Description

The M16C/80 group of single-chip microcomputers are built using the high-performance silicon gate CMOS process using a M16C/80 Series CPU core and are packaged in a 100-pin and 144-pin plastic molded QFP. The peripheral functions of 100-pin and 144-pin are common. These single-chip microcomputers operate using sophisticated instructions featuring a high level of instruction efficiency. With 16M bytes of address space, they are capable of executing instructions at high speed. They also feature a built-in multiplier and DMAC, making them ideal for controlling office, communications, industrial equipment, and other high-speed processing applications.

Features

• Memory capacity	ROM (See ROM expansion figure.) RAM 10 to 24 Kbytes
• Shortest instruction execution time	50ns (f(X_{IN})=20MHz)
• Supply voltage	4.2 to 5.5V (f(X_{IN})=20MHz) Mask ROM, external ROM and flash memory versions 2.7 to 5.5V (f(X_{IN})=10MHz) Mask ROM, external ROM and flash memory versions
• Low power consumption	45mA (M30800MC-XXXFP) (f(X_{IN}) = 20MHz without software wait,Vcc=5V)
• Interrupts	29 internal and 8 external interrupt sources, 5 software interrupt sources; 7 levels (including key input interrupt)
• Multifunction 16-bit timer	5 output timers + 6 input timers
• Serial I/O	5 channels for UART or clock synchronous
• DMAC	4 channels (trigger: 31 sources)
• DRAMC	Used for EDO, FP, CAS before RAS refresh, self-refresh
• A-D converter	10 bits X 8 channels (Expandable up to 10 channels)
• D-A converter	8 bits X 2 channels
• CRC calculation circuit	1 circuit
• X-Y converter	1 circuit
• Watchdog timer	1 line
• Programmable I/O	87 lines:100-pin version, 123 lines:144-pin version
• Input port	1 line ($P8_5$ shared with $\overline{NMI}$ pin)
• Memory expansion	Available (16M bytes)
• Chip select output	4 lines
• Clock generating circuit	2 built-in clock generation circuits (built-in feedback resistance, and external ceramic or quartz oscillator)

Specifications written in this manual are believed to be accurate, but are not guaranteed to be entirely free of error. Specifications in this manual may be changed for functional or performance improvements. Please make sure your manual is the latest edition.

Applications

Audio, cameras, office equipment, communications equipment, portable equipment, etc.

------Table of Contents------

CPU ... 13
Reset ... 18
Processor Mode ... 26
Clock Generating Circuit ... 43
Protection ... 55
Interrupts ... 56
Watchdog Timer ... 78
DMAC ... 80
Timer ... 91
Serial I/O ... 123
A-D Converter ... 165
D-A Converter ... 175
CRC Calculation Circuit ... 177
X-Y converter ... 179
DRAM controller ... 182
Programmable I/O Ports ... 189
Usage Precaution ... 207
Electric characteristics ... 218
Flash memory version ... 265

Pin Configuration

Figures 1.1.1 and 1.1.2 show the pin configuration (top view) for 100-pin and Figure 1.1.3 shows the pin configuration (top view) for 144-pin.

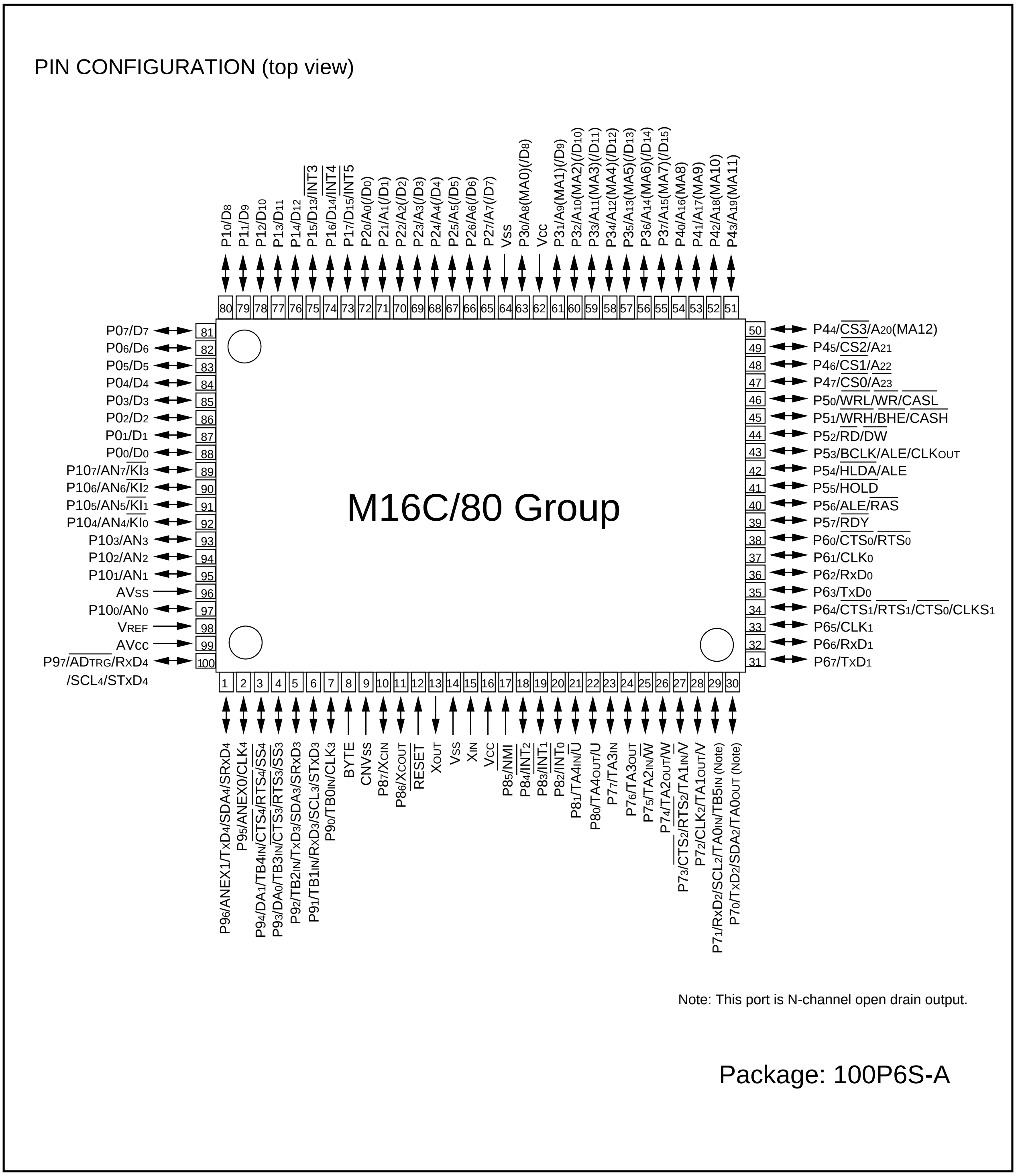
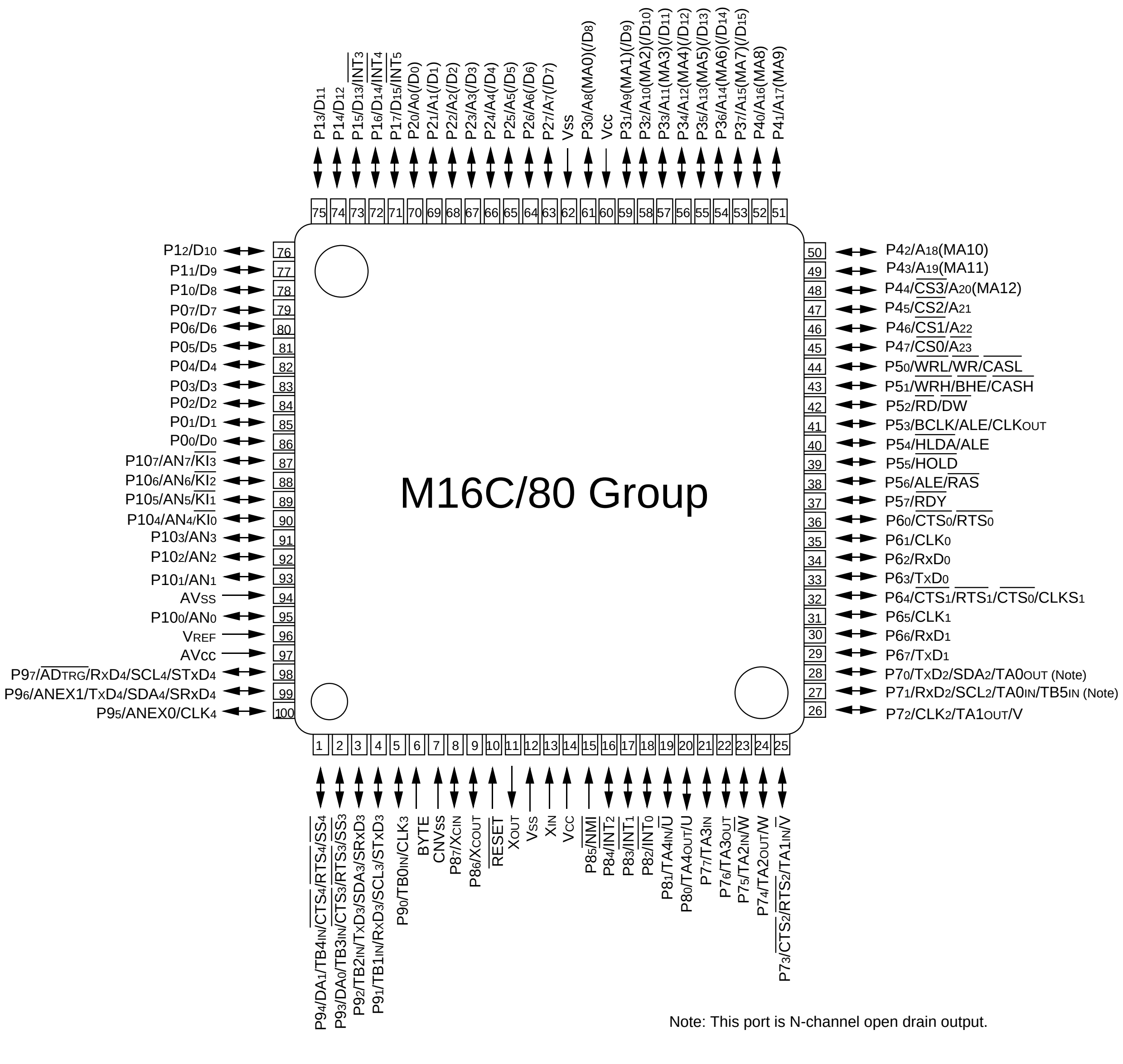


