TOSHIBA

TOSHIBA Original CMOS 32-Bit Microcontroller

TLCS-900/H1 Series

TMP92CH21FG

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”.

CMOS 32-bit Microcontroller

TMP92CH21FG/JTMP92CH21

1. Outline and Device Characteristics

The TMP92CH21 is a high-speed advanced 32-bit Microcontroller developed for controlling equipment which processes mass data.

The TMP92CH21 has a high-performance CPU (900/H1 CPU) and various built-in I/Os.

The TMP92CH21FG is housed in a 144-pin flat package. The JTMP92CH21 is a chip form product.

Device characteristics are as follows:

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

- Compatible with TLCS-900/L1 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 f_{SYS} = 20 MHz)

(3) Internal memory
- Internal RAM: 16 Kbytes (can be used for program, data and display memory)
- Internal ROM: 8 Kbytes (used as boot program)
  Possible downloading of user program through either USB,
  UART or NAND flash.

(4) External memory expansion
- Expandable up to 512 Mbytes (shared program/data area)
- Can simultaneously support 8,- 16- or 32-bit width external data bus
  ... dynamic data bus sizing
- Separate bus system

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

(6) 8-bit timers: 4 channels

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

(8) General-purpose serial interface: 2 channels
- UART/synchronous mode: 2 channels (channel 0 and 1)
- IrDA ver.1.0 (115 kbps) mode selectable: 1 channel (channel 0)

(9) USB (universal serial bus) controller: 1 channel
- Compliant with USB ver.1.1
- Full-speed (12 MHz) (Low-speed is not supported.)
- Endpoints spec
  Endpoint 0: Control 64 bytes* 1-FIFO
  Endpoint 1: BULK (out) 64 bytes* 2-FIFO
  Endpoint 2: BULK (in) 64 bytes* 2-FIFO
  Endpoint 3: Interrupt (in) 8 bytes* 1-FIFO
- Descriptor RAM: 384 bytes

(10) I^2S (Inter-IC sound) interface: 1 channel
- I^2S bus mode/SIO mode selectable (Master, transmission only)
- 32-byte FIFO buffer

(11) LCD controller
- Supports up to 4096 color for TFT, 256 color, 16, 8, 4 gray levels and B/W for STN
- Shift register/built-in RAM LCD driver

(12) SDRAM controller: 1 channel
- Supports 16 M, 64 M, 128 M, 256 M, and up to 512-Mbit SDR (Single Data Rate)-SDRAM
- Possible to execute instruction on SDRAM

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

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

(15) 10-bit AD converter: 4 channels

(16) Touch screen interface
- Available to reduce external components

(17) Watchdog timer

(18) Melody/alarm generator
- Melody: Output of clock 4 to 5461 Hz
- Alarm: Output of 8 kinds of alarm pattern and 5 kinds of interval interrupt

(19) MMU
- Expandable up to 512 Mbytes (3 local area/8 bank method)
- Independent bank for each program, read data, write data and LCD display data

(20) Interrupts: 50 interrupt
- 9 CPU interrupts: Software interrupt instruction and illegal instruction
- 34 internal interrupts: Seven selectable priority levels
- 7 external interrupts: Seven selectable priority levels (6-edge selectable)

(21) Input/output ports: 82 pins (Except Data bus (16bit), Address bus (24bit) and $\overline{RD}$ pin)

(22) NAND flash interface: 2 channels
- Direct NAND flash connection capability
- ECC calculation (for SLC- type)

(23) Stand-by function
- Three HALT modes: IDLE2 (programmable), IDLE1, STOP
- Each pin status programmable for stand-by mode

(24) Triple-clock controller
- Clock doubler (PLL) supplies 48 MHz for USB, 36 MHz system-clock for others
- Clock gear function: Select high-frequency clock fc to fc/16
- RTC (fs = 32.768 kHz)

(25) Operating voltage:
- VCC = 3.0 V to 3.6 V (fc max = 40 MHz)
- VCC = 2.7 V to 3.6 V (fc max = 27 MHz)

(26) Package:
- 144-pin QFP (P-LQFP144 -1616-0.40C)
- 144-pin chip form is also available. For details, contact your local Toshiba sales representative.

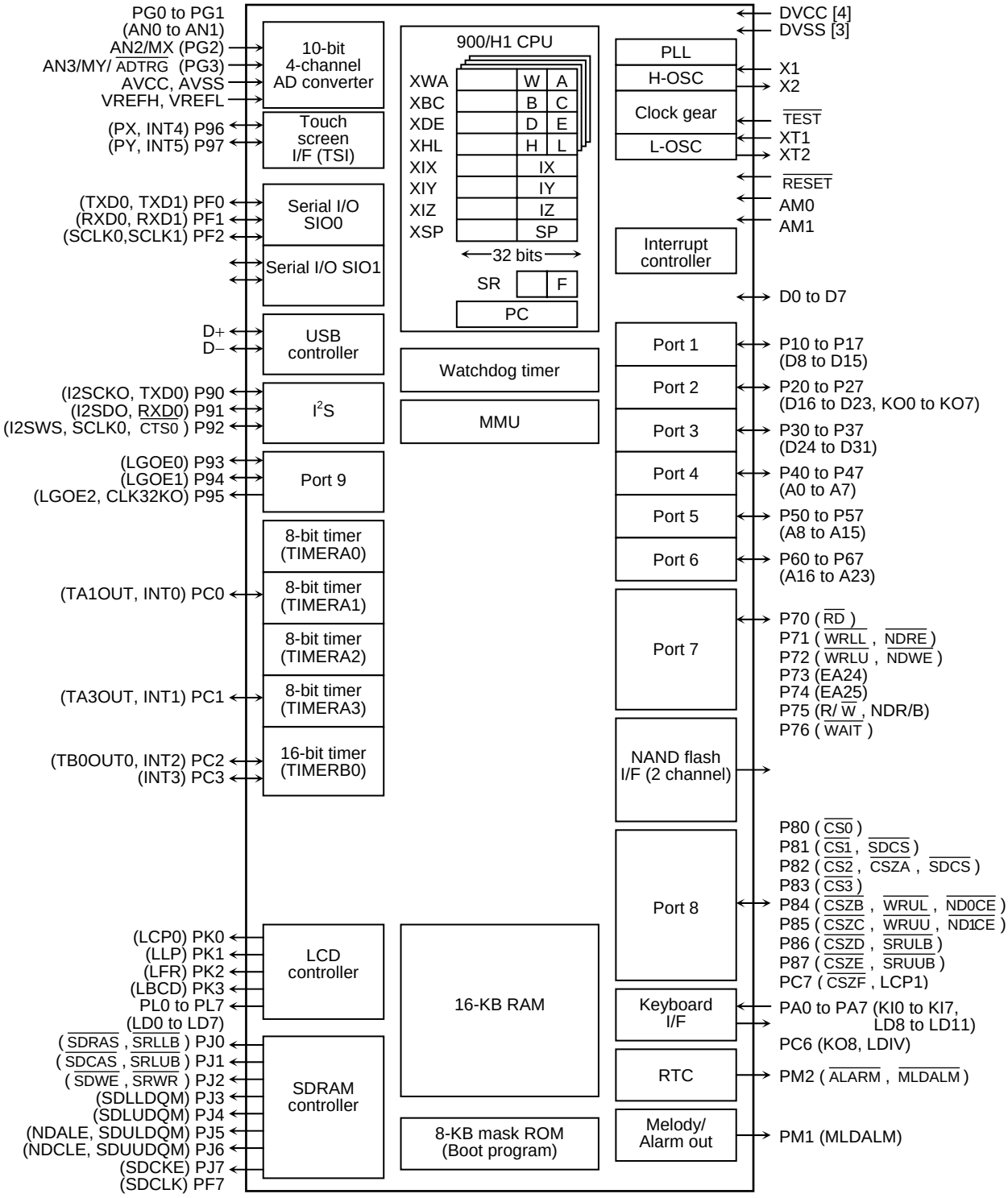


Figure 1.1 TMP92CH21 Block Diagram

2. Pin Assignment and Functions

The assignment of input/output pins for the TMP92CH21FG, their names and functions are as follows:

2.1 Pin Assignment

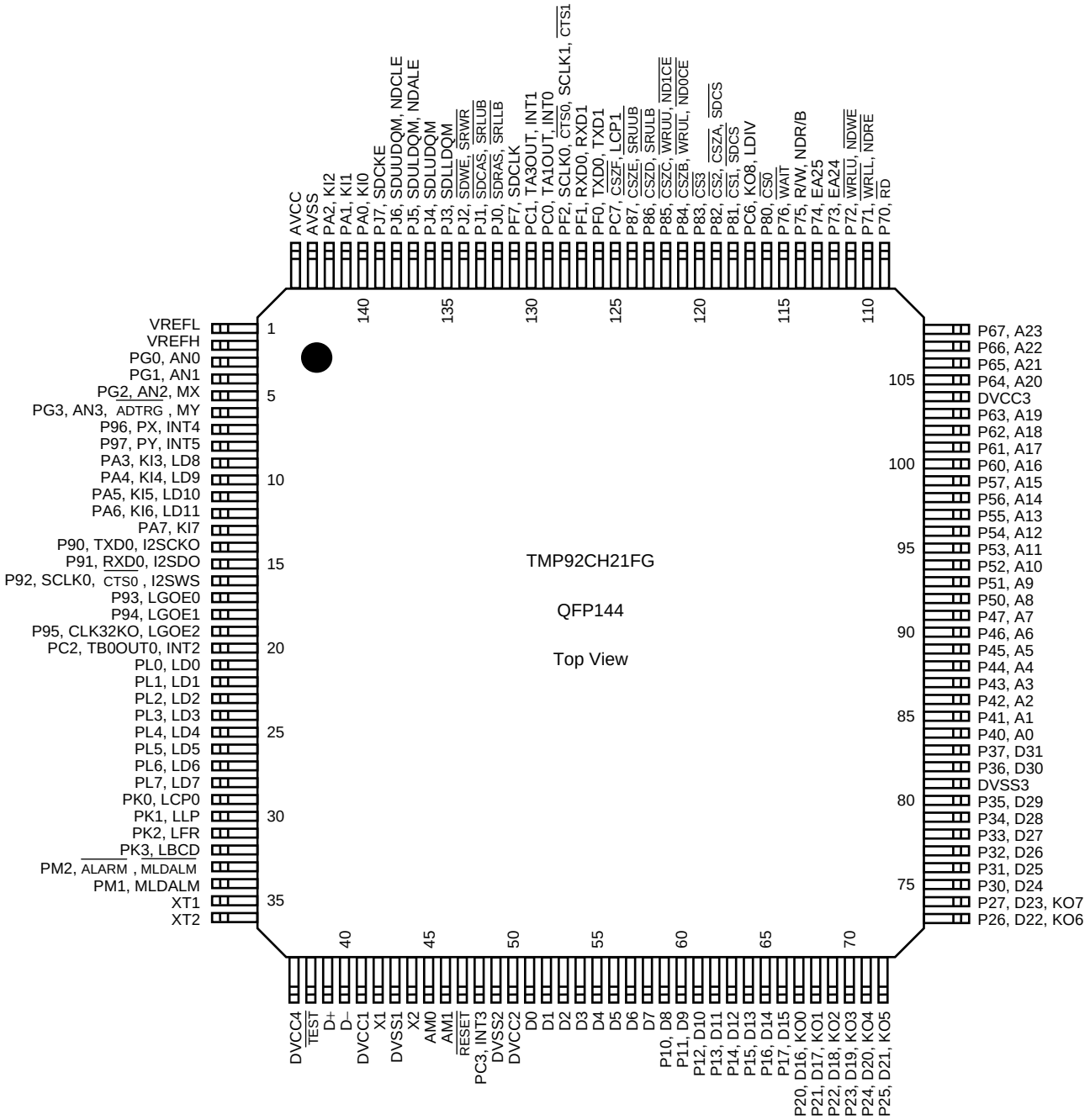


Figure 2.1.1 Pin Assignment Diagram (144-pin QFP)

2.2 PAD Assignment

(Chip size 5.98 mm × 6.42 mm)

Table 2.2.1 Pad Assignment Diagram (144-pin chip)

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	−2852	2671	49	DVSS2	−488	−3072	97	P55	2848	815
2	VREFH	−2852	2546	50	DVCC2	−338	−3072	98	P56	2848	941
3	PG0	−2852	2421	51	D0	−200	−3072	99	P57	2848	1066
4	PG1	−2852	2296	52	D1	−75	−3072	100	P60	2848	1191
5	PG2	−2852	2171	53	D2	49	−3072	101	P61	2848	1316
6	PG3	−2852	2045	54	D3	174	−3072	102	P62	2848	1441
7	P96	−2852	1920	55	D4	300	−3072	103	P63	2848	1566
8	P97	−2852	1795	56	D5	425	−3072	104	DVCC3	2848	1692
9	PA3	−2852	1270	57	D6	550	−3072	105	P64	2848	1823
10	PA4	−2852	1145	58	D7	675	−3072	106	P65	2848	1974
11	PA5	−2852	1020	59	P10	800	−3072	107	P66	2848	2130
12	PA6	−2852	895	60	P11	925	−3072	108	P67	2848	2292
13	PA7	−2852	769	61	P12	1050	−3072	109	P70	2460	3065
14	P90	−2852	644	62	P13	1176	−3072	110	P71	2295	3065
15	P91	−2852	519	63	P14	1301	−3072	111	P72	2127	3065
16	P92	−2852	394	64	P15	1426	−3072	112	P73	1964	3065
17	P93	−2852	269	65	P16	1551	−3072	113	P74	1807	3065
18	P94	−2852	144	66	P17	1676	−3072	114	P75	1654	3065
19	P95	−2852	18	67	P20	1801	−3072	115	P76	1506	3065
20	PC2	−2852	−106	68	P21	1927	−3072	116	P80	1361	3065
21	PL0	−2852	−231	69	P22	2052	−3072	117	PC6	1226	3065
22	PL1	−2852	−356	70	P23	2177	−3072	118	P81	1101	3065
23	PL2	−2852	−481	71	P24	2303	−3072	119	P82	976	3065
24	PL3	−2852	−606	72	P25	2460	−3072	120	P83	851	3065
25	PL4	−2852	−732	73	P26	2848	−2279	121	P84	726	3065
26	PL5	−2852	−857	74	P27	2848	−2138	122	P85	600	3065
27	PL6	−2852	−982	75	P30	2848	−1982	123	P86	475	3065
28	PL7	−2852	−1107	76	P31	2848	−1831	124	P87	350	3065
29	PK0	−2852	−1232	77	P32	2848	−1687	125	PC7	225	3065
30	PK1	−2852	−1357	78	P33	2848	−1562	126	PF0	100	3065
31	PK2	−2852	−1482	79	P34	2848	−1437	127	PF1	−24	3065
32	PK3	−2852	−1608	80	P35	2848	−1311	128	PF2	−150	3065
33	PM2	−2852	−1892	81	DVSS3	2848	−1186	129	PC0	−275	3065
34	PM1	−2852	−2017	82	P36	2848	−1061	130	PC1	−400	3065
35	XT1	−2852	−2142	83	P37	2848	−936	131	PF7	−525	3065
36	XT2	−2852	−2444	84	P40	2848	−811	132	PJ0	−650	3065
37	DVCC4	−2465	−3072	85	P41	2848	−686	133	PJ1	−775	3065
38	$\overline{\text{TEST}}$	−2339	−3072	86	P42	2848	−560	134	PJ2	−901	3065
39	D+	−2062	−3072	87	P43	2848	−435	135	PJ3	−1026	3065
40	D−	−1875	−3072	88	P44	2848	−310	136	PJ4	−1151	3065
41	DVCC1	−1598	−3072	89	P45	2848	−185	137	PJ5	−1276	3065
42	X1	−1472	−3072	90	P46	2848	−60	138	PJ6	−1401	3065
43	DVSS1	−1347	−3072	91	P47	2848	65	139	PJ7	−1526	3065
44	X2	−1126	−3072	92	P50	2848	190	140	PA0	−1652	3065
45	AM0	−1001	−3072	93	P51	2848	315	141	PA1	−1777	3065
46	AM1	−876	−3072	94	P52	2848	440	142	PA2	−1902	3065
47	$\overline{\text{RESET}}$	−750	−3072	95	P53	2848	565	143	AVSS	−2275	3065
48	PC3	−625	−3072	96	P54	2848	690	144	AVCC	−2400	3065

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 (1/5)

Pin Name	Number of Pins	I/O	Function
D0 to D7	8	I/O	Data: Data bus 0 to 7
P10 to P17	8	I/O	Port 1: I/O port input or output specifiable in units of bits
D8 to D15		I/O	Data: Data bus 8 to 15
P20 to P27	8	I/O	Port 2: I/O port input or output specifiable in units of bits
D16 to D23		I/O	Data: Data bus 16 to 23
KO0 to KO7		Output	Key output 0 to 7: Pins used of key-scan strobe (Open-drain output programmable)
P30 to P37	8	I/O	Port 3: I/O port input or output specifiable in units of bits
D24 to D31		I/O	Data24: Data bus 24 to 31
P40 to P47	8	Output	Port 4: Output port
A0 to A7		Output	Address: Address bus 0 to 7
P50 to P57	8	Output	Port 5: Output port
A8 to A15		Output	Address: Address bus 8 to 15
P60 to P67	8	I/O	Port 6: I/O port input or output specifiable in units of bits
A16 to A23		Output	Address: Address bus 16 to 23
P70	1	Output	Port70: Output port
$\overline{RD}$		Output	Read: Outputs strobe signal to read external memory
P71	1	I/O	Port 71: I/O port
$\overline{WRLL}$		Output	Write: Output strobe signal for writing data on pins D0 to D7
$\overline{NDRE}$		Output	NAND flash read: Outputs strobe signal to read external NAND flash
P72	1	I/O	Port 72: I/O port
$\overline{WRLU}$		Output	Write: Output strobe signal for writing data on pins D8 to D15
$\overline{NDWE}$		Output	Write Enable for NAND flash
P73	1	Output	Port 73: Output port
EA24		Output	Extended Address 24
P74	1	Output	Port 74: Output port
EA25		Output	Extended Address 25
P75	1	I/O	Port 75: I/O port
R/$\overline{W}$		Output	Read/Write: 1 represents read or dummy cycle; 0 represents write cycle
NDR/B		Input	NAND flash ready (1)/Busy (0) input
P76	1	I/O	Port 76: I/O port
$\overline{WAIT}$		Input	Wait: Signal used to request CPU bus wait

Table 2.3.2 Pin Names and Functions (2/5)

