	
TB0RUN	←	1	0	X	X	–	0	X	0	Enable the TB0RG0H/L double buffer. (The duty and frequency are changed on an INTTB01 interrupt.)
TB0FFCR	←	1	1	0	0	1	1	1	0	Set the mode to invert TB0FF0 at the match with TB0RG0H/L/TB0RG1H/L. Set TB0FF0 to 0.
TB0MOD	←	0	0	1	0	0	1	*	*	Select the Prescaler output clock as the input clock and disable the capture function.
		(** = 01, 10, 11)								
PCCR	←	X	1	–	X	–	X	–	–	Set PC6 to function as TB0OUT0.
PCFC	←	X	1	–	X	–	X	–	–	
TB0RUN	←	1	0	X	X	–	1	X	1	Start TMRB0.

X: Don't care, –: No change

3.18 PSB (Power supply backup)

The power supply input of TMP92C820 is divided into three systems as follows;

- Analog power supply input (AVCC to AVSS)
- Digital power supply input (DVCC to DVSS)
- Digital power supply input for RTC (RTCVCC to DVSS)

The individual power supply input is isolated from each other.

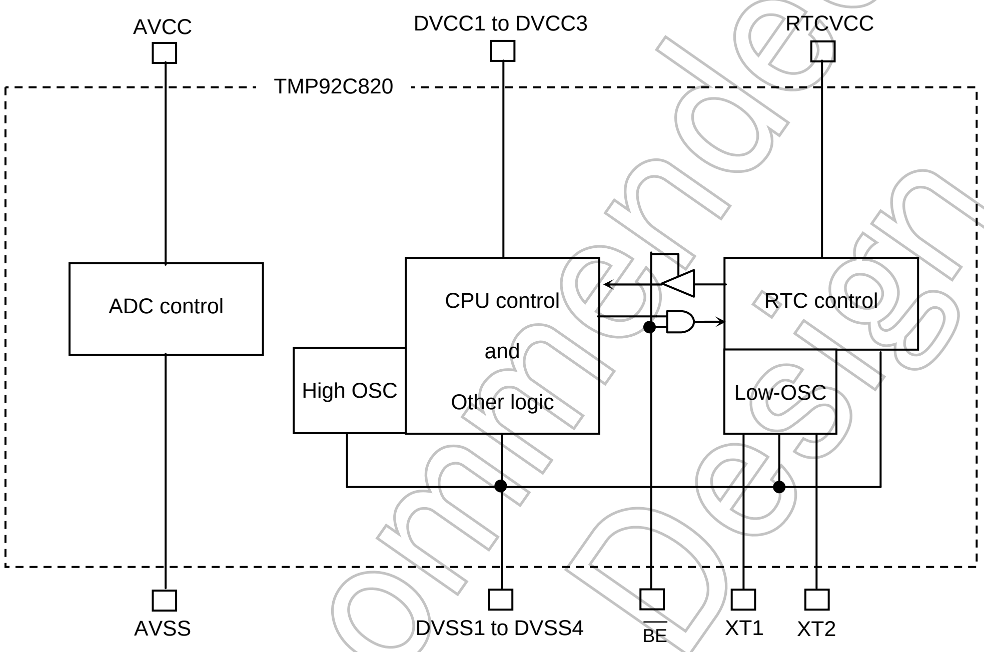


Figure 3.18.1 Power Supply Input System

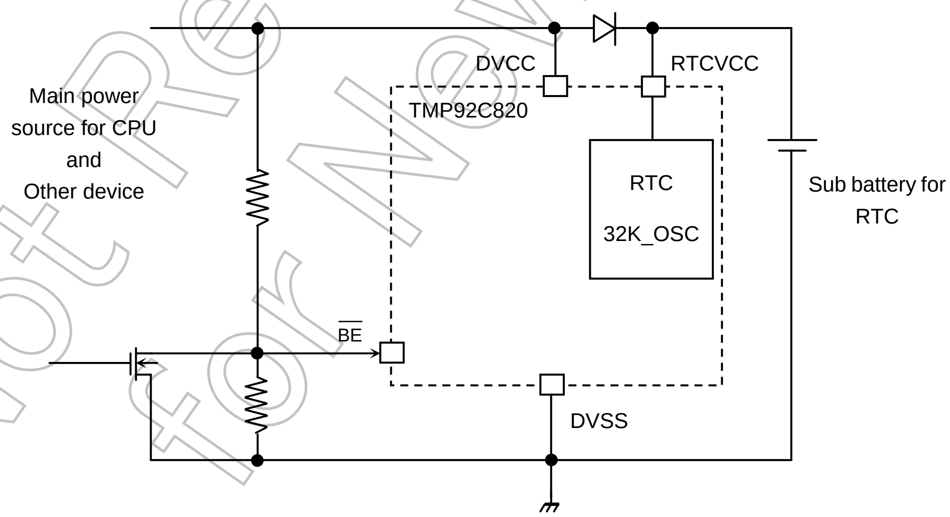


Figure 3.18.2 Outside Circuit Example for PSB

The TMP92C820 has the power supply backup mode which is designed to work for only RTC under sub battery supply. TMP92C820 enters the power supply backup mode using the $\overline{BE}$ (Backup enable signal pin) and the $\overline{RESET}$.

Figure 3.18.3 to Figure 3.18.4 show the timing diagram of $\overline{BE}$ and $\overline{RESET}$.

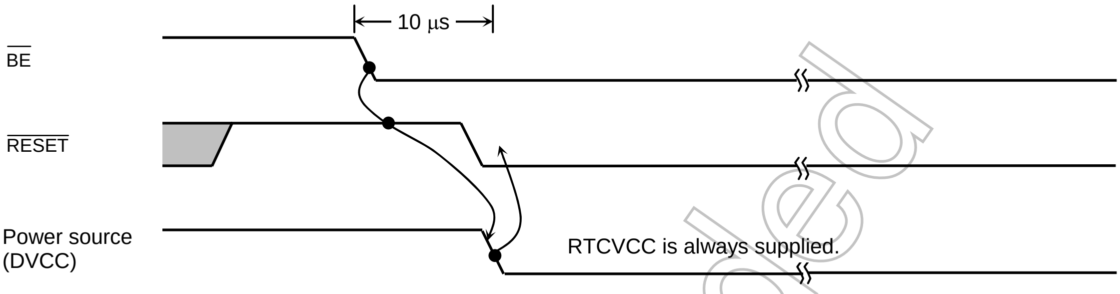


Figure 3.18.3 Normal Mode to PSB Mode

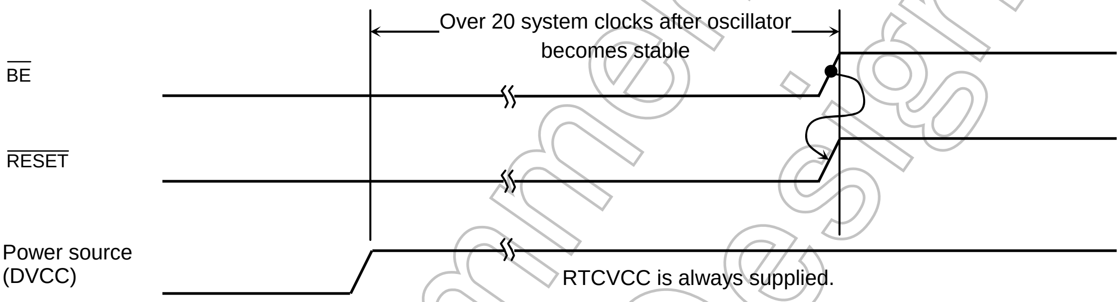


Figure 3.18.4 Normal Mode from PSB Mode

Backup enable pin ($\overline{BE}$)

RTC can work under BE = "L". It is prohibited to access to RTC registers when BE = "L". In addition, low-frequency oscillator (fs) isn't provided except RTC circuit. Under this condition, only internal RTC circuit operates, output function ($\overline{ALARM}$, INTRTC) is prohibited.

Caution

1) Because it might waste power consumption if control signal is "H" level with no-power supply to DVcc, control signal usually set "L" level or high impedance. However, when using backup function with no-power supply to DVcc, $\overline{BE}$ pin must be input "L" level.
2) When $\overline{BE}$ pin is set to "L", low-frequency oscillator operates forcibly and RTC operates too. Therefore, don't set to $\overline{BE}$ = "L", when low-frequency oscillator and RTC are not operating.
3) When releasing $\overline{RESET}$, please confirm $\overline{BE}$ pin to be "H" level completely before releasing $\overline{RESET}$.

4. Electrical Characteristics

4.1 Absolute Maximum Ratings

Parameter	Symbol	Rating	Unit
Power supply voltage	V_{CC}	−0.5 to 4.0	V
Input voltage	V_{IN}	−0.5 to V_{CC} + 0.5	
Output current (Per pin)	I_{OL}	2	mA
Output current (Per pin)	I_{OH}	−2	
Output current (Total)	ΣI_{OL}	80	
Output current (Total)	ΣI_{OH}	−80	
Power dissipation (Ta = 85°C)	P_D	600	mW
Soldering temperature (10 s)	Tsolder	260	°C
Storage temperature	Tstg	−65 to +150	
Operation temperature	Topr	−20 to +70	

Note: The absolute maximum ratings are rated values that must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any absolute maximum rating is exceeded, the device may break down or its performance may be degraded, causing it to catch fire or explode resulting in injury to the user. Thus, when designing products that include this device, ensure that no absolute maximum rating value will ever be exceeded.

Solderability of lead-free products

Test parameter	Test condition	Note
Solderability	(1) Use of Sn-37Pb solder Bath Solder bath temperature =230ºC, Dipping time = 5 seconds The number of times = one, Use of R-type flux	Pass: solderability rate until forming ≥95%
	(2) Use of Sn-3.0Ag-0.5Cu solder bath Solder bath temperature =245 ºC, Dipping time = 5 seconds The number of times = one, Use of R-type flux (use of lead-free)	

4.2 DC Electrical Characteristics

V_{CC} = 3.3 ± 0.3 V/X1 = 4 to 40 MHz/Ta = −20 to 70°C

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Power supply voltage (DVCC = AVCC = RTCVCC) (DVSS = AVSS = 0 V)	V_{CC}	X1 = 4 to 40 MHz (Internal 2 to 20 MHz) XT1 = 30 to 34 kHz	3.0		3.6	V
Input low voltage D0 to D7 P10 to P17 (D8 to D15) P20 to P27 (D16 to D23) P30 to P37 (D24 to D31)	V_{IL0}		−0.3		0.6	V
Input low voltage P40 to P47 P50 to P57 P60 to P67 P76 P95 PF0, PF3 PG0 to PG4 PL0 to PL7	V_{IL1}				$0.3V_{CC}$	
Input low voltage P90 to P94, P96 PA0 to PA7 PC0, PC1, PC3, PC5, PC6 PF1, PF2, PF4, PF5 $\overline{BE}$ $\overline{RESET}$	V_{IL2}				$0.25V_{CC}$	
Input low voltage AM0 to AM1	V_{IL3}				0.3	
Input low voltage X1, XT1	V_{IL4}				$0.2V_{CC}$	
Input high voltage D0 to D 7 P10 to P17 (D8 to D15) P20 to P27 (D16 to D23) P30 to P37 (D24 to D31)	V_{IH0}		2.0		$V_{CC} + 0.3$	V
Input high voltage P40 to P47 P50 to P57 P60 to P67 P76 P95 PF0, PF3 PG0 to PG4 PL0 to PL7	V_{IH1}		$0.7 \times V_{CC}$			
Input high voltage P90 to P94, P96 PA0 to PA7 PC0, PC1, PC3, PC5, PC6 PF1, PF2, PF4, PF5 $\overline{BE}$ $\overline{RESET}$	V_{IH2}		$0.75 \times V_{CC}$			
Input high voltage AM0 to AM1	V_{IH3}		$V_{CC} - 0.3$			
Input high voltage X1, XT1	V_{IH4}		$0.8 \times V_{CC}$			

V_{CC} = 3.3 ± 0.3 V/X1 = 4 to 40 MHz/Ta = −20 to 70°C

Parameter	Symbol	Condition	Min	Typ.	Max	Unit
Output low voltage	V_{OL}	I_{OL} = 1.6 mA			0.45	V
Output high voltage	V_{OH}	I_{OH} = −400 μA	2.4			V
Input leakage current	I_{LI}	0.0 ≤ V_{in} ≤ V_{CC}		0.02	±5	μA
Output leakage current	I_{LO}	0.2 ≤ V_{in} ≤ V_{CC} − 0.2		0.05	±10	μA
Power down voltage at STOP (for internal RAM backup)	V_{STOP}	V_{IL2} = 0.2 V_{CC}, V_{IH2} = 0.8 V_{CC}	1.8		3.6	V
Pull-up resistor $\overline{RESET}$	R_{RST}		100		400	kΩ
Programmable pull-up resistor	R_{KH}					
Pin capacitance	C_{IO}	fc = 1 MHz			10	pF
Schmitt width	V_{TH}	P90 to P94, P96, PA0 to PA7, PC0, PC1, PC3, PC5, PC6, PF1, PF2, PF4, PF5, $\overline{BE}$, $\overline{RESET}$	0.4	1.0		V
Operating current (NORMAL)	ICC	DV_{CC} = 3.6 V, X1 = 40 MHz (Internal 20 MHz)		37.0	60	mA
IDLE2 mode	ICC_{IDLE2}			26.0	39	mA
IDLE1 mode	ICC_{IDLE1}			2.7	5.0	mA
STOP	ICC_{STOP}	DV_{CC} = 3.6 V		0.4	15	μA
SLOW	ICCS	DV_{CC} = 3.6 V, XT1 = 32.768 kHz (Internal 15.8625 kHz)		43.0	100	μA
SLOW, IDEL2 mode	$ICCS_{IDLE2}$			30.0	70	μA
SLOW, IDLE1 mode	$ICCS_{IDLE1}$			8.0	40	μA
RTC V_{CC} power dissipation	ICCRTC	$RTCV_{CC}$ = 3.6 V, XT1 = 32.768 kHz		4.0	7.0	μA
		$RTCV_{CC}$ = 2.0 V, XT1 = 32.768 kHz		1.0	2.0	

4.3 AC Characteristics

4.3.1 Basic Bus Cycle

Read cycle V_{CC} = 3.3 ± 0.3 V/X1 = 4 to 40 MHz/Ta = –20 to 70°C

No.	Parameter	Symbol	Min	Max	at 20 MHz	at 16 MHz	Unit
1	OSC period (X1/X2)	t_{OSC}	25	250	25	31.25	ns
2	System clock period (= T)	t_{CYC}	50	500	50	62.5	ns
3	SDCLK low width	t_{CL}	0.5T – 15		10	16	ns
4	SDCLK low width	t_{CH}	0.5T – 15		10	16	ns
5-1	A0 to A23 valid → D0 to D31 input at 0 waits	t_{AD}		2.0T – 30	70	95	ns
5-2	A0 to A23 valid → D0 to D31 input at 1 wait	t_{AD3}		3.0T – 30	120	157.5	ns
6-1	$\overline{RD}$ fall → D0 to D31 input at 0 waits	t_{RD}		1.5T – 30	45	63.75	ns
6-2	$\overline{RD}$ fall → D0 to D31 input at 1 wait	t_{RD3}		2.5T – 30	95	126.25	ns
7-1	$\overline{RD}$ low width at 0 waits	t_{RR}	1.5T – 20		55	74	ns
7-2	$\overline{RD}$ low width at 1 wait	t_{RR3}	2.5T – 20		105	136	ns
8	A0 to A23 valid → $\overline{RD}$ fall	t_{AR}	0.5T – 20		5	11	ns
9	RD rise → SDCLK rise	t_{RK}	0.5T – 20		5	11	ns
10	A0 to A23 valid → D0 to D31 hold	t_{HA}	0		0	0	ns
11	$\overline{RD}$ rise → D0 to D31 hold	t_{HR}	0		0	0	ns
12	$\overline{WAIT}$ setup time	t_{TK}	15		15	15	ns
13	$\overline{WAIT}$ hold time	t_{KT}	5		5	5	ns
14	Data byte control access time for SRAM	t_{SBA}		1.5T – 30	45	63.75	ns

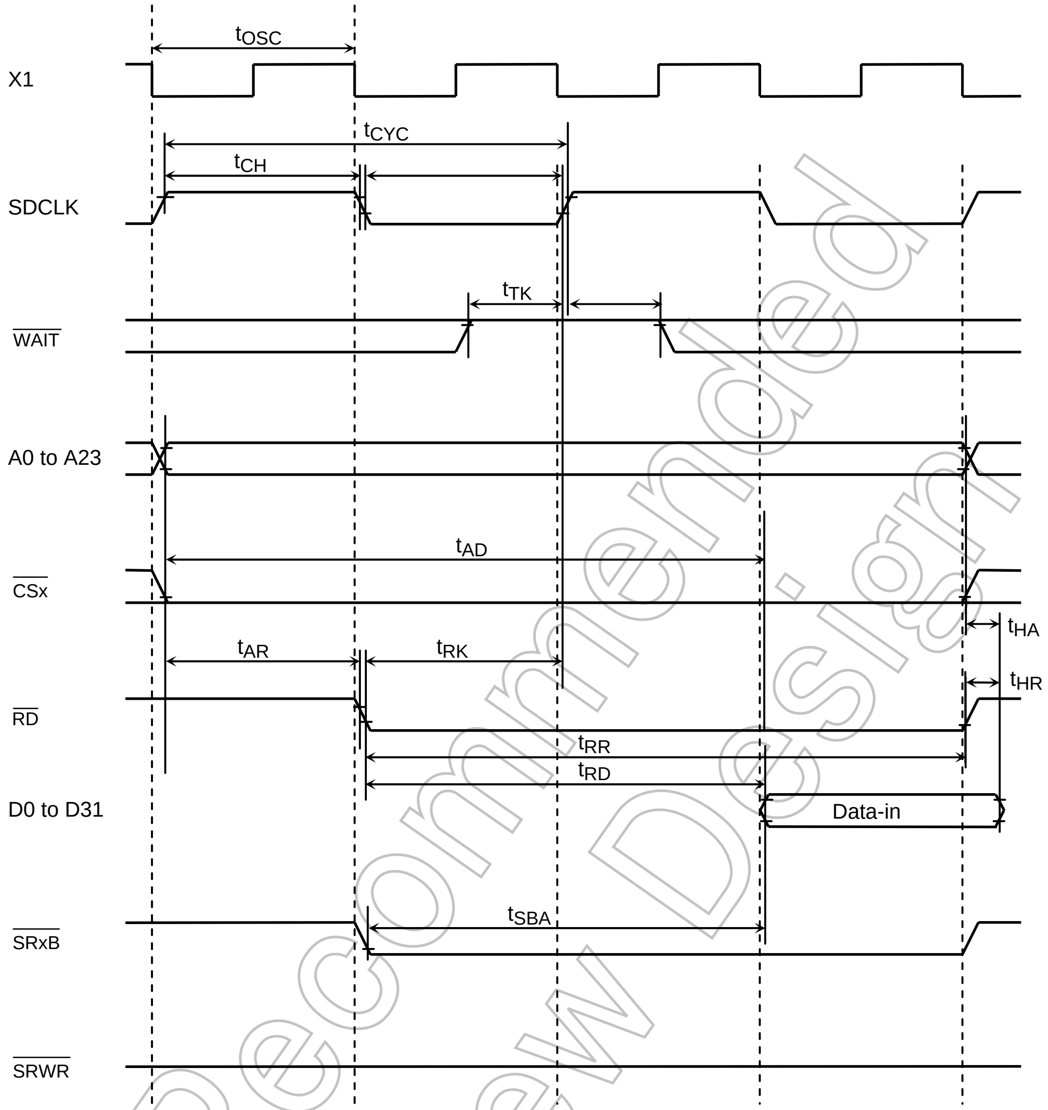
Write cycle V_{CC} = 3.3 ± 0.3 V/X1 = 4 to 40 MHz/Ta = –20 to 70°C