Figure 1.1.1. Pin configuration for 100-pin version (top view) (1)

PIN CONFIGURATION (top view)

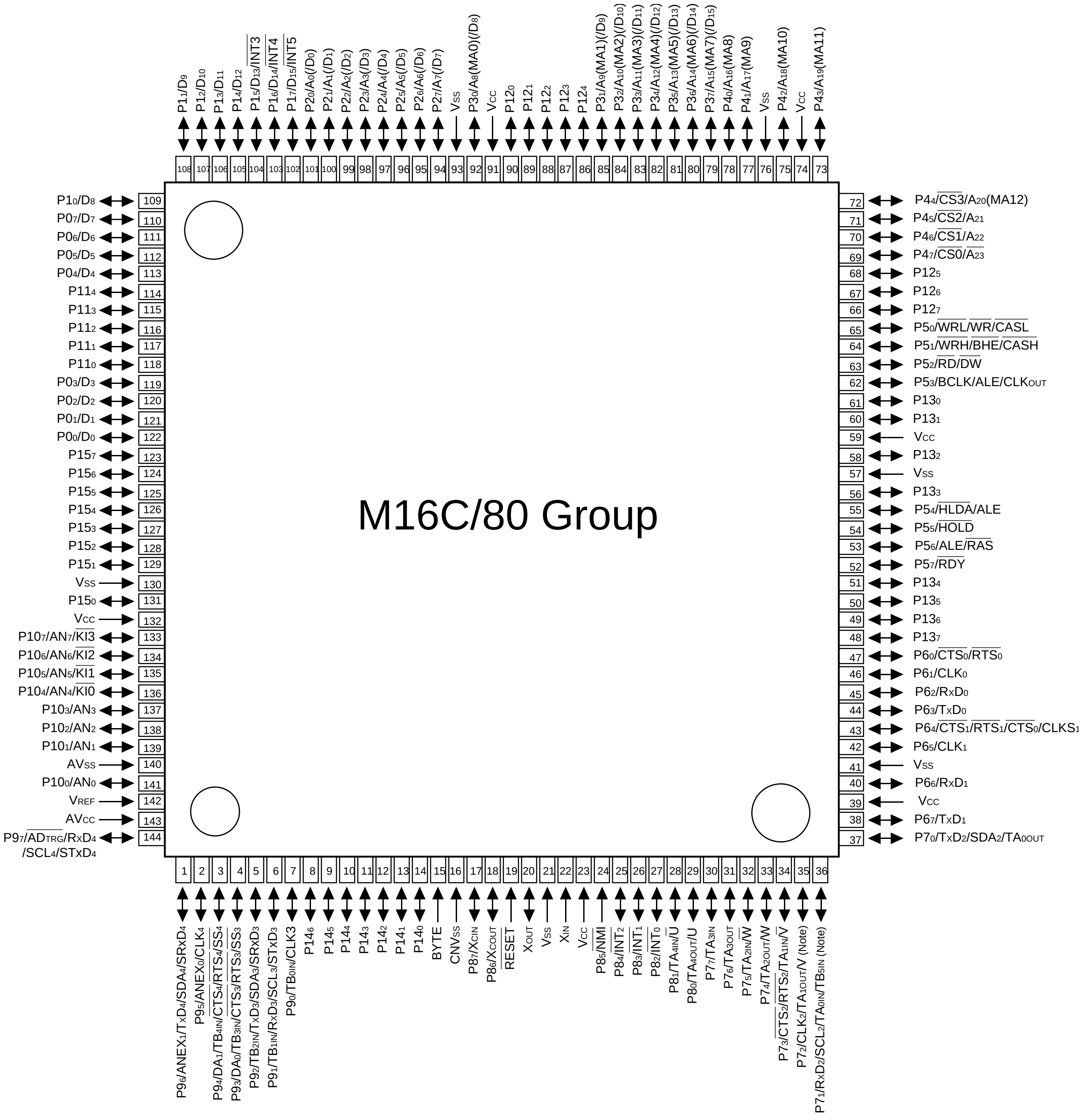


Note: This port is N-channel open drain output.