Pin Name	Number of Pins	I/O	Function
P80 $\overline{CS0}$	1	Output Output	Port80: Output port Chip select 0: Outputs “low” when address is within specified address area
P81 $\overline{CS1}$ $\overline{SDCS}$	1	Output Output Output	Port81: 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 address area
P82 $\overline{CS2}$ $\overline{CSZA}$ $\overline{SDCS}$	1	Output Output Output Output	Port82: Output port Chip select 2: Outputs “Low” when address is within specified address area Expand chip select: ZA: Outputs “0” when address is within specified address area Chip select for SDRAM: Outputs “0” when address is within SDRAM address area
P83 $\overline{CS3}$	1	Output Output	Port83: Output port Chip select 3: Outputs “low” when address is within specified address area
P84 $\overline{WRUL}$ $\overline{CSZB}$ $\overline{ND0CE}$	1	Output Output Output Output	Port84: Output port Write: Output strobe signal for writing data on pins D16 to D23 Expand chip select: ZB: Outputs “0” when address is within specified address area Chip select for NAND flash 0: Outputs “0” when NAND flash 0 is enabled
P85 $\overline{WRUU}$ $\overline{CSZC}$ $\overline{ND1CE}$	1	Output Output Output Output	Port85: Output port Write: Output strobe signal for writing data on pins D24 to D31 Expand chip select: ZC: Outputs “0” when address is within specified address area Chip select for NAND flash 1: Outputs “0” when NAND flash 1 is enabled
P86 $\overline{CSZD}$ $\overline{SRULB}$	1	Output Output Output	Port86: Output port Expand chip select: ZD: outputs “0” when address is within specified address area Data enable for SRAM on pins D16 to D23
P87 $\overline{CSZE}$ $\overline{SRUUB}$	1	Output Output Output	Port87: Output port Expand chip select: ZE: Outputs “0” when address is within specified address area Data enable for SRAM on pins D24 to D31
P90 TXD0 I2SCKO	1	I/O Output Output	Port90: I/O port Serial 0 send data: Open-drain output programmable I^2S clock output
P91 RXD0 I2SDO	1	I/O Input Output	Port91: I/O port (Schmitt-input) Serial 0 receive data I^2S data output
P92 SCLK0 $\overline{CTS0}$ I2SWS	1	I/O I/O Input Output	Port92: I/O port (Schmitt-input) Serial 0 clock I/O Serial 0 data send enable (Clear to send) I^2S word select output
P93 LGOE0	1	I/O Output	Port93: I/O port Output enable-0 for external TFT-LCD driver
P94 LGOE1	1	I/O Output	Port94: I/O port Output enable-1 for external TFT-LCD driver
P95 CLK32KO LGOE2	1	Output Output Output	Port95: Output port Output fs (32.768 kHz) clock Output enable-2 for external TFT-LCD driver
P96 INT4 PX	1	Input Input Output	Port 96: Input port (Schmitt-input) Interrupt request pin4: Interrupt request with programmable rising/falling edge X-Plus: Pin connectted to X+ for touch screen panel
P97 INT5 PY	1	Input Input Output	Port 97: Input port (Schmitt-input) Interrupt request pin5: Interrupt request with programmable rising/falling edge Y-Plus: Pin connectted to Y+ for touch screen panel
PA0 to PA2 KI0 to KI2	3	Input Input	Port: A0 to A2 port: Pin used to input ports (Schmitt input, with pull-up resistor) Key input 0 to 2: Pin used for key-on wakeup 0 to 2
PA3 to PA6 KI3 to KI6 LD8 to LD11	4	Input Input Output	Port: A3 to A6 port: Pin used to input ports (Schmitt input, with pull-up resistor) Key input 3 to 6: Pin used for key-on wakeup 3 to 6 Data bus 8 to 11for LCD driver
PA7 KI7	1	Input Input	Port: A7 port: Pin used to input ports (Schmitt input, with pull-up resistor) Key input 7: Pin used for key-on wakeup 7

Table 2.3.3 Pin Names and Functions (3/5)

Pin Name	Number of Pins	I/O	Function
PC0	1	I/O	Port C0: I/O port (Schmitt-input)
INT0		Input	Interrupt request pin 0: Interrupt request pin with programmable level/rising/falling edge
TA1OUT		Output	8-bit timer 1 output: Timer 1 output
PC1	1	I/O	Port C1: I/O port (Schmitt-input)
INT1		Input	Interrupt request pin 1: Interrupt request pin with programmable rising/falling edge
TA3OUT		Output	8-bit timer 3 output: Timer 3 output
PC2	1	I/O	Port C2: I/O port (Schmitt-input)
INT2		Input	Interrupt request pin 2: Interrupt request pin with programmable rising/falling edge
TB0OUT0		Output	Timer B0 output
PC3	1	I/O	Port C3: I/O port (Schmitt-input)
INT3		Input	Interrupt request pin 3: Interrupt request pin with programmable rising/falling edge
PC6	1	I/O	Port C6: I/O port
KO8		Output	Key Output 8: Pin used of key-scan strobe (Open-drain output programmable)
LDIV		Output	Data invert enable for external TFT-LCD driver
PC7	1	I/O	Port C7: I/O port
$\overline{\text{CSZF}}$		Output	Expand chip select: ZF: Outputs “0” when address is within specified address area
LCP1		Output	Shift-clock-1 for external TFT-LCD driver
PF0	1	I/O	Port F0: I/O port (Schmitt-input)
TXD0		Output	Serial 0 send data: Open-drain output programmable
TXD1		Output	Serial 1 send data: Open-drain output programmable
PF1	1	I/O	Port F1: I/O port (Schmitt-input)
RXD0		Input	Serial 0 receive data
RXD1		Input	Serial 1 receive data
PF2	1	I/O	Port F2: I/O port (Schmitt-input)
SCLK0		I/O	Serial 0 clock I/O
$\overline{\text{CTS0}}$		Input	Serial 0 data send enable (Clear to send)
SCLK1		I/O	Serial 1 clock I/O
$\overline{\text{CTS1}}$		Input	Serial 1 data send enable (Clear to send)
PF7	1	Output	Port F7: Output port
SDCLK		Output	Clock for SDRAM (When SDRAM is not used, SDCLK can be used as system clock)
PG0 to PG1	2	Input	Port G0 to G1 port: Pin used to input ports
AN0 to AN1		Input	Analog input 0 to 1: Pin used to Input to AD conveter
PG2	1	Input	Port G2 port: Pin used to input ports
AN2		Input	Analog input 2: Pin used to Input to AD conveter
MX		Output	X-Minus: Pin connectted to X– for touch screen panel
PG3	1	Input	Port G3 port: Pin used to input ports
AN3		Input	Analog input 3: Pin used to input to AD conveter
MY		Output	Y-Minus: Pin connectted to Y– for touch screen panel
$\overline{\text{ADTRG}}$		Intput	AD trigger: Signal used to request AD start

Table 2.3.4 Pin Names and Functions (4/5)

Pin Name	Number of Pins	I/O	Function
PJ0	1	Output	Port J0: Output port
$\overline{\text{SDRAS}}$		Output	Row address strobe for SDRAM
$\overline{\text{SRLLB}}$		Output	Data enable for SRAM on pins D0 to D7
PJ1	1	Output	Port J1: Output port
$\overline{\text{SDCAS}}$		Output	Column address strobe for SDRAM
$\overline{\text{SRLUB}}$		Output	Data enable for SRAM on pins D8 to D15
PJ2	1	Output	Port J2: Output port
$\overline{\text{SDWE}}$		Output	Write enable for SDRAM
$\overline{\text{SRWR}}$		Output	Write for SRAM: Strobe signal for writing data
PJ3	1	Output	Port J3: Output port
SDLLDQM		Output	Data enable for SDRAM on pins D0 to D7
PJ4	1	Output	Port J4: Output port
SDLUDQM		Output	Data enable for SDRAM on pins D8 to D15
PJ5	1	I/O	Port J5: I/O port
SDULDQM		Output	Data enable for SDRAM on pins D16 to D23
NDALE		Output	Address latch enable for NAND flash
PJ6	1	I/O	Port J6: I/O port
SDUUDQM		Output	Data enable for SDRAM on pins D24 to D31
NDCLE		Output	Command latch enable for NAND flash
PJ7	1	Output	Port J7: Output port
SDCKE		Output	Clock enable for SDRAM
PK0	1	Output	Port K0: Output port
LCP0		Output	LCD driver output pin
PK1	1	Output	Port K1: Output port
LLP		Output	LCD driver output pin
PK2	1	Output	Port K2: Output port
LFR		Output	LCD driver output pin
PK3	1	Output	Port K3: Output port
LBCD		Output	LCD driver output pin
PL0 to PL3	4	Output	Port L0 to L3: Output port
LD0 to LD3		Output	Data bus for LCD driver
PL4 to PL7	4	I/O	Port L4 to L7: I/O port
LD4 to LD7		Output	Data bus for LCD driver
$\overline{\text{TEST}}$	1	Input	Connect to VCC.
PM1	1	Output	Port M1: Output port
MLDALM		Output	Melody/alarm output pin
PM2	1	Output	Port M2: Output port
$\overline{\text{ALARM}}$		Output	RTC alarm output pin
$\overline{\text{MLDALM}}$		Output	Melody/alarm output pin (inverted)

Note: The output functions SDULDQM, NDALE of PJ5-pin and SDUUDQM, NDCLE of PJ6-pin cannot be used simultaneously. Therefore, 32-bit SDRAM and NAND-Flash cannot be used at the same time.

Table 2.3.5 Pin Names and Functions (5/5)

Pin Name	Number of Pins	I/O	Function
D+, D–	2	I/O	USB-data connecting pin Connect pull-up resistor to both pins to avoid through current when USB is not in use.
AM0, AM1	2	Input	Operation mode: Fix to AM1 = “0”, AM0 = “1” for 16-bit external bus starting Fix to AM1 = “1”, AM0 = “0” for 32-bit external bus starting Fix to AM1 = “1”, AM0 = “1” for BOOT (32-bit internal MROM) starting
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 TMP92CH21 (with pull-up resistor, Schmitt input)
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	4	–	Power supply pins (All V_{CC} pins should be connected to the power supply pin)
DVSS	3	–	GND pins (0 V) (All pins should be connected to GND (0 V))

Note: Use a 9.0 MHz oscillator at pins X1/X2 when USB is used.

3. Operation

This section describes the basic components, functions and operation of the TMP92CH21.

3.1 CPU

The TMP92CH21 contains an advanced high-speed 32-bit CPU (TLCS-900/H1 CPU)

3.1.1 CPU Outline

The TLCS-900/H1 CPU is a high-speed, high-performance CPU based on the TLCS-900/L1 CPU. The TLCS-900/H1 CPU has an expanded 32-bit internal data bus to process instructions more quickly.

The following is an outline of the CPU:

Table 3.1.1 TMP92CH21 Outline

Parameter	TMP92CH21
Width of CPU address bus	24 bits
Width of CPU data bus	32 bits
Internal operating frequency	Max 20 MHz
Minimum bus cycle	1-clock access (50 ns at f_{SYS} = 20MHz)
Internal RAM	32-bit 1-clock access
Internal boot ROM	32-bit 2-clock access
Internal I/O	8- or 16-bit 2-clock access or 8- or 16-bit 5 to 6-clock access
External SRAM, Masked ROM	8- or 16- or 32-bit 2-clock access (waits can be inserted)
External SDRAM	16- or 32-bit min. 1-clock access
External NAND flash	8-bit min. 4-clock access (waits can be inserted)
Minimum instruction execution cycle	1-clock (50 ns at f_{SYS} =20MHz)
Conditional jump	2-clock (100 ns at f_{SYS} =20MHz)
Instruction queue buffer	12 bytes
Instruction set	Compatible with TLCS-900/L1 (LDX instruction is deleted)
CPU mode	Maximum mode only
Micro DMA	8 channels

3.1.2 Reset Operation

When resetting the TMP92CH21, 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 fc = 40 MHz).

At reset, since the clock doubler (PLL) is bypassed and the clock-gear is set to 1/16, the system clock operates at 1.25 MHz (fc = 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 <IFF2:0> of the status register (SR) to 111 (thereby setting the interrupt level mask register to level 7).
- Clears bits <RFP1:0> of the status register to 00 (there by selecting register bank 0).

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 shown in the "Special Function Register" table 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 reset is released as soon as external reset is released.

Memory controller operation cannot be ensured until the power supply becomes stable after power-on reset. External RAM data provided before turning on the TMP92CH21 may be corrupted because the control signals are unstable until the power supply becomes stable after power on reset.

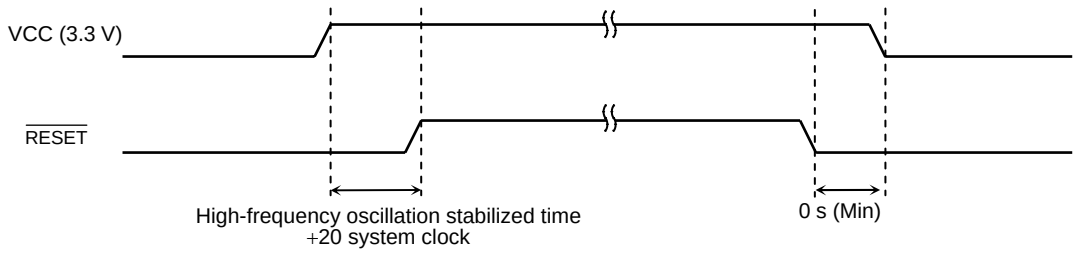


Figure 3.1.1 Power on Reset Timing Example

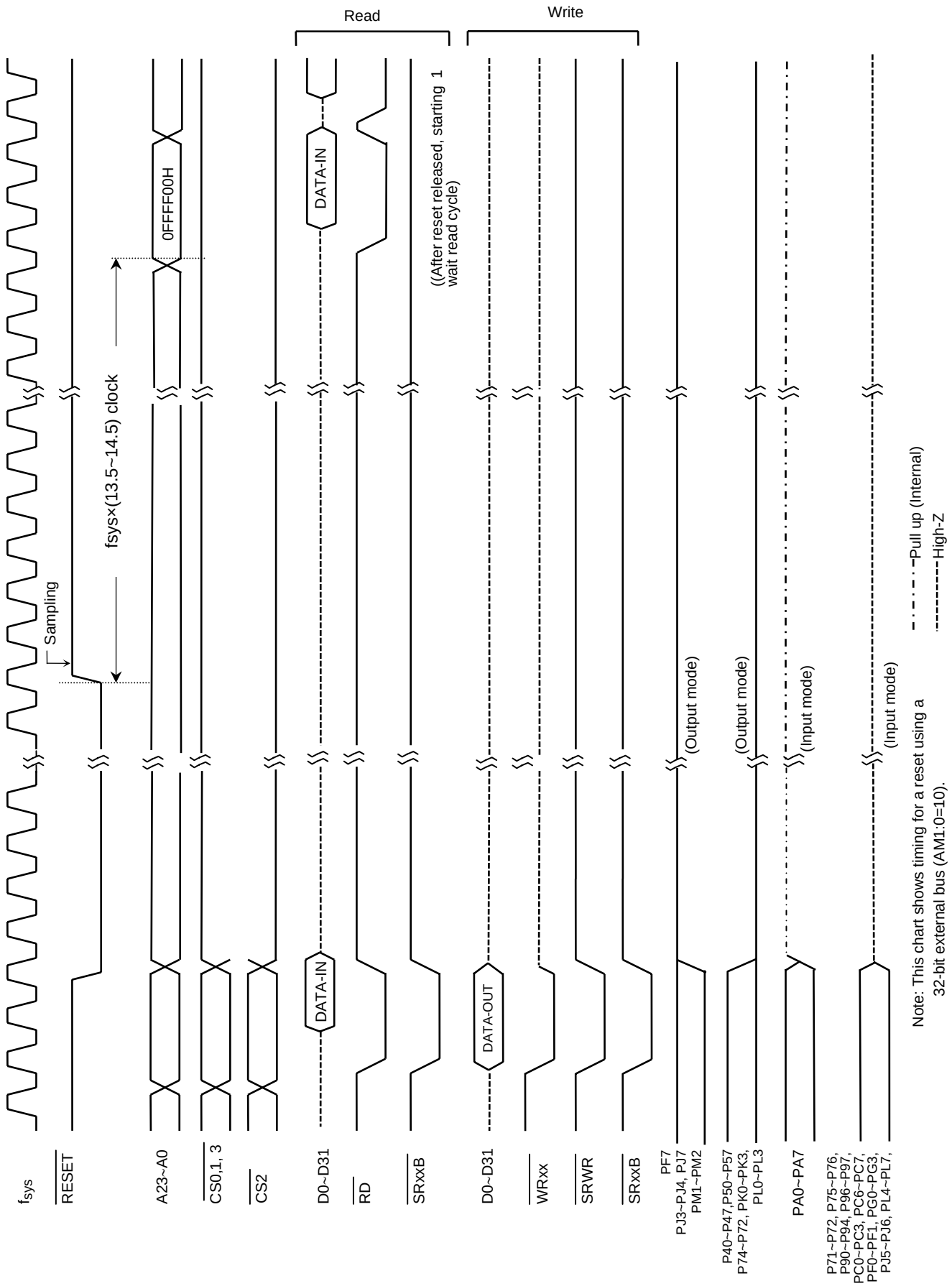


Figure 3.1.2 TMP92CH21 Reset Timing Chart

3.1.3 Setting of AM0 and AM1

Set AM1 and AM0 pins as shown in Table 3.1.2 according to system usage.

Table 3.1.2 Operation Mode Setup Table

Operation Mode	Mode Setup Input Pin		
	$\overline{\text{RESET}}$	AM1	AM0
16-bit external bus starting (MULTI 16 mode)		0	1
32-bit external bus starting (MULTI 32 mode)		1	0
Boot (32-bit internal MROM) starting (BOOT mode)		1	1

3.2 Memory Map

Figure 3.2.1 is a memory map of the TMP92CH21.

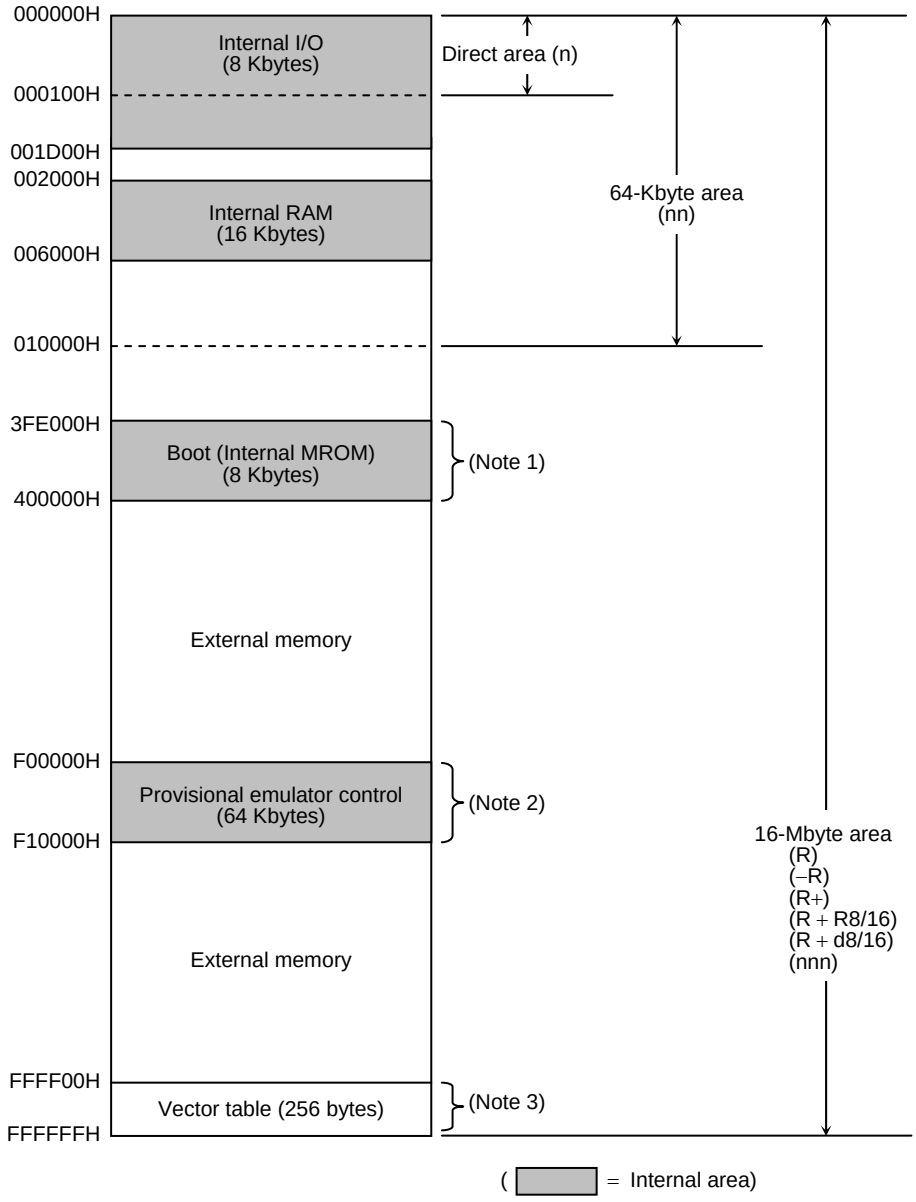


Figure 3.2.1 Memory Map

Note 1: Boot program (Internal MROM) is mapped only for BOOT mode. For other starting modes, its area (3FE000H to 3FFFFFH) is mapped to external-memory.