No.	Parameter	Symbol	Min	Max	at 20 MHz	at 16 MHz	Unit
15-1	D0 to D31 valid → $\overline{WRxx}$ rise at 0 waits	t_{DW}	1.25T – 35		28	43	ns
15-2	D0 to D31 valid → $\overline{WRxx}$ rise at 1 wait	t_{DW3}	2.25T – 35		78	106	ns
16-1	$\overline{WRxx}$ low width at 0 waits	t_{WW}	1.25T – 30		33	48	ns
16-2	$\overline{WRxx}$ low width at 1 wait	t_{WW3}	2.25T – 30		83	111	ns
17	A0 to A23 valid → WR fall	t_{AW}	0.5T – 20		5	11	ns
18	$\overline{WRxx}$ fall → SDCLK rise	t_{WK}	0.5T – 20		5	11	ns
19	$\overline{WRxx}$ rise → A0 to A23 hold	t_{WA}	0.25T – 5		8	11	ns
20	$\overline{WRxx}$ rise → D0 to D31 hold	t_{WD}	0.25T – 5		8	11	ns
21	$\overline{RD}$ rise → D0 to D31 output	t_{RDO}	0.5T – 5		20	26.25	ns
22	Write pulse width for SRAM	t_{SWP}	1.25T – 30		32.5	48.125	ns
23	Data byte control to end of write for SRAM	t_{SBW}	1.25T – 30		32.5	48.125	ns
24	Address setup time for SRAM	t_{SAS}	0.5T – 20		5	11.25	ns
25	Write recovery time for SRAM	t_{SWR}	0.25T – 5		7.5	10.625	ns
26	Data setup time for SRAM	t_{SDS}	1.25T – 35		27.5	43.125	ns
27	Data hold time for SRAM	t_{SDH}	0.25T – 5		7.5	10.625	ns

AC condition

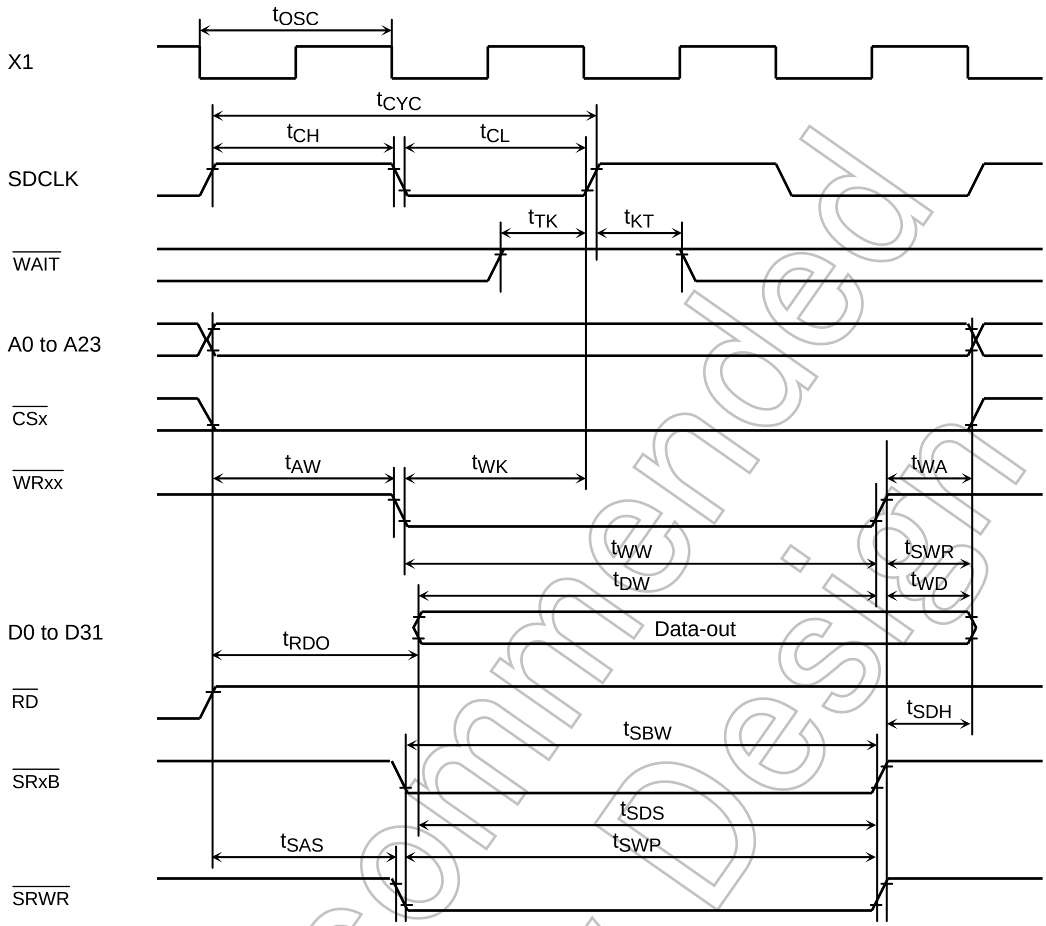
- Output: High = 0.7 V_{CC}, Low = 0.3 V_{CC}, C_L = 50 pF
- Input: High = 0.9 V_{CC}, Low = 0.1 V_{CC}

(1) Read cycle (0 waits)



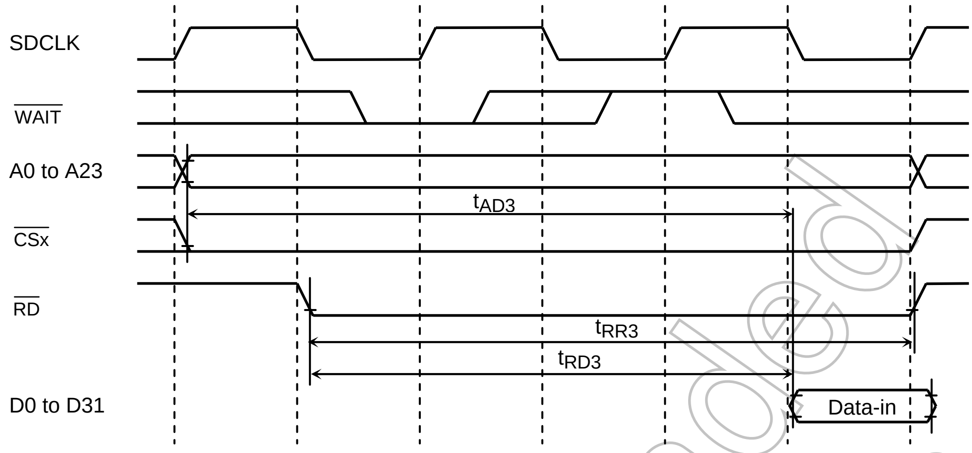
Note: The phase relation between X1 input signal and the other signals is unsettled. The timing chart above is an example.

(2) Write cycle (0 waits)

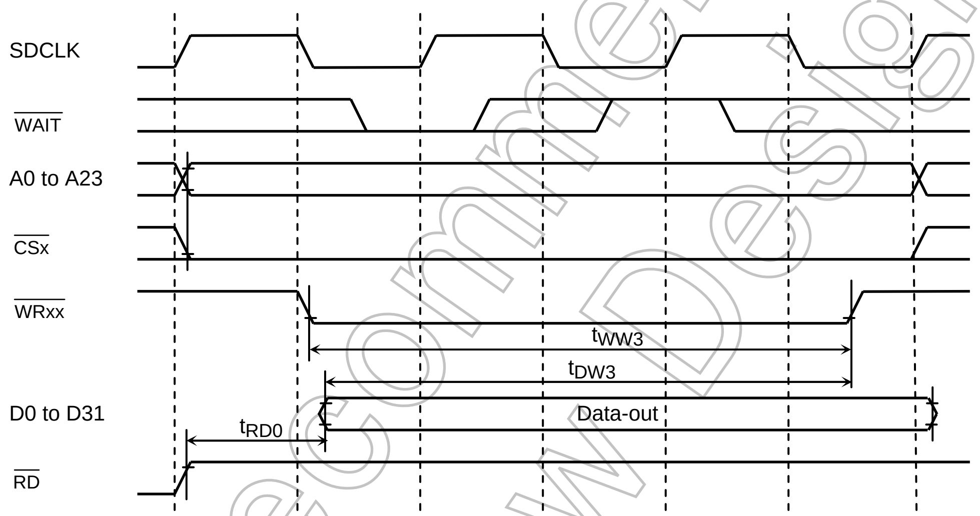


Note: The phase relation between X1 input signal and the other signals is unsettled. The timing chart above is an example.

(3) Read cycle (1 wait)



(4) Write cycle (1 wait)



4.3.2 Page ROM Read Cycle

(1) 3-2-2-2 mode

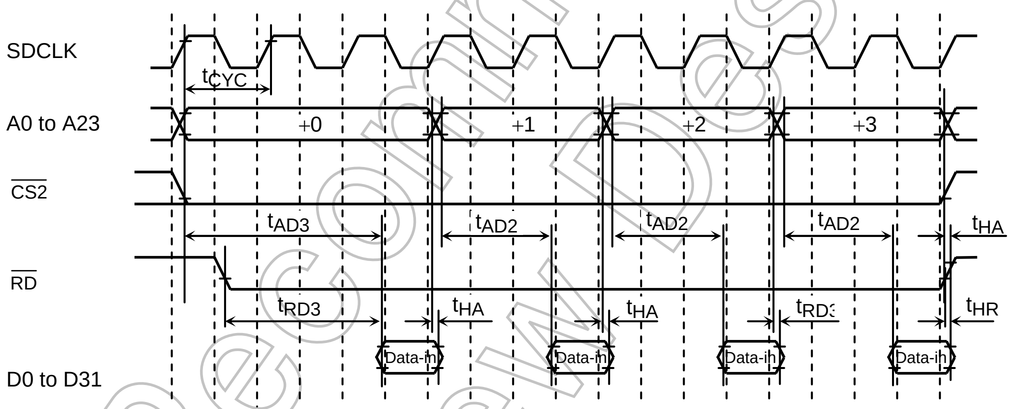
V_{CC} = 3.3 ± 0.3 V/X1 = 4 to 40 MHz/Ta = −20 to 70°C

No.	Parameter	Symbol	Min	Max	at 20 MHz	at 16 MHz	Unit
1	System clock period (= T)	t_{CYC}	50	500	50	62.5	ns
2	A0 and A1 → D0 to D31 input	t_{AD2}		2.0T − 50	50	75	ns
3	A2 to A23 → D0 to D31 input	t_{AD3}		3.0T − 50	100	138	ns
4	RD fall → D0 to D31 input	t_{RD3}		2.5T − 45	80	111	ns
5	A0 to A23 invalid → D0 to D31 hold	t_{HA}	0		0	0	ns
6	RD rise → D0 to D31 hold	t_{HR}	0		0	0	ns

AC condition

- Output: High = $0.7V_{CC}$, Low = $0.3V_{CC}$, C_L = 50 pF
- Input: High = $0.9V_{CC}$, Low = $0.1V_{CC}$

(2) Page ROM read cycle (3-2-2-2 mode)



4.4 SDRAM Controller AC Electrical Characteristics

V_{CC} = 3.3 ± 0.3 V/X1 = 4 to 40 MHz/Ta = −20 to 70°C

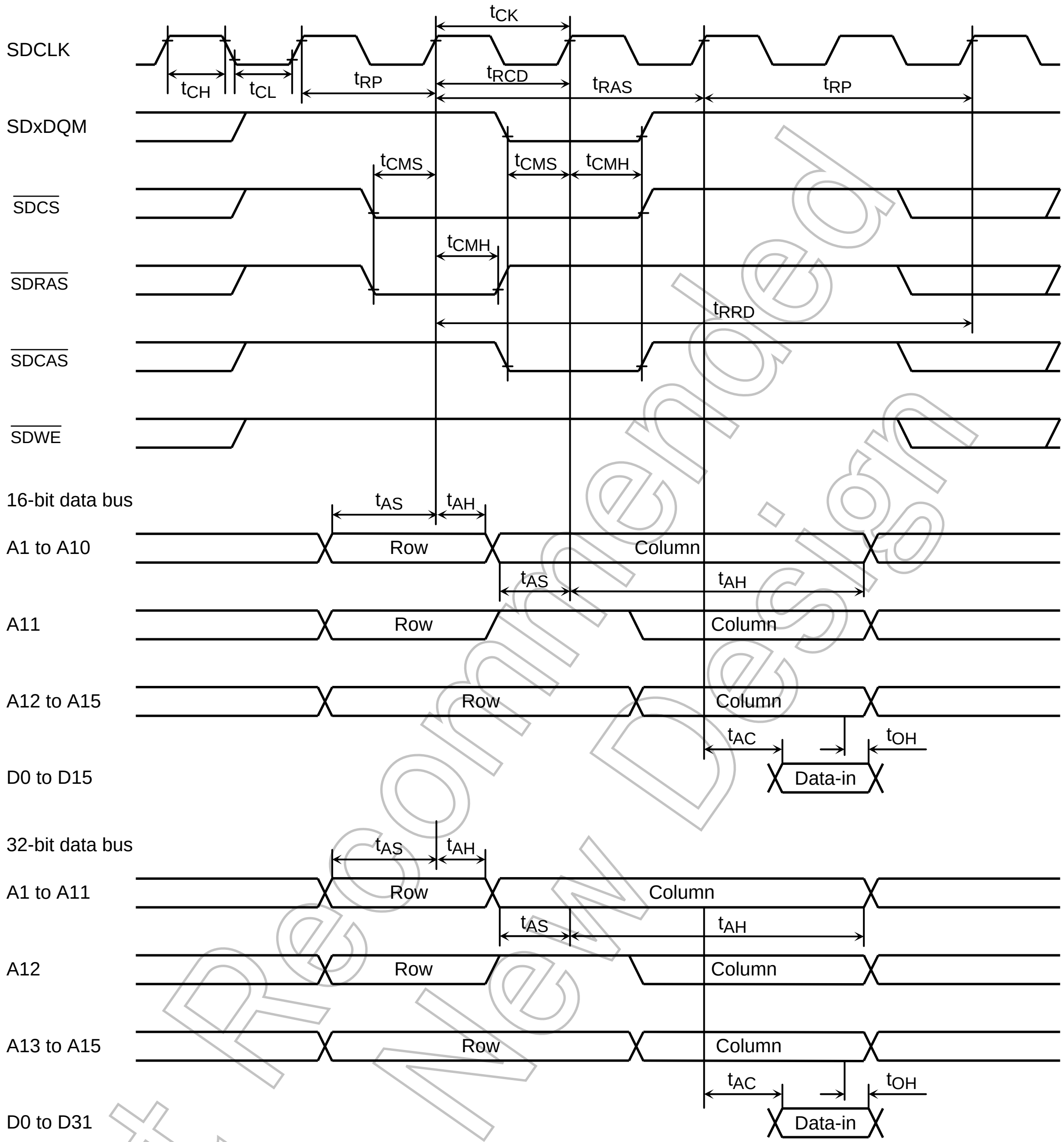
No.	Parameter	Symbol	Variable		at 20 MHz		at 16 MHz		Unit
			Min	Max	Min	Max	Min	Max	
1	Ref/active to Ref/active command period	t_{RC}	2T		100		125		ns
2	Active to precharge command period	t_{RAS}	2T		100		125		ns
3	Active to read/write command delay time	t_{RCD}	T		50		62.5		ns
4	Precharge to active command period	t_{RP}	T		50		62.5		ns
5	Active to active command period	t_{RRD}	3T		150		187.5		ns
6	Write recovery time (CL* = 2)	t_{WR}	T		50		62.5		ns
7	CLK cycle time (CL* = 2)	t_{CK}	T		50		62.5		ns
8	CLK high level width	t_{CH}	0.5T − 15		10		16.25		ns
9	CLK low level width	t_{CL}	0.5T − 15		10		16.25		ns
10	Access time from CLK (CL* = 2)	t_{AC}		T − 30		20		32.5	ns
11	Output data hold time	t_{OH}	0		0		0		ns
12	Data-in setup time	t_{DS}	T − 35		15		27.5		ns
13	Data-in hold time	t_{DH}	T − 5		45		57.50		ns
14	Address setup time	t_{AS}	0.75T − 35		2.5		11.88		ns
15	Address hold time	t_{AH}	3		3		3		ns
16	CKE setup time	t_{CKS}	0.5T − 15		10		16.25		ns
17	Command setup time	t_{CMS}	0.5T − 15		10		16.25		ns
18	Command hold time	t_{CMH}	0.5T − 15		10		16.25		ns
19	Mode register set cycle time	t_{RSC}	T		50		62.5		ns

Note 1: CL* is CAS latency.