Package: 100P6Q-A

Figure 1.1.2. Pin configuration for 100-pin version (top view) (2)

PIN CONFIGURATION (top view)


M16C/80 Group
108 $P1_1/D_9$
107 $P1_2/D_{10}$
106 $P1_3/D_{11}$
105 $P1_4/D_{12}$
104 $P1_5/D_{13}/\overline{INT3}$
103 $P1_6/D_{14}/\overline{INT4}$
102 $P1_7/D_{15}/\overline{INT5}$
101 $P2_0/A_0(/D_0)$
100 $P2_1/A_1(/D_1)$
99 $P2_2/A_2(/D_2)$
98 $P2_3/A_3(/D_3)$
97 $P2_4/A_4(/D_4)$
96 $P2_5/A_5(/D_5)$
95 $P2_6/A_6(/D_6)$
94 $P2_7/A_7(/D_7)$
93 V_{SS}
92 $P3_0/A_8(MA0)(/D_8)$
91 V_{CC}
90 $P12_0$
89 $P12_1$
88 $P12_2$
87 $P12_3$
86 $P12_4$
85 $P3_1/A_9(MA1)(/D_9)$
84 $P3_2/A_{10}(MA2)(/D_{10})$
83 $P3_3/A_{11}(MA3)(/D_{11})$
82 $P3_4/A_{12}(MA4)(/D_{12})$
81 $P3_5/A_{13}(MA5)(/D_{13})$
80 $P3_6/A_{14}(MA6)(/D_{14})$
79 $P3_7/A_{15}(MA7)(/D_{15})$
78 $P4_0/A_{16}(MA8)$
77 $P4_1/A_{17}(MA9)$
76 V_{SS}
75 $P4_2/A_{18}(MA10)$
74 V_{CC}
73 $P4_3/A_{19}(MA11)$
72 $P4_4/\overline{CS3}/A_{20}(MA12)$
71 $P4_5/\overline{CS2}/A_{21}$
70 $P4_6/\overline{CS1}/A_{22}$
69 $P4_7/\overline{CS0}/A_{23}$
68 $P12_5$
67 $P12_6$
66 $P12_7$
65 $P5_0/\overline{WRL}/\overline{WR}/\overline{CASL}$
64 $P5_1/\overline{WRH}/\overline{BHE}/\overline{CASH}$
63 $P5_2/\overline{RD}/\overline{DW}$
62 $P5_3/BCLK/ALE/CLK_{OUT}$
61 $P13_0$
60 $P13_1$
59 V_{CC}
58 $P13_2$
57 V_{SS}
56 $P13_3$
55 $P5_4/\overline{HLDA}/ALE$
54 $P5_5/\overline{HOLD}$
53 $P5_6/ALE/\overline{RAS}$
52 $P5_7/\overline{RDY}$
51 $P13_4$
50 $P13_5$
49 $P13_6$
48 $P13_7$
47 $P6_0/\overline{CTS_0}/\overline{RTS_0}$
46 $P6_1/CLK_0$
45 $P6_2/RxD_0$
44 $P6_3/TxD_0$
43 $P6_4/\overline{CTS_1}/\overline{RTS_1}/\overline{CTS_0}/CLKS_1$
42 $P6_5/CLK_1$
41 V_{SS}
40 $P6_6/RxD_1$
39 V_{CC}
38 $P6_7/TxD_1$
37 $P7_0/TxD_2/SDA_2/TA0_{OUT}$
109 $P1_0/D_8$
110 $P0_7/D_7$
111 $P0_6/D_6$
112 $P0_5/D_5$
113 $P0_4/D_4$
114 $P11_4$
115 $P11_3$
116 $P11_2$
117 $P11_1$
118 $P11_0$
119 $P0_3/D_3$
120 $P0_2/D_2$
121 $P0_1/D_1$
122 $P0_0/D_0$
123 $P15_7$
124 $P15_6$
125 $P15_5$
126 $P15_4$
127 $P15_3$
128 $P15_2$
129 $P15_1$
130 V_{SS}
131 $P15_0$
132 V_{CC}
133 $P10_7/AN_7/\overline{KI3}$
134 $P10_6/AN_6/\overline{KI2}$
135 $P10_5/AN_5/\overline{KI1}$
136 $P10_4/AN_4/\overline{KI0}$
137 $P10_3/AN_3$
138 $P10_2/AN_2$
139 $P10_1/AN_1$
140 AV_{SS}
141 $P10_0/AN_0$
142 V_{REF}
143 AV_{CC}
144 $P9_7/\overline{AD_{TRG}}/RxD_4/SCL_4/STxD_4$
1 $P9_6/ANEX_1/TxD_4/SDA_4/SRxD_4$
2 $P9_5/ANEX_0/CLK_4$
3 $P9_4/DA_1/TB4_{IN}/\overline{CTS_4}/\overline{RTS_4}/\overline{SS_4}$
4 $P9_3/DA_0/TB3_{IN}/\overline{CTS_3}/\overline{RTS_3}/\overline{SS_3}$
5 $P9_2/TB2_{IN}/TxD_3/SDA_3/SRxD_3$
6 $P9_1/TB1_{IN}/RxD_3/SCL_3/STxD_3$
7 $P9_0/TB0_{IN}/CLK3$
8 $P14_6$
9 $P14_5$
10 $P14_4$
11 $P14_3$
12 $P14_2$
13 $P14_1$
14 $P14_0$
15 BYTE
16 CNV_{SS}
17 $P8_7/X_{CIN}$
18 $P8_6/X_{COUT}$
19 $\overline{RESET}$
20 X_{OUT}
21 V_{SS}
22 X_{IN}
23 V_{CC}
24 $P8_5/\overline{NMI}$
25 $P8_4/\overline{INT_2}$
26 $P8_3/\overline{INT_1}$
27 $P8_2/\overline{INT_0}$
28 $P8_1/TA4_{IN}/\overline{U}$
29 $P8_0/TA4_{OUT}/U$
30 $P7_7/TA3_{IN}$
31 $P7_6/TA3_{OUT}$
32 $P7_5/TA2_{IN}/\overline{W}$
33 $P7_4/TA2_{OUT}/W$
34 $P7_3/\overline{CTS_2}/\overline{RTS_2}/TA1_{IN}/\overline{V}$
35 $P7_2/CLK_2/TA1_{OUT}/V$ (Note)
36 $P7_1/RxD_2/SCL_2/TA0_{IN}/TB5_{IN}$ (Note)