Note 2: The Provisional emulator control area, mapped F00000H to F0FFFFH after reset, is for emulator use and so is not available. When emulator $\overline{WR}$ signal and $\overline{RD}$ signal are asserted, this area is accessed. Ensure external memory is used.

Note 3: Do not use the last 16-byte area (FFFFF0H to FFFFFFH). This area is reserved for an emulator.

3.21.2 Hardware Specification for Internal Boot ROM

(1) Memory map

Figure 3.21.1 shows a memory map of BOOT mode.

An 8-Kbyte ROM is built-in and it is mapped to address 3FE000H to 3FFFFFH.

In MULTI mode, the boot ROM is not mapped and its area is mapped as an external area.

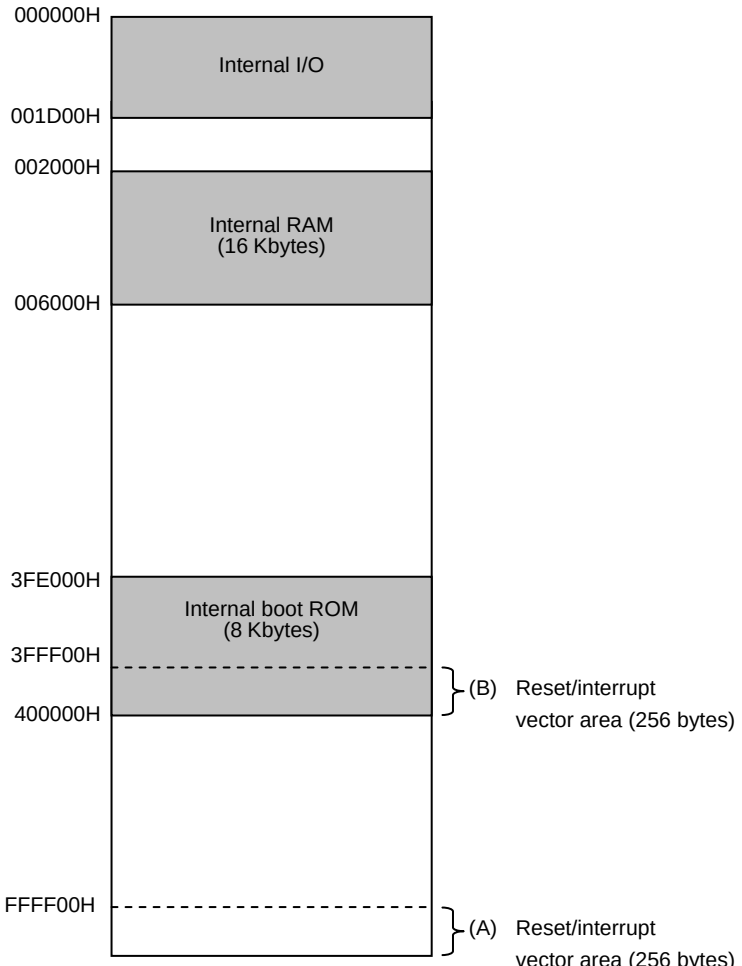


Figure 3.21.1 Memory Map of BOOT Mode

(2) Reset/interrupt address conversion circuit

A reset/interrupt vector address conversion circuit is included.

This function allows for individual reset/interrupt vector areas. For details, refer to section 3.6.5, Internal Boot ROM Control.

(3) Clearing boot ROM

After boot sequence in BOOT mode, the application system program may continue to run without reset asserting. In this case, any external memory which is mapped to address 3FE000H to 3FFFFFH cannot be accessed because the boot ROM is assigned here.

So, an internal boot ROM can be cleared by setting BROMCR<ROMLESS> to "1".

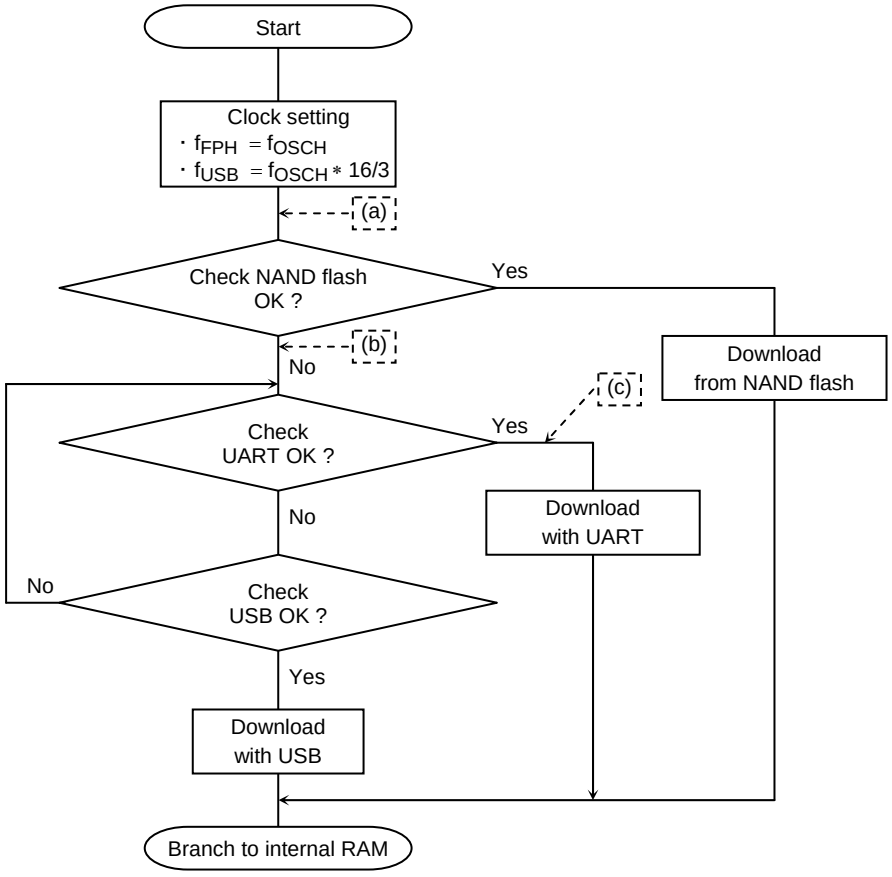
For the details, refer to section 3.6.5, Internal Boot ROM Control.

3.21.3 Outline of Boot Operation

There are 3 downloading methods: NAND flash, UART and USB.

After reset, a boot program in the boot ROM operates as shown in the Figure 3.21.2 flow chart.

Internal RAM use is the same regardless of downloading method, and is shown in Figure 3.21.3.



Note 1: When USB downloading is used, a special USB device driver and application software are needed on the PC.

Note 2: When UART downloading is used, special application software is needed on the PC.

Note 3: (a), (b) and (c) on the flow chart show the points at which external port pins are set. Refer to Table 3.21.3 for details.

Figure 3.21.2 Flow Chart Outline of Internal Boot ROM

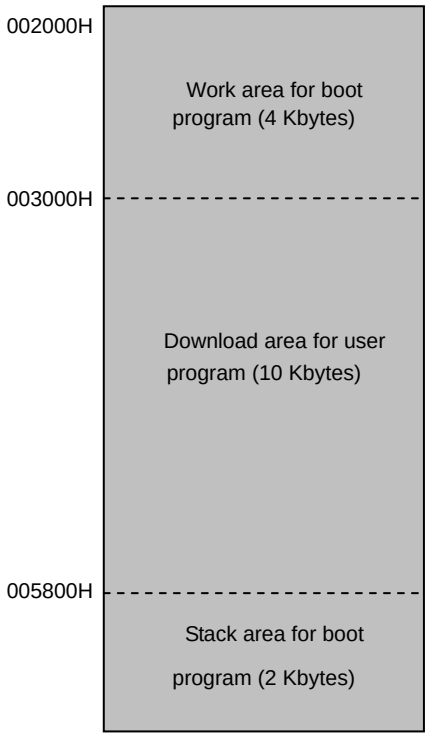


Figure 3.21.3 Internal RAM Use

(1) Port setting

The boot program port settings are shown in Table 3.21.3, and Table 3.21.4 shows PCB design. These port settings must be carefully noted when designing an application system.

The remaining ports are not set, so they maintain their status after reset.

Table 3.21.3 Port Setting

Port		Function	I/O	Port Setting by Boot Program		
				(a)	(b)	(c)
NAND flash	P71	$\overline{\text{NDRE}}$	Output	Set to the function pin shown left	No change from (a)	No change from (a)
	P72	$\overline{\text{NDWE}}$	Output			
	P75	NDR/$\overline{\text{B}}$	Input			
	P84	$\overline{\text{ND0CE}}$	Output			
	PJ5	NDALE	Output			
	PJ6	NDCLE	Output			
	–	D7 to D0	I/O	No change		
UART	PF0	TXD1	Output	No change to input port status after reset	No change from (a)	Set to the TXD1 output pin
	PF1	RXD1	Input	Set to the RXD1 input pin		No change from (a)
USB	–	D +	I/O	No change		
	–	D –	I/O			
	PC6	PUCTL	Output	No change to input port status after reset	Set to the output port pin	No change from (b)

Table 3.21.4 How to Design PCB

Port		Function	I/O	Boot Method		
				NAND flash	UART	USB
NAND flash	P71	$\overline{\text{NDRE}}$	Output	Connect to NAND flash and pull-up by 100 kΩ resistor because this pin is changed to input port by reset.	Not affected by UART boot. If the NAND flash is not used in the system, ensure no conflict with the I/O direction shown left.	Not affected by USB boot. If the NAND flash is not used in the system, ensure no conflict with the I/O direction shown left.
	P72	$\overline{\text{NDWE}}$	Output			
	P75	NDR/$\overline{\text{B}}$	Input	Connect to NAND flash and pull-up by 2 kΩ resistor because R/B pin of NAND flash has open-drain output buffer.		
	P84	$\overline{\text{ND0CE}}$	Output	Connect to NAND flash.		
	PJ5	NDALE	Output			
	PJ6	NDCLE	Output			
	–	D7 to D0	I/O			
UART	PF0	TXD1	Output	Not affected by NAND flash boot.	Connect to level shifter.	Not affected by USB boot.
	PF1	RXD1	Input			Pull-up by 100 kΩ to avoid UART executing.
USB	–	D +	I/O	Not affected NAND flash boot.	Not affected by UART boot.	Connect to USB connector, add dumping resistor (27Ω) and 1.5 kΩ pull-up which can be switched ON/OFF.
	–	D –	I/O			Connect to USB connector and add dumping resistor (27Ω).
	PC6	PUCTL	Output			Used to control ON/OFF pull-up resistor of D + pin. The switch should be ON by “1”. As this pin changes to input port by reset, add100 kΩ pull-down.

Note 1: When booting method is either NAND flash or UART and USB is used in the system, ensure the D + pin pull-up resistor is not on in the BOOT mode.

Note 2: When booting method is USB, do not start UART application software on the PC.

Note 3: When booting method is UART, do not connect the USB connector.

(2) I/O registers setting by boot program

Table 3.21.5 shows I/O register setting by boot program.

Take particular note of these set values when using an application system program which continues to run without asserting a reset after a boot sequence is executed .

Also take note of the status of the CPU registers and internal RAM following execution of a boot sequence.

Table 3.21.5 I/O Register Setting by Boot Program

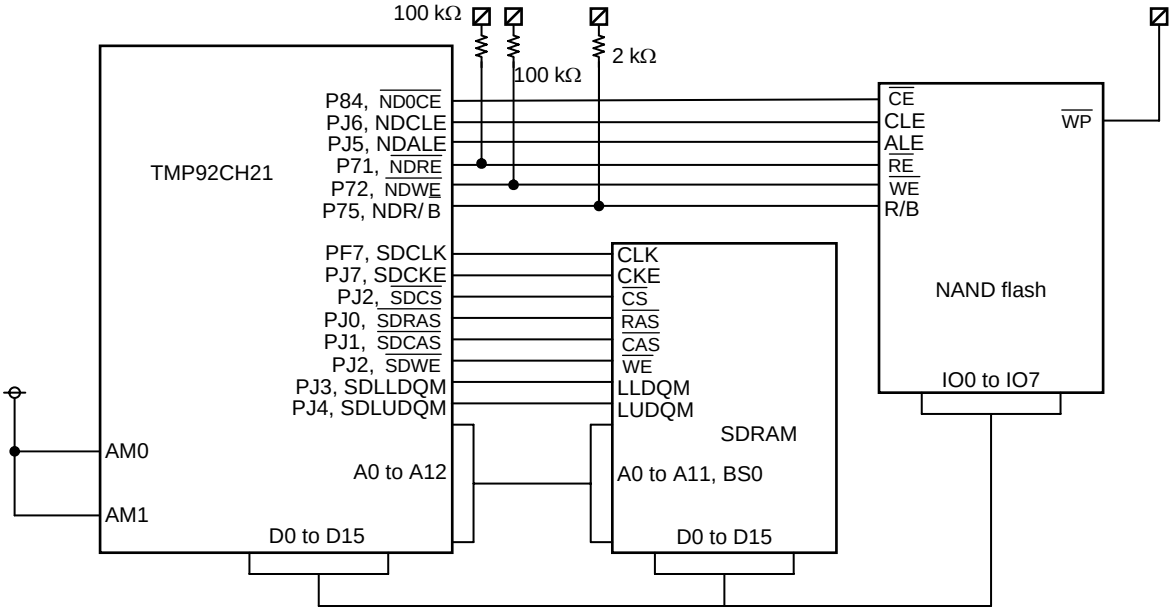
Symbol	Set Value	Set Content
WDMOD	00H	Stop watchdog timer.
WDCR	B1H	Disable watchdog timer.
SYSCR0	80H	Set system clock.
SYSCR1	00H	Set system clock.
SYSCR2	2CH	Set system clock.
PLLCR0	40H	Where USB is used for boot, set to use PLL output clock for f_{FPH}.
	00H	Where USB is not used for boot, set not to use PLL output clock for f_{FPH}.
PLLCR1	80H	Set to PLL ON. Not affected by boot method.
INTEUSB	04H	Set USB interrupt level.
INTETC01	44H	Set INTTC interrupt level.

Note: The setting values for NAND flash, UART and USB are not shown. Set each register where these functions are used in the system.

3.21.4 Download from NAND flash

(1) Connection example

Figure 3.21.4 shows an example of NAND flash. (A 16-bit SDRAM is used as program memory).



Note 1: The values of the pull-up resistors are recommended values.

Note 2: The $\overline{WP}$ (Write protect) pin of NAND flash is not supported by the TMP92CH21. If necessary, it must be prepared on an external circuit.

Figure 3.21.4 Example of NAND Flash Connection

(2) Supported NAND flash

The boot program is designed based on SmartMedia™ physical format specification Ver1.20. Table 3.21.6 shows supported memory devices and device codes.

Table 3.21.6 Supported Memory

Memory Size [Mbyte]	NAND Flash 3.3 V Model	Masked ROM 3.3 V Model
1	Not supported	
2		
4	OK (E3H)	OK (D5H)
8	OK (E6H)	OK (D6H)
16	OK (73H)	OK (57H)
32	OK (75H)	OK (58H)
64	OK (76H)	OK (D9H)
128	OK (79H)	OK (DAH)

(3) Data format

The download data consists of the boot identification code (4 bytes), user program size (2 bytes) and user program (max 10 Kbytes). These should be assigned (programmed) to NAND flash as shown in Figure 3.21.5. Also program the ECC code in the redundant area of the NAND flash, the block status area and thedata status area .

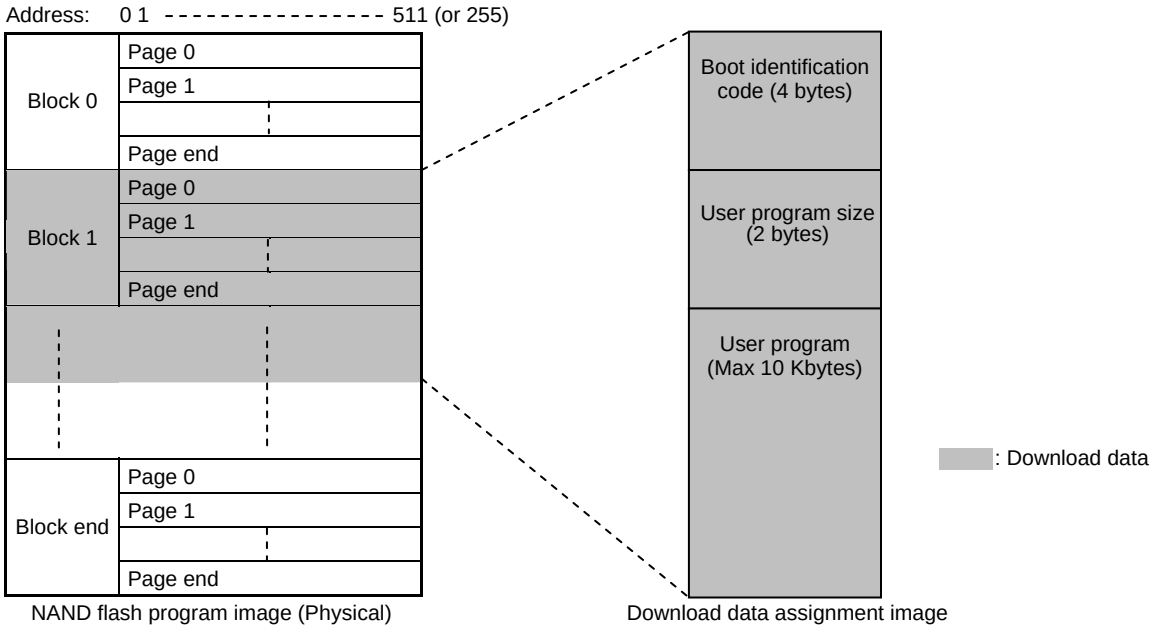


Figure 3.21.5 Download Data Image

a) Boot identification code (4 bytes)

The boot program initially checks the boot identification code. If the boot characters in ASCII code are read from the first 4 bytes in page 0, block 1 of the NAND flash, the boot program will recognize the boot method as NAND flash.

42H (“B”)
4FH (“O”)
4FH (“O”)
54H (“T”)

Figure 3.21.6 Boot Identification Code

b) User program size (2 bytes)

The program size should be programmed to the next 2 bytes. The first byte is the lower 8 bits and the second is the upper 8 bits. This size indicates only the user program size; it does not include the boot identification code (4 bytes) and user program size (2 bytes).

This must be less than or equal to 10 Kbytes. So, the maximum number is 2800H.

Size (Lower 8 bits)
Size (Upper 8 bits)

Figure 3.21.7 User Program Size

c) User program (max 10 Kbytes)

This refers to a user program that is loaded to internal RAM.

When creating a user program, note the following points.

- Set start address to 3000H

  Beforehand, program (write) the user program to NAND flash in binary format.

  An example explaining how to make a binary format file is given below.

Example: How to convert from Intel Hex format file to binary format file

The following is an example of display in text editor when an Intel Hex format file is opened.

: 103000000607F100030000F201030000B1F16010B7

: 00000001FF

In fact, their data are as below because ASCII code is used for Intel Hex format files.

3A313033303030303030363037463130303033303030304632303130333030303042313631363031304237 0D0A3A303030303030303146460D0A

So, first convert the above data to binary format using the table below.