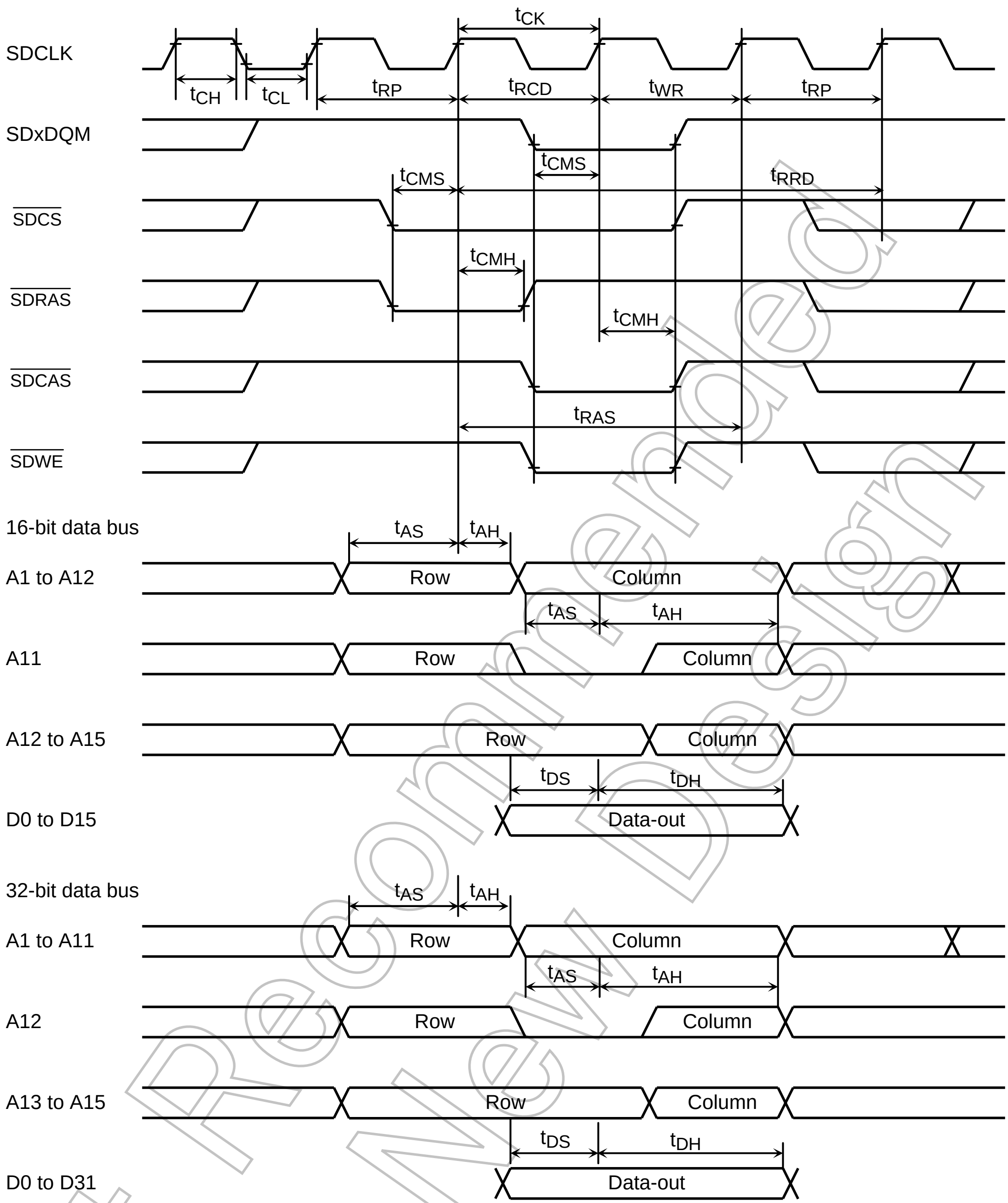
Note 2: AC measuring conditions

- Output level: High = 0.7 V_{CC}, Low = 0.3 V_{CC}, C_L = 50 pF
- Input level: High = 0.9 V_{CC}, Low = 0.1 V_{CC}.

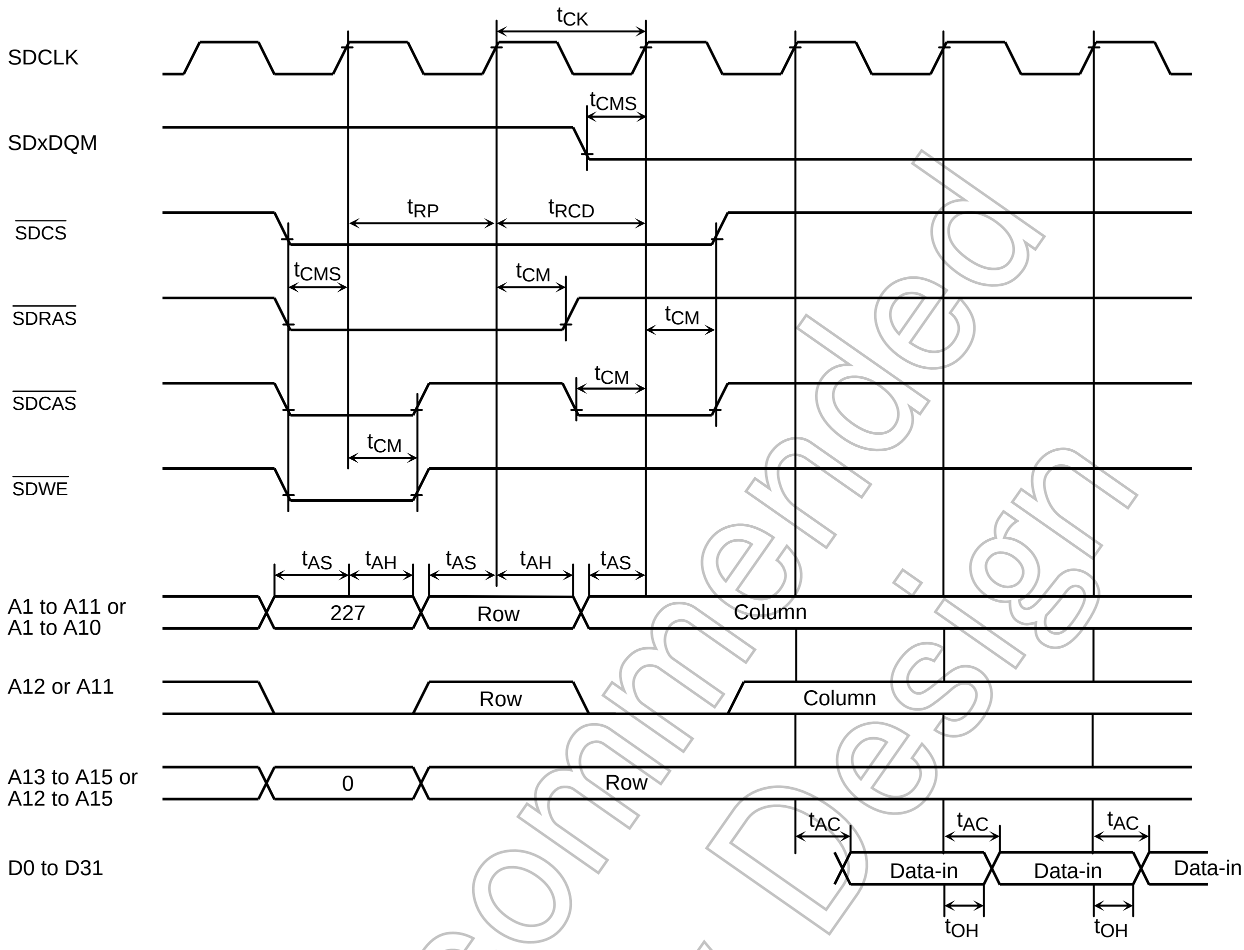
- SDRAM read timing (CPU access or LCDC normal access)



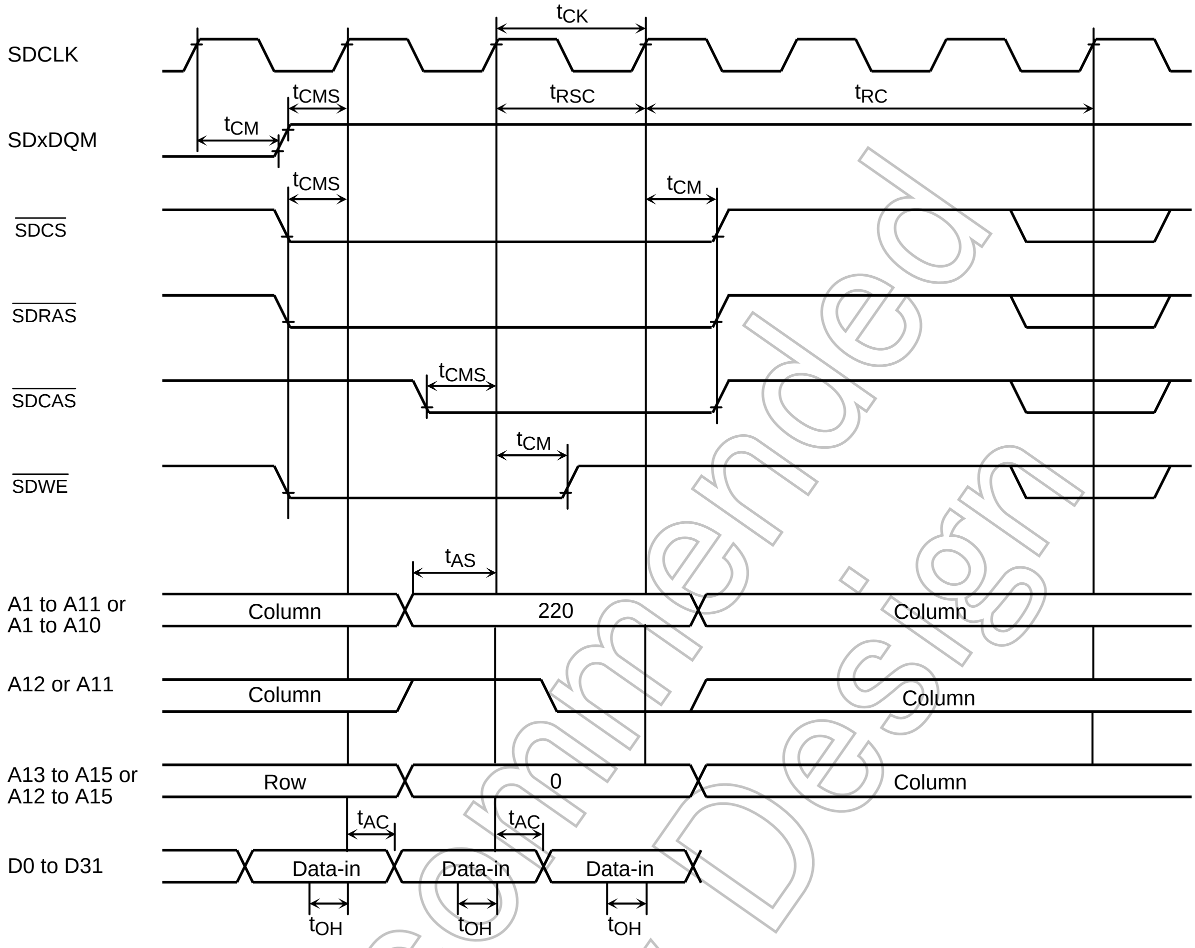
- SDRAM write timing (CPU access)



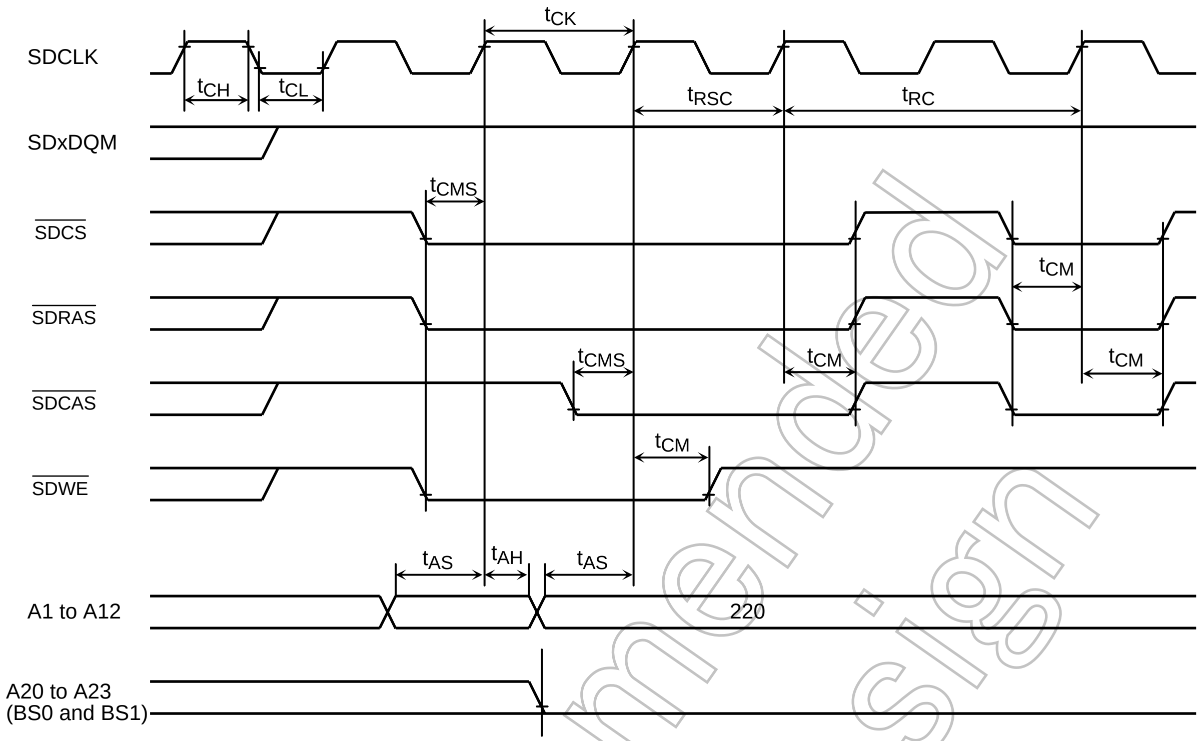
- SDRAM burst read timing (Start of burst cycle)



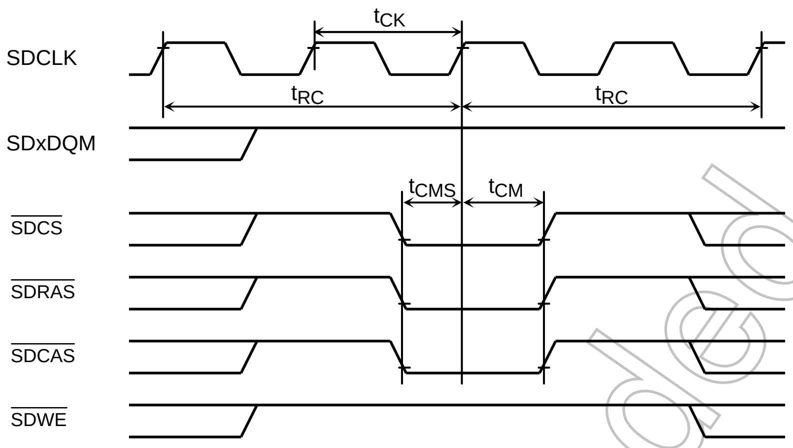
- SDRAM burst read timing (End of burst cycle)



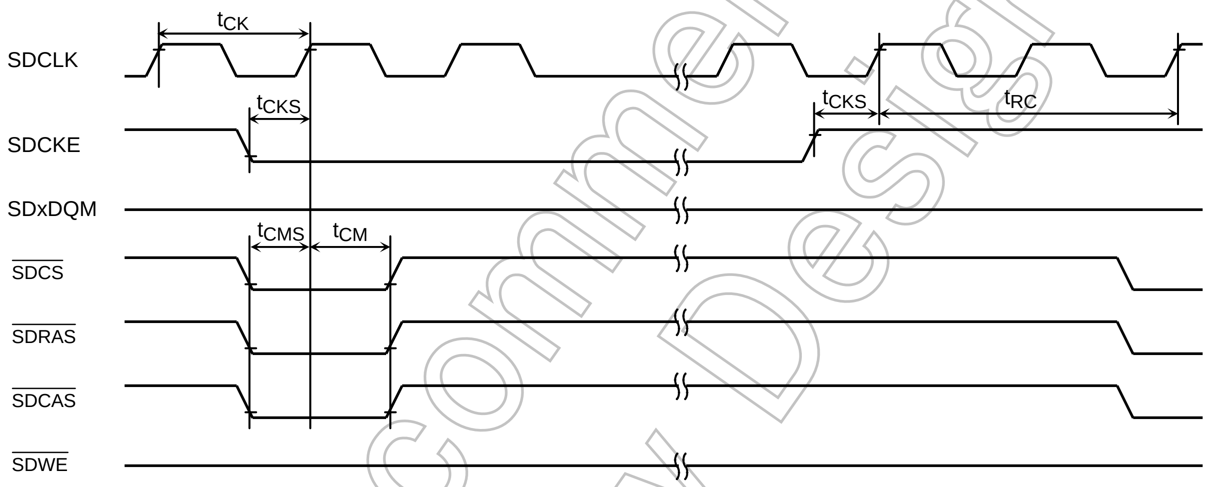
- SDRAM initialize timing



- SDRAM refresh timing



- SDRAM self refresh timing



4.5 AD Conversion Characteristics

Parameter		Symbol	Min	Typ.	Max	Unit
Analog reference voltage (+)		V_{REFH}	V_{CC} − 0.2	V_{CC}	V_{CC}	V
Analog reference voltage (−)		V_{REFL}	VSS	VSS	VSS + 0.2	V
AD converter power supply voltage		A_{VCC}	V_{CC}	V_{CC}	V_{CC}	V
AD converter ground		A_{VSS}	VSS	VSS	VSS	V
Analog input voltage		A_{VIN}	VREFL		VREFH	V
Analog current for analog reference voltage	<VREFON> = 1	I_{REF}		0.8	1.2	mA
Analog current for analog reference voltage	<VREFON> = 0	I_{REF}		0.02	5.0	UA
Total error (Quantize error of ±0.5 LSB is included)		E_T		±1.0	±4.0	LSB

4.6 Event Counter (TI0, TI4, TI8, TI9, TIA, and TIB)

Parameter	Symbol	Variable		20 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
Clock cycle	T_{VCK}	8T + 100		500		600		ns
Clock low width	T_{VCKL}	4T + 40		240		290		ns
Clock high width	T_{VCKH}	4T + 40		240		290		ns