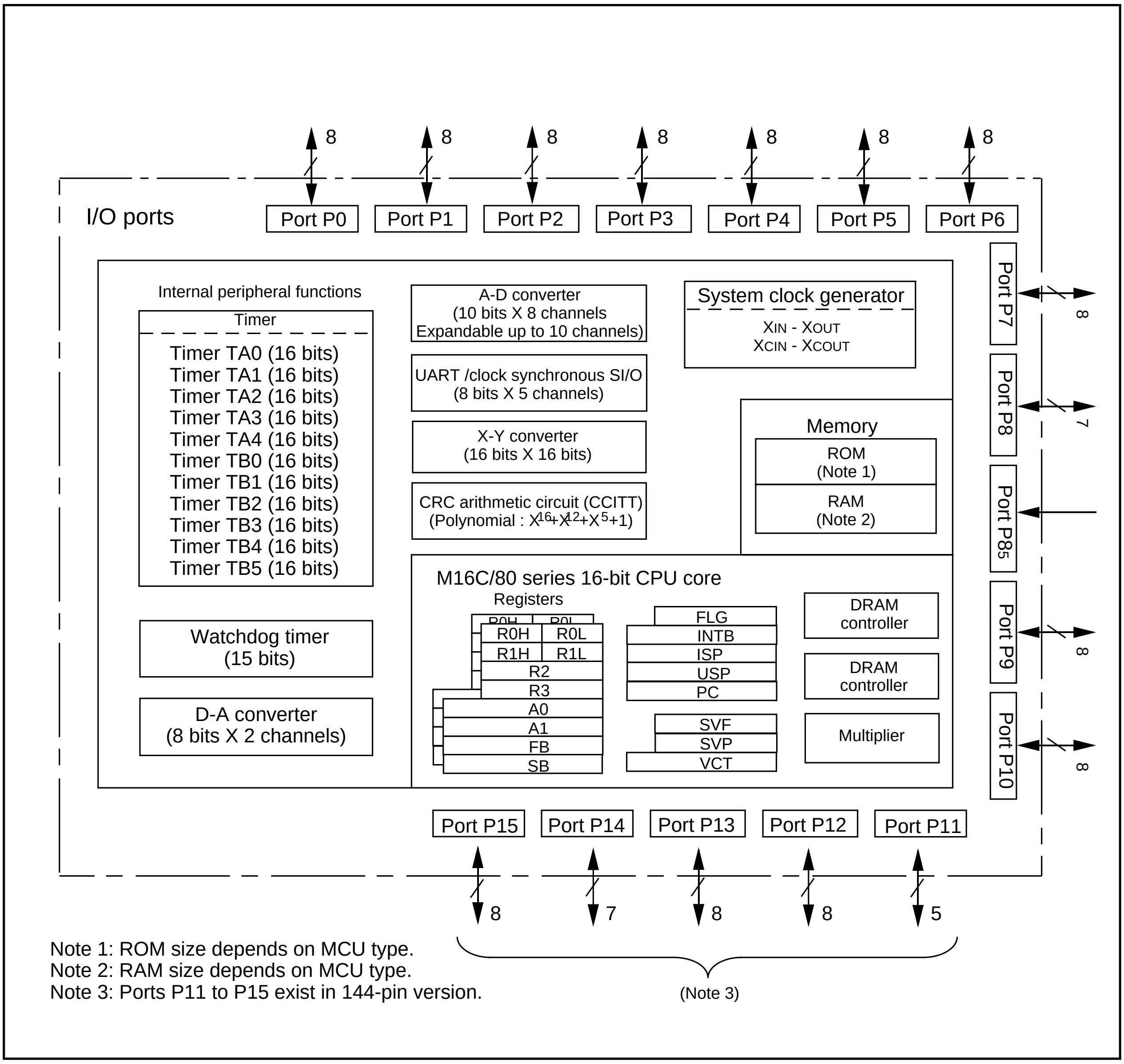
Note: This port is N-channel open drain output.

Package: 144P6Q-A

Figure 1.1.3. Pin configuration for 144-pin version (top view)

Block Diagram

Figure 1.1.4 is a block diagram of the M16C/80 group.



Note 1: ROM size depends on MCU type.
Note 2: RAM size depends on MCU type.
Note 3: Ports P11 to P15 exist in 144-pin version.

Figure 1.1.4. Block diagram of the M16C/80 group

Performance Outline

Table 1.1.1 is a performance outline of M16C/80 group.

Table 1.1.1. Performance outline of M16C/80 group

Item		Performance
Number of basic instructions		106 instructions
Shortest instruction execution time		50ns(f(X_{IN})=20MHz)
Memory capacity	ROM	See ROM expansion figure.
	RAM	10 to 24 K bytes
I/O port	100-pin	P0 to P10 (except $P8_5$) 8-bit x 10, 7-bit x 1
	144-pin	P0 to P15 (except $P8_5$) 8-bit x 13, 7-bit x 2, 5-bit x 1
Input port	$P8_5$	1 bit x 1
Multifunction timer	TA0, TA1, TA2, TA3,TA4 TB0, TB1, TB2, TB3, TB4, TB5	16 bits x 5 16 bits x 6
Serial I/O	UART0, UART1, UART2, UART3, UART4	(UART or clock synchronous) x 5
A-D converter		10 bits x (8 + 2) channels
D-A converter		8 bits x 2
DMAC		4 channels
DRAM controller		CAS before RAS refresh, self-refresh, EDO, FP
CRC calculation circuit		CRC-CCITT
X-Y converter		16 bits X 16 bits
Watchdog timer		15 bits x 1 (with prescaler)
Interrupt		29 internal and 8 external sources, 5 software sources, 7 levels
Clock generating circuit		2 built-in clock generation circuits (built-in feedback resistance, and external ceramic or quartz oscillator)
Supply voltage		4.2 to 5.5V (f(X_{IN})=20MHz) Mask ROM, external ROM and flash memory versions 2.7 to 5.5V (f(X_{IN})=10MHz) Mask ROM, external ROM and flash memory versions
Power consumption		45mA (f(X_{IN}) = 20MHz without software wait,Vcc=5V) Mask ROM 128 Kbytes version
I/O characteristics	I/O withstand voltage	5V
	Output current	5mA
Memory expansion		Available (up to 16M bytes)
Operating ambient temperature		–40 to 85^{o}C
Device configuration		CMOS high performance silicon gate
Package		100-pin and 144-pin plastic mold QFP

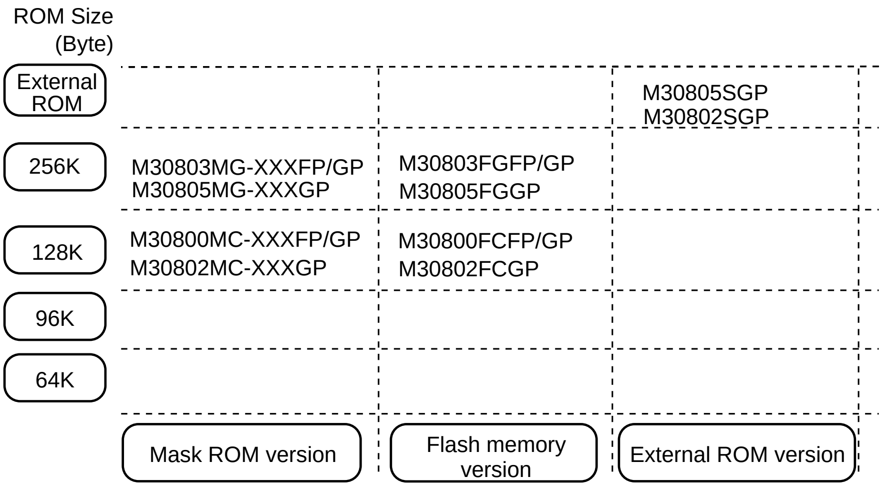
Mitsubishi plans to release the following products in the M16C/80 group:

(1) Support for mask ROM version, external ROM version and flash memory version

(2) ROM capacity

(3) Package

100P6S-A	: Plastic molded QFP (mask ROM version and flash memory version)
100P6Q-A	: Plastic molded QFP (mask ROM version and flash memory version)
144P6Q-A	: Plastic molded QFP (mask ROM version and flash memory version)



ROM Size (Byte)	Mask ROM version	Flash memory version	External ROM version
External ROM			M30805SGP M30802SGP
256K	M30803MG-XXXFP/GP M30805MG-XXXGP	M30803FGFP/GP M30805FGGP	
128K	M30800MC-XXXFP/GP M30802MC-XXXGP	M30800FCFP/GP M30802FCGP	
96K			
64K			

Figure 1.1.5. ROM expansion

The M16C/80 group products currently supported are listed in Table 1.1.2.