Before (ASCII)	After (Binary)
3A	3A (Only 3A should not be converted.)
30 to 39	0 to 9
41 or 61	A
42 or 62	B
43 or 63	C
44 or 64	D
45 or 65	E
46 or 66	F
0D0A	Delete

Next, delete characters other than data
(Start mark, data number, address, record type and checksum).

The Intel Hex format and its meaning are given below.

Data record 3A 10 3000 00 0607F100030000F201030000B1F16010 B7

- 3A: : (Start mark)
- 10: Data number
- 3000: Address
- 00: Record type
- 0607F100030000F201030000B1F16010: Data
- B7: Checksum

End record 3A 00 0000 01 FF

- 3A: : (Start mark)
- 00: Data number
- 0000: Address
- 01: Record type
- FF: Data

(4) Error check item

The items checked by the boot program are given below.

If an error occurs in any check, the boot program will cancel downloading from NAND flash and skip to the next operation (recognizing UART or USB).

a) Supported NAND flash

The boot program reads a device code from NAND flash and checks whether it is supported or not.

b) Boot identification code

c) User program size

The boot program checks whether it is less than or equal to 10 Kbytes.

d) Block status area

The boot program checks whether each block is normal or not. If the block status area on first page of any block has 2-bit or more "0" data, it is an error.

e) Data status area

The boot program checks whether each data status is correct or not. If the data status area has 4-bit or more "0" data, it is an error.

f) ECC error

The boot program reads both calculated code from NDFC and ECC code in NAND flash and checks whether they are correctable or not.

g) NAND flash R/B

The boot program checks whether NDR/B pin is normal or not in each action.

If the busy status is longer than 70 [μs] at f_{FPH} = 40 MHz, it is an error.

(5) ECC error check

a) Calculation ECC code

The NDFC (NAND flash controller) is used for calculation of ECC code.

b) ECC code correction

The boot program operates as below.

1. Compares both calculated ECC code from NDFC and ECC code in NAND flash.
2. Evaluates and corrects according to the following cases.

Case (a): No data error → (OK) Next operation

Case (b): 1-bit data error → (OK) Error correction and next operation

Case (c): 2-bit or more data error → (Error) Termination

Case (d): ECC code 1-bit error → (OK) Next operation

Case (e): ECC code 2-bit or more error → (Error) Termination

For reference, details of calculation flow are given below.

1) Make XOR data by calculating exclusive OR after both ECC code from NDFC and NAND flash are placed to 4-byte data as below.
   Lower 2 bytes: Line parity
   Upper 2 bytes: Column parity
   (Valid data of column parity is lower 6-bit in upper 2 bytes)
2) If XOR data equals "0", it will terminate normally because the ECC code is the same, but if not, they are checked as to whether they are correctable or not.
3) If XOR data does not have 2-bit or more "1" data, it will terminate normally because of the ECC code 1-bit error.
4) If the effective data (2-bit width from bit0 to bit21 in XOR data) equals either 01B or 10B, it corrects data because they are correctable.
   If the effective data has either 00B or 11B, it terminates abnormally because they are not correctable.

Example 1: If the XOR data equals 0026A65AH, shown below in binary,
0000000000 10 01 10 10 10 01 10 01 01 10 10B
all effective data (2-bit width from bit0 to bit21) equals either 01B or 10B. So, this is evaluated as being correctable.

Example 2: If the XOR data equals 002EA65AH, shown below in binary,
0000000000 10 11 10 10 10 01 10 01 01 10 10B
bit18 and bit19 are 11B, so this is evaluated as being uncorrectable.

5) Data correcting takes error line information from line parity in XOR data and error bit information from column parity and inverts the bit.

Example: If the XOR data equals 0026A65AH, line parity is shown below in binary.
10 10 01 10 01 01 10 10B
If 10B is converted to 1B and 01B is converted to 0B,
they become 1 1 0 1 0 0 1 1B and meaning the 212th byte.
In the same manner, error bit information becomes bit5.
As a result, it inverts bit5 of 212th byte.

3.21.5 Download with UART

(1) Connection example

Figure 3.21.8 shows an example of UART. (A 16-bit NOR flash is used as program memory.)

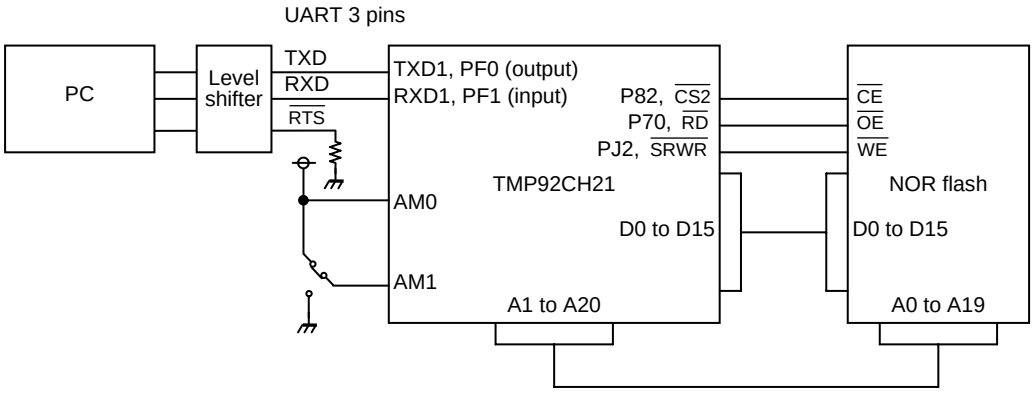


Figure 3.21.8 Example of UART

(2) UART interface specification

SIO channel 1 is used to download.

The following shows the UART communication format in BOOT mode.

Before booting, the PC side must also be setup in the same way.

The default baud rate is 9600 bps, but it can be changed to other values as shown in Table 3.21.9.

Serial transfer mode	:	UART (Asynchronous communication) mode, full duplex communication
Data length	:	8 bits
Parity bit	:	None
STOP bit	:	1 bit
Handshake	:	None
Baud rate (Default)	:	9600 bps

(3) UART data transfer format

Table 3.21.7 to Table 3.21.12 show the supported frequency, data transfer format, baud rate modification commands, operation commands, version management information, and frequency measurement result with data storing location, respectively.

Please also refer to the description of boot program operation in the following pages.

Table 3.21.7 Supported Frequency (f_{OSCH})

6.00 MHz	8.00 MHz	9.00 MHz	10.00 MHz	16.00 MHz	20.00 MHz	22.579 MHz	25.00 MHz	32.00 MHz	33.868 MHz	36.00 MHz	40.00 MHz

Note: Internal PLL (Clock multiplier) is not used.

Table 3.21.8 Transfer Format

	Byte Number to Transfer	Transfer Data from PC to TMP92CH21	Baud Rate	Transfer Data from TMP92CH21 to PC
Boot ROM	1st byte	Matching data (5AH)	9600 bps	– (Frequency measurement and baud rate auto set)
	2nd byte	–		OK: Echo back data (5AH) Error: Nothing transmitted
	3rd byte to 6th byte	–		Version management information (Refer to Table 3.21.11)
	7th byte	–		Frequency information (Refer to Table 3.21.12)
	8th byte	Baud rate modification command (Refer to Table 3.21.9)		–
	9th byte	–		OK: Echo back data Error: Error code × 3
	10th byte to n'th – 4 byte	User program Intel Hex format (binary)	New baud rate	Error: Stop operation by checksum error
	n'th – 3 byte	–		OK: SUM (High) (Refer (6) – c)
	n'th – 2 byte	–		OK: SUM (Low)
	n'th – 1 byte	User program start command (C0H) (Refer to Table 3.21.10)		–
	n'th byte	–		OK: Echo back data (C0H) Error: Error code × 3
RAM	–	Branch to user program start address		

"Error code × 3" means sending error code 3 times. For example, when error code is 62H, TMP92CH21 sends 62H 3 times. (For error code, refer to (4)-b.)

Table 3.21.9 Baud Rate Modification Command

Baud Rate (bps)	9600	19200	38400	57600	115200
Modification Command	28H	18H	07H	06H	03H

Note 1: If f_{OSCH} is either 16.0, 20.0, 20.58 or 25.0 MHz, 115200 bps is not supported.

Note 2: If f_{OSCH} is 10.0 MHz, both 57600 and 115200 bps are not supported.

Note 3: If f_{OSCH} is 6.00, 8.00 or 9.00 MHz, then 38400, 57600 and 115200 bps are not supported.

Table 3.21.10 Operation Command

Operation Command	Operation
C0H	Start user program

Table 3.21.11 Version Management Information

Version Information	ASCII Code
FRM1	46H, 52H, 4DH, 31H

Table 3.21.12 Frequency Measurement Result Data

f_{OSCH} [MHz]	6.000	8.000	9.000	10.000	16.000	20.000
2000H (RAM storing address)	09H	0AH	08H	0BH	00H	01H
	22.579	25.000	32.000	33.868	36.000	40.000
	02H	03H	04H	05H	06H	07H

(4) Description of UART boot program operation

The boot program receives data that is sent from the PC by UART, and loads it to internal RAM.

If the transferring terminates normally, it calculates SUM and sends the result to the PC before staring to execute the user program. The starting address to execute is the address received first . This boot program enables user's own on-board programming.

a) Operation procedure

1. Connect the serial cable . Make sure to perform connection before resetting the micro controller.
2. Set both AM1 and AM0 pins to "1" and reset the micro controller.
3. The receive data in the first byte is the matching data. When the boot program starts, it goes to a state in which it waits for the matching data to be received. Upon receiving the matching data, it automatically adjusts the serial channels' initial baud rate to 9600 bps. The matching data is 5AH.
4. The second byte is used to echo back 5AH to the PC upon completion of the automatic baud rate setting in the first byte. If the device fails in automatic baud rate setting, it goes to an idle state.
5. The third through sixth bytes are used to send the boot program's version management information in ASCII code. The PC should check that the correct version of the boot program is used.

6. The seventh byte is used to send information of the measured frequency. The PC should check that the frequency of the resonator is measured correctly.
7. The receive data in the eighth byte is the baud rate modification data. The five kinds of baud rate modification data shown in Table 3.21.9 are available. Even when you do not change the baud rate, be sure to send the initial baud rate data (28H: 9600 bps). Baud rate modification becomes effective after the echo back transmission is completed.
8. The ninth byte is used to echo back the received data to the PC when the data received in the eighth byte is one of the baud rate modification data corresponding to the device's operating frequency. Then the baud rate is changed. If the received baud rate data does not correspond to the device's operating frequency, the device goes to an idle state after sending 3 bytes of baud rate modification error code (62H).
9. The receive data in the 10th byte through n'th – 4 bytes is received as binary data in Intel Hex format. No received data is echoed back to the PC.
   The boot program processing routine ignores the received data until it receives the start mark (3AH for " : ") in Intel Hex format. Nor does it send error code to the PC. After receiving the start mark, the routine receives a range of data from the data length to checksum and writes the received data to the specified RAM addresses successively.
   After receiving one record of data from start mark to checksum, the routine goes to a start mark waiting state again.
   If a receive error or checksum error of Intel Hex format occurs, the device goes to an IDLE state without returning error code to the PC.
   Because the boot program processing routine executes a SUM calculation routine upon detecting the end record, the controller should be placed in a SUM waiting state after sending the end record to the device.
10. The n'th – 3 bytes and the n'th – 2 bytes are the SUM value that is sent to the PC in order of upper byte and lower byte. For details on how to calculate the SUM, refer to "notes on SUM" in the latter pages of this manual. The SUM calculation is performed only when no write error, receive error, or Intel Hex format error has been encountered after detecting the end record. Soon after calculation of SUM, the device sends the SUM data to the PC. The PC should determine whether writing to the RAM has terminated normally depending on whether the SUM value is received after sending the end record to the device.
11. After sending the SUM, the device goes to a state waiting for the user program start code. If the SUM value is correct, the PC should send the user program start command to the n'th – 1 byte. The user program start command is C0H.
12. The n'th byte is used to echo back the user program start code to the PC. After sending the echo back to the PC, the stack pointer is set to 5FFFH and the boot program jumps to the 1st address that is received as data in Intel Hex format.
13. If the user program start code is wrong or a receive error occurs, the device goes to an idle state after returning 3 bytes of error code to the PC.

b) Error code

The boot program sends the processing status to the PC using various codes.

The error codes are listed in the table below.

Table 3.21.13 Error Codes

Error Code	Meaning of Error Code
62H	Baud rate modification error occurred.
64H	Operation command error occurred.
A1H	Framing error in received data occurred.
A3H	Overrun error in received data occurred

Note 1: When a receive error occurs when receiving the user program, the device does not send the error code to the PC.

Note 2: After sending the error code, the device goes to an IDLE state.

c) Notes on SUM

1. Calculation method

SUM consists of byte + byte... + byte, the sum of which is returned in words as the result. Namely, data is read out in bytes, the sum of which is calculated, with the result returned in words.

Example:

If the data to be calculated consists of the 4 bytes shown to the left, SUM of the data is:

A1H + B2H + C3H + D4H = 02EAH

SUM (HIGH) = 02H

SUM (LOW) = EAH

2. Calculation data

The data from which SUM is calculated is the RAM data from the first address received to the last address received.

The received RAM write data is not the only data to be calculated for SUM. Even when the received addresses are noncontiguous and there are some unwritten areas, data in the entire memory area is calculated. The user program should not contain unwritten gaps.

d) Notes on Intel Hex format (Binary)

1. After receiving the checksum of a record, the device waits for the start mark (3AH for " : ") of the next record. Therefore, the device ignores all data received between records during that time unless the data is 3AH.
2. Make sure that once the PC program has finished sending the checksum of the end record, it does not send anything and waits for 2 bytes of data to be received (upper and lower bytes of SUM). This is because after receiving the checksum of the end record, the boot program calculates the SUM and returns the calculated SUM in 2 bytes to the PC.
3. Writing to areas outside the device's internal RAM causes incorrect operation. Therefore, when an extended record is transmitted, be sure to set a paragraph address to 0000H.
4. Always make sure the first record type is an extended record, because the initial value of the address pointer is 00H.
5. The user program is assigned to the address from 3000H to 57FFH and it should be within 10 Kbytes.
6. Transmit a user program not by the ASCII code but by binary. An example explaining how to make binary format file is given below.

Example: How to convert from Intel Hex format file to binary format file.

The following is an example of display in text editor where an Intel Hex format file is opened.

: 103000000607F100030000F201030000B1F16010B7

: 00000001FF

In fact, their data are as below because ASCII code is used for Intel Hex format files.

3A313033303030303030363037463130303033303030304632303130333030303042314631363031304237 0D0A3A303030303030303146460D0A

So, first convert the above data to binary format using the table below.

Before (ASCII)	After (Binary)
3A	3A (Only 3A should not be converted.)
30 to 39	0 to 9
41 or 61	A
42 or 62	B
43 or 63	C
44 or 64	D
45 or 65	E
46 or 66	F
0D0A	Delete it

The Intel Hex format and its meaning are given below.

Data record 3A 10 3000 00 0607F100030000F201030000B1F16010 B7

- 3A: : (Start mark)
- 10: Data number
- 3000: Address
- 00: Record type
- 0607F100030000F201030000B1F16010: Data
- B7: Checksum

End record 3A 00 0000 01 FF

- 3A: : (Start mark)
- 00: Data number
- 0000: Address
- 01: Record type
- FF: Data

e) Error when receiving user program

If the following errors occur in Intel Hex format when receiving the user program, the device goes to an idle state.

When the record type is not 00H, 01H, and 02H

When a checksum error occurs

f) Error between frequency measurement and baud rate

The boot program measures the resonator frequency when receiving matching data. If the error is under 3%, the boot program decides on that frequency. Since there is an overlap between the margin of 3% for 32.000 MHz and 33.868 MHz, the boundary is set at the intermediate value between the two. The baud rate is set based on the measured frequency. Each baud rate includes a set error shown in Table 3.21.14. For example, in the case of 20.000 MHz and 9600 bps, the baud rate is actually set at 9615.38 bps with an error of 0.2%. To establish communication, the sum of the baud rate set error shown in Table 3.21.14 and frequency error must be under 3%.

Table 3.21.14 Setting Error of Each Baud Rate (%)

	9600 bps	19200 bps	38400 bps	57600 bps	115200 bps
6.000 MHz	0.2	0.2	–	–	–
8.000 MHz	0.2	0.2	–	–	–
9.000 MHz	0.2	–0.7	–	–	–
10.000 MHz	0.2	0.2	–1.4	–	–
16.000 MHz	0.2	0.2	0.2	–0.8	–
20.000 MHz	0.2	0.2	0.2	1.0	–
22.579 MHz	–0.7	–0.7	–0.7	0.1	–
25.000 MHz	0.5	–0.8	0.5	0.5	–
32.000 MHz	0.2	0.2	0.2	0.7	–0.8
33.868 MHz	0.3	0.3	0.3	–0.7	–0.7
36.000 MHz	0.2	–0.7	0.2	0.2	0.2
40.000 MHz	0.2	0.2	0.2	–0.3	1.0

–: Not supported

(5) Further notes

a) Handshake function

The TMP92CH21 has a $\overline{\text{CTS}}$ pin, but boot programs do not use it.

b) RS-232C connector

When the boot program is running, do not connect or disconnect an RS-232C connector.

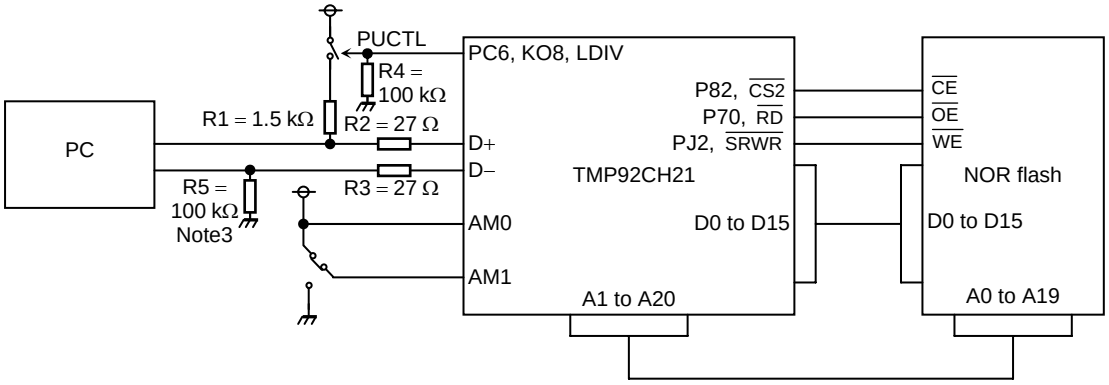
c) Software on PC

Special application software is needed on the PC.

3.21.6 Download with USB

(1) Connection example

Figure 3.21.9 shows an example of USB. (16-bit NOR flash is used as program memory.)



Note 1: The values of pull-up / pull-down resistors are recommended values.

Note 2: The PC6 (KO8, LDIV) pin is assigned as PUCTL (Control to pull-up) for USB. So, note whether it is used as KO8 or LDIV.

Note 3: Pull-down resistor R2 is used only to fix the level for the flow current. If there is no ON/ OFF control by port for example, confirm operation by actual setting, and set the value to ensure the USB connection is not cut.

Figure 3.21.9 USB Connecting Example

(2) USB interface specification outline

For USB booting, make sure the oscillator is 9 MHz.

The baud rate is fixed at full speed (12 MHz).

The boot function is employed using the following 2 transfer types.

Table 3.21.15 Transfer Types Used by Boot Program

Transfer Type	Purpose
Control	Used as transmitting for standard request or vendor request
Bulk	Used as transmitting for vendor request or user program

An outline flowchart is given below.