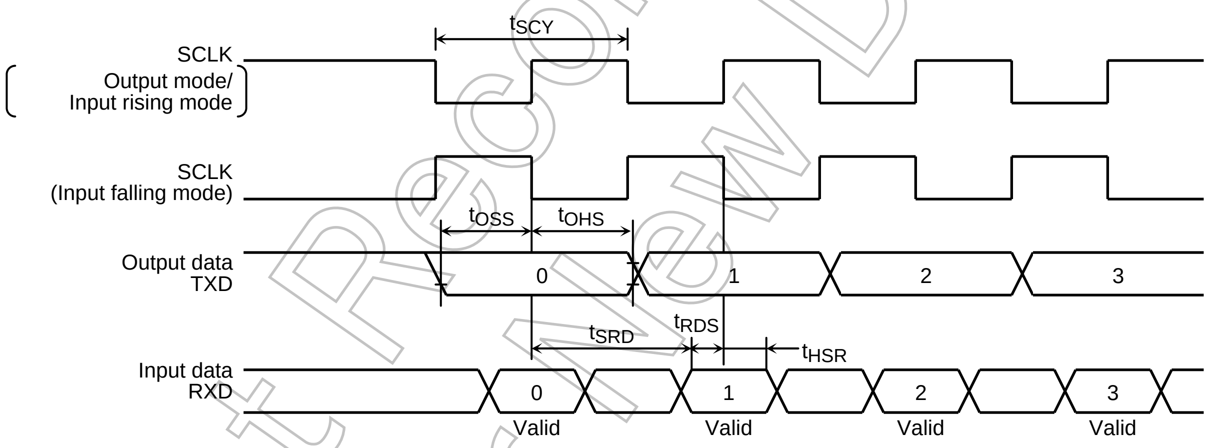
4.7 Serial Channel Timing

(1) SCLK input mode (I/O interface mode)

Parameter	Symbol	Variable		20 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK cycle	T_{SCY}	16T		0.8		1.0		μs
Output data → SCLK rise	T_{OSS}	$T_{SCY}/2 - 4T - 110$		90		140		ns
SCLK rise → Output data hold	T_{OHS}	$T_{SCY}/2 + 2T + 0$		500		625		
SCLK rise → Input data hold	T_{HSR}	0		0		0		
SCLK rise → Input data valid	T_{SRD}		$T_{SCY} - 0$		800		1000	
Input data → SCLK rise	T_{RDS}	0		0				

(2) SCLK output mode (I/O interface mode)

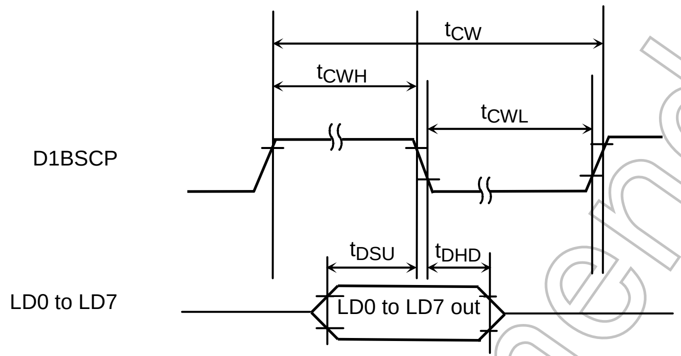
Parameter	Symbol	Variable		20 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
SCLK cycle (Programmable)	T_{SCY}	16T	8192T	0.8	409.6	1.0	512	μs
Output data → SCLK rise	T_{OSS}	$T_{SCY}/2 - 40$		360		460		ns
SCLK rise → Output data hold	T_{OHS}	$T_{SCY}/2 - 40$		360		460		
SCLK rise → Input data hold	T_{HSR}	0		0		0		
SCLK rise → Input data valid	T_{SRD}		$T_{SCY} - 1T - 180$		570		757.5	
Input data → SCLK rise	T_{RDS}	0		0		0		



4.8 Interrupt Operation

Parameter	Symbol	Variable		20 MHz		16 MHz		Unit
		Min	Max	Min	Max	Min	Max	
INT0 to INT3 low width	T_{INTAL}	4T + 40		200		290		ns
INT0 to INT3 high width	T_{INTAH}	4T + 40		200		290		

4.9 LCD Controller SR Mode



V_{CC} = 3.3 ± 0.3 V/X1 = 4 to 40 MHz/Ta = –20 to 70°C

No.	Parameter	Symbol	Variable		20 MHz (tm = 0)		16 MHz (tm = 0)		Unit
			Min	Max	Min	Max	Min	Max	
1	Data vaild → D1BSCP fall	t_{DSU}	0.5T – 20 + tm		5		11.25		ns
2	D1BSCP fall → Data hold	t_{DHD}	0.5T – 5 + tm		20		26.25		ns
3	D1SBCP → Clock high width	t_{CWH}	0.5T – 10 + tm		15		21.25		ns
4	D1BSCP → Clock low width	t_{CWL}	0.5T – 10 + tm		15		21.25		ns
5	D1BSCP → Clock cycle	t_{CW}	T + 2tm		50		62.5		ns

Note: tm = $(2^{scpw} – 1)$ ×, e.g., if Scpw = 3 (8 clock mode) and 20 MHz, tm = $(2^3 – 1)$ × 50 = 350

4.10 Recommended Oscillation Circuit

The TMP92C820 has been evaluate by below the oscillator vender below. Use this information when selecting external parts.

Note: The total load value of the oscillation is the sum of external loads (C1 and C2) and the floating load of the actual assembled board. There is a possibility of operating error when using C1 and C2 values in the table below. When designing the board, design the minimum length pattern around the oscillator. We also recommend that oscillator evaluation be carried out using the actual board.

(1) Connection example

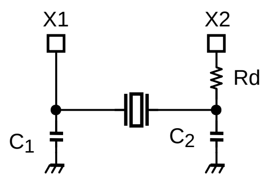


Figure 4.10.1 High-frequency Oscillator

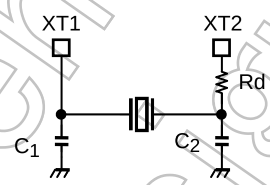


Figure 4.10.2 Low-frequency Oscillator

(2) TMP92C820 recommended ceramic oscillator: Murata Manufacturing Co., Ltd; JAPAN

MCU	Oscillation Frequency [MHz]	Type	Oscillator Product number	Parameter of Elements				Running Condition	
				C1 [pF] Note1	C2 [pF] Note1	Rf [Ω]	Rd [Ω]	Voltage [V]	Tc [°C]
TMP92C820FG	2.000	SMD	CSTCC2M00G56-R0	(47)	(47)	Open	0	1.8~2.7	−20~ +80
	4.000	SMD	CSTCR4M00G55-R0	(39)	(39)	Open	0	2.7~3.6	
		Lead	CSTLS4M00G56-B0	(47)	(47)	Open	0		
	6.000	SMD	CSTCR6M00G55-R0	(39)	(39)	Open	0		
		Lead	CSTLS6M00G56-B0	(47)	(47)	Open	0		
	10.000	SMD	CSTCE10M0G52-R0	(10)	(10)	Open	0		
		Lead	CSTLS10M0G53-B0	(15)	(15)	Open	0		
			CSTLS10M0G53-B0	(15)	(15)	Open	0	1.8~2.7	
	12.000	SMD	CSTCE12M5G52-R0	(10)	(10)	Open	0		
	20.000	SMD	CSTCG20M0V53-R0	(15)	(15)	Open	0	2.7~3.6	

Note 1: The figure in parentheses ( ) under C1 and C2 is the built-in condenser type.

Note 2: The product numbers and specifications of the oscillators made by Murata Manufacturing Co., Ltd. are subject to change. For up-to-date information, please refer to the following URL: http://www.murata.co.jp/

5. Table of Special Function Registers (SFRs)

The SFRs (Special function registers) include the I/O ports and peripheral control registers allocated to the 8-Kbyte address space from 000000H to 001FFFH.

(1) I/O port

(2) I/O port control

(3) Interrupt control

(4) DMA controller

(5) Memory controller

(6) MMU

(7) Clock gear

(8) LCD controller

(9) SDRAM controller

(10) 8-bit timer

(11) 16-bit timer

(12) UART/serial channel

(13) I²C bus/serial channel

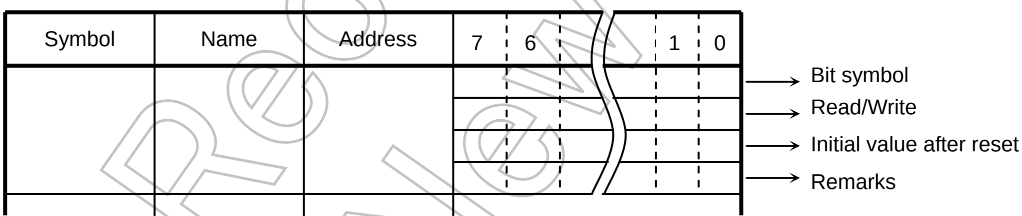
(14) AD converter

(15) Watchdog timer

(16) RTC (Real time clock)

(17) Melody/alarm generator

Table layout



Note: "Prohibit RMW" in the table means that you cannot use RMW instructions on these registers.

Example) When setting bit0 only of the register P0CR, the instruction "SET 0, (PxCR)" cannot be used. The LD (Transfer) instruction must be used to write all eight bits.

Read/Write

R/W: Both read and write are possible.

R: Only read is possible.

W: Only write is possible.

W*: Both read and write are possible (when this bit is read as 1).

Prohibit RMW: Read-modify-write instructions are prohibited. (EX, ADD, ADC, BUS, SBC, INC, DEC,AND, OR, XOR, STCF, RES, SET, CHG, TSET, RLC, RRC, RL, RR, SLA, SRA, SLL, SRL, RLD, RRD instructions are read-modify-write instructions.)

Prohibit RMW *: Read-modify-write is prohibited when controlling the pull-up resistor.

Table 5.1 I/O Register Address Map

[1] Port

Address	Name
0000H	
1H	
2H	
3H	
4H	P1
5H	
6H	P1CR
7H	P1FC
8H	P2
9H	
AH	P2CR
BH	P2FC
CH	P3
DH	
EH	P3CR
FH	P3FC

Address	Name
0010H	P4
1H	
2H	P4CR
3H	P4FC
4H	P5
5H	
6H	P5CR
7H	P5FC
8H	P6
9H	
AH	P6CR
BH	P6FC
CH	P7
DH	
EH	P7CR
FH	P7FC

Address	Name
0020H	P8
1H	P8FC2
2H	
3H	P8FC
4H	P9
5H	P9ODE
6H	P9CR
7H	P9FC
8H	PA
9H	
AH	
BH	PAFC
CH	
DH	
EH	
FH	

Address	Name
0030H	PC
1H	
2H	PCCR
3H	PCFC
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	PF
DH	
EH	PFCR
FH	PFFC

Address	Name
0040H	PG
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	PJ
DH	PJFC2
EH	
FH	PJFC

Address	Name
0050H	PK
1H	
2H	
3H	PKFC
4H	PL
5H	
6H	PLCR
7H	PLFC
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Note: Do not access un-named addresses.

[2] INTC

Address	Name
00D0H	INTE12
1H	INTE3
2H	
3H	
4H	INTETA01
5H	INTETA23
6H	
7H	
8H	INTETB01
9H	
AH	INTETBO0
BH	INTES0
CH	INTES1
DH	
EH	
FH	

Address	Name
00E0H	Reserved
1H	Reserved
2H	Reserved
3H	INTESB0
4H	Reserved
5H	INTALM01
6H	INTALM23
7H	INTALM4
8H	INTERTC
9H	INTEKEY
AH	INTLCD
BH	Reserved
CH	Reserved
DH	INTES2
EH	INTEP0
FH	

Address	Name
00F0H	INTE0AD
1H	INTETC01
2H	INTETC23
3H	INTETC45
4H	INTETC67
5H	SIMC
6H	IIMC
7H	INTWDT
8H	INTCLR
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[3] DMAC

Address	Name
0100H	DMA0V
1H	DMA1V
2H	DMA2V
3H	DMA3V
4H	DMA4V
5H	DMA5V
6H	DMA6V
7H	DMA7V
8H	DMAB
9H	DMAR
AH	Reserved
BH	
CH	
DH	
EH	
FH	

[4] MEMC

Address	Name
0140H	B0CSL
1H	B0CSH
2H	MAMR0
3H	MSAR0
4H	B1CSL
5H	B1CSH
6H	MAMR1
7H	MSAR1
8H	B2CSL
9H	B2CSH
AH	MAMR2
BH	MSAR2
CH	B3CSL
DH	B3CSH
EH	MAMR3
FH	MSAR3

Address	Name
0150H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	BEXCSL
9H	BEXCSH
AH	
BH	
CH	
DH	
EH	
FH	

Address	Name
0160H	
1H	
2H	
3H	
4H	
5H	
6H	PMEMCR
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[5] MMU

Address	Name
01D0H	LOCAL0
1H	LOCAL1
2H	LOCAL2
3H	LOCAL3
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Note: Do not access un-named addresses.

[6] CGEAR

Address	Name
10E0H	SYSCR0
1H	SYSCR1
2H	SYSCR2
3H	EMCCR0
4H	EMCCR1
5H	EMCCR2
6H	Reserved
7H	
8H	Reserved
9H	Reserved
AH	
BH	
CH	
DH	
EH	
FH	

[7] LCDC-1

Address	Name
0200H	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	Reserved
EH	
FH	

Address	Name
0210H	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	

[7] LCDC-2

Address	Name
0220H	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
0230H	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

Address	Name
0240H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	Reserved
9H	Reserved
AH	Reserved
BH	Reserved
CH	Reserved
DH	Reserved
EH	Reserved
FH	Reserved

Note: Do not access un-named addresses.

[8] SDRAMC

Address	Name
0250H	SDACR
1H	SDRCR
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[9] 8-bit timer

Address	Name
1100H	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	

[10] 16-bit timer

Address	Name
1180H	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

[11] SIO

Address	Name
1200H	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
1210H	SC2BUF
1H	SC2CR
2H	SC2MOD0
3H	BR2CR
4H	BR2ADD
5H	SC2MOD1
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[12] SBI

Address	Name
1240H	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 un-named addresses.

[13] 10-bit ADC

Address	Name
12A0H	ADREG0L
1H	ADREG0H
2H	ADREG1L
3H	ADREG1H
4H	ADREG2L
5H	ADREG2H
6H	ADREG3L
7H	ADREG3H
8H	ADREG4L
9H	ADREG4H
AH	Reserved
BH	Reserved
CH	Reserved
DH	Reserved
EH	Reserved
FH	Reserved

Address	Name
12B0H	
1H	
2H	
3H	
4H	
5H	
6H	
7H	
8H	ADMOD0
9H	ADMOD1
AH	ADMOD2
BH	Reserved
CH	
DH	
EH	
FH	

[14] WDT

Address	Name
1300H	WDMOD
1H	WDCR
2H	
3H	
4H	
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[15] RTC

Address	Name
1320H	SECR
1H	MINR
2H	HOURR
3H	DAYR
4H	DATER
5H	MONTHR
6H	YEARR
7H	PAGER
8H	RESTR
9H	
AH	
BH	
CH	
DH	
EH	
FH	

[16] MLD

Address	Name
1330H	ALM
1H	MELALMC
2H	MELFL
3H	MELFH
4H	ALMINT
5H	
6H	
7H	
8H	
9H	
AH	
BH	
CH	
DH	
EH	
FH	

Note: Do not access un-named addresses.

(1) I/O port

Symbol	Name	Address	7	6	5	4	3	2	1	0
P1	Port 1	0004H	P17	P16	P15	P14	P13	P12	P11	P10
			R/W							
			Data from external port (Output latch register is cleared to 0)							
P2	Port 2	0008H	P27	P26	P25	P24	P23	P22	P21	P20
			R/W							
			Data from external port (Output latch register is cleared to 0)							
P3	Port 3	000CH	P37	P36	P35	P34	P33	P32	P31	P30
			R/W							
			Data from external port (Output latch register is cleared to 0)							
P4	Port 4	0010H	P47	P46	P45	P44	P43	P42	P41	P40
			R/W							
			Data from external port (Output latch register is cleared to 0)							
P5	Port 5	0014H	P57	P56	P55	P54	P53	P52	P51	P50
			R/W							
			Data from external port (Output latch register is cleared to 0)							
P6	Port 6	0018H	P67	P66	P65	P64	P63	P62	P61	P60
			R/W							
			Data from external port (Output latch register is cleared to 0)							
			–							
P7	Port 7	001CH		P76	P75	P74	P73	P72	P71	P70
				R/W						
				Data from external port Note1	1	1	1	1	1	1
P8	Port 8	0020H	P87	P86	P85	P84	P83	P82	P81	P80
			R/W							
			1	1	1	1	1	0	1	1
P9	Port 9	0024H		P96	P95	P94	P93	P92	P91	P90
				R/W						
				Data from external port (Output latch register is set to 1)						
PA	Port A	0028H	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
			R							
			Data from external port							
PC	Port C	0030H		PC6	PC5		PC3		PC1	PC0
				R/W			R/W		R/W	
				Data from external port Note2			Data from external port Note2		Data from external port Note2	
PF	Port F	003CH			PF5	PF4	PF3	PF2	PF1	PF0
					R/W					
					Data from external port (Output latch register is set to 1)					
PG	Port G	0040H				PG4	PG3	PG2	PG1	PG0
						R				
						Data from external port				
PJ	Port J	004CH	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
			R/W							
			1	1	1	1	1	1	1	1
PK	Port K	0050H		PK6		PK4	PK3	PK2	PK1	PK0
				R/W		R/W				
				1		1	1	1	1	1
PL	Port L	0054H	PL7	PL6	PL5	PL4	PL3	PL2	PL1	PL0
			R/W							
			Data from external port (Output latch register is set to 1)							

Note 1: Output latch register is cleared to 0.