Table 1.1.2. M16C/80 group

As of August, 2000

Type No	ROM capacity	RAM capacity	Package type	Remarks
M30800MC-XXXFP	128K bytes	10K bytes	100P6S-A	Mask ROM version
M30800MC-XXXGP			100P6Q-A	
M30803MG-XXXFP	256K bytes	20K bytes	100P6S-A	
M30803MG-XXXGP			100P6Q-A	
M30800FCFP **	128K bytes	10K bytes	100P6S-A	Flash memory version
M30800FCGP **			100P6Q-A	
M30803FGFP **	256K bytes	20K bytes	100P6S-A	
M30803FGGP **			100P6Q-A	
M30802MC-XXXGP	128K bytes	10K bytes	144P6Q-A	Mask ROM version
M30805MG-XXXGP	256K bytes	20K bytes		
M30802FCGP **	128K bytes	10K bytes		Flash memory version
M30805FGGP **	256K bytes	20K bytes		
M30802SGP	————	10K bytes		External ROM version
M30805SGP	————	24K bytes		

** :Under development

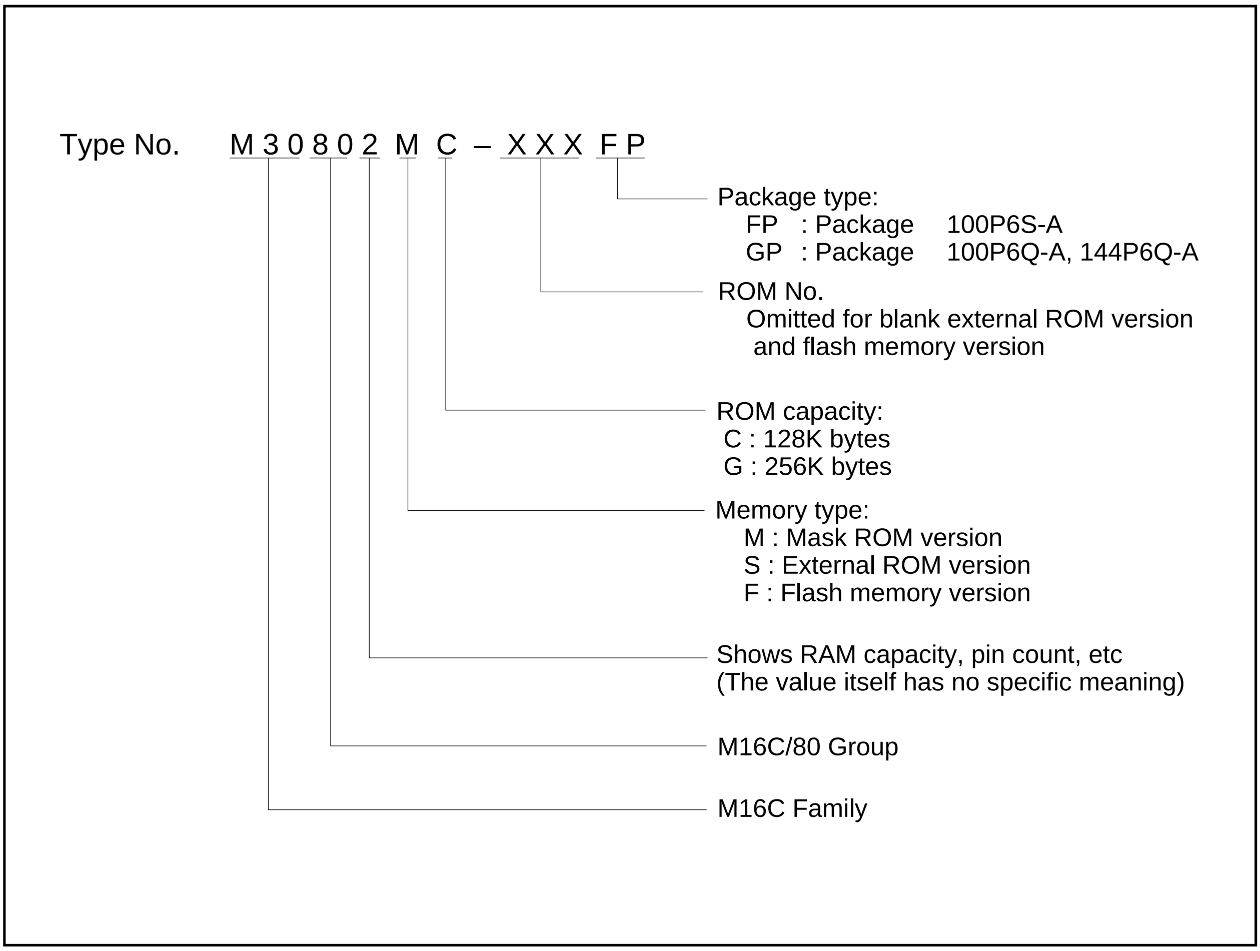

Type No. M 3 0 8 0 2 M C – X X X F P
Package type:
FP : Package 100P6S-A
GP : Package 100P6Q-A, 144P6Q-A
ROM No.
Omitted for blank external ROM version and flash memory version
ROM capacity:
C : 128K bytes
G : 256K bytes
Memory type:
M : Mask ROM version
S : External ROM version
F : Flash memory version
Shows RAM capacity, pin count, etc
(The value itself has no specific meaning)
M16C/80 Group
M16C Family

Figure 1.1.6. Type No., memory size, and package

Pin Description (1)

Pin name	Signal name	I/O type	Function
V_{CC}, V_{SS}	Power supply input		Supply 4.2 (2.7) to 5.5 V to the V_{CC} pin. Supply 0 V to the V_{SS} pin.
CNV_{SS}	CNV_{SS}	I	This pin switches between processor modes. Connect it to the V_{SS} when operating in single-chip or memory expansion mode after reset. Connect it to the V_{CC} when in microprocessor mode after reset.
$\overline{RESET}$	Reset input	I	An “L” on this input resets the microcomputer.
X_{IN} X_{OUT}	Clock input Clock output	I O	These pins are provided for the main clock generating circuit. Connect a ceramic resonator or crystal between the X_{IN} and the X_{OUT} pins. To use an externally derived clock, input it to the X_{IN} pin and leave the X_{OUT} pin open.
BYTE	External data bus width select input	I	This pin selects the width of an data bus in the external area 3. A 16--bit width is selected when this input is “L”; an 8-bit width is selected when this input is “H”. This input must be fixed to either “H” or “L”. When not using the external bus, connect this pin to V_{SS}.
AV_{CC}	Analog power supply input		This pin is a power supply input for the A-D converter. Connect this pin to V_{CC}.
AV_{SS}	Analog power supply input		This pin is a power supply input for the A-D converter. Connect this pin to V_{SS}.
V_{REF}	Reference voltage input	I	This pin is a reference voltage input for the A-D converter.
$P0_0$ to $P0_7$	I/O port P0	I/O	This is an 8-bit CMOS I/O port. It has an input/output port direction register that allows the user to set each pin for input or output individually. When set for input in single chip mode, the user can specify in units of four bits via software whether or not they are tied to a pull-up resistance. In memory expansion and microprocessor mode, an built-in pull-up resistance cannot be used. However, it is possible to select pull-up resistance presence to the usable port as I/O port by setting.
D_0 to D_7		I/O	When set as a separate bus, these pins input and output data (D_0–D_7).
$P1_0$ to $P1_7$	I/O port P1	I/O	This is an 8-bit I/O port equivalent to P0. $P1_5$ to $P1_7$ also function as external interrupt pins as selected by software.
D_8 to D_{15}		I/O	When set as a separate bus, these pins input and output data (D_8–D_{15}).
$P2_0$ to $P2_7$	I/O port P2	I/O	This is an 8-bit I/O port equivalent to P0.
A_0 to A_7		O	These pins output 8 low-order address bits (A_0–A_7).
A_0/D_0 to A_7/D_7		I/O	If a multiplexed bus is set, these pins input and output data (D_0–D_7) and output 8 low-order address bits (A_0–A_7) separated in time by multiplexing.
$P3_0$ to $P3_7$	I/O port P3	I/O	This is an 8-bit I/O port equivalent to P0.
A_8 to A_{15}		O	These pins output 8 middle-order address bits (A_8–A_{15}).
A_8/D_8 to A_{15}/D_{15}		I/O	If the external bus is set as a 16-bit wide multiplexed bus, these pins input and output data (D_8–D_{15}) and output 8 middle-order address bits (A_8–A_{15}) separated in time by multiplexing.
MA0 to MA7		O	If accessing to DRAM area, these pins output row address and column address separated in time by multiplexing.