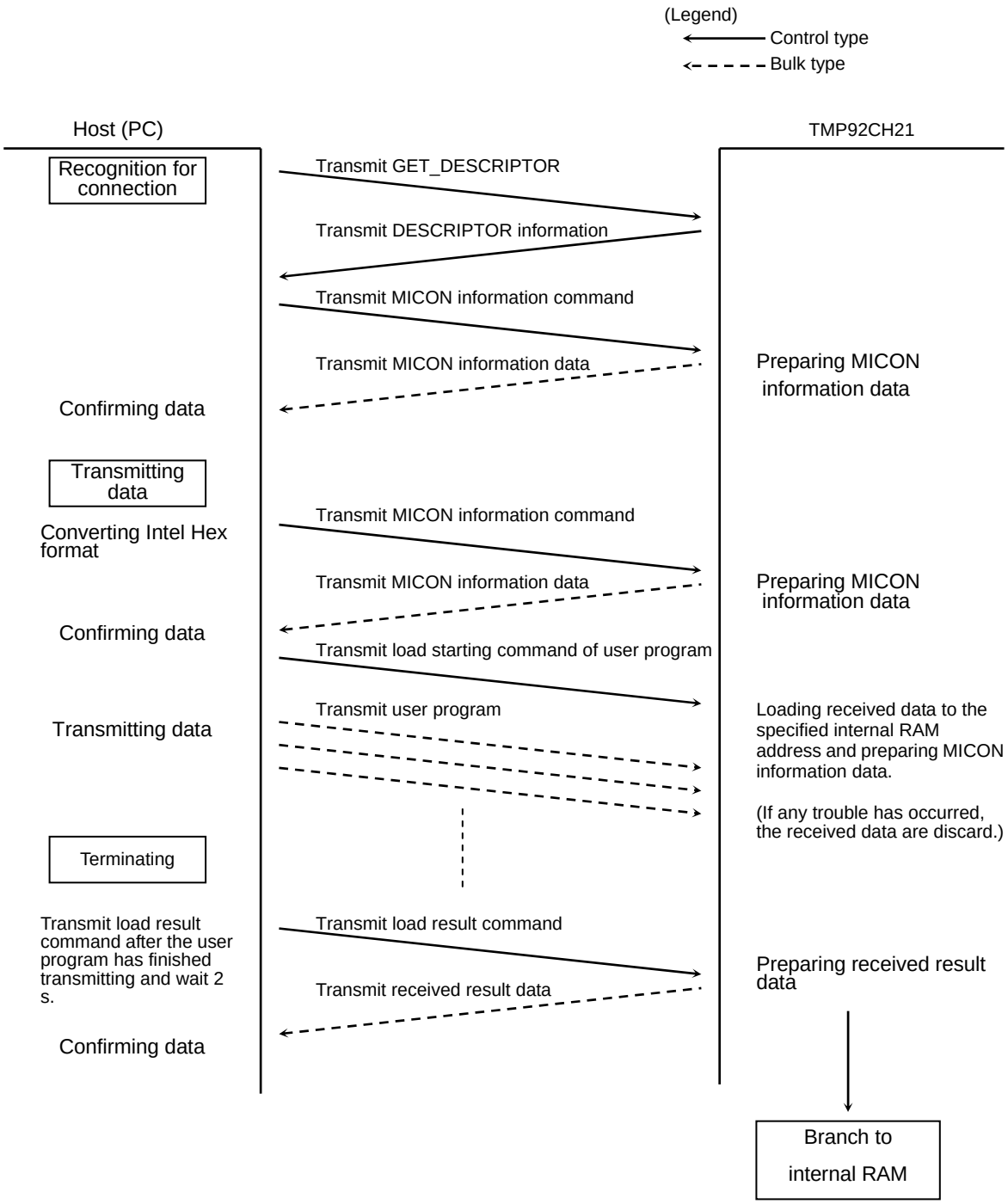


Figure 3.21.10 Outline Flowchart

The vendor request command table is shown below.

Table 3.21.16 Vendor Request Command Table

Command Name	Value of Request	Outline	Notes
MICON (Microcomputer) information command	00H	Transmit microcomputer information	This is transmitted after a setup stage is terminated by bulk in transfer type.
Load starting command of user program	02H	Receive user program	Substitute size of user program to wIndex. The user program should be received after a setup stage is terminated by bulk out transfer type.
Transmit result command	04H	Transmit the result	This is transmitted after a setup stage is terminated by bulk in transfer type.

The data structure of setup command is shown below.

Table 3.21.17 Data Structure of Setup Command

Field Name	Value	Meaning
bmRequestType	40H	D7 0: Host to device D6-D5 2: Vender D4-D0 0: Device
bRequest	00H, 02H, 04H	00H: MICON information 02H: Start to transmit user program 04H: Result for user program received
wValue	00H to FFFFH	Own data number (Not used by boot program)
wIndex	00H to FFFFH	Size of user program (Used when a user program starts to be transmitted.)
wLength	0000H	Fixed

The standard request command table is shown below.

Table 3.21.18 The Standard Request Command Table

Standard Request	Response Medthod
GET_STATUS	By hardware, automatically
CLEAR_FEATURE	
SET_FEATURE	
SET_ADDRESS	
GET_DISCRIPTOR	
SET_DISCRIPTOR	Not supported
GET_CONFIGRATION	By hardware, automatically
SET_CONFIGRATION	
GET_INTERFACE	
SET_INTERFACE	
SYNCH_FRAME	Ignored

The information transmitted by GET_DISCRIPTOR is shown below.

Table 3.21.19 Information Transmitted by GET_DISCRIPTOR

Device Descriptor

Field Name	Value	Meaning
Blength	12H	18 bytes
BdescriptorType	01H	Device descriptor
BcdUSB	0110H	USB Version 1.1
BdeviceClass	00H	Device class is not used
BdeviceSubClass	00H	Sub command is not used
BdeviceProtocol	00H	Protocol is not used
BmaxPacketSize0	40H	EP0 max packet size 64 bytes
IdVendor	0930H	Vendor ID
IdProduct	6504H	Product ID (0)
BcdDevice	0001H	Device version (v 0.1)
Imanufacturer	00H	Index value of string descriptor in which producer is shown
Iproduct	00H	Index value of string descriptor in which product name is shown
IserialNumber	00H	Index value of string descriptor in which product number is shown
BnumConfigurations	01H	Configuration is 1

Configuration Descriptor

Field Name	Value	Meaning
bLength	09H	9 bytes
bDescriptorType	02H	Configuration descriptor
wTotalLength	0020H	Total length (32 bytes) in which each descriptor of configuration descriptor, interface and endpoint is added.
bNumInterfaces	01H	Interface is 1
bConfigurationValue	01H	Configuration number 1
iConfiguration	00H	Index value of string descriptor in which this configuration name is shown (Not used).
bmAttributes	80H	Bus power
MaxPower	31H	Maximum power consumption (49 mA)

Interface Descriptor

Field Name	Value	Meaning
bLength	09H	9 bytes
bDescriptorType	04H	Interface descriptor
bInterfaceNumber	00H	Interface number 0
bAlternateSetting	00H	Alternate setting number 0
bNumEndpoints	02H	Endpoint is 2
bInterfaceClass	FFH	Specified device
bInterfaceSubClass	00H	
bInterfaceProtocol	50H	Bulk only protocol
iIinterface	00H	Index value of string descriptor in which this interface name is shown (Not used).

Endpoint Descriptor

Field Name	Value	Meaning
<Endpoint 1>		
blength	07H	7 bytes
bDescriptorType	05H	Endpoint descriptor
bEndpointAddress	01H	EP1 is OUT
bmAttributes	02H	Bulk transfer
wMaxPacketSize	0040H	Payload 64 bytes
bInterval	00H	(Ignored for bulk transfer)
<Endpoint 2>		
bLength	07H	7 bytes
bDescriptor	05H	Endpoint descriptor
bEndpointAddress	82H	EP2 is IN
bmAttributes	02H	Bulk transfer
wMaxPacketSize	0040H	Payload 64 bytes
bInterval	00H	(Ignored for bulk transfer)

The information transmitted by the MICON information command is shown below.

Table 3.21.20 Information Transmitted by MICON Information Command

Micon Information	ASCII Code
"TMP92CH21FG"	54H, 4DH, 50H, 39H, 32H, 43H, 48H, 32H, 31H, 46H, 47H, 20H, 20H, 20H, 20H

The information transmitted by the result information command is shown below.

Table 3.21.21 Information Transmitted by Result Information Command

Result	Value	Error Condition
No error	00H	
Not received user program error	02H	When a user program is received without receiving user program starting command.
Received except Intel Hex format error	04H	When the first data of the user program is not " : " (3AH).
Over user program size error	06H	When more than the value of wIndex is received.
Received incorrect address error	08H	When the user program address is incorrect. When the user program size is over 10 Kbytes
Protocol error or other error	0AH	When start or result of user program is received first. When check SUM is incorrect in Intel Hex file. When record type is incorrect in Intel Hex file. When address length is more than 2 in Intel Hex file. When end record length is not 0 in Intel Hex file.

(3) Description of USB boot program operation

The boot program provides the following RAM loader function.

The data, which is transmitted by the PC in Intel Hex format, is loaded to the internal RAM.

After loading normally, the user program will begin to execute. The first received address is set as the starting address.

By this function, this boot program enables the user's own on-board programming.

a. Operational procedure

1. Connect the USB cable.
2. Set both AM1 and AM0 pin to "1" and reset the micro controller.
3. On the PC side, recognize USB connection and confirm sub information by GET_DISCRIPTOR.
4. On the PC side, transmit MICON information command by vendor request and confirm MICON information data by Bulk IN after a setup stage is finished.
5. The boot program prepares MICON information in ASCII code after MICON information command is received.
6. On the PC side, convert user program into binary format.
7. On the PC side, transmit load-starting command by vendor request and transmit user program by Bulk OUT after a setup stage is finished.
8. On the PC side, wait 2 seconds and transmit load result command by vendor request. Confirm the result by bulk in after a setup stage is finished.
9. The boot program prepares the result after load result command is received.
10. If the result is not normal, the boot program cannot be returned normally. In this case, terminate device driver on the PC and retry from step 2.

b. Notes on user program format (Binary)

1. After receiving the checksum of a record, the device waits for the start mark (3AH for ": ") of the next record.The device therefore ignores all data received between records during that time unless the data is 3AH.
2. The first record type is not needed as an address record because the initial value of the address pointer is 00H.
3. The user program is assigned to the address from 3000H to 57FFH and it should be within 10 Kbytes.
4. In the user program, change the Intel Hex format file (usually ASCII code) to binary format and transfer it. The example below explains how to make a binary format file. (This is the same as with UART.)
   Make sure that the maximum data number of 1 record is FAH for the user program.

Example: Transfer data case of writing 16 bytes data from address 3000H by Intel Hex format file.

The following is an example of display in text editor where an Intel Hex format file is opened.

: 103000000607F100030000F201030000B1F16010B7

: 00000001FF

In fact, their data are as below because ASCII code is used for Intel Hex format files.

3A313033303030303030363037463130303033303030304632303130333030303030

42314631363031304237 0D0A3A3030303030303031 46460D0A

So, first convert the above data to binary format using the table below.

Before (ASCII)	After (Binary)
3A	3A (Only 3A should not be converted.)
30 to 39	0 to 9
41 or 61	A
42 or 62	B
43 or 63	C
44 or 64	D
45 or 65	E
46 or 66	F
0D0A	Delete it

The Intel Hex format and its meaning are given below.

Data record 3A 10 3000 00 0607F100030000F201030000B1F16010 B7

- 3A: : (Start mark)
- 10: Data number
- 3000: Address
- 00: Record type
- 0607F100030000F201030000B1F16010: Data
- B7: Checksum

End record 3A 00 0000 01 FF

- 3A: : (Start mark)
- 00: Data number
- 0000: Address
- 01: Record type
- FF: Data

(4) Further notes

a) USB connector

When the boot program is running, do not connect or disconnect the USB connector.

b) Software on PC

Special USB device driver and application software is needed on the PC.

4. Electrical Characteristics

4.1 Maximum Ratings

Parameter	Symbol	Rating	Unit
Power Supply Voltage	V_{CC}	−0.5 to 4.0	V
Input Voltage	V_{IN}	−0.5 to VCC + 0.5	V
Output Current	I_{OL}	2	mA
Output Current	I_{OH}	−2	mA
Output Current (Total)	ΣI_{OL}	80	mA
Output Current (Total)	ΣI_{OH}	−80	mA
Power Dissipation (Ta = 85°C)	P_D	600	mW
Soldering Temperature (10 s)	T_{SOLDER}	260	°C
Storage Temperature	T_{STG}	−65 to 150	°C
Operation Temperature	T_{OPR}	−20 to 70	°C

Note: The maximum ratings are rated values which must not be exceeded during operation, even for an instant. Any one of the ratings must not be exceeded. If any maximum rating is exceeded, 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 which include this device, ensure that no 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 =230°C, 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°C, Dipping time = 5 seconds The number of times = one, Use of R-type flux (use lead-free)	

4.2 DC Electrical Characteristics (1/2)

V_{CC} = 3.3 ± 0.3V/X1 = 6 to 40 MHz/Ta = −20 to 70°C
V_{CC} = 2.7 − 3.6V/X1 = 6 to 27 MHz/Ta = −20 to 70°C

Parameter	Symbol	Min	Typ.	Max	Unit	Condition	
Power supply voltage (DVCC = AVCC) (DVSS = AVSS = 0 V)	V_{CC}	3.0		3.6	V	X1 = 6 to 40 MHz	XT1 = 30 to 34 kHz
		2.7				X1 = 6 to 27 MHz	
Input low voltage for 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 for P60 to P67, P71 to P72, P75 to P76, P90, P93 to P94, PC6 to PC7, PG0 to PG3, PJ5 to PJ6, PL4 to PL7	V_{IL1}			0.3 × V_{CC}			
Input low voltage for P91 to P92, P96 to P97, PA0 to PA7, PC0 to PC3, PF0 to PF2, $\overline{RESET}$	V_{IL2}			0.25 × V_{CC}			
Input low voltage for AM0 to AM1	V_{IL3}			0.3			
Input low voltage for X1, XT1	V_{IL4}			0.2 × V_{CC}			
Input high voltage for D0 to D7, 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 for P60 to P67, P71 to P72, P75 to P76, P90, P93 to P94, PC6 to PC7, PG0 to PG3, PJ5 to PJ6, PL4 to PL7	V_{IH1}	0.7 × V_{CC}					
Input high voltage for P91 to P92, P96 to P97, PA0 to PA7, PC0 to PC3, PF0 to PF2, $\overline{RESET}$	V_{IH2}	0.75 × V_{CC}					
Input high voltage for AM0 to AM1	V_{IH3}	V_{CC} − 0.3					
Input high voltage for X1, XT1	V_{IH4}	0.8 × V_{CC}					

DC Electrical Characteristics (2/2)

Parameter	Symbol	Min	Typ.	Max	Unit	Condition	
Output low voltage	V_{OL}			0.45	V	I_{OL} = 1.6 mA	
Output high voltage	V_{OH1}	2.4				I_{OH} = –400 μA	
	V_{OH2}	0.9 × V_{CC}				I_{OH} = –20 μA	
Output low voltage for MX, MY pins	$V_{OL (T)}$			0.2		$I_{OL (T)}$ = 6.6 mA	V_{CC} = 3.0 to 3.6 V
Output high voltage for PX, PY pins	$V_{OH (T)}$	V_{CC} – 0.2				$I_{OH (T)}$ = –6.6 mA	
Input leakage current	I_{LI}		0.02	±5	μA	0.0 ≤ V_{IN} ≤ V_{CC}	
Output leakage current	I_{LO}		0.05	±10	μA	0.2 ≤ V_{IN} ≤ V_{CC} – 0.2 V	
Power down voltage at STOP (for internal RAM backup)	V_{STOP}	1.8		3.6	V	V_{IL2} = 0.2 × V_{CC}, V_{IH2} = 0.8 × V_{CC}	
Pull-up resistor for $\overline{RESET}$, PA0 to PA7	R_{RST}	80		500	kΩ		
Programmable pull down resistor for P96	R_{KH}						
Pin capacitance	C_{IO}			10	pF	fc = 1 MHz	
Schmitt width for P91 to P92, P96 to P97, PA0 to PA7,PC0 to PC3, PF0 to PF2, $\overline{RESET}$	V_{TH}	0.4	1.0		V		
NORMAL (Note 2)	I_{CC}		33	65	mA	V_{CC} = 3.6 V, fc = 40 MHz	
IDLE2			16	26			
IDLE1			4.3	8.7			
SLOW (Note 2)			25.2	110	μA	Ta ≤ 70°C	V_{CC} = 3.6 V, fs = 32 kHz
				70		Ta ≤ 50°C	
IDLE2			15.1	80		Ta ≤ 70°C	
				30		Ta ≤ 50°C	
IDLE1			4.3	60		Ta ≤ 70°C	
				20		Ta ≤ 50°C	
STOP			0.2	50		Ta ≤ 70°C	V_{CC} = 3.6 V
				15		Ta ≤ 50°C	

Note 1: Typical values are for when Ta = 25°C and VCC = 3.3 V unless otherwise noted.

Note 2: ICC measurement conditions (NORMAL, SLOW):

All functions are operational; output pins are opened and input pins are fixed. C_L = 30 pF is loaded to data and address bus.