Note 2: Output latch register is set to 1

(2) I/O port control (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P1CR	Port 1 control register	0006H (Prohibit RMW)	P17C	P16C	P15C	P14C	P13C	P12C	P11C	P10C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P1FC	Port 1 function register	0007H (Prohibit RMW)								P1F
										W
										1
										0: Port 1:Data bus (D8 to D15)
P2CR	Port 2 control register	000AH (Prohibit RMW)	P27C	P26C	P25C	P24C	P23C	P22C	P21C	P20C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P2FC	Port 2 function register	000BH (Prohibit RMW)								P2F
										W
										0/1
										0: Port 1: Data bus (D16 to D23)
P3CR	Port 3 control register	000EH (Prohibit RMW)	P37C	P36C	P35C	P34C	P33C	P32C	P31C	P30C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P3FC	Port 3 function register	000FH (Prohibit RMW)								P3F
										W
										0/1
										0: Port 1: Data bus (D24 to D31)
P4CR	Port 4 control register	0012H (Prohibit RMW)	P47C	P46C	P45C	P44C	P43C	P42C	P41C	P40C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P4FC	Port 4 function register	0013H (Prohibit RMW)	P47F	P46F	P45F	P44F	P43F	P42F	P41F	P40F
			W							
			1	1	1	1	1	1	1	1
			0: Port 1: Address bus (A0 to A7)							
P5CR	Port 5 control register	0016H (Prohibit RMW)	P57C	P56C	P55C	P54C	P53C	P52C	P51C	P50C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P5FC	Port 5 function register	0017H (Prohibit RMW)	P57F	P56F	P55F	P54F	P53F	P52F	P51F	P50F
			W							
			1	1	1	1	1	1	1	1
			0: Port 1: Address bus (A8 to A15)							
P6CR	Port 6 control register	001AH (Prohibit RMW)	P67C	P66C	P65C	P64C	P63C	P62C	P61C	P60C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
P6FC	Port 6 function register	001BH (Prohibit RMW)	P67F	P66F	P65F	P64F	P63F	P62F	P61F	P60F
				W						
			1	1	1	1	1	1	1	1
			0: Port 1: Address bus (A16 to A23)							

I/O port control (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
P7CR	Port 7 control register	001EH (Prohibit RMW)		P76C						
				W						
				0						
				0: Input 1: Output						
P7FC	Port 7 function register	001FH (Prohibit RMW)		P76F	P75F	P74F	P73F	P72F	P71F	P70F
				W						
				0	0	0	0	0	0	1
				0: Port 1: $\overline{\text{WAIT}}$	0: Port 1: R/$\overline{\text{W}}$	0: Port 1: $\overline{\text{WRUU}}$	0: Port 1: $\overline{\text{WRUL}}$	0: Port 1: $\overline{\text{WRLU}}$	0: Port 1: $\overline{\text{WRLL}}$	0: Port 1: $\overline{\text{RD}}$
P8FC	Port 8 function register	0023H (Prohibit RMW)	P87F	–	P85F	P84F	P83F	P82F	P81F	P80F
			W							
			1	0	0	0	0	0	0	0
			0: Port 1: SDCLK	Always write 0.	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: $\overline{\text{CS0}}$
P8FC2	Port 8 function register 2	0021H (Prohibit RMW)	–	P86F2	P85F2	P84F2	–	P82F2	P81F2	P80F2
			W							
			0	0	0	0	0	0	0	0
			Always write “0”	0: <P86F> 1: $\overline{\text{CS2D}}$	0: <P85F> 1: $\overline{\text{CS2C}}$	0: <P84F> 1: $\overline{\text{CS2B}}$	Always write “0”.	0: <P82F> 1: $\overline{\text{CS2A}}$	0: <P81F> 1: $\overline{\text{SDCSL}}$	0: <P80F> 1: $\overline{\text{SDCSH}}$
P9CR	Port 9 control register	0026H (Prohibit RMW)		P96C	P95C	P94C	P93C	P92C	P91C	P90C
				W						
				0	0	0	0	0	0	0
				0: Input 1: Output						
P9FC	Port 9 function register	0027H (Prohibit RMW)		P96F	P95F	P94F	P93F	P92F	P91F	P90F
				W						
				0	0	0	0	0	0	0
				0: Port 1: RXD2, $\overline{\text{CSEXA}}$	0: Port 1: TXD2, $\overline{\text{CS2G}}$	0: Port 1: $\overline{\text{CS2F}}$	0: Port 1: $\overline{\text{CS2E}}$	0: Port, SI 1: SCL Note	0: Port 1: SO, SDA	0: Port, SCK input 1: SCK output Note
P9ODE	Port 9 ODE register	0025H (Prohibit RMW)			P95ODE	–	–	P92ODE	P91ODE	
					W					
					0	0	0	0	0	
					0: 3 states 1: Open drain	Always write “0”	Always write “0”	0: 3 states 1: Open drain	0: 3 states 1: Open drain	
PAFC	Port A function register	002BH (Prohibit RMW)	PA7F	PA6F	PA5F	PA4F	PA3F	PA2F	PA1F	PA0F
			W							
			0	0	0	0	0	0	0	0
			0: KEY-IN disable 1: KEY-IN enable							
PCCR	Port C control register	0032H (Prohibit RMW)		PC6C	PC5C		PC3C		PC1C	PC0C
				W			W		W	
				0	0		0		0	0
				0: Input 1: Output			0: Input 1: Output		0: Input 1: Output	
PCFC	Port C function register	0033H (Prohibit RMW)		PC6F	PC5F		PC3F		PC1F	PC0F
				W			W		W	
				0	0		1		0	0
				0: Port 1: INT3 TB0OUT0	0: Port 1: INT2 TA3OUT		0: Port 1: INT0		0: Port 1: INT1 TA1OUT	0: Port 1: TA0IN

Note : When using SI and SCK input function, set P9FC<P92F, P90F> to “0” (Function setting).

I/O port control (3/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
PFCR	Port F control register	003EH (Prohibit RMW)			PF5C	PF4C	PF3C	PF2C	PF1C	PF0C
					W					
					0	0	0	0	0	0
					0: Input 1: Output					
PFFC	Port F function register	003FH (Prohibit RMW)			PF5F		PF3F	PF2F		PF0F
					W		W			W
					0		0	0		0
					0: Port 1: SCLK1 output		0: Port 1: TXD1	0: Port 1: SCLK0 output		0: Port 1: TXD0
PJFC	Port J function register	004FH (Prohibit RMW)	PJ7F	PJ6F	PJ5F	PJ4F	PJ3F	PJ2F	PJ1F	PJ0F
			W							
			0	0	0	0	0	0	0	0
			0: Port 1: SDCKE	0:Port 1: SDUUDQM	0: Port 1: SDULDQM	0: Port 1: SDLUDQM	0: Port 1: SDLLDQM	0: Port 1: $\overline{SDWE}$	0: Port 1: $\overline{SDCAS}$	0: Port 1: $\overline{SDRAS}$
PJFC2	Port J function register 2	004DH (Prohibit RMW)	–	PF6F2	PF5F2	PF4F2	PF3F2	PF2F2	–	–
			W							
			0	0	0	0	0	0	0	0
			Always write "0".	0: <PJ6F> 1: $\overline{SRUUB}$	0: <PJ5F> 1: $\overline{SRULB}$	0: <PJ4F> 1: $\overline{SRLUB}$	0: <PJ3F> 1: $\overline{SRLLB}$	0: <PJ2F> 1: $\overline{SRWR}$	Always write "0".	Always write "0".
PKFC	Port K function register	0053H (Prohibit RMW)		PK6F		PK4F	PK3F	PK2F	PK1F	PK0F
				W		W				
				0		0	0	0	0	0
				0: Port 1: $\overline{ALARM}$ at <PK6> = 1 1: $\overline{MLDALM}$ at <PK6> = 0		0: Port 1: DOFFB	0: Port 1: DLEBCD	0: Port 1: D3BFR	0: Port 1: D2BLP	0: Port 1: D1BSCP
PLCR	Port L control register	0056H (Prohibit RMW)	PL7C	PL6C	PL5C	PL4C	PL3C	PL2C	PL1C	PL0C
			W							
			0	0	0	0	0	0	0	0
			0: Input 1: Output							
PLFC	Port L function register	0057H (Prohibit RMW)	PL7F	PL6F	PL5F	PL4F	PL3F	PL2F	PL1F	PL0F
			W							
			0	0	0	0	0	0	0	0
			0: Port 1: Data bus for LCDC (LD7 to LD0)							

(3) Interrupt control (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE12	INT1& INT2 enable	00D0H	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	Level of request interrupt			1: INT1	Level of request interrupt		
INTE3	INT3 enable	00D1H	–				INT3			
			–	–	–	–	I3C	I3M2	I3M1	I3M0
			–	–			R	R/W		
			–	–	–	–	0	0	0	0
			Always write “0”.				1: INT3	Level of request interrupt		
INTETA01	INTTA0& INTTA1 enable	00D4H	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	Level of request interrupt			1: INTTA0	Level of request interrupt		
INTETA23	INTTA2& INTTA3 enable	00D5H	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
			1: INTTA3	Level of request interrupt			1: INTTA2	Level of request interrupt		
INTETB01	INTTB0& INTTB1 enable	00D8H	INTTB1 (TMRB1)				INTTB0 (TMRB0)			
			ITB1C	ITB1M2	ITB1M1	ITB1M0	ITB0C	ITB0M2	ITB0M1	ITB0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTB1	Level of request interrupt			1: INTTB0	Level of request interrupt		
INTETBO0	INTTBO0 (Overflow) enable	00DAH	–				INTTBO0			
			–	–	–	–	ITBO0C	ITBO0M2	ITBO0M1	ITBO0M0
			–	–			R	R/W		
			–	–	–	–	0	0	0	0
			Always write “0”.				1: INTTBO0	Level of request interrupt		
INTES0	INTRX0& INTTX0 enable	00DBH	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	Level of request interrupt			1: INTRX0	Level of request interrupt		
INTES1	INTRX1& INTTX1 enable	00DCH	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	Level of request interrupt			1: INTRX1	Level of request interrupt		
INTESB0	INTSBE0 enable	00E3H	–				INTSBE0			
			–	–	–	–	ISBE0C	ISBE0M2	ISBE0M1	ISBE0M0
			–	–			R	R/W		
			–	–	–	–	0	0	0	0
			Always write “0”.				1: INTSBE0	Level of request interrupt		
INTEALM01	INTALM0 & INTALM1 enable	00E5H	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	Level of request interrupt			1: INTALM0	Level of request interrupt		
INTEALM23	INTALM2 & INTALM3 enable	00E6H	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	Level of request interrupt			1: INTALM2	Level of request interrupt		

Interrupt control (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTEALM4	INTALM4 enable	00E7H	–				INTALM4			
			–	–	–	–	IA4C	IA4M2	IA4M1	IA4M0
			–	–			R	R/W		
			–	–	–	–	0	0	0	0
			Always write “0”.				1: INTALM4	Level of request interrupt		
INTERTC	INTRTC enable	00E8H	–				INTRTC			
			–	–	–	–	IRC	IRM2	IRM1	IRM0
			–	–			R	R/W		
			–	–	–	–	0	0	0	0
			Always write “0”.				1: INTRTC	Level of request interrupt		
INTECKEY	INTKEY enable	00E9H	–				INTKEY			
			–	–	–	–	IKC	IKM2	IKM1	IKM0
			–	–			R	R/W		
			–	–	–	–	0	0	0	0
			Always write “0”.				1: INTKEY	Level of request interrupt		
INTLCD	INTLCD enable	00EAH	–				INTLCD			
			–	–	–	–	ILCD1C	ILCDM2	ILCDM1	ILCDM0
			–	–			R	R/W		
			–	–	–	–	0	0	0	0
			Always write “0”.				1: INTLCD	Level of request interrupt		
INTES2	INTRX2& INTTX2 enable	00EDH	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	Level of request interrupt			1: INTRX2	Level of request interrupt		
INTEP0	INTP0 enable	00EEH	–				INTP0			
			–	–	–	–	IP0C	IP0M2	IP0M1	IP0M0
			–	–			R	R/W		
			–	–	–	–	0	0	0	0
			Always write “0”.				1: INTP0	Level of request interrupt		

Interrupt control (3/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
INTE0AD	INT0& INTAD enable	00F0H	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	Level of request interrupt			1: INT0	Level of request interrupt		
INTETC01	INTTC0& INTTC1 enable	00F1H	INTTC1 (DMA1)				INTTC0 (DMA0)			
			ITC1C	ITC1M2	ITC1M1	ITC1M0	ITC0C	ITC0M2	ITC0M1	ITC0M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTC1	Level of request interrupt			1: INTTC0	Level of request interrupt		
INTETC23	INTTC2& INTTC3 enable	00F2H	INTTC3 (DMA3)				INTTC2 (DMA2)			
			ITC3C	ITC3M2	ITC3M1	ITC3M0	ITC2C	ITC2M2	ITC2M1	ITC2M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTC3	Level of request interrupt			1: INTTC2	Level of request interrupt		
INTETC45	INTTC4& INTTC5 enable	00F3H	INTTC5 (DMA5)				INTTC4 (DMA4)			
			ITC5C	ITC5M2	ITC5M1	ITC5M0	ITC4C	ITC4M2	ITC4M1	ITC4M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTC5	Level of request interrupt			1: INTTC4	Level of request interrupt		
INTETC67	INTTC6& INTTC7 enable	00F4H	INTTC7 (DMA7)				INTTC6 (DMA6)			
			ITC7C	ITC7M2	ITC7M1	ITC7M0	ITC6C	ITC6M2	ITC6M1	ITC6M0
			R	R/W			R	R/W		
			0	0	0	0	0	0	0	0
			1: INTTC7	Level of request interrupt			1: INTTC6	Level of request interrupt		
SIMC	SIO interrupt mode control	00F5H (Prohibit RMW)						IR2LE	IR1LE	IR0LE
								W		
								1	1	1
								0: INTRX2 edge mode 1: INTRX2 level mode	0: INTRX1 edge mode 1: INTRX1 level mode	0: INTRX0 edge mode 1: INTRX0 level mode
IIMC	Interrupt input mode control	00F6H (Prohibit RMW)			I3EDGE	I2EDGE	I1EDGE	I0EDGE	I0LE	–
					W				R/W	
					0	0	0	0	0	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 l level mode	Always write “0”.
INTWDT	INTWD	00F7H	–				INTWD			
			–	–	–	–	ITCWD	–	–	–
			–	–			R	–		
			–	–	–	–	0	–	–	–
			Always write “0”.				1: INTWD	–		
INTCLR	Interrupt clear control	00F8H (Prohibit RMW)	CLRV7	CLRV6	CLRV5	CLRV4	CLRV3	CLRV2	CLRV1	CLRV0
			W							
			0	0	0	0	0	0	0	0
			Interrupt vector							

(4) DMA controller

Symbol	Name	Address	7	6	5	4	3	2	1	0
DMA0V	DMA0 start vector	0100H			DMA0V5	DMA0V4	DMA0V3	DMA0V2	DMA0V1	DMA0V0
					R/W					
					0	0	0	0	0	0
					DMA0 start vector					
DMA1V	DMA1 start vector	0101H			DMA1V5	DMA1V4	DMA1V3	DMA1V2	DMA1V1	DMA1V0
					R/W					
					0	0	0	0	0	0
					DMA1 start vector					
DMA2V	DMA2 start vector	0102H			DMA2V5	DMA2V4	DMA2V3	DMA2V2	DMA2V1	DMA2V0
					R/W					
					0	0	0	0	0	0
					DMA2 start vector					
DMA3V	DMA3 start vector	0103H			DMA3V5	DMA3V4	DMA3V3	DMA3V2	DMA3V1	DMA3V0
					R/W					
					0	0	0	0	0	0
					DMA3 start vector					
DMA4V	DMA4 start vector	0104H			DMA4V5	DMA4V4	DMA4V3	DMA4V2	DMA4V1	DMA4V0
					R/W					
					0	0	0	0	0	0
					DMA4 start vector					
DMA5V	DMA5 start vector	0105H			DMA5V5	DMA5V4	DMA5V3	DMA5V2	DMA5V1	DMA5V0
					R/W					
					0	0	0	0	0	0
					DMA5 start vector					
DMA6V	DMA6 start vector	0106H			DMA6V5	DMA6V4	DMA6V3	DMA6V2	DMA6V1	DMA6V0
					R/W					
					0	0	0	0	0	0
					DMA6 start vector					
DMA7V	DMA7 start vector	0107H			DMA7V5	DMA7V4	DMA7V3	DMA7V2	DMA7V1	DMA7V0
					R/W					
					0	0	0	0	0	0
					DMA7 start vector					
DMAB	DMA burst	0108H	DBST7	DBST6	DBST5	DBST4	DBST3	DBST2	DBST1	DBST0
			R/W							
			0	0	0	0	0	0	0	0
			1: DMA request on burst mode							
DMAR	DMA request	0109H (Prohibit RMW)	DREQ7	DREQ6	DREQ5	DREQ4	DREQ3	DREQ2	DREQ1	DREQ0
			R/W							
			0	0	0	0	0	0	0	0
			1: DMA request in software							

(5) Memory controller (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
B0CSL	BLOCK0 MEMC control register low	0140H (Prohibit RMW)		B0WW2	B0WW1	B0WW0		B0WR2	B0WR1	B0WR0
				W				W		
				0	1	0		0	1	0
				Write waits 001: 2 states (0 waits) 010: 3 states (1 wait) 101: 4 states (2 waits) 110: 5 states (3 waits) 111: 6 states (4 waits) 011: $\overline{\text{WAIT}}$ pin input mode Others: (Reserved)				Read waits 001: 2 states (0 waits) 010: 3 states (1 wait) 101: 4 states (2 waits) 110: 5 states (3 waits) 111: 6 states (4 waits) 011: $\overline{\text{WAIT}}$ pin input mode Others: (Reserved)		
B0CSH	BLOCK0 MEMCT control register high	0141H (Prohibit RMW)	B0E			B0REC	B0OM1	B0OM0	B0BUS1	B0BUS0
			W			W				
			0			0	0	0	0	0
			CS select 0: Disable 1: enable			0: No insert dummy cycle (Default) 1: Insert dummy cycle	00: ROM/SRAM 01: Reserved 10: Reserved 11: Reserved		Data bus width 00: 8 bits 01: 16 bits 10: 32 bits 11: Reserved	
B1CSL	BLOCK1 MEMC control register low	0144H (Prohibit RMW)		B1WW2	B1WW1	B1WW0		B1WR2	B1WR1	B1WR0
				W				W		
				0	1	0		0	1	0
				Write waits 001: 2 states (0 waits) 010: 3 states (1 wait) 101: 4 states (2 waits) 110: 5 states (3 waits) 111: 6 states (4 waits) 011: $\overline{\text{WAIT}}$ pin input mode Others: (Reserved)				Read waits 001: 2 states (0 waits) 010: 3 states (1 wait) 101: 4 states (2 waits) 110: 5 states (3 waits) 111: 6 states (4 waits) 011: $\overline{\text{WAIT}}$ pin input mode Others: (Reserved)		
B1CSH	BLOCK1 MEMC control register high	0145H (Prohibit RMW)	B1E			B1REC	B1OM1	B1OM0	B1BUS1	B1BUS0
			W			W				
			0			0	0	0	0	0
			CS select 0: Disable 1: Enable			0: No insert dummy cycle (Default) 1: Insert dummy cycle	00: ROM/SRAM 01: Reserved 10: Reserved 11: SDRAM		Data bus width 00: 8 bits 01: 16 bits 10: 32 bits 11: Reserved	
B2CSL	BLOCK2 MEMC control register low	0148H (Prohibit RMW)		B2WW2	B2WW1	B2WW0		B2WR2	B2WR1	B2WR0
				W				W		
				0	1	0		0	1	0
				Write waits 001: 2 states (0 waits) 010: 3 states (1 wait) 101: 4 states (2 waits) 110: 5 states (3 waits) 111: 6 states (4 waits) 011: $\overline{\text{WAIT}}$ pin input mode Others: (Reserved)				Read waits 001: 2 states (0 waits) 010: 3 states (1 wait) 101: 4 states (2 waits) 110: 5 states (3 waits) 111: 6 states (4 waits) 011: $\overline{\text{WAIT}}$ pin input mode Others: (Reserved)		
B2CSH	BLOCK2 MEMC control register high	0149H (Prohibit RMW)	B2E	B2M		B2REC	B2OM1	B2OM0	B2BUS1	B2BUS0
			W			W				
			1	0		0	0	0	0/1	0/1
			CS select 0: Disable 1: Enable	0: 16 Mbytes 1: Sets area		0: No insert dummy cycle (Default) 1: Insert dummy cycle	00: ROM/SRAM 01: Reserved 10: Reserved 11: Reserved		Data bus width 00: 8 bits 01: 16 bits 10: 32 bits 11: Reserved	