Pin Description (2)

Pin name	Signal name	I/O type	Function
$P4_0$ to $P4_7$	I/O port P4	I/O	This is an 8-bit I/O port equivalent to P0.
A_{16} to A_{22}, $\overline{A_{23}}$		O	These pins output 8 high-order address bits (A_{16}–A_{22}, $\overline{A_{23}}$). Highest address bit ($\overline{A_{23}}$) outputs inversely.
$\overline{CS_0}$ to $\overline{CS_3}$		O	These pins output $\overline{CS_0}$–$\overline{CS_3}$ signals. $\overline{CS_0}$–$\overline{CS_3}$ are chip select signals used to specify an access space.
MA8 to MA12		O	If accessing to DRAM area, these pins output row address and column address separated in time by multiplexing.
$P5_0$ to $P5_7$	I/O port P5	I/O	This is an 8-bit I/O port equivalent to P0. $P5_3$ in this port outputs a divide-by-8 or divide-by-32 clock of X_{IN} or a clock of the same frequency as X_{CIN} as selected by software.
$\overline{\text{WRL}}$ / $\overline{\text{WR}}$, $\overline{\text{WRH}}$ / $\overline{\text{BHE}}$, $\overline{\text{RD}}$, BCLK, $\overline{\text{HLDA}}$, $\overline{\text{HOLD}}$, ALE, $\overline{\text{RDY}}$		O O O O O I O I	Output $\overline{\text{WRL}}$, $\overline{\text{WRH}}$ ($\overline{\text{WR}}$ and $\overline{\text{BHE}}$), $\overline{\text{RD}}$, BCLK, $\overline{\text{HLDA}}$, and ALE signals. $\overline{\text{WRL}}$ and $\overline{\text{WRH}}$, and $\overline{\text{BHE}}$ and $\overline{\text{WR}}$ can be switched using software control. ■ $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, and $\overline{\text{RD}}$ selected With a 16-bit external data bus, data is written to even addresses when the $\overline{\text{WRL}}$ signal is “L” and to the odd addresses when the $\overline{\text{WRH}}$ signal is “L”. Data is read when $\overline{\text{RD}}$ is “L”. ■ $\overline{\text{WR}}$, $\overline{\text{BHE}}$, and $\overline{\text{RD}}$ selected Data is written when $\overline{\text{WR}}$ is “L”. Data is read when $\overline{\text{RD}}$ is “L”. Odd addresses are accessed when $\overline{\text{BHE}}$ is “L”. Use this mode when using an 8-bit external data bus. While the input level at the $\overline{\text{HOLD}}$ pin is “L”, the microcomputer is placed in the hold state. While in the hold state, $\overline{\text{HLDA}}$ outputs an “L” level. ALE is used to latch the address. While the input level of the $\overline{\text{RDY}}$ pin is “L”, the bus of microcomputer is in the wait state.
$\overline{\text{DW}}$, $\overline{\text{CASL}}$, $\overline{\text{CASH}}$, $\overline{\text{RAS}}$		O O O O	When accessing to DRAM area while $\overline{\text{DW}}$ signal is “L”, write to DRAM. $\overline{\text{CASL}}$ and $\overline{\text{CASH}}$ show timing when latching to line address. When $\overline{\text{CASL}}$ accesses to even address, and $\overline{\text{CASH}}$ to odd, these two pins become “L”. $\overline{\text{RAS}}$ signal shows timing when latching to row address.
$P6_0$ to $P6_7$	I/O port P6	I/O	This is an 8-bit I/O port equivalent to P0. When set for input in single chip mode, microprocessor mode and memory expansion mode the user can specify in units of four bits via software whether or not they are tied to a pull-up resistance. Pins in this port also function as UART0 and UART1 I/O pins as selected by software.
$P7_0$ to $P7_7$	I/O port P7	I/O	This is an 8-bit I/O port equivalent to P6 ($P7_0$ and $P7_1$ are N-channel open drain output). Pins in this port also function as timer A_0–A_3, timer B5 or UART2 I/O pins as selected by software.
$P8_0$ to $P8_4$, $P8_6$, $P8_7$, $P8_5$	I/O port P8 I/O port $P8_5$	I/O I/O I/O I	$P8_0$ to $P8_4$, $P8_6$, and $P8_7$ are I/O ports with the same functions as P6. Using software, they can be made to function as the I/O pins for timer A4 and the input pins for external interrupts. $P8_6$ and $P8_7$ can be set using software to function as the I/O pins for a sub clock generation circuit. In this case, connect a quartz oscillator between $P8_6$ (X_{COUT} pin) and $P8_7$ (X_{CIN} pin). $P8_5$ is an input-only port that also functions for $\overline{\text{NMI}}$. The $\overline{\text{NMI}}$ interrupt is generated when the input at this pin changes from “H” to “L”. The $\overline{\text{NMI}}$ function cannot be canceled using software. The pull-up cannot be set for this pin.
$P9_0$ to $P9_7$	I/O port P9	I/O	This is an 8-bit I/O port equivalent to P6. Pins in this port also function as UART3 and UART4 I/O pins, Timer B0–B4 input pins, D-A converter output pins, A-D converter extended input pins, or A-D trigger input pins as selected by software.
$P10_0$ to $P10_7$	I/O port P10	I/O	This is an 8-bit I/O port equivalent to P6. Pins in this port also function as A-D converter input pins. Furthermore, $P10_4$–$P10_7$ also function as input pins for the key input interrupt function.

Pin Description (3)

Pin name	Signal name	I/O type	Function
$P11_0$ to $P11_4$ (Note)	I/O port P11	II/O	This is an 5-bit I/O port equivalent to P6.
$P12_0$ to $P12_7$ (Note)	I/O port P12	II/O	This is an 8-bit I/O port equivalent to P6.
$P13_0$ to $P13_7$ (Note)	I/O port P13	II/O	This is an 8-bit I/O port equivalent to P6.
$P14_0$ to $P14_6$ (Note)	I/O port P14	II/O	This is an 7-bit I/O port equivalent to P6.
$P15_0$ to $P15_7$ (Note)	I/O port P15	II/O	This is an 8-bit I/O port equivalent to P6.

Note : Port P11 to P15 exist in 144-pin version.

Operation of Functional Blocks

The M16C/80 group accommodates certain units in a single chip. These units include ROM and RAM to store instructions and data and the central processing unit (CPU) to execute arithmetic/logic operations. Also included are peripheral units such as timers, serial I/O, D-A converter, DMAC, CRC calculation circuit, A-D converter, DRAM controller and I/O ports.

The following explains each unit.

Memory

Figure 1.2.1 is a memory map of the M16C/80 group. The address space extends the 16 Mbytes from address 000000_{16} to $FFFFFF_{16}$. From $FFFFFF_{16}$ down is ROM. For example, in the M30800MC-XXXFP, there is 128K bytes of internal ROM from $FE0000_{16}$ to $FFFFFF_{16}$. The vector table for fixed interrupts such as the reset and $\overline{\text{NMI}}$ are mapped to $FFFFDC_{16}$ to $FFFFFF_{16}$. The starting address of the interrupt routine is stored here. The address of the vector table for timer interrupts, etc., can be set as desired using the internal register (INTB). See the section on interrupts for details.

From 000400_{16} up is RAM. For example, in the M30800MC-XXXFP, 10 Kbytes of internal RAM is mapped to the space from 000400_{16} to $002BFF_{16}$. In addition to storing data, the RAM also stores the stack used when calling subroutines and when interrupts are generated.