4.3 AC Characteristics

4.3.1 Basic Bus Cycle

Read cycle

No.	Parameter	Symbol	Variable		40 MHz	36 MHz	27 MHz	Unit
			Min	Max				
1	OSC period (X1/X2)	t_{OSC}	25	166.7	25	27.7	37.0	ns
2	System clock period (= T)	t_{CYC}	50	333.3	50	55.5	74.0	
3	SDCLK low width	t_{CL}	0.5 T – 15		10	12.7	22	
4	SDCLK high width	t_{CH}	0.5 T – 15		10	12.7	22	
5-1	A0 to A23 valid → D0 to D31 Input at 0 waits	$t_{AD\ (3.0\ V)}$		2.0 T – 30	70	81	–	
		$t_{AD\ (2.7\ V)}$		2.0 T – 35	–	–	113	
5-2	A0 to A23 valid → D0 to D31 Input at 1 wait	$t_{AD3\ (3.0\ V)}$		3.0 T – 30	120	136.5	–	
		$t_{AD3\ (2.7\ V)}$		3.0 T – 35	–	–	187	
6-1	$\overline{RD}$ falling → D0 to D31 Input at 0 waits	t_{RD}		1.5 T – 30	45	53.3	81	
6-2	$\overline{RD}$ falling → D0 to D31 Input at 1 wait	t_{RD3}		2.5 T – 30	95	108.8	155	
7-1	$\overline{RD}$ low width at 0 waits	t_{RR}	1.5 T – 20		55	63.2	91	
7-2	$\overline{RD}$ low width at 1 wait	t_{RR3}	2.5 T – 20		105	118.8	165	
8	A0 to A23 valid → $\overline{RD}$ falling	t_{AR}	0.5 T – 20		5	7.7	17	
9	$\overline{RD}$ falling → SDCLK rising	t_{RK}	0.5 T – 20		5	7.7	17	
10	A0 to A23 valid → D0 to D31 hold	t_{HA}	0		0	0	0	
11	$\overline{RD}$ rising → D0 to D31 hold	t_{HR}	0		0	0	0	
12	$\overline{WAIT}$ setup time	t_{TK}	15		15	15	15	
13	$\overline{WAIT}$ hold time	t_{KT}	5		5	5	5	
14	Data byte control access time for SRAM	t_{SBA}		1.5 T – 30	45	53.3	81	
15	$\overline{RD}$ high width	t_{RRH}	0.5 T – 15		10	12.7	22	

Write cycle

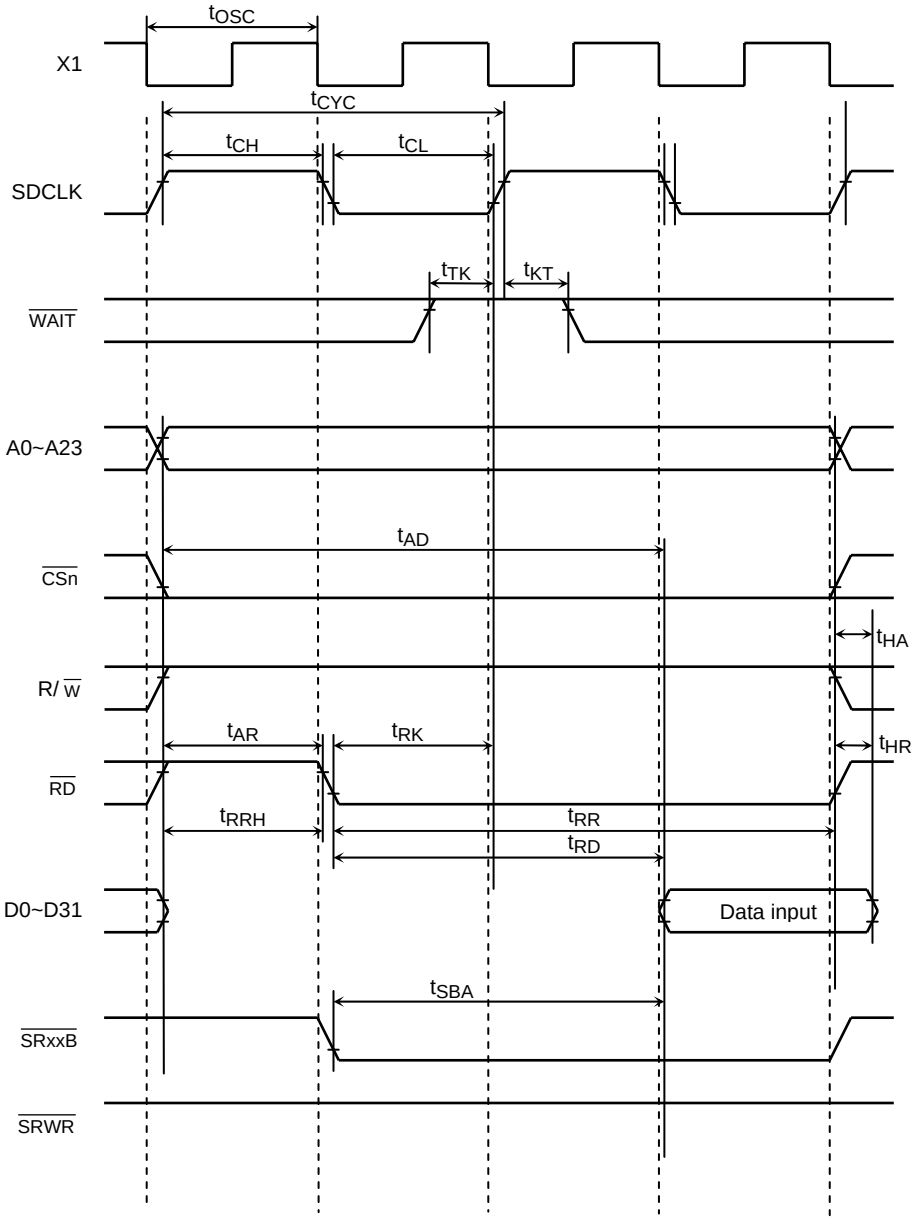
No.	Parameter	Symbol	Variable		40 MHz	36 MHz	27 MHz	Unit
			Min	Max				
16-1	D0 to D31 valid → $\overline{WRxx}$ rising at 0 waits	t_{DW}	1.25T – 35		27.5	34.3	57.5	ns
16-2	D0 to D31 valid → $\overline{WRxx}$ rising at 1 wait	t_{DW3}	2.25T – 35		77.5	89.8	131.5	
17-1	$\overline{WRxx}$ low width at 0 waits	t_{WW}	1.25T – 30		32.5	34.3	62.5	
17-2	$\overline{WRxx}$ low width at 1 wait	t_{WW3}	2.25T – 30		82.5	89.8	136.5	
18	A0 to A23 valid → $\overline{WR}$ falling	t_{AW}	0.5T – 20		5	7.7	17	
19	$\overline{WRxx}$ falling → SDCLK rising	t_{WK}	0.5T – 20		5	7.7	17	
20	$\overline{WRxx}$ rising → A0 to A23 hold	t_{WA}	0.25T – 5		7.5	8.8	13.5	
21	$\overline{WRxx}$ rising → D0 to D31 hold	t_{WD}	0.25T – 5		7.5	8.8	13.5	
22	$\overline{RD}$ rising → D0 to D31 output	$t_{RDO\ (3.0\ V)}$	0.5T – 5		20	22.7	–	
		$t_{RDO\ (2.7\ V)}$	0.5T – 7		–	–	30	
23	Write pulse width for SRAM	t_{SWP}	1.25T – 30		32.5	39.3	62.5	
24	Data byte control to end of write for SRAM	t_{SBW}	1.25T – 30		32.5	39.3	62.5	
25	Address setup time for SRAM	t_{SAS}	0.5T – 20		5	7.7	17	
26	Write recovery time for SRAM	t_{SWR}	0.25T – 5		7.5	8.8	13.5	
27	Data setup time for SRAM	t_{SDS}	1.25T – 35		27.5	34.3	57.5	
28	Data hold time for SRAM	t_{SDH}	0.25T – 5		7.5	8.8	13.5	

AC measuring condition

- Output: High = 0.7 VCC, Low = 0.3 VCC, C_L = 50 pF
- Input: High = 0.9 VCC, Low = 0.1 VCC

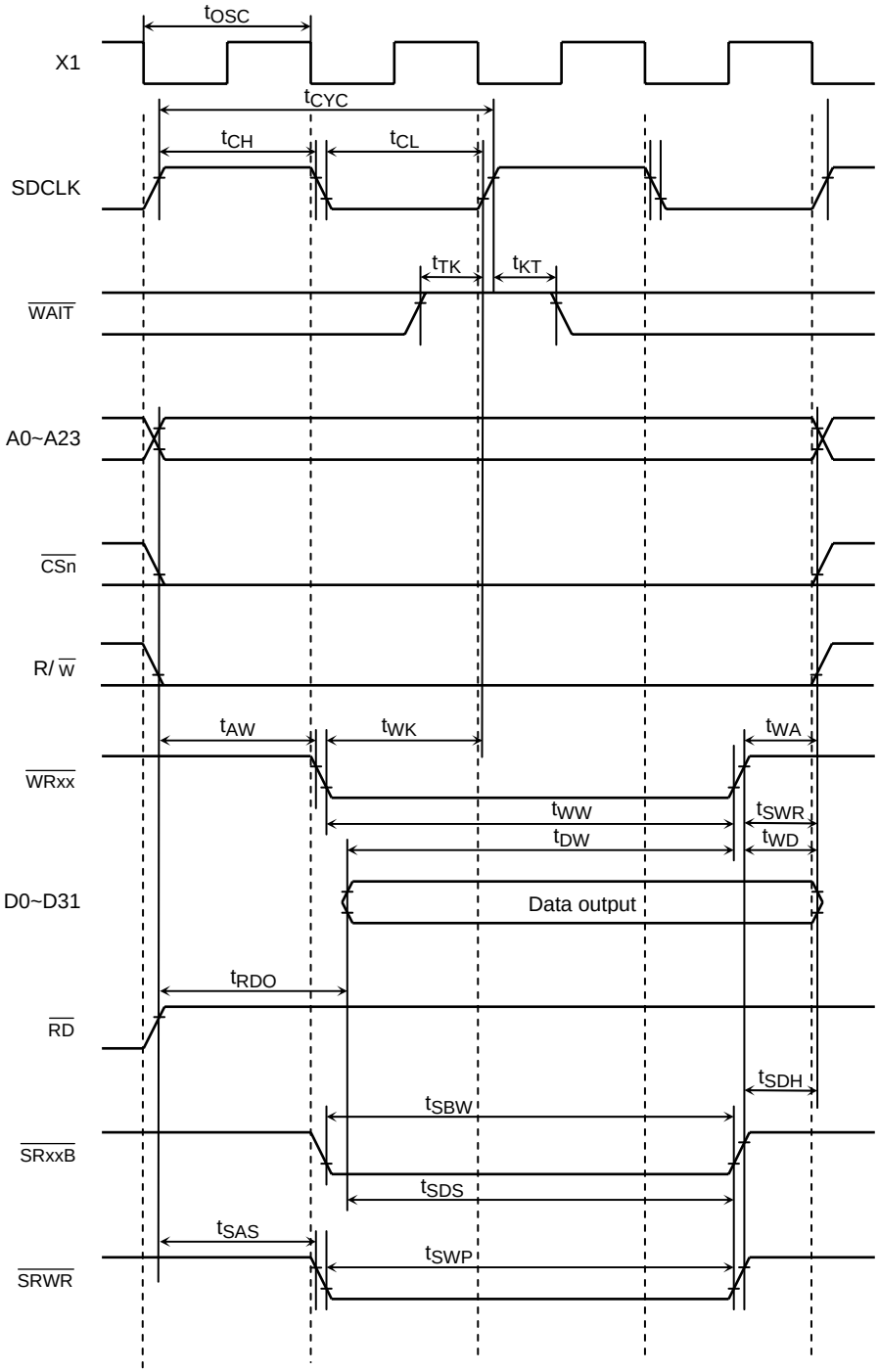
Note: The figures in the “Variable” column cover the whole VCC range (2.7 V to 3.6 V).
Exceptions are shown by the VCC (min), “(3.0 V)” or “(2.7 V)”, added to the “Symbol” column.

(1) Read cycle (0 waits)



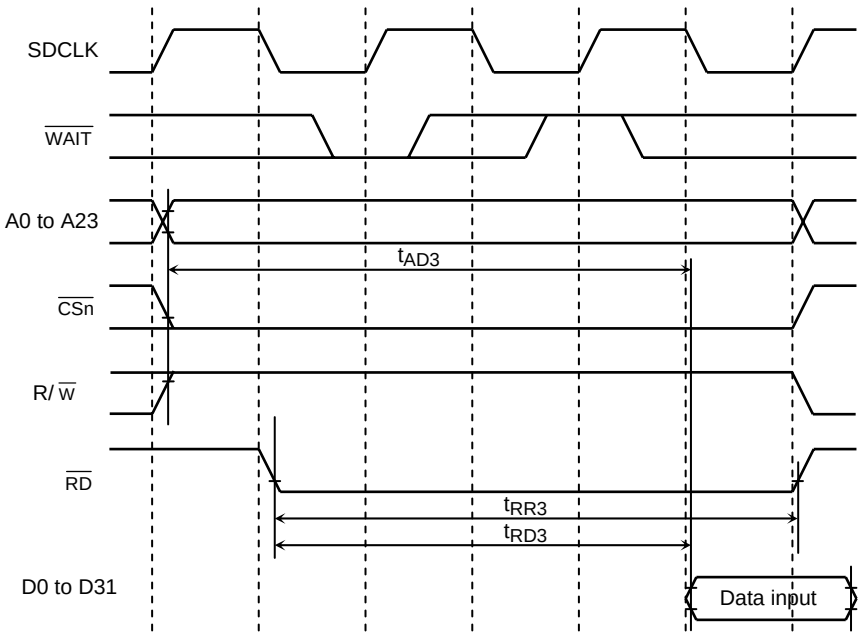
Note: The phase relation between X1 input signal and the other signals is undefined.
The above timing chart is an example.

(2) Write cycle (0 waits)

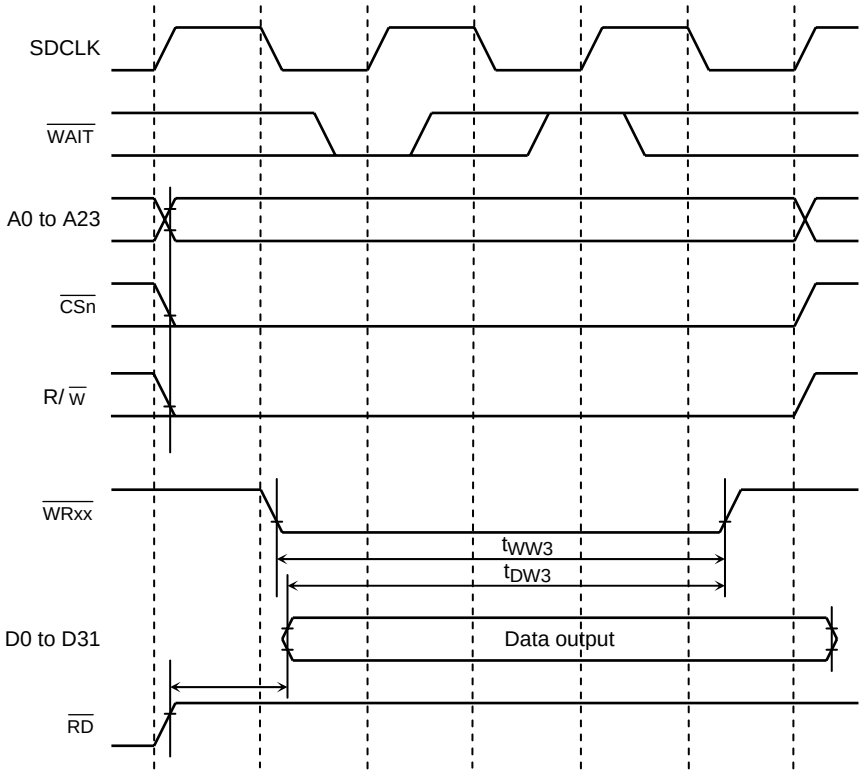


Note: The phase relation between X1 input signal and the other signals is undefined.
The above timing chart is an example.

(3) Read cycle (1 wait)



(4) Write cycle (1 wait)



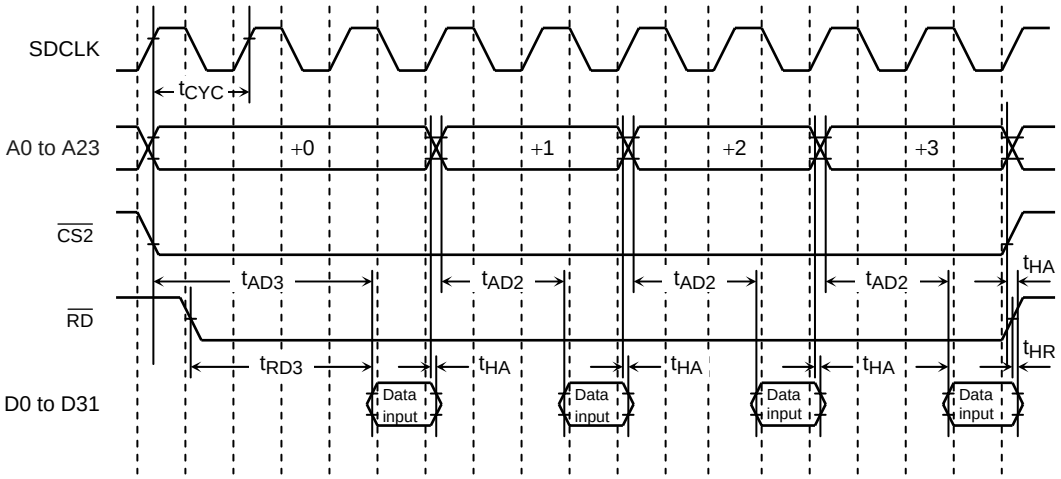
4.3.2 Page ROM Read Cycle

(1) 3-2-2-2 mode

No.	Parameter	Symbol	Variable		40 MHz	36 MHz	27 MHz	Unit
			Min	Max				
1	System clock period (= T)	t_{CYC}	50	166.7	50	55.5	74	ns
2	A0, A1 → D0 to D31 input	t_{AD2}		2.0T – 50	50	61	98	
3	A2 to A23 → D0 to D31 input	t_{AD3}		3.0T – 50	100	116.5	172	
4	$\overline{RD}$ falling → D0 to D31 input	t_{RD3}		2.5T – 45	80	93.8	140	
5	A0 to A23 Invalid → D0 to D31 hold	t_{HA}	0		0	0	0	
6	$\overline{RD}$ rising → D0 to D31 hold	t_{HR}	0		0	0	0	

AC measuring condition

- Output: High = 0.7 VCC, Low = 0.3 VCC, C_L = 50 pF
- Input: High = 0.9 VCC, Low = 0.1 VCC



4.3.3 SDRAM Controller AC Characteristics

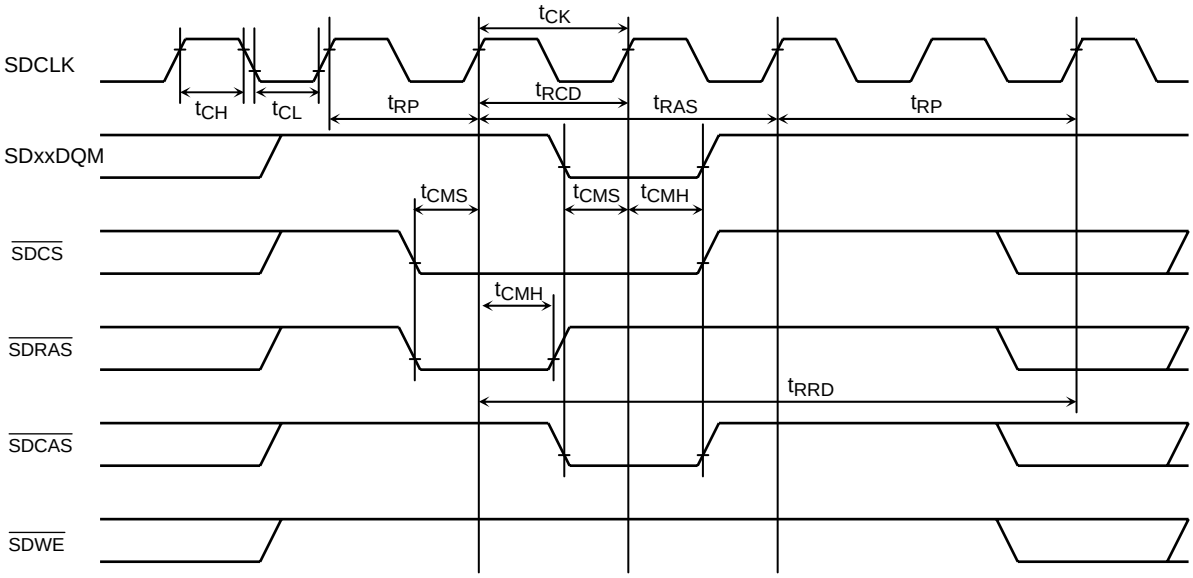
No.	Parameter	Symbol	Variable		40 MHz	36 MHz	27 MHz	Unit
			Min	Max				
1	Ref/active to ref/active command period	t_{RC}	2T		100	111	148	ns
2	Active to precharge command period	t_{RAS}	2T	12210	100	111	148	
3	Active to read/write command delay time	t_{RCD}	T		50	55.5	74	
4	Precharge to active command period	t_{RP}	T		50	55.5	74	
5	Active to active command period	t_{RRD}	3T		150	166.5	222	
6	Write recovery time (CL* = 2)	t_{WR}	T		50	55.5	74	
7	Clock cycle time (CL* = 2)	t_{CK}	T		50	55.5	74	
8	Clock high level width	t_{CH}	0.5T – 15		10	12.7	22	
9	Clock low level width	t_{CL}	0.5T – 15		10	12.7	22	
10	Access time from clock (CL* =2)	t_{AC}		T – 30	20	25.5	44	
11	Output data hold time	t_{OH}	0		0	0	0	
12	Data in setup time	t_{DS}	T – 35		15	20.5	39	
13	Data in hold time	t_{DH}	T – 5		45	50.5	69	
14	Address setup time	t_{AS}	0.75T – 30		7.5	11.6	25.5	
15	Address hold time	t_{AH}	0.25T – 9		3.5	4.8	9.5	
16	CKE setup time	t_{CKS}	0.5T – 15		10	12.7	22	
17	Command setup time	t_{CMS}	0.5T – 15		10	12.7	22	
18	Command hold time	t_{CMH}	0.5T – 15		10	12.7	22	
19	Mode register set cycle time	t_{RSC}	T		50	55.5	74	

CL*: CAS latency.