Memory controller (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
B3CSL	BLOCK3 MEMC control register low	014CH (Prohibit RMW)		B3WW2	B3WW1	B3WW0		B3WR2	B3WR1	B3WR0
				W				W		
				0	1	0		0	1	0
				Write waits 001: 2 states (0 waits) 010: 3 states (1 wait) 101: 4 states (2 waits) 110: 5 states (3 waits) 111: 6 states (4 waits) 011: $\overline{\text{WAIT}}$ pin input mode Others: (Reserved)				Read waits 001: 2 states (0 waits) 010: 3 states (1 wait) 101: 4 states (2 waits) 110: 5 states (3 waits) 111: 6 states (4 waits) 011: $\overline{\text{WAIT}}$ pin input mode Others: (Reserved)		
B3CSH	BLOCK3 MEMC control register high	014DH (Prohibit RMW)	B3E			B3REC	B3OM1	B3OM0	B3BUS1	B3BUS0
			W			W				
			0			0	0	0	0	0
			CS select 0: Disable 1: Enable			0: No insert dummy cycle (Default) 1: Insert dummy cycle	00: ROM/SRAM 01: Reserved 10: Reserved 11: Reserved		Data bus width 00: 8 bits 01: 16 bits 10: 32 bits 11: Reserved	
BEXCSL	BLOCK EX MEMC control register low	0158H Prohibit RMW		BEXWW2	BEXWW1	BEXWW0		BEXWR2	BEXWR1	BEXWR0
				W				W		
				0	1	0		0	1	0
				Write waits 001: 2 states (0 waits) 010: 3 states (1 wait) 101: 4 states (2 waits) 110: 5 states (3 waits) 111: 6 states (4 waits) 011: $\overline{\text{WAIT}}$ pin input mode Others: (Reserved)				Read waits 001: 2 states (0 waits) 010: 3 states (1 wait) 101: 4 states (2 waits) 110: 5 states (3 waits) 111: 6 states (4 waits) 011: $\overline{\text{WAIT}}$ pin input mode Others: (Reserved)		
BEXCSH	BLOCK EX MEMC control register high	0159H (Prohibit RMW)					BEXOM1	BEXOM0	BEXBUS1	BEXBUS0
							W			
							0	0	0	0
							00: ROM/SRAM 01: Reserved 10: Reserved 11: Reserved		Data bus width 00: 8 bits 01: 16 bits 10: 32 bits 11: Reserved	
PMEMCR	Page ROM control register	0166H				OPGE	OPWR1	OPWR0	PR1	PR0
						R/W				
						0	0	0	1	0
						ROM page access 0: Disable 1: Enable	Wait number on page 00: 1 state (n-1-1-1 mode) 01: 2 states (n-2-2-2 mode) 10: 3 states (n-3-3-3 mode) 11: (Reserved)		Byte number in a page 00: 64 bytes 01: 32 bytes 10: 16 bytes (Default) 11: 8 bytes	

Memory control (3/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
MAMR0	Memory register 0	0142H	M0V20	M0V19	M0V18	M0V17	M0V16	M0V15	M0V14-9	M0V8
			R/W							
			1	1	1	1	1	1	1	1
			0: Compare enable 1: Compare disable							
MSAR0	Memory start address register 0	0143H	M0S23	M0S22	M0S21	M0S20	M0S19	M0S18	M0S17	M0S16
			R/W							
			1	1	1	1	1	1	1	1
			Set start address A23 to A16							
MAMR1	Memory address mask register 1	0146H	M1V21	M1V20	M1V19	M1V18	M1V17	M1V16	MV15-9	M1V8
			R/W							
			1	1	1	1	1	1	1	1
			0: Compare enable 1: Compare disable							
MSAR1	Memory start address register 1	0147H	M1S23	M1S22	M1S21	M1S20	M1S19	M1S18	M1S17	M1S16
			R/W							
			1	1	1	1	1	1	1	1
			Set start address A23 to A16							
MAMR2	Memory register 2	014AH	M2V22	M2V21	M2V20	M2V19	M2V18	M2V17	M2V16	M2V15
			R/W							
			1	1	1	1	1	1	1	1
			0: Compare enable 1: Compare disable							
MSAR2	Memory start address register 2	014BH	M2S23	M2S22	M2S21	M2S20	M2S19	M2S18	M2S17	M2S16
			R/W							
			1	1	1	1	1	1	1	1
			Set start address A23 to A16							
MAMR3	Memory register 3	014EH	M3V22	M3V21	M3V20	M3V19	M3V18	M3V17	M3V16	M3V15
			R/W							
			1	1	1	1	1	1	1	1
			0: Compare enable 1: Compare disable							
MSAR3	Memory start address register 3	014FH	M3S23	M3S22	M3S21	M3S20	M3S19	M3S18	M3S17	M3S16
			R/W							
			1	1	1	1	1	1	1	1
			Set start address A23 to A16							

(6) MMU

Symbol	Name	Address	7	6	5	4	3	2	1	0
LOCAL0	LOCAL0 register	01D0H	L0E					L0EA22	L0EA21	L0EA20
			R/W					R/W		
			0					0	0	0
			Use BANK for LOCAL0 0: Not use 1: Use					Setting BANK number for LOCAL0		
LOCAL1	LOCAL1 register	01D1H	L1E					L1EA23	L1EA22	L1EA21
			R/W					R/W		
			0					0	0	0
			Use BANK for LOCAL1 0: Not use 1: Use					Setting BANK number for LOCAL1		
LOCAL2	LOCAL2 register	01D2H	L2E					L2EA23	L2EA22	L2EA21
			R/W					R/W		
			0					0	0	0
			Use BANK for LOCAL2 0: Disable 1: Enable					Setting BANK number for LOCAL2		
LOCAL3	LOCAL3 register	01D3H	L3E			L3EA26	L3EA25	L3EA24	L3EA23	L3EA22
			R/W			R/W				
			0			0	0	0	0	0
			Use BANK for LOCAL3 0: Disable 1: Enable			00000 to 00011: $\overline{CS2B}$ 00100 to 00111: $\overline{CS2C}$ 01000 to 01011: $\overline{CS2D}$		01100 to 01111: $\overline{CS2E}$ 10000 to 10011: $\overline{CS2F}$ 10100 to 10111: $\overline{CS2G}$ 11000 to 11111: Set prohibition		

(7) Clock gear (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SYSCR0	System clock control register 0	10E0H	XEN	XTEN				WUEF		
			R/W					R/W		
			1	1				0		
			High-frequency oscillator (fc) 0: Stop 1: Oscillation	Low-frequency oscillator (fs) 0: Stop 1: Oscillation				Warm-up timer 0: Write Don't care 1: Write start timer 0: Read end warm up 1: Read do not end warm up		
SYSCR1	System clock control register 1	10E1H					SYSCK	GEAR2	GEAR1	GEAR0
							R/W			
							0	1	0	0
							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)		
SYSCR2	System clock control register 2	10E2H	–		WUPTM1	WUPTM0	HALTM1	HALTM0	SELDRV	DRVE
			R/W		R/W					
			0		1	0	1	1	0	0
			Always write “0”.		Warm-up timer 00: Reserved 01: 2^8/inputted frequency 10: 2^{14}/inputted frequency 11: 2^{16}/inputted frequency		HALT mode 00: Reserved 01: STOP mode 10: IDLE1 mode 11: IDLE2 mode		<DRVE> mode select 0: Stop 1: IDLE1	Pin state control in STOP/ IDLE1 mode 0: I/O off 1: Remains the state before halt

Clock gear (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
EMCCR0	EMC control register 0	10E3H	PROTECT					EXTIN	DRVOSCH	DRVOSCL
			R					R/W		
			0					0	1	1
			Protect flag 0: OFF 1: ON					1: External clock	fc oscillator driver ability 1: Normal 0: Weak	fs oscillator driver ability 1: Normal 0: Weak
EMCCR1	EMC control register 1	10E4H	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							
EMCCR2	EMC control register 2	10E5H								

(8) LCD controller (1/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
LCDMODE	LCD mode register	0200H	BAE	AAE	SCPW1	SCPW0	TA3LCDCK	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: BaseSCP 01: 2 clocks 10: 4 clocks 11: 8 clocks		Select low frequency 0: fs (32 kHz) 1: TA3OUT	Byte-number/common 0: 512 bytes 1: 1024 bytes	Display RAM selection 0: SRAM 1: SDRAM	Mode selection 0: RAM 1: SR
LCDDVM	Divide FRM register	0201H	FMN7	FMN6	FMN5	FMN4	FMN3	FMN2	FMN1	FMN0
			R/W							
			0	0	0	0	0	0	0	0
			Setting DVM bit7 to 0							
LCDSIZE	LCD size register	0202H	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 mode 000: 128 0001: 160 0010: 200 0011: 240 0100: 320 0101: 400 0110: 480 Others: Reserved				Setting the LCD segment number for SR mode 0000: 128 0001: 160 0010: 240 0011: 320 0100: 400 0101: 480 0110: 560 0111: 640 Others: Reserved			
LCDCTL	LCD control register	0203H	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	LD bus output control 0: OFF (= Normal) 1: ON (= ALL 0)	Divided FR mode 0: Disable 1: Enable	Always write "0".	Setting bit 9 for f_{FP} [9:0]	Type selection of LCD driver with built-in RAM 0: Sequential access 1:Random access	Setting bit 8 for f_{FP} [9:0]	Start control in SR mode 0: Stop 1: Start
LCDFFP	LCD frequency register	0204H	FP7	FP6	FP5	FP4	FP3	FP2	FP1	FP0
			R/W							
			0	0	0	0	0	0	0	0
			f_{FP} set value bit7 to 0							
LCDGL	LCD gray level register	0205H							GRAY1	GRAY0
									R/W	
									0	0
									00: Monochrome 01: 4 levels 10: 8 levels 11: 16 levels	

LCD controller (2/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
LCDCM	LCD cursor mode register	0206H	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	0207H				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	0208H				CH4	CH3	CH2	CH1	CH0
						R/W				
						0	0	0	0	0
						Cursor height (Y size) 00000: 1 dot (Min) 11111: 32 dots (Max)				
LCDCP	LCD cursor APB register	0209H					APB3	APB2	APB1	APB0
							R/W			
							0	0	0	0
							Setting bit3 to 0 for cursor absolute position			
LCDCPL	LCD cursor AP register low	020AH	CAP7	CAP6	CAP5	CAP4	CAP3	CAP2	CAP1	CAP0
			R/W							
			0	0	0	0	0	0	0	0
			Setting bit7 to 0 for cursor absolute position							
LCDCPM	LCD cursor AP register medium	020BH	CAP15	CAP14	CAP13	CAP12	CAP11	CAP10	CAP9	CAP8
			R/W							
			0	0	0	0	0	0	0	0
			Setting bit15 to 8 for cursor absolute position							
LCDCPH	LCD cursor AP register high	020CH	CAP23	CAP22	CAP21	CAP20	CAP19	CAP18	CAP17	CAP16
			R/W							
			0	1	0	0	0	0	0	0
			Setting bit23 to 16 for cursor absolute position							

LCD controller (3/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
LSARAM	A area start address register medium	0210H	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 high	0211H	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 medium	0212H	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 high	0213H	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 medium	0214H	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 high	0215H	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 medium	0216H	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 high	0217H	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 low	0218H	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 medium	0219H	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 high	021AH	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 (4/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
LG0L	LCD gray level data setting register low	0220H	–	–	–	–	–	–	–	–
			R/W							
			0	0	0	0	0	0	0	0
LG0H	LCD gray level data setting register high	0221H	–	–	–	–	–	–	–	–
			R/W							
			0	0	0	0	0	0	0	0
LG1L	LCD gray level data setting register low	0222H	–	–	–	–	–	–	–	–
			R/W							
			0	0	0	0	0	0	0	0
LG1H	LCD gray level data setting register high	0223H	–	–	–	–	–	–	–	–
			R/W							
			1	0	0	0	0	0	0	0
LG2L	LCD gray level data setting register low	0224H	–	–	–	–	–	–	–	–
			R/W							
			1	0	0	0	0	0	0	0
LG2H	LCD gray level data setting register high	0225H	–	–	–	–	–	–	–	–
			R/W							
			1	0	0	0	0	0	0	0
LG3L	LCD gray level data setting register low	0226H	–	–	–	–	–	–	–	–
			R/W							
			1	0	0	0	0	0	0	0
LG3H	LCD gray level data setting register high	0227H	–	–	–	–	–	–	–	–
			R/W							
			1	0	0	0	1	0	0	0
LG4L	LCD gray level data setting register low	0228H	–	–	–	–	–	–	–	–
			R/W							
			1	0	0	0	1	0	0	0
LG4H	LCD gray level data setting register high	0229H	–	–	–	–	–	–	–	–
			R/W							
			1	0	0	0	1	0	0	0
LG5L	LCD gray level data setting register low	022AH	–	–	–	–	–	–	–	–
			R/W							
			1	0	0	0	1	0	1	0
LG5H	LCD gray level data setting register high	022BH	–	–	–	–	–	–	–	–
			R/W							
			1	0	0	0	1	0	0	0
LG6L	LCD gray level data setting register low	022CH	–	–	–	–	–	–	–	–
			R/W							
			1	0	0	0	1	0	1	0
LG6H	LCD gray level data setting register high	022DH	–	–	–	–	–	–	–	–
			R/W							
			1	0	0	0	1	0	1	0

LCD controller (5/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
LG7L	LCD gray level data setting register low	022EH	–	–	–	–	–	–	–	–
			R/W							
			1	0	1	0	1	0	1	0
LG7H	LCD gray level data setting register high	022FH	–	–	–	–	–	–	–	–
			R/W							
			1	0	0	0	1	0	1	0
LG8L	LCD gray level data setting register low	0230H	–	–	–	–	–	–	–	–
			R/W							
			1	0	1	0	1	0	1	0
LG8H	LCD gray level data setting register high	0231H	–	–	–	–	–	–	–	–
			R/W							
			1	0	1	0	1	0	1	0
LG9L	LCD gray level data setting register low	0232H	–	–	–	–	–	–	–	–
			R/W							
			0	1	0	1	0	1	0	1
LG9H	LCD gray level data setting register high	0233H	–	–	–	–	–	–	–	–
			R/W							
			1	1	0	1	0	1	0	1
LGAL	LCD gray level data setting register low	0234H	–	–	–	–	–	–	–	–
			R/W							
			1	1	0	1	0	1	0	1
LGAH	LCD gray level data setting register high	0235H	–	–	–	–	–	–	–	–
			R/W							
			1	1	0	1	0	1	0	1
LGBL	LCD gray level data setting register low	0236H	–	–	–	–	–	–	–	–
			R/W							
			1	1	0	1	0	1	0	1
LGBH	LCD gray level data setting register high	0237H	–	–	–	–	–	–	–	–
			R/W							
			1	1	0	1	1	1	0	1
LGCL	LCD gray level data setting register low	0238H	–	–	–	–	–	–	–	–
			R/W							
			1	1	0	1	1	1	0	1
LGCH	LCD gray level data setting register high	0239H	–	–	–	–	–	–	–	–
			R/W							
			1	1	0	1	1	1	0	1
LGDL	LCD gray level data setting register low	023AH	–	–	–	–	–	–	–	–
			R/W							
			1	1	0	1	1	1	0	1
LGDH	LCD gray level data setting register high	023BH	–	–	–	–	–	–	–	–
			R/W							
			1	1	1	1	1	1	0	1

LCD controller (6/6)

Symbol	Name	Address	7	6	5	4	3	2	1	0
LGEL	LCD gray level data setting register low	023CH	–	–	–	–	–	–	–	–
			R/W							
			1	1	0	1	1	1	0	1
LGEH	LCD gray level data setting register high	023DH	–	–	–	–	–	–	–	–
			R/W							
			1	1	0	1	1	1	0	1
LGFL	LCD gray level data setting register low	023EH	–	–	–	–	–	–	–	–
			R/W							
			1	1	1	1	1	1	1	1
LGFH	LCD gray level data setting register high	023FH	–	–	–	–	–	–	–	–
			R/W							
			1	1	1	1	1	1	1	1

(9) SDRAM controller

Symbol	Name	Address	7	6	5	4	3	2	1	0
SDACR	SDRAM address control	0250H	SDINI		SDBUS1	SDBU0		SMUXW1	SMUXW0	SMAC
			R/W		R/W			R/W		
			0		0	0		0	0	0
			Auto initialize 0: Disable 1: Enable		Selecting structure of data bus 00: 16 bits × 1 01: 16 bits × 2 10: 32 bits × 1			Selecting address multiplex type 00: Type A 01: Type B 10: Type C 11: Reserved		SDRAM controller 0: Disable 1: Enable
SDRCR	SDRAM refresh control	0251H	SFRC	SRS2	SRS1	SRS0	SASFRC			SRC
			R/W							R/W
			0	0	0	0	0			0
			Self refresh 0: Disable 1: Enable	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			Interval refresh 0: Disable 1: Enable

(10) 8-bit timer

Symbol	Name	Address	7	6	5	4	3	2	1	0
TA01RUN	TMRA01 RUN register	1100H	TA0RDE				I2TA01	TA01PRUN	TA1RUN	TA0RUN
			R/W				R/W			
			0				0	0	0	0
			Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	TMRA01 Prescaler	Up counter (UC1)	Up counter (UC0)
								0: Stop and clear 1: Run (Count up)		
TA0REG	8-bit timer register 0	1102H Prohibit RMW	–							
			W							
			Undefined							
TA1REG	8-bit timer register 1	1103H Prohibit RMW	–							
			W							
			Undefined							
TA01MOD	TMRA01 mode register	1104H	TA01M1	TA01M0	PWM01	PWM00	TA1CLK1	TA1CLK0	TA0CLK1	TA0CLK0
			R/W							
			0	0	0	0	0	0	0	0
			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	
TA1FFCR	TMRA1 flip-flop control register	1105H Prohibit RMW					TA1FFC1	TA1FFC0	TA1FFIE	TA1FFIS
							W		R/W	
							1	1	0	0
							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
TA23RUN	TMRA23 RUN register	1108H	TA2RDE				I2TA23	TA23PRUN	TA3RUN	TA2RUN
			R/W				R/W			
			0				0	0	0	0
			Double buffer 0: Disable 1: Enable				IDLE2 0: Stop 1: Operate	TMRA23 Prescaler	Up counter (UC3)	Up counter (UC2)
								0: Stop and clear 1: Run (Count up)		
TA2REG	8-bit timer register 2	110AH Prohibit RMW	–							
			W							
			Undefined							
TA3REG	8-bit timer register 3	110BH Prohibit RMW	–							
			W							
			Undefined							
TA23MOD	TMRA23 mode register	110CH	TA23M1	TA23M0	PWM21	PWM20	TA3CLK1	TA3CLK0	TA2CLK1	TA2CLK0
			R/W							
			0	0	0	0	0	0	0	0
			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: TA2TRG 01: ϕT1 10: ϕT16 11: ϕT256		Source clock for TMRA2 00: Reserved 01: ϕT1 10: ϕT4 11: ϕT16	
TA3FFCR	TMRA3 flip-flop control register	110DH Prohibit RMW					TA3FFC1	TA3FFC0	TA3FFIE	TA3FFIS
							W		R/W	
							1	1	0	0
							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