The SFR area is mapped to 000000_{16} to $0003FF_{16}$. This area accommodates the control registers for peripheral devices such as I/O ports, A-D converter, serial I/O, and timers, etc. Figure 1.5.1 to 1.5.4 are location of peripheral unit control registers. Any part of the SFR area that is not occupied is reserved and cannot be used for other purposes.

The special page vector table is mapped to $FFFE00_{16}$ to $FFFFDB_{16}$. If the starting addresses of subroutines or the destination addresses of jumps are stored here, subroutine call instructions and jump instructions can be used as 2-byte instructions, reducing the number of program steps.

In memory expansion mode and microprocessor mode, a part of the spaces are reserved and cannot be used. For example, in the M30800MC-XXXFP, the following spaces cannot be used.

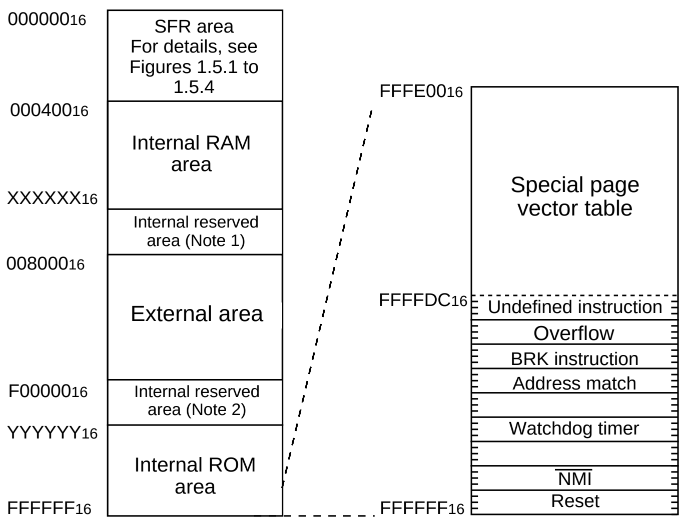
- The space between $002C00_{16}$ and 008000_{16} (Memory expansion and microprocessor modes)
- The space between $F00000_{16}$ and $FDFFFF_{16}$ (Memory expansion mode)

<100-pin version>

Type No.	Address $XXXXX_{16}$	Address $YYYYY_{16}$
M30800MC/FC	$002BFF_{16}$	$FE0000_{16}$
M30803MG/FG	$0053FF_{16}$	$FC0000_{16}$

<144-pin version>

Type No.	Address $XXXXX_{16}$	Address $YYYYY_{16}$
M30802MC/FC	$002BFF_{16}$	$FE0000_{16}$
M30805MG/FG	$0053FF_{16}$	$FC0000_{16}$
M30802S	$002BFF_{16}$	———
M30805S	$0063FF_{16}$	———



Note 1: During memory expansion and microprocessor modes, can not be used.
Note 2: In memory expansion mode, can not be used.

Figure 1.2.1. Memory map

Central Processing Unit (CPU)

The CPU has a total of 28 registers shown in Figure 1.3.1. Eight of these registers (R0, R1, R2, R3, A0, A1, SB and FB) come in two sets; therefore, these have two register banks.

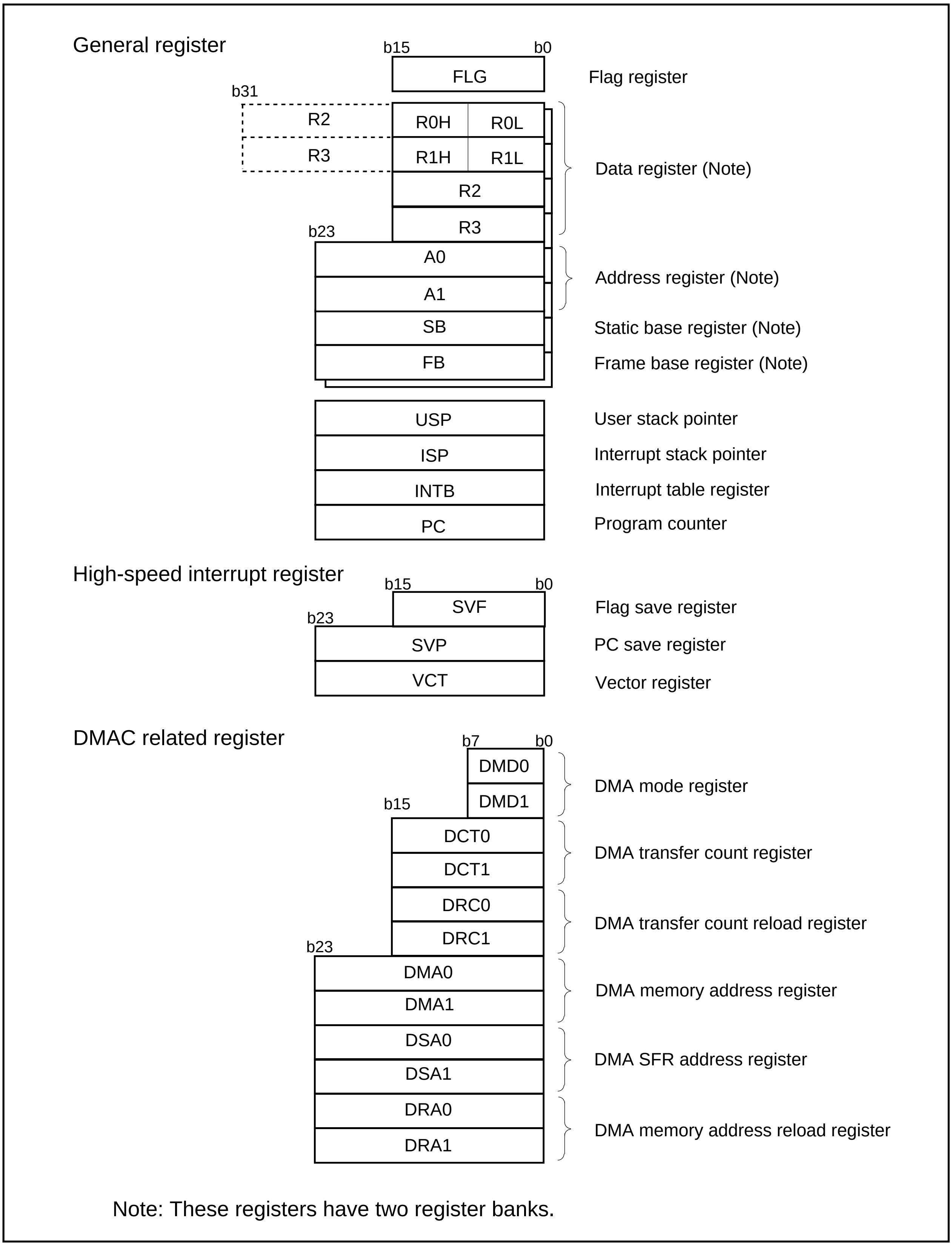


Figure 1.3.1. Central processing unit register

(1) Data registers (R0, R0H, R0L, R1, R1H, R1L, R2, R3, R2R0 and R3R1)

Data registers (R0, R1, R2, and R3) are configured with 16 bits, and are used primarily for transfer and arithmetic/logic operations.

Registers R0 and R1 each can be used as separate 8-bit data registers, high-order bits as (R0H/R1H), and low-order bits as (R0L/R1L). Registers R2 and R0, as well as R3 and R1 can use as 32-bit data registers (R2R0/R3R1).

(2) Address registers (A0 and A1)

Address registers (A0 and A1) are configured with 24 bits, and have functions equivalent to those of data registers. These registers can also be used for address register indirect addressing and address register relative addressing.