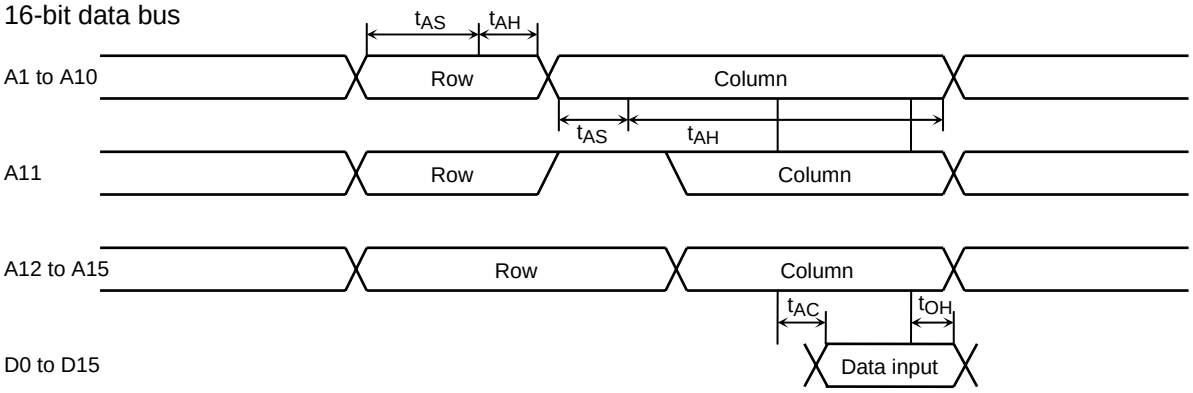
AC measuring conditions

- Output level: High = 0.7 VCC, Low = 0.3 VCC, C_L = 50 pF
- Input level: High = 0.9 VCC, Low = 0.1 VCC

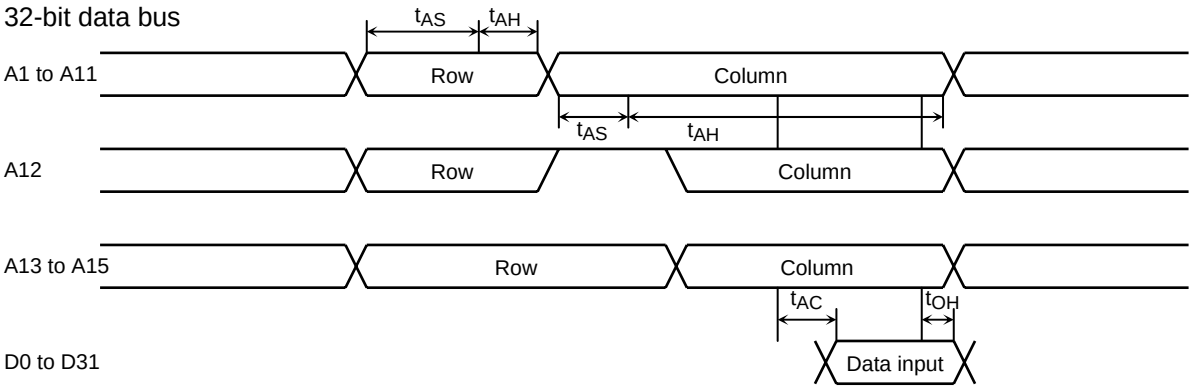
(1) SDRAM read timing (CPU access or LCDC normal access)



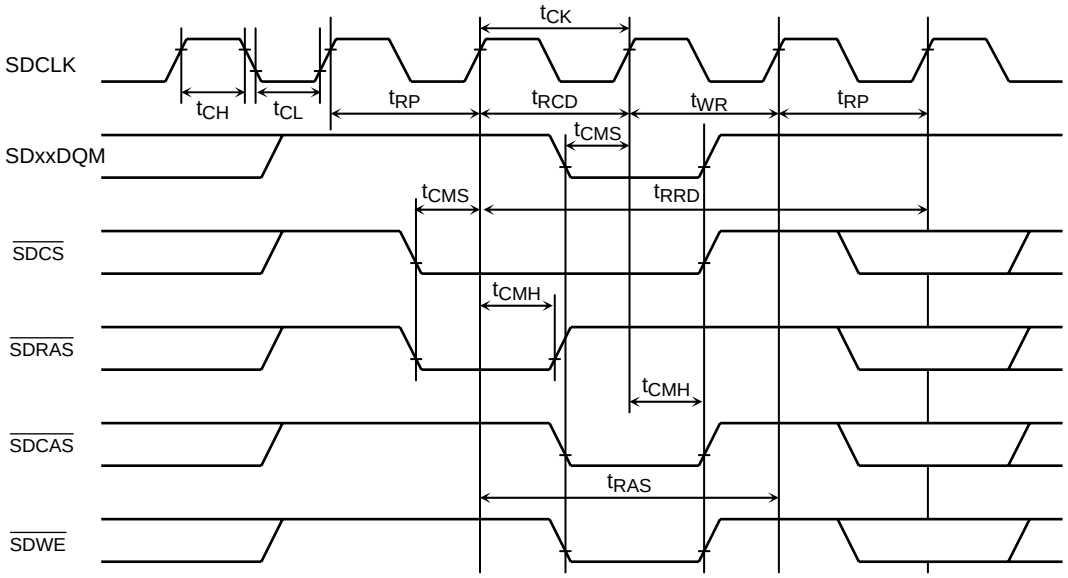
16-bit data bus



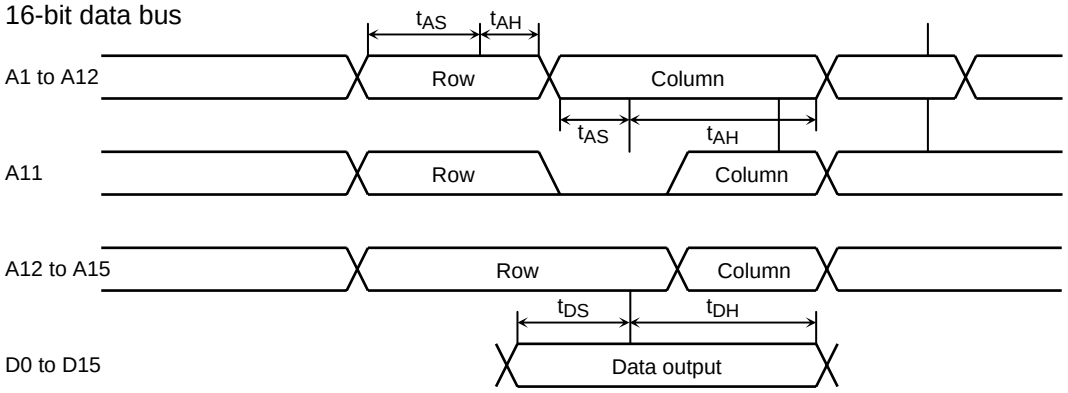
32-bit data bus



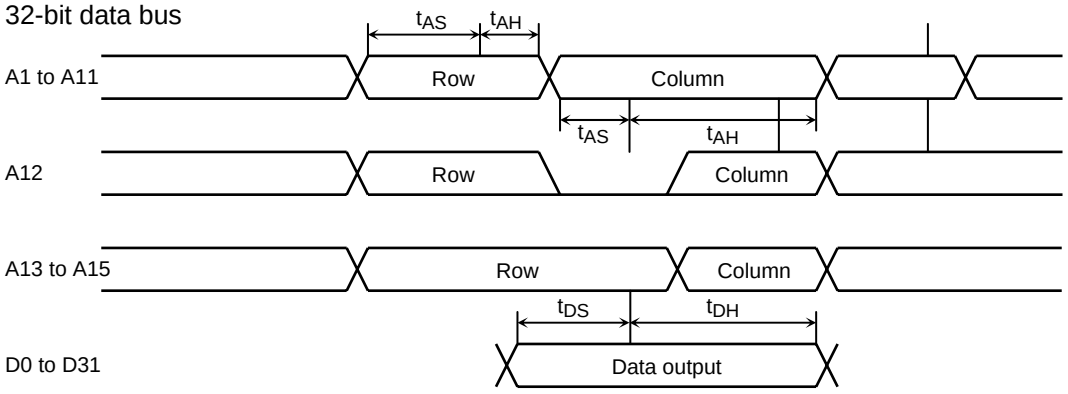
(2) SDRAM write timing (CPU access)



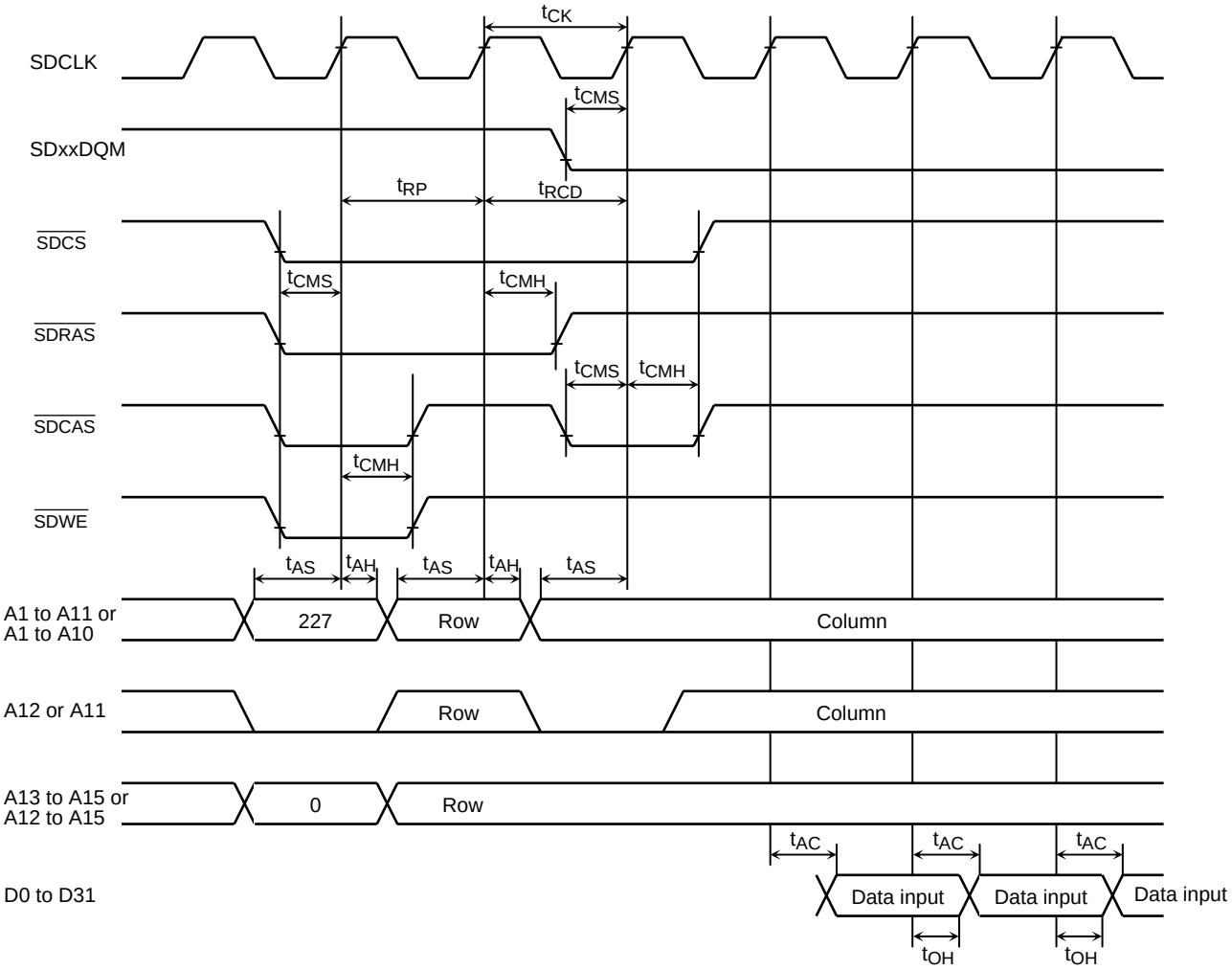
16-bit data bus



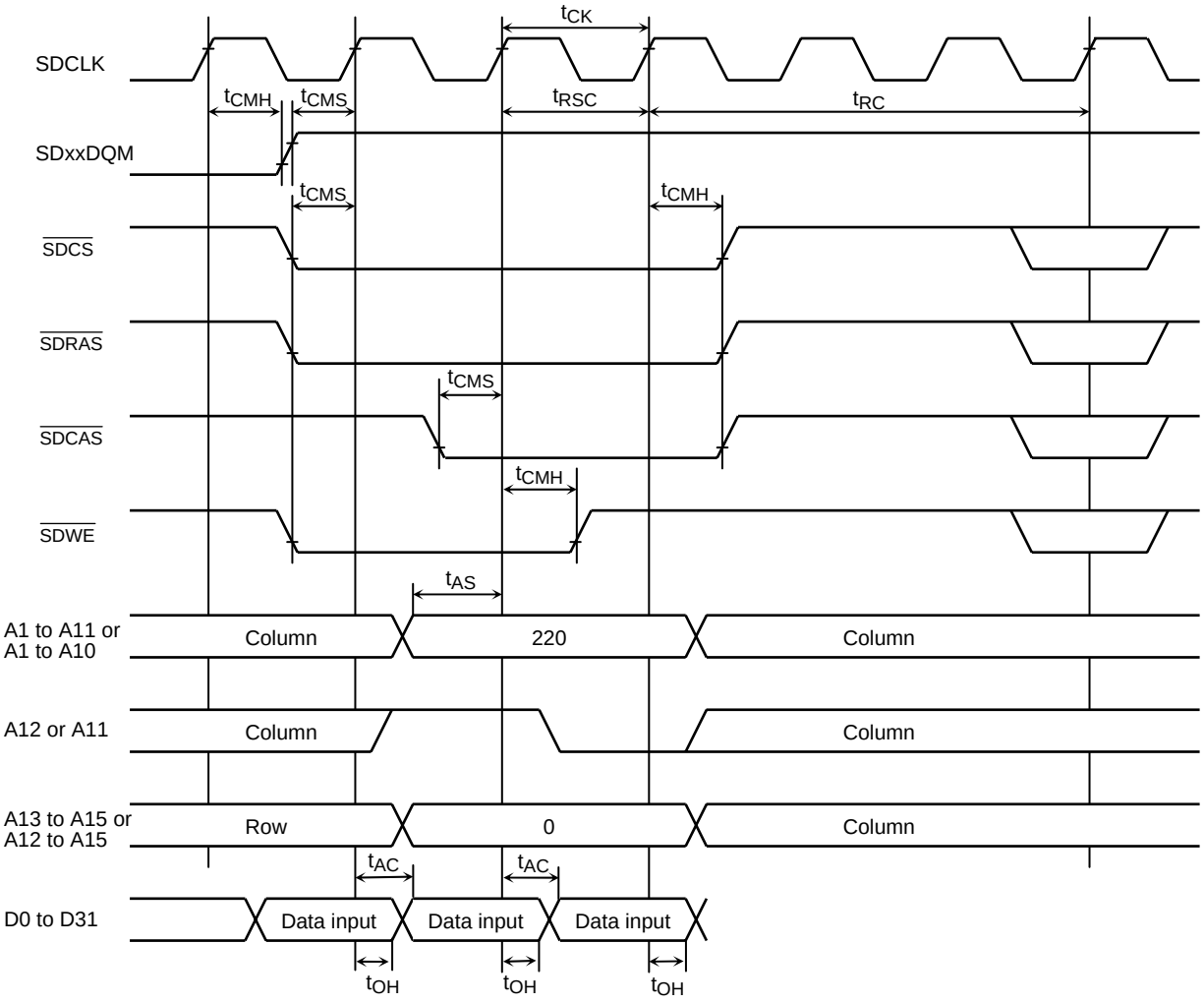
32-bit data bus



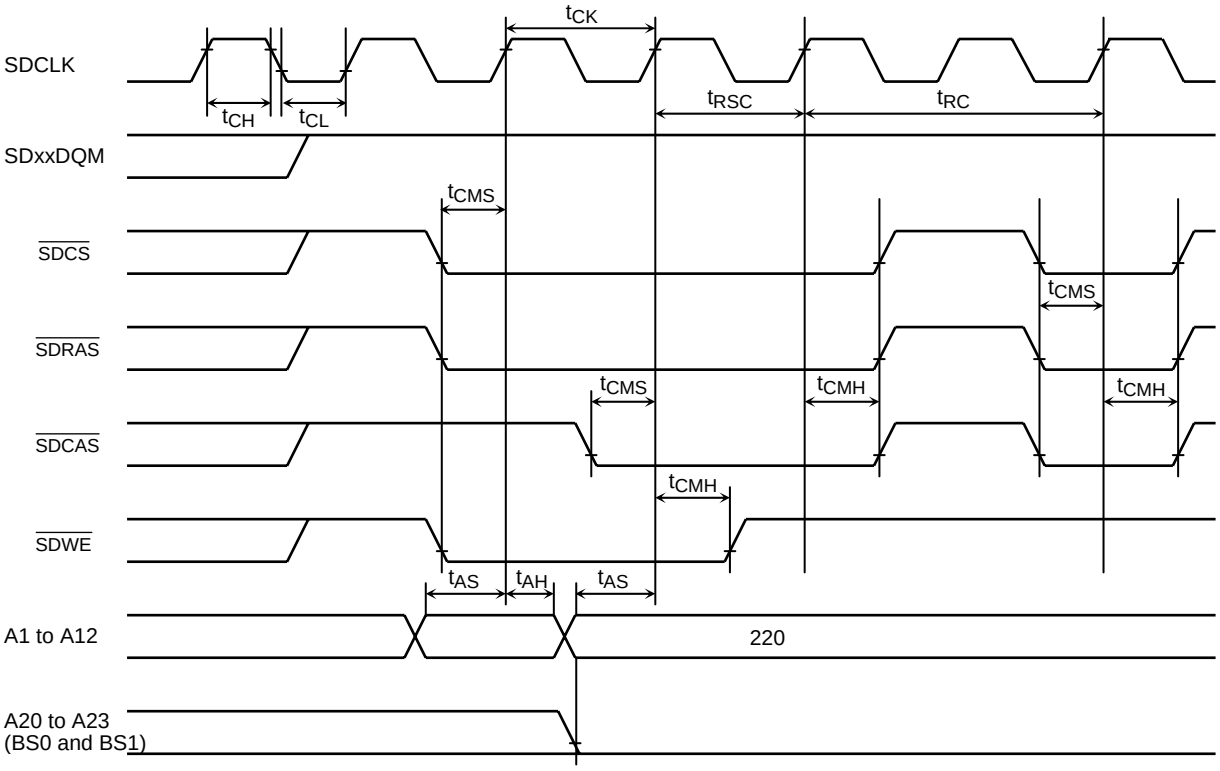
(3) SDRAM burst read timing (Start of burst cycle)