(11) 16-bit timer

Symbol	Name	Address	7	6	5	4	3	2	1	0
TB0RUN	TMRB0 RUN register	1180H	TB0RDE	–			I2TB0	TB0PRUN		TB0RUN
			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	TMRB0 prescaler		Up counter (UC10)
								0: Stop and clear 1: Run (Count up)		
TB0MOD	TMRB0 mode register	1182H Prohibit RMW	–	–	TB0CP0I	TB0CPM1	TB0CPM0	TB0CLE	TB0CLK1	TB0CLK0
			R/W		W	R/W				
			0	0	1	0	0	0	0	0
			Always write "0".		Execute software capture 0: Software capture 1: Undefined	Capture timing 00: Disable 01: Reserved 10: Reserved 11: TA1OUT↑ TA1OUT↓		Control up counter 0: Disable clearing 1: Enable clearing	TMRB0 source clock 00: Reserved 01: ϕT1 10: ϕT4 11: ϕT16	
TB0FFCR	TMRB0 flip-flop control register	1183H Prohibit RMW	–	–	TB0C1T1	TB0C0T1	TB0E1T1	TB0E0T1	TB0FF0C1	TB0FF0C0
			W		R/W				W*	
			1	1	0	0	0	0	1	1
			Always write "11".		TB0FF0 inversion trigger 0: Disable trigger 1: Enable trigger				Control TB0FF0 00: Invert 01: Set 10: Clear 11: Don't care * Always read as 11.	
					Invert when the UC10 value is loaded in to TB0CP1H/L.	Invert when the UC10 value is loaded in to TB0CP0H/L.	Invert when the UC10 value matches the value in TB0RG1H/L.	Invert when the UC10 value matches the value in TB0RG0H/L.		
TB0RG0L	16-bit timer register 0 low	1188H Prohibit RMW	–							
			W							
			Undefined							
TB0RG0H	16-bit timer register 0 high	1189H Prohibit RMW	–							
			W							
			Undefined							
TB0RG1L	16-bit timer register 1 low	118AH Prohibit RMW	–							
			W							
			Undefined							
TB0RG1H	16-bit timer register 1 high	118BH Prohibit RMW	–							
			W							
			Undefined							
TB0CP0L	Capture register 0 low	118CH	–							
			R							
			Undefined							
TB0CP0H	Capture register 0 high	118DH	–							
			R							
			Undefined							
TB0CP1L	Capture register 1 low	118EH	–							
			R							
			Undefined							
TB0CP1H	Capture register 1 high	118FH	–							
			R							
			Undefined							

(12) UART/serial channel (1/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC0BUF	Serial channel 0 buffer register	1200H 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 register	1201H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Clear to 0 after reading)			R/W	
			Undefined	0	0	0	0	0	0	0
			Receive data bit8	Parity 0: Odd 1: Even	Parity 0: Disable 1: Enable	1: Error Overrun	Parity	Framing	0: SCLK0↑ 1: SCLK0↓	0: Baud rate generator 1: SCLK0 pin input
SC0MOD0	Serial channel 0 mode 0 register	1202H	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Trans-mission data bit8	0: CTS disable 1: CTS enable	0: Receive disable 1: Receive enable	Wakeup 0: Disable 1: Enable	00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		00: TA0TRG 01: Baud rate generator 10: Internal clock f_{IO} 11: External clock (SCLK0 input)	
BR0CR	Serial channel 0 baud rate control register	1203H	–	BR0ADDE	BR0CK1	BR0CK0	BR0S3	BR0S2	BR0S1	BR0S0
			R/W							
			0	0	0	0	0	0	0	0
			Always write "0".	(16 – K)/16 divided 0: Disable 1: Enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Divided frequency setting			
BR0ADD	Serial channel 0 K setting register	1204H					BR0K3	BR0K2	BR0K1	BR0K0
							R/W			
							0	0	0	0
							Sets frequency divisor "K" (divided by N + (16 – K)/16).			
SC0MOD1	Serial channel 0 mode 1 register	1205H	I2S0	FDPX0						
			R/W	R/W						
			0	0						
			IDLE2 0: Stop 1: Operate	Duplex 1: Full duplex 0: Half duplex						
SIRCR	IrDA control register	1207H	PLSEL	RXSEL	TXEN	RXEN	SIRWD3	SIRWD2	SIRWD1	SIRWD0
			R/W							
			0	0	0	0	0	0	0	0
			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) + 100ns Can be set: 1 to 14 Can not be set: 0 and 15			

UART/serial channel (2/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC1BUF	Serial channel 1 buffer register	1208H 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 register	1209H	RB8	EVEN	PE	OERR	PERR	FERR	SCLKS	IOC
			R	R/W		R (Clear to 0 after reading)			R/W	
			Undefined	0	0	0	0	0	0	0
			Receive data bit8	Parity 0: Odd 1: Even	Parity 0: Disable 1: Enable	1: Error			0: SCLK1↑ 1: SCLK1↓	0: Baud rate generator 1: SCLK1 pin input
						Overrun	Parity	Framing		
SC1MOD0	Serial channel 1 mode 0 register	120AH	TB8	CTSE	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Trans-mission data bit8	0: CTS disable 1: CTS enable	0: Receive disable 1: Receive enable	Wakeup 0: Disable 1: Enable	00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		00: TA0TRG 01: Baud rate generator 10: Internal clock f_{IO} 11: External clock (SCLK1 input)	
BR1CR	Serial channel 1 baud rate control register	120BH	–	BR1ADDE	BR1CK1	BR1CK0	BR1S3	BR1S2	BR1S1	BR1S0
			R/W							
			0	0	0	0	0	0	0	0
			Always write "0".	(16 – K)/16 divided 0: Disable 1: Enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Divided frequency setting			
BR1ADD	Serial channel 1 K setting register	120CH					BR1K3	BR1K2	BR1K1	BR1K0
							R/W			
							0	0	0	0
							Sets frequency divisor "K" (divided by N + (16 – K)/16)			
SC1MOD1	Serial channel 1 mode 1 register	120DH	I2S1	FDPX1						
			R/W							
			0	0						
			IDLE2 0: Stop 1: Operate	Duplex 1: Full duplex 0: Half duplex						

UART/serial channel (3/3)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SC2BUF	Serial channel 2 buffer register	1210H 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 register	1211H	RB8	EVEN	PE	OERR	PERR	FERR	–	–
			R	R/W		R (Clear to 0 after reading)			R/W	
			Undefined	0	0	0	0	0	0	0
			Receive data bit8	Parity 0: Odd 1: Even	Parity 0: Disable 1: Enable	1:Error			Always write “0”.	Always write “0”.
						Overrun	Parity	Framing		
SC2MOD0	Serial channel 2 mode 0 register	1212H	TB8	–	RXE	WU	SM1	SM0	SC1	SC0
			R/W							
			0	0	0	0	0	0	0	0
			Trans-mission data bit8	Always write “0”.	0: Receive disable 1: Receive enable	Wakeup 0: Disable 1: Enable	00: I/O interface mode 01: 7-bit UART mode 10: 8-bit UART mode 11: 9-bit UART mode		00: TA0REG 01: Baud rate generator 10: Internal clock f_{IO} 11: Reserved	
BR2CR	Serial channel 2 baud rate control register	1213H	–	BR2ADDE	BR2CK1	BR2CK0	BR2S3	BR2S2	BR2S1	BR2S0
			R/W							
			0	0	0	0	0	0	0	0
			Always write “0”.	(16 – K)/16 divided 0: Disable 1: Enable	00: ϕT0 01: ϕT2 10: ϕT8 11: ϕT32		Divided frequency setting			
BR2ADD	Serial channel 2 K setting register	1214H					BR2K3	BR2K2	BR2K1	BR2K0
							R/W			
							0	0	0	0
							Sets frequency divisor “K” (divided by N + (16 – K)/16)			
SC2MOD1	Serial channel 2 mode 1 register	1215H	I2S2	FDPX2						
			R/W							
			0	0						
			IDLE2 0: Stop 1: Operate	Duplex 1: Full duplex 0: Half duplex						

(13) I²C bus/serial channel (1/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SBI0CR1	SBI0 control register 1	I²C mode 1240H (Prohibit RMW)	BC2	BC1	BC0	ACK		SCK2	SCK1	SCK0/ SWRMON
			W			R/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			Acknow -ledge mode 0: Disable 1: Enable		Setting of the divide value "n" 000: 5 001: 6 010: 7 011: 8 100: 9 101: 10 110: 11 111: Reserved		
		SIO mode 1240H (Prohibit RMW)	SIOS	SIOINH	SIOM1	SIOM0		SCK2	SCK1	SCK0
			W					W		
			0	0	0	0		0	0	0
			Transfer 0: Stop 1: Start	Transfer 0: Continue 1: Abort	Transfer mode 00: 8 bits transmit 10: 8 bits transmit/receive 11: 8 bits receive			Setting of the divide value "n" 000: 4 001: 5 010: 6 011: 7 100: 8 101: 9 110: 10 111: External clock SCK0		
SBI0DBR	SBI0 buffer register	1241H (Prohibit RMW)	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
			R (Receiving)/W (Transmission)							
			Undefined							
I2C0AR	I²CBUS0 Address register	1242H (Prohibit RMW)	SA6	SA5	SA4	SA3	SA2	SA1	SA0	ALS
			W							
			0	0	0	0	0	0	0	0
			Setting slave address							Address recognition 0: Enable 1: Disable
SBI0CR2	SBI0 control register 2	I²C mode 1243H (Prohibit RMW)	MST	TRX	BB	PIN	SBIM1	SBIM0	SWRST1	SWRST0
			W							
			0	0	0	1	0	0	0	0
			0: Slave 1: Master	0: Receive 1: Transmit	Start/Stop condition generation 0: Stop 1: Start	INTSBE0 interrupt 0: Request 1: Cancel	Operation mode selection 00: Port mode 01: SIO mode 10: I²C mode 11: Reserved		Software reset generate write "10" and "01", then an internal reset signal is generated.	
		SIO mode 1243H (Prohibit RMW)					SBIM1	SBIM0	–	–
							W			
							0	0	0	0
							Operation mode selection 00: Port mode 01: SIO mode 10: I²C mode 11: Reserved		Always write "0".	Always write "0".
SBI0SR	SBI0 status register	I²C mode 1243H (Prohibit RMW)	MST	TRX	BB	PIN	AL	AAS	AD0	LRB
			R							
			0	0	0	1	0	0	0	0
			0: Slave 1: Master	0: Receive 1: Transmit	Bus status monitor 0: Free 1: Busy	INTSBE0 interrupt 0: Request 1: Cancel	Arbitration lost detection 1: Detect	Slave address match detection monitor 1: Detect	General call detection 1: Detect	Last receive bit monitor 0: 0 1: 1
		SIO mode 1243H (Prohibit RMW)					SIOF	SEF		
							R			
							0	0		
							Transfer status 0: Stopped 1: In progress	Shift status 0: Stopped 1: In progress		

I²C bus/serial channel (2/2)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SBI0BR0	SBI0 baud rate register 0	I²C mode 1244H (Prohibit RMW)	–	I2SBI0						
			W	R/W						
			0	0						
			Always write “0”.	IDLE2 0: Stop 1: Operate						
		SBI mode 1244H (Prohibit RMW)	–	–						
			W	R/W						
			0	0						
			Always write “0”.	Always write “0”.						
SBI0BR1	SBI0 baud rate register 1	1245H (Prohibit RMW)	P4EN	–						
			W							
			0	0						
			Clock control 0: Stop 1: Operate	Always write “0”.						

(14) AD converter

Symbol	Name	Address	7	6	5	4	3	2	1	0
ADMOD0	AD mode control register 0	12B8H	EOCF	ADBF	–	–	ITM0	REPET	SCAN	ADS
			R		R/W					
			0	0	0	0	0	0	0	0
			AD conversion end flag 1: End	AD conversion busy flag 1: Busy	Always write "0".	Always write "0".	0: Every 1 time 1: Every 4 times	Repeat mode 0: Single mode 1: Repeat mode	Scan mode 0: Fixed channel mode 1: Channel scan mode	AD conversion start 1: Start Always read as "0"
ADMOD1	AD mode control register 1	12B9H	VREFON	I2AD	–	–	–	ADCH2	ADCH1	ADCH0
			R/W							
			0	0	0	0	0	0	0	0
			Ladder resistance 0: OFF 1: ON	IDLE2 0: Stop 1: Operate	Always write "0".	Always write "0".	Always write "0".	Input channel 000: AN0 AN0 001: AN1 AN0 → AN1 010: AN2 AN0 → AN1 → AN2 011: AN3 AN0 → AN1 → AN2 → AN3 100: AN4 AN0 → AN1 → AN2 → AN3 → AN4		
ADMOD2	AD mode control register 1	12BAH								ADTRG
										R/W
										0
										AD external trigger start control 0: Disable 1: Enable
ADREG0L	AD result register 0 low	12A0H	ADR01	ADR00						ADR0RF
			R							R
			Undefined							0
ADREG0H	AD result register 0 high	12A1H	ADR09	ADR08	ADR07	ADR06	ADR05	ADR04	ADR03	ADR02
			R							
			Undefined							
ADREG1L	AD result register 1 low	12A2H	ADR11	ADR10						ADR1RF
			R							R
			Undefined							0
ADREG1H	AD result register 1 high	12A3H	ADR19	ADR18	ADR17	ADR16	ADR15	ADR14	ADR13	ADR12
			R							
			Undefined							
ADREG2L	AD result register 2 low	12A4H	ADR21	ADR20						ADR2RF
			R							R
			Undefined							0
ADREG2H	AD result register 2 high	12A5H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
			R							
			Undefined							
ADREG3L	AD result register 3 low	12A6H	ADR31	ADR30						ADR3RF
			R							R
			Undefined							0
ADREG3H	AD result register 3 high	12A7H	ADR39	ADR38	ADR37	ADR36	ADR35	ADR34	ADR33	ADR32
			R							
			Undefined							
ADREG4L	AD result register 4 low	12A8H	ADR21	ADR20						ADR4RF
			R							R
			Undefined							0
ADREG4H	AD result register 4 high	12A9H	ADR29	ADR28	ADR27	ADR26	ADR25	ADR24	ADR23	ADR22
			R							
			Undefined							

(15) Watchdog timer

Symbol	Name	Address	7	6	5	4	3	2	1	0
WDMOD	WDT mode register	1300H	WDTE	WDTP1	WDTP0		–	I2WDT	RESCR	–
			R/W				R/W			
			1	0	0		0	0	0	0
			WDT control 1: Enable	WDT select detecting time 00: $2^{15}/f_{IO}$ 01: $2^{17}/f_{IO}$ 10: $2^{19}/f_{IO}$ 11: $2^{21}/f_{IO}$			Always write “0”.	IDLE2 0: Stop 1: Operate	1: Internally connects WDT out to the reset pin	Always write “0”.
WDCR	WDT control register	1301H Prohibit RMW	–							
			W							
			–							
			B1H: WDT disable code 4EH: WDT clear code							

(16) RTC (Real time clock)

Symbol	Name	Address	7	6	5	4	3	2	1	0
SECR	Second register	1320H		SE6	SE5	SE4	SE3	SE2	SE1	SE0
				R/W						
				Undefined						
			"0" is read.	40 sec	20 sec	10 sec	8 sec	4 sec	2 sec	1 sec
MINR	Minute register	1321H		MI6	MI5	MI4	MI3	MI2	MI1	MI0
				R/W						
				Undefined						
			"0" is read.	40 min.	20 min.	10 min.	8 min.	4 min.	2 min.	1 min.
HOURR	Hour register	1322H			HO5	HO4	HO3	HO2	HO1	HO0
					R/W					
					Undefined					
			"0" is read.		20 hour (PM/AM)	10 hour	8 hour	4 hour	2 hour	1 hour
DAYR	Day register	1323H						WE2	WE1	WE0
								R/W		
								Undefined		
			"0" is read.					W2	W1	W0
DATER	Date register	1324H			DA5	DA4	DA3	DA2	DA1	DA0
					R/W					
					Undefined					
			"0" is read.		20 day	10 day	8 day	4 day	2 day	1 day
MONTHR	Month register	1325H				MO4	MO3	MO2	MO1	MO0
						R/W				
						Undefined				
		PAGE 0	"0" is read.			10 month	8 month	4 month	2 month	1 month
		PAGE 1	"0" is read.							0: Indicator for 12 hours 1: Indicator for 24 hours
YEARR	Year register	1326H	YE7	YE6	YE5	YE4	YE3	YE2	YE1	YE0
			R/W							
			Undefined							
		PAGE 0	80 year	40 year	20 year	10 year	8 year	4 year	2 year	1 year
		PAGE 1	"0" is read.						Leap year setting 00: Leap year 01: One year after 10: Tow year after 11: Three year after	
PAGER	Page register	1327H Prohibit RMW	INTENA			ADJUST	ENATMR	ENAALM		PAGE
			R/W			W	R/W			R/W
			0			Undefined				Undefined
			INTRTC 0:disable 1:enable	"0" is read.		0:Don't care 1:Adjust	Clock 0:disable 1:enable	Alarm 0:disable 1:enable	"0" is read.	PAGE setting
RESTR	Reset register	1328H Prohibit RMW	DIS1HZ	DIS16HZ	RSTTMR	RSTALM	RE3	RE2	RE1	RE0
			W							
			Undefined							
			1 Hz 0:disable 1:enable	16 Hz 0:disable 1:enable	1: Reset clock	1: Reset alarm	Always write "0".			

(17) Melody/alarm generator

Symbol	Name	Address	7	6	5	4	3	2	1	0
ALM	Alarm-pattern register	1330H	AL8	AL7	AL6	AL5	AL4	AL3	AL2	AL1
			R/W							
			0	0	0	0	0	0	0	0
			Alarm pattern set							
MELALMC	Melody/ Alarm control register	1331H	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	0
			Free-run counter control 00: Hold 01: Restart 10: Clear 11: Clear & start		Alarm frequency invert 1: Invert	Always write “0”.				Output frequency 0: Alarm 1: Melody
MELFL	Melody frequency L-register	1332H	ML7	ML6	ML5	ML4	ML3	ML2	ML1	ML0
			R/W							
			0	0	0	0	0	0	0	0
			Melody frequency set (Low 8 bits)							
MELFH	Melody frequency H-register	1333H	MELON				ML11	ML10	ML9	ML8
			R/W				R/W			
			0				0	0	0	0
			Melody counter control 0: Stop & clear 1: Start				Melody frequency set (Upper 4 bits)			
ALMINT	Alarm interrupt enable register	1334H			–	IALM4E	IALM3E	IALM2E	IALM1E	IALM0E
					R/W	R/W				
					0	0	0	0	0	0
					Always write “0”.	INTALM4 to INTALM0 alarm interrupt enable				

6. Port Section Equivalent Circuit Diagram

- Reading the circuit diagram

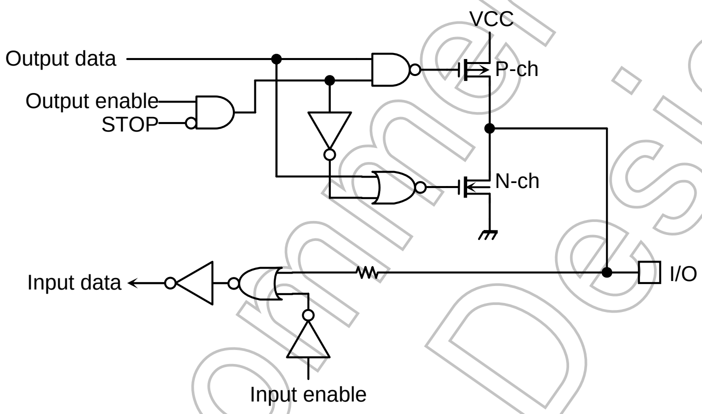
Basically, the gate symbols written are the same as those used for the standard CMOS logic IC [74HCXX] series.

The dedicated signal is described below.

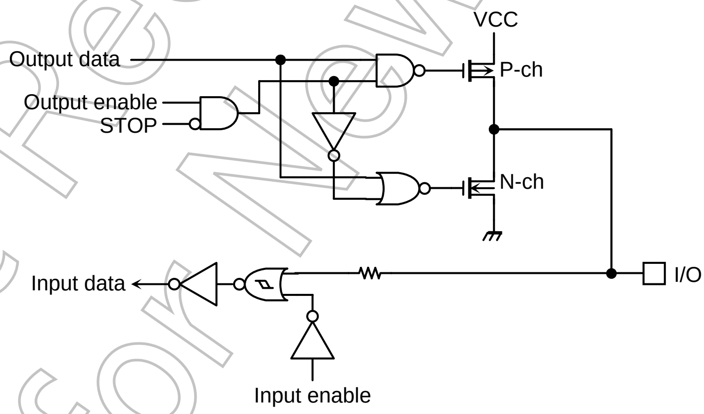
STOP: This signal becomes active "1" when the HALT mode setting register is set to the STOP mode and the CPU executes the HALT instruction. When the drive enable bit <DRVE> is set to "1", however, STOP remains at "0".

The input protection resistance ranges from several tens of ohms to several hundreds of ohms.

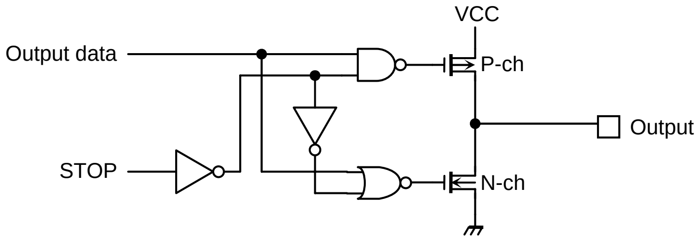
- D0 to D7, P1 (D8 to D15), P2 (D16 to D23), P3 (D24 to D31), P4 (A0 to A7), P5 (A8 to A15), P6 (A16 to A23), P76 and PL0 to PL7



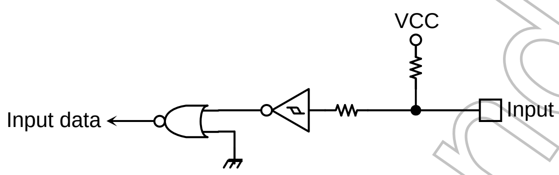
- P90, P96, PC0, PC1, PC3, PC5, PC6, PF1, PF2, PF4, PF5



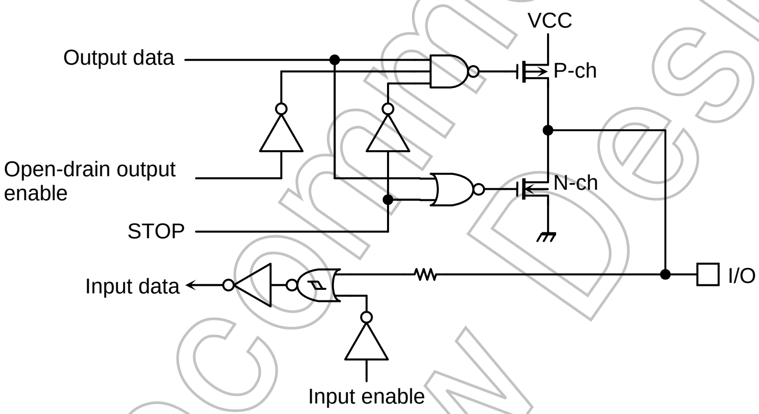
- P70 to P75, P80 to P87, PJ0 to PJ7, PK0 to PK4 and PK6



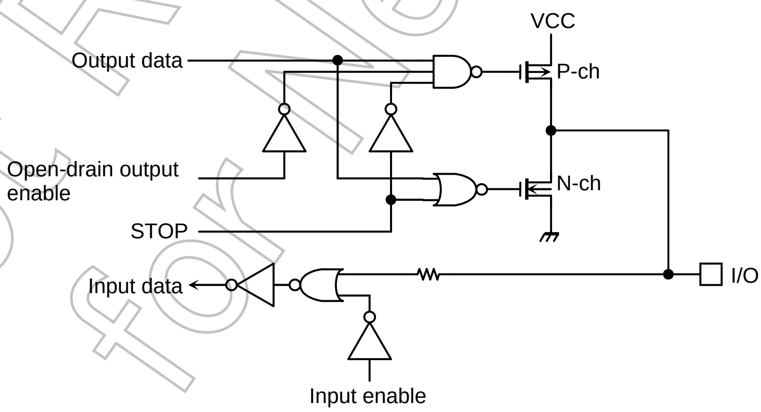
- PA



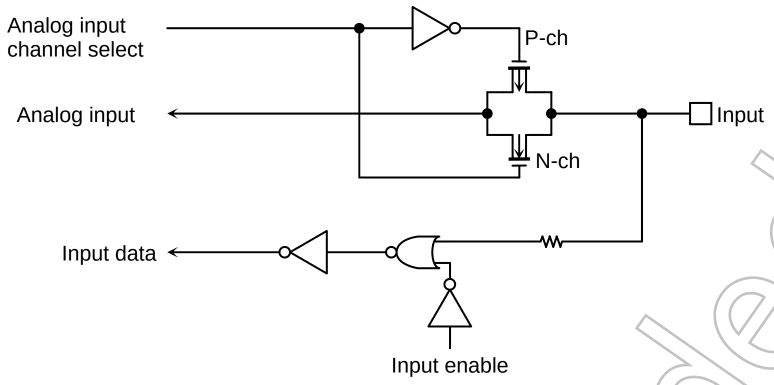
- P91 (SO/SDA), P92 (SI/SCL), P93 and P94



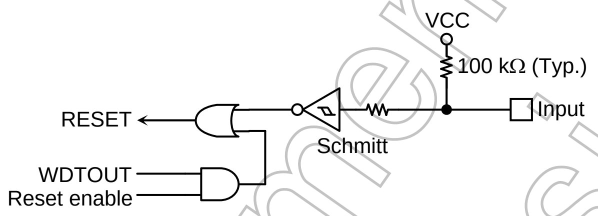
- P95 (TXD2), PF0, PF3



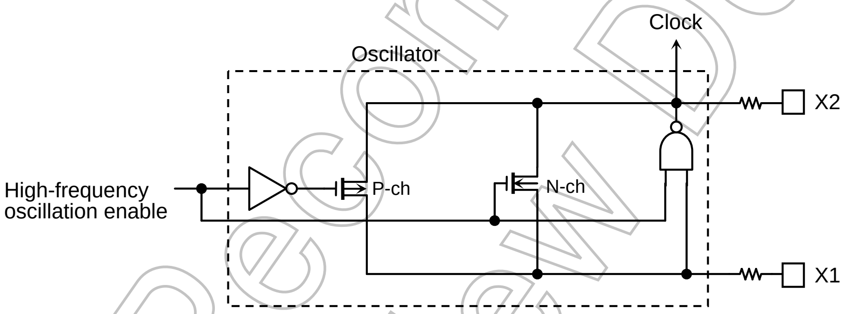
■ PG (AN0 to AN4)



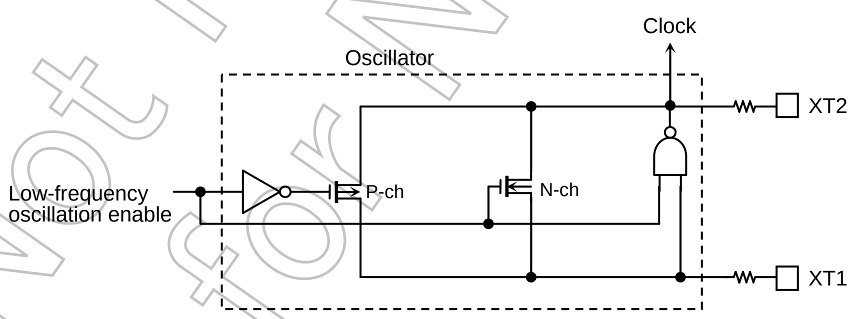
■ $\overline{\text{RESET}}$



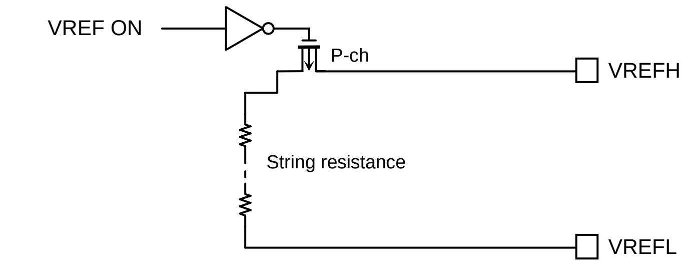
■ X1 and X2



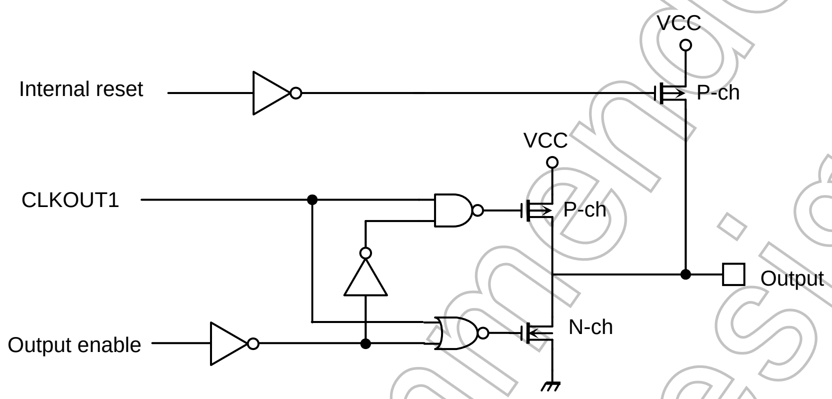
■ XT1 and XT2



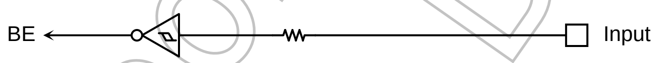
- VREFH and VREFL



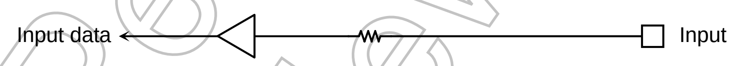
- SDCLK



- $\overline{\mathrm{BE}}$



- AM0 to AM1



7. Points to Note and Restrictions

7.1 Notation

(1) The notation for built-in/I/O registers is as follows register symbol <Bit symbol>

Example: TA01RUN<TA0RUN> denotes bit TA0RUN of register TA01RUN.

(2) Read-modify-write instructions (RMW)

An instruction in which the CPU reads data from memory and writes the data to the same memory location in one instruction.

Example 1: SET 3, (TA01RUN); Set bit3 of TA01RUN.

Example 2: INC 1, (100H); Increment the data at 100H.

- Examples of read-modify-write instructions on the TLCS-900

Exchange instruction

EX (mem), R	

Arithmetic operations

ADD (mem), R/#	ADC (mem), R/#
SUB (mem), R/#	SBC (mem), R/#
INC #3, (mem)	DEC #3, (mem)

Logic operations

AND (mem), R/#	OR (mem), R/#
XOR (mem), R/#	

Bit manipulation operations

STCF#3/A, (mem)	RES #3, (mem)
SET #3, (mem)	CHG #3, (mem)
TSET#3, (mem)	

Rotate and shift operations

RLC (mem)	RRC (mem)
RL (mem)	RR (mem)
SLA (mem)	SRA (mem)
SLL (mem)	SRL (mem)
RLD (mem)	RRD (mem)

(3) fc, fs, f_{FPH}, f_{SYS} and one state

The clock frequency input on ins X1 and 2 is called f_{OSCH}. The clock selected by DFMCR0<ACT1:0> is called fc.

The clock selected by SYSCR1<SYSCK> is called f_{FPH}. The clock frequency give by f_{FPH} divided by 2 is called f_{SYS}.

One cycle of f_{SYS} is referred to as one state.

7.2 Points to Note

(1) AM0 and AM1 pins

This pin is connected to the V_{CC} or the V_{SS} pin. Do not alter the level when the pin is active.

(2) EMU0 and EMU1

Open pins.

(3) Reserved address areas

The TMP92C820 does not have any reserved areas.

(4) Warm-up counter

The warm-up counter operates when STOP mode is released, even if the system is using an external oscillator. As a result a time equivalent to the warm-up time elapses between input of the release request and output of the system clock.

(5) Programmable pull-up resistance

The programmable pull-up resistor can be turned ON/OFF by a program when the ports are set for use as input ports. When the ports are set for use as output ports, they cannot be turned ON/OFF by a program. The data registers (e.g., P5) are used to turn the pull-up/pull-down resistors ON/OFF. Consequently read-modify-write instructions are prohibited.

(6) Watchdog timer

The watchdog timer starts operation immediately after a reset is released. When the watchdog timer is not to be used, disable it.

(7) AD converter

The string resistor between the VREFH and VREFL pins can be cut by a program so as to reduce power consumption. When STOP mode is used, disable the resistor using the program before the HALT instruction is executed.

(8) CPU (Micro DMA)

Only the "LDC cr, r" and "LDC r, cr" instructions can be used to access the control registers in the CPU. (e.g., The transfer source address register (DMASn).)

(9) Undefined SFR

The value of an undefined bit in an SFR is undefined when read.

(10) POP SR instruction

Please execute the POP SR instruction during DI condition.

(11) Releasing the HALT mode by requesting an interruption

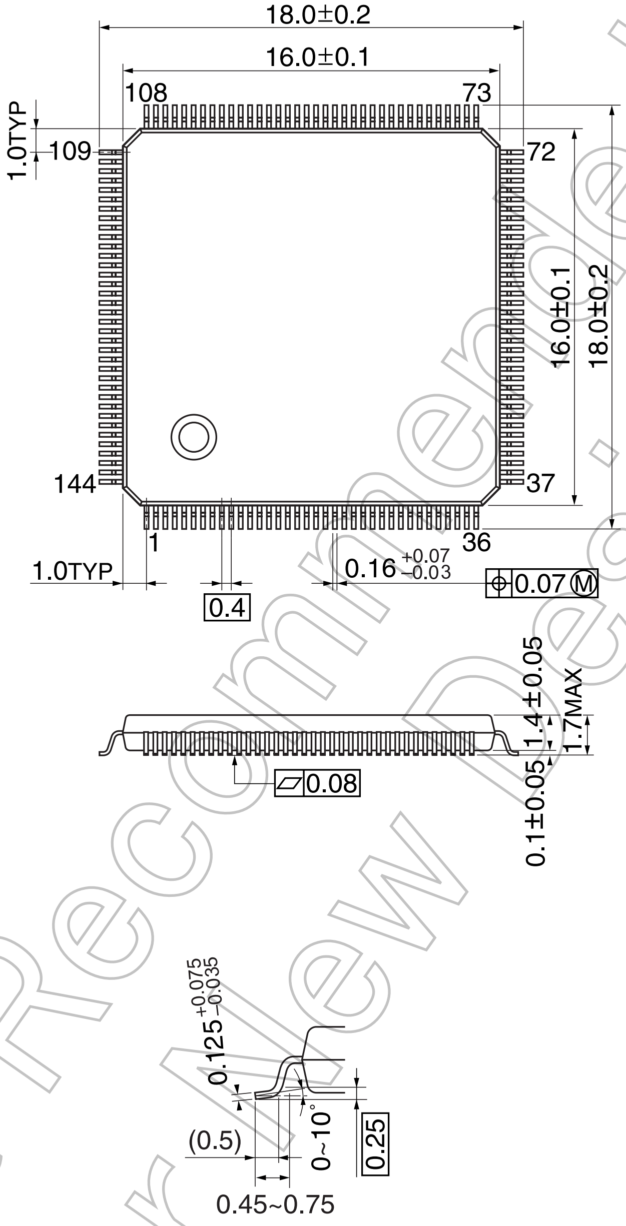
Usually, interrupts can release all halts status. However, the interrupts = (INT0 to INT3, INTKEY, INTRTC and INTALM0 to INTALM4) which can release the HALT mode may not be able to do so if they are input during the period CPU is shifting to the HALT mode (for about 3 clocks of f_{FPH}) with IDLE1 or STOP mode (IDLE2 is not applicable). (In this case, an interrupt request is kept on hold internally.)

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

8. Package Dimensions

P-LQFP144-1616-0.40C

Unit: mm



Note: Palladium plating