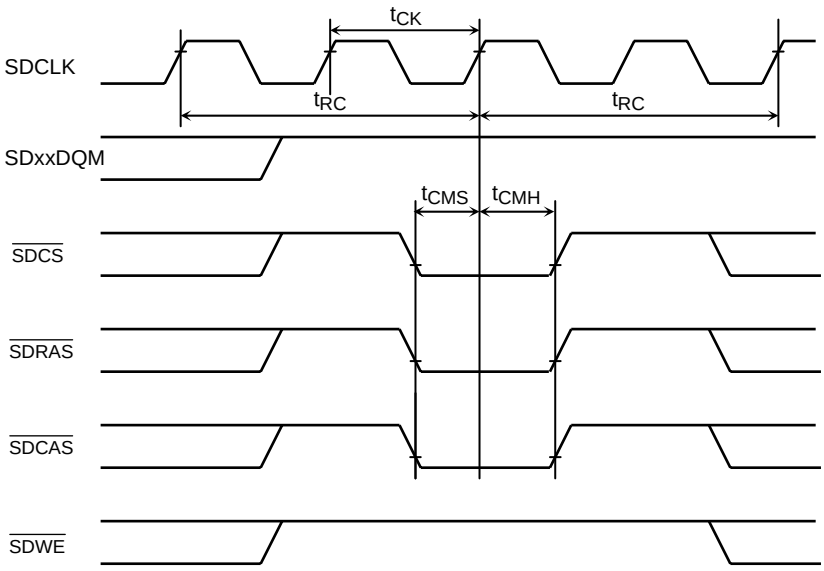
(4) SDRAM burst read timing (End of burst cycle)



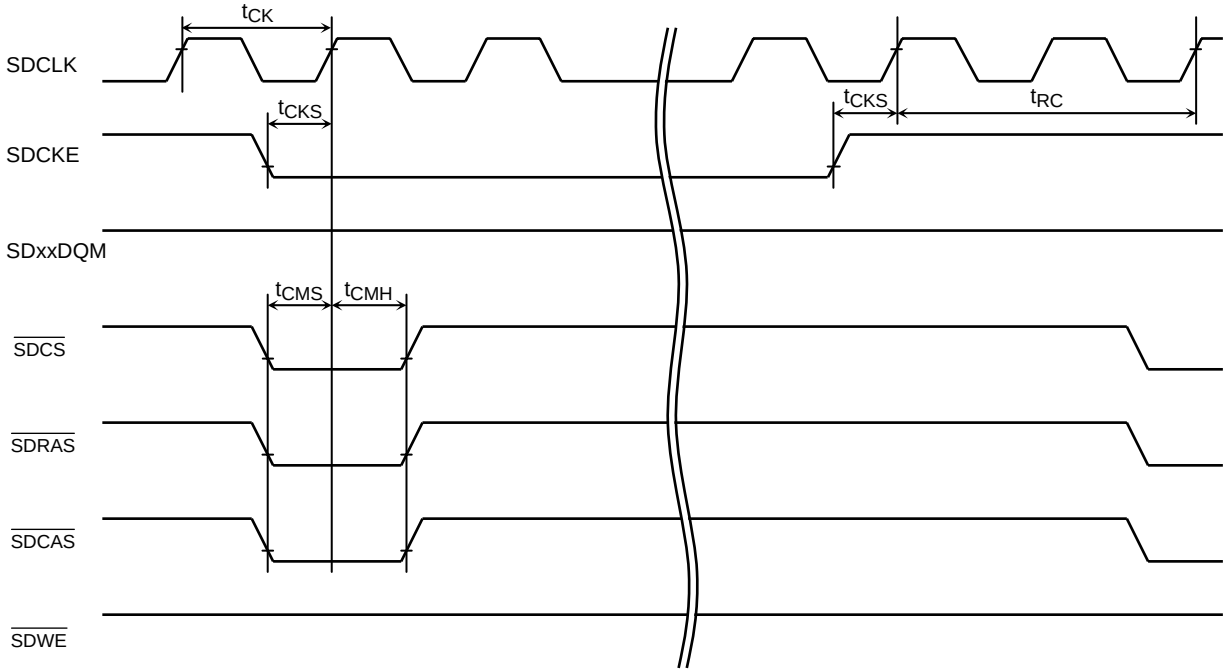
(5) SDRAM initialize timing



(6) SDRAM refresh timing



(7) SDRAM self refresh timing



4.3.4 NAND Flash Controller AC Characteristics

No.	Parameter	Symbol	Variable		40 MHz	36 MHz	27 MHz	Unit
			Min	Max				
1	$\overline{\text{NDRE}}$ low width	t_{RP}	(1 + n) T – 12		38	43.5	62	ns
2	$\overline{\text{NDRE}}$ data access time	$t_{REA (3.0 V)}$		(1 + n) T – 25	25	30.5	–	
		$t_{REA (2.7 V)}$		(1 + n) T – 30	–	–	44	
3	Read data hold time	t_{OH}	0		0	0	0	
4	$\overline{\text{NDWE}}$ low width	t_{WP}	(0.75 + n) T – 20		17.5	21.6	35.5	
5	Write data setup time	t_{DS}	(3.25 + n) T – 30		132.5	150.3	210.5	
6	Write data hold time	t_{DH}	0.25 T – 2		10.5	11.8	16.5	

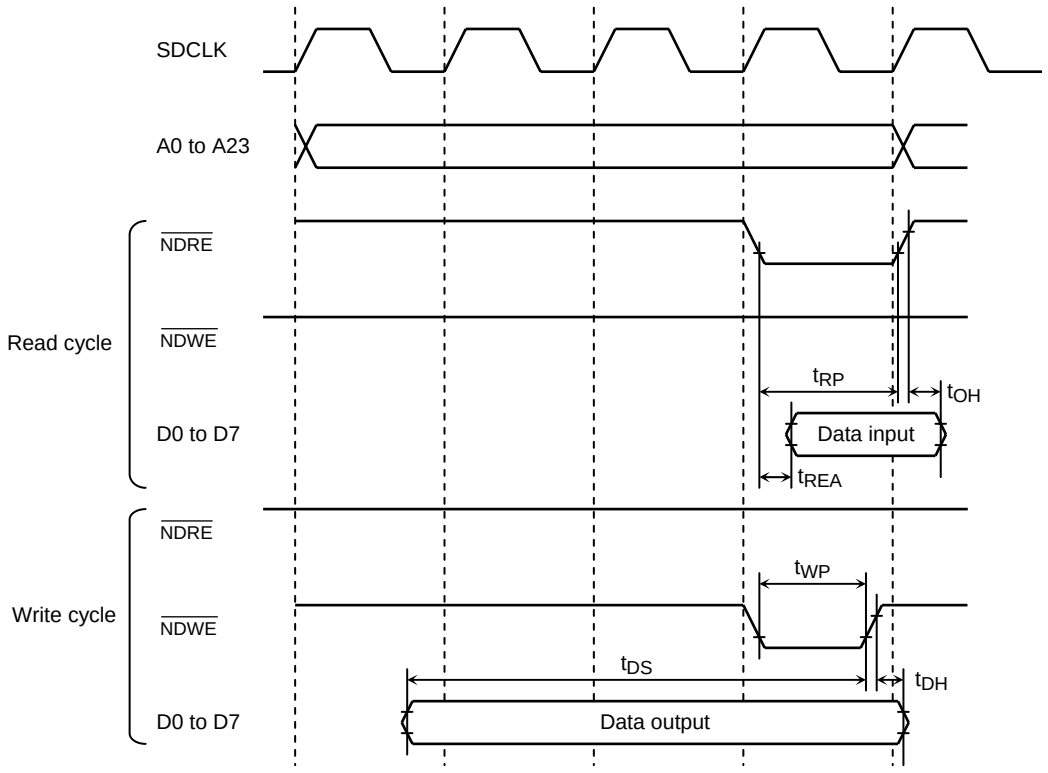
AC measuring conditions

- Output level: High = 0.7 VCC, Low = 0.3 VCC, C_L = 50 pF
- Input level: High = 0.9 VCC, Low = 0.1 VCC

Note 1: The “n” shown in “Variable” refers to the wait number which is set to NDnFSPR<SPW3:0> register.

Example: When NDnFSPR<SPW3:0> = “0001”, t_{RP} = (1 + n) T – 12 = 2T – 12

Note 2: The figures in the “Variable” column cover the whole VCC range (2.7 to 3.6V).
Exceptions are shown by the VCC (min), “(3.0 V)” or “ (2.7 V)”, added to the ”Symbol” column.



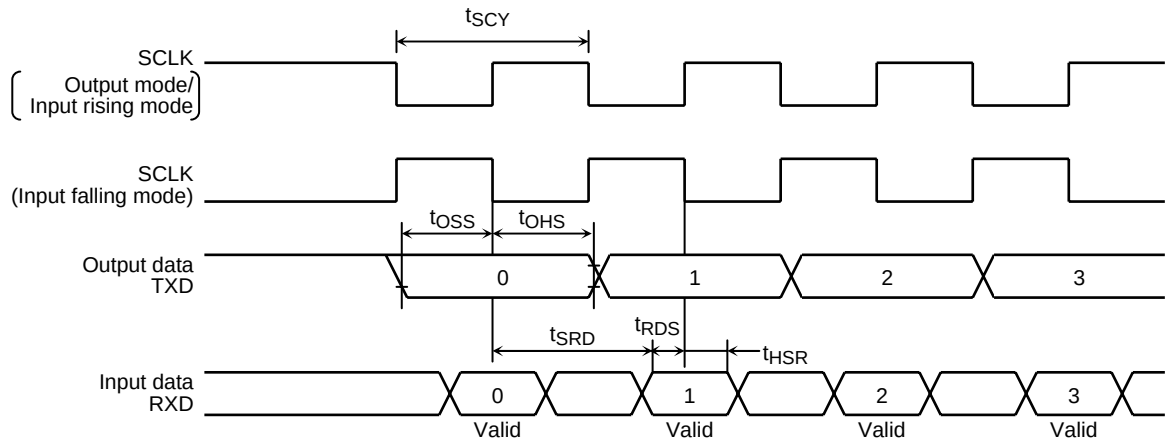
4.3.5 Serial Channel Timing

(1) SCLK input mode (I/O interface mode)

Parameter	Symbol	Variable		40 MHz	36 MHz	27 MHz	Unit
		Min	Max				
SCLK cycle	t_{SCY}	16T		0.8	0.888	1.184	μs
Output data → SCLK rising/falling	t_{OSS}	$t_{SCY}/2 - 4T - 110$		90	114	186	ns
SCLK rising/falling → Output data hold	t_{OHS}	$t_{SCY}/2 + 2T + 0$		500	554	740	
SCLK rising/falling → Input data hold	t_{HSR}	3 T + 10		160	175	232	
SCLK rising/falling → Input data valid	t_{SRD}		$t_{SCY} - 0$	800	888	1184	
Input data valid → SCLK rising/falling	t_{RDS}	0		0	0	0	

(2) SCLK output mode (I/O Interface mode)

Parameter	Symbol	Variable		40 MHz	36 MHz	27 MHz	Unit
		Min	Max				
SCLK cycle (Programmable)	t_{SCY}	16 T	8192T	0.8	0.888	1.184	μs
Output data → SCLK rising/falling	t_{OSS}	$t_{SCY}/2 - 40$		360	404	552	ns
SCLK rising/falling → Output data hold	t_{OHS}	$t_{SCY}/2 - 40$		360	404	552	
SCLK rising/falling → Input data hold	t_{HSR}	0		0	0	0	
SCLK rising/falling → Input data valid	t_{SRD}		$t_{SCY} - 1T - 180$	570	654	967	
Input data valid → SCLK rising/falling	t_{RDS}	1 T + 180		230	233	253	

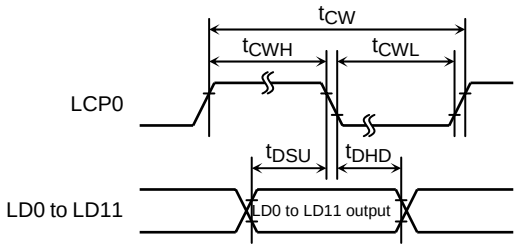


4.3.6 Interrupt Operation

Parameter	Symbol	Variable		40 MHz	36 MHz	27 MHz	Unit
		Min	Max				
INT0 to INT5 low width	t_{INTAL}	4 T + 40		240	262	336	ns
INT0 to INT5 high width	t_{INTAH}	4 T + 40		240	262	336	

4.3.7 LCD Controller (SR mode)

Parameter	Symbol	Variable		40 MHz	36 MHz	27 MHz	Unit
		Min	Max				
LCP0 clock period (= tm)	t_{CW}	2 T		100	111	148	ns
LCP0 high width	t_{CWH}	0.5 tm – 12		38	43.5	62	
LCP0 low width	t_{CWL}	0.5 tm – 12		38	43.5	62	
Data valid → LCP0 falling	t_{DSU}	0.5 tm – 20		30	35.5	54	
LCP0 falling → Data hold	t_{DHD}	0.5 tm – 5		45	50.5	69	

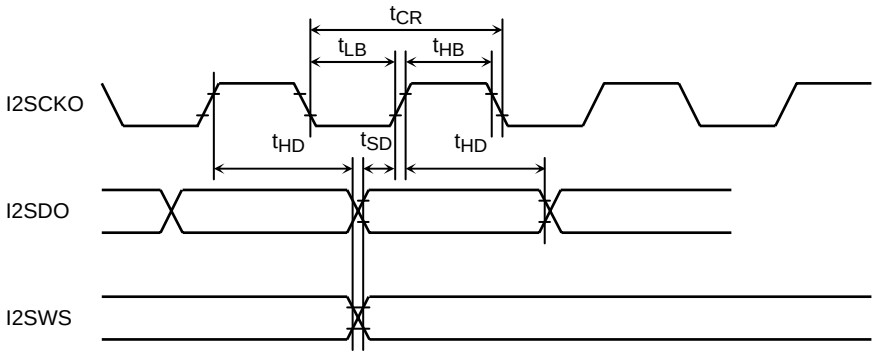


4.3.8 I^2S Timing (I^2S, SIO Mode)

Parameter	Symbol	Variable		40 MHz	36 MHz	27 MHz	Unit
		Min	Max				
I2SCKO clock period	t_{CR}	T		50	55	74	ns
I2SCKO high width	t_{HB}	0.5 t_{CR} – 15		10	12	22	
I2SCKO low width	t_{LB}	0.5 t_{CR} – 15		10	12	22	
I2SDO, I2SWS setup time	t_{SD}	0.5 t_{CR} – 15		10	12	22	
I2SDO, I2SWS hold time	t_{HD}	0.5 t_{CR} – 5		20	22	32	

AC measuring conditions

- Output level: High = 0.7 VCC, Low = 0.3 VCC, C_L = 10 pF

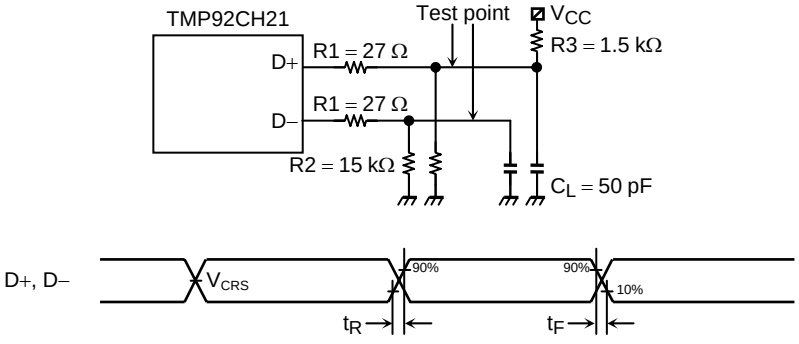


4.3.9 USB Timing (Full-speed)

V_{CC} = 3.3 ± 0.3 V/f_{USB} = 48 MHz/Ta = –20 to 70°C

Parameter	Symbol	Min	Max	Unit
Rising time for D+, D–	t_R	4	20	ns
Falling time for D+, D–	t_F	4	20	
Output signal crossover voltage	V_{CRS}	1.3	2.0	V

AC measuring conditions



4.4 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}	V_{SS}	V_{SS}	V_{SS} + 0.2	
AD converter power supply voltage	AV_{CC}	V_{CC}	V_{CC}	V_{CC}	
AD converter ground	AV_{SS}	V_{SS}	V_{SS}	V_{SS}	
Analog input voltage	AV_{IN}	V_{REFL}		V_{REFH}	
Analog current for analog reference voltage <VREFON> = 1	I_{REF}		0.8	1.35	mA
Analog current for analog reference voltage <VREFON> = 0			0.02	5.0	μA
Total error (Quantize error of ± 0.5 LSB is included.)	E_T		±1.0	±4.0	LSB

Note 1: 1LSB = (VREFH – VREFL) / 1024 [V]

Note 2: Minimum frequency for operation

AD converter operation is guaranteed only when using fc (high-frequency oscillator). fs is not guaranteed. However, operation is guaranteed if the clock frequency selected by the clock gear is over 4MHz.

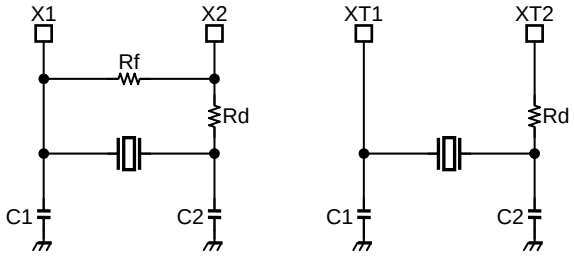
Note 3: The value for Icc includes the current which flows through the AV_{CC} pin.

4.5 Recommended Oscillation Circuit

The TMP92CH21 has been evaluated by the oscillator vender below. Use this information when selecting external parts.

Note: The total load value of the oscillator 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



High-frequency oscillator Low-frequency oscillator

(2) Recommended ceramic oscillator: Murata Manufacturing Co., Ltd.

MCU	Oscillation Frequency [MHZ]	Oscillator Product Number	Parameter of Elements				Running Condition	
			C1 [pF]	C2 [pF]	Rd [Ω]	Rf [Ω]	Voltage [V]	T_C [°C]
TMP92CH21FG	2.00	CSTCC2M00G56-R0	(47)	(47)	0	Open	1.8 ~ 2.7	-20 ~ +80
	4.00	CSTCR4M00G55-R0	(39)	(39)			2.7 ~ 3.6	
		CSTLS4M00G56-B0	(47)	(47)				
	6.00	CSTCR6M00G55-R0	(39)	(39)				
		CSTLS6M00G56-B0	(47)	(47)				
	9.00	CSTCE9M00G55-R0	(33)	(33)				
		CSTLS9M00G56-B0	(47)	(47)				
	10.00	CSTCE10M0G52-R0	(10)	(10)			1.8 ~ 2.7	
		CSTCE10M0G55-R0	(33)	(33)			2.7 ~ 3.6	
		CSTLS10M0G53-B0	(15)	(15)			1.8 ~ 2.7	
		CSTLS10M0G56-B0	(47)	(47)			2.7 ~ 3.6	
	12.00	CSTCE12M5G52-R0	(10)	(10)			1.8 ~ 2.7	
	20.00	CSTCE20M0V53-R0	(15)	(15)			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

(3) Recommended ceramic oscillator: TDK Co., Ltd.

MCU	Oscillation Frequency [MHZ]	Oscillator Product Number	Parameter of Elements				Running Condition	
			C1 [pF]	C2 [pF]	Rf [Ω]	Rd [Ω]	Voltage [V]	T_C [°C]
TMP92CH21FG	4.00	FCR4.0MC5	–	–	–	–	2.7 ~ 3.6	–20 ~ 70
	6.00	FCR6.0MC5	–	–	–	–		
	10.00	FCR10.MC5	–	–	–	–		
	20.00	CCR20.0MXC7	–	–	–	–		
	40.00	CCR40.0MXC7	–	–	–	–		

Note: The product numbers and specifications of the oscillators made by TDL Co.,Ltd. are subject change. For up-to-date information, please refer to the following URL;

http://www.tdk.co.jp