(3) Static base register (SB)

Static base register (SB) is configured with 24 bits, and is used for SB relative addressing.

(4) Frame base register (FB)

Frame base register (FB) is configured with 24 bits, and is used for FB relative addressing.

(5) Program counter (PC)

Program counter (PC) is configured with 24 bits, indicating the address of an instruction to be executed.

(6) Interrupt table register (INTB)

Interrupt table register (INTB) is configured with 24 bits, indicating the start address of an interrupt vector table.

(7) User stack pointer (USP), interrupt stack pointer (ISP)

Stack pointer comes in two types: user stack pointer (USP) and interrupt stack pointer (ISP), each configured with 24 bits.

Your desired type of stack pointer (USP or ISP) can be selected by a stack pointer select flag (U flag).

This flag is located at the position of bit 7 in the flag register (FLG).

Set USP and ISP to an even number so that execution efficiency is increased.

(8) Save flag register (SVF)

This register consists of 16 bits and is used to save the flag register when a high-speed interrupt is generated.

(9) Save PC register (SVP)

This register consists of 24 bits and is used to save the program counter when a high-speed interrupt is generated.

(10) Vector register (VCT)

This register consists of 24 bits and is used to indicate the jump address when a high-speed interrupt is generated.

(11) DMA mode registers (DMD0/DMD1)

These registers consist of 8 bits and are used to set the transfer mode, etc. for DMA.

(12) DMA transfer count registers (DCT0/DCT1)

These registers consist of 16 bits and are used to set the number of DMA transfers performed.

(13) DMA transfer count reload registers (DRC0/DRC1)

These registers consist of 16 bits and are used to reload the DMA transfer count registers.

(14) DMA memory address registers (DMA0/DMA1)

These registers consist of 24 bits and are used to set a memory address at the source or destination of DMA transfer.

(15) DMA SFR address registers (DSA0/DSA1)

These registers consist of 24 bits and are used to set a fixed address at the source or destination of DMA transfer.

(16) DMA memory address reload registers (DRA0/DRA1)

These registers consist of 24 bits and are used to reload the DMA memory address registers.

(17) Flag register (FLG)

Flag register (FLG) is configured with 11 bits, each bit is used as a flag. Figure 1.3.2 shows the flag register (FLG). The following explains the function of each flag:

- **Bit 0: Carry flag (C flag)**
  This flag retains a carry, borrow, or shift-out bit that has occurred in the arithmetic/logic unit.

- **Bit 1: Debug flag (D flag)**
  This flag enables a single-step interrupt.
  When this flag is “1”, a single-step interrupt is generated after instruction execution. This flag is cleared to “0” when the interrupt is acknowledged.

- **Bit 2: Zero flag (Z flag)**
  This flag is set to “1” when an arithmetic operation resulted in 0; otherwise, cleared to “0”.

- **Bit 3: Sign flag (S flag)**
  This flag is set to “1” when an arithmetic operation resulted in a negative value; otherwise, cleared to “0”.

- **Bit 4: Register bank select flag (B flag)**
  This flag chooses a register bank. Register bank 0 is selected when this flag is “0” ; register bank 1 is selected when this flag is “1”.

- **Bit 5: Overflow flag (O flag)**
  This flag is set to “1” when an arithmetic operation resulted in overflow; otherwise, cleared to “0”.

- **Bit 6: Interrupt enable flag (I flag)**
  This flag enables a maskable interrupt.
  An interrupt is disabled when this flag is “0”, and is enabled when this flag is “1”. This flag is cleared to “0” when the interrupt is acknowledged.

- **Bit 7: Stack pointer select flag (U flag)**
  Interrupt stack pointer (ISP) is selected when this flag is “0” ; user stack pointer (USP) is selected when this flag is “1”.
  This flag is cleared to “0” when a hardware interrupt is acknowledged or an INT instruction of software interrupt Nos. 0 to 31 is executed.

- **Bits 8 to 11: Reserved area**

• Bits 12 to 14: Processor interrupt priority level (IPL)

Processor interrupt priority level (IPL) is configured with three bits, for specification of up to eight processor interrupt priority levels from level 0 to level 7.

If a requested interrupt has priority greater than the processor interrupt priority level (IPL), the interrupt is enabled.

• Bit 15: Reserved area

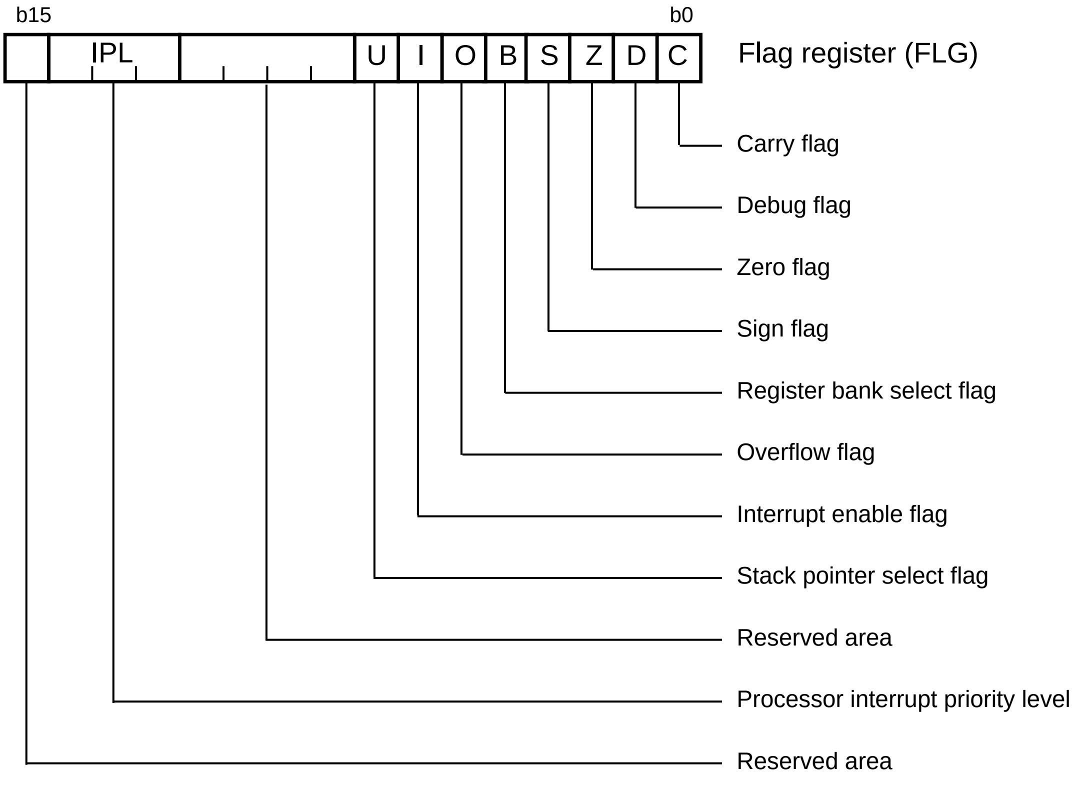


Figure 1.3.2. Flag register (FLG)

Reset

There are two kinds of resets; hardware and software. In both cases, operation is the same after the reset. (See “Software Reset” for details of software resets.) This section explains on hardware resets.

When the supply voltage is in the range where operation is guaranteed, a reset is effected by holding the reset pin level “L” (0.2VCC max.) for at least 20 cycles. When the reset pin level is then returned to the “H” level while main clock is stable, the reset status is cancelled and program execution resumes from the address in the reset vector table.

Figure 1.4.1 shows the example reset circuit. Figure 1.4.2 shows the reset sequence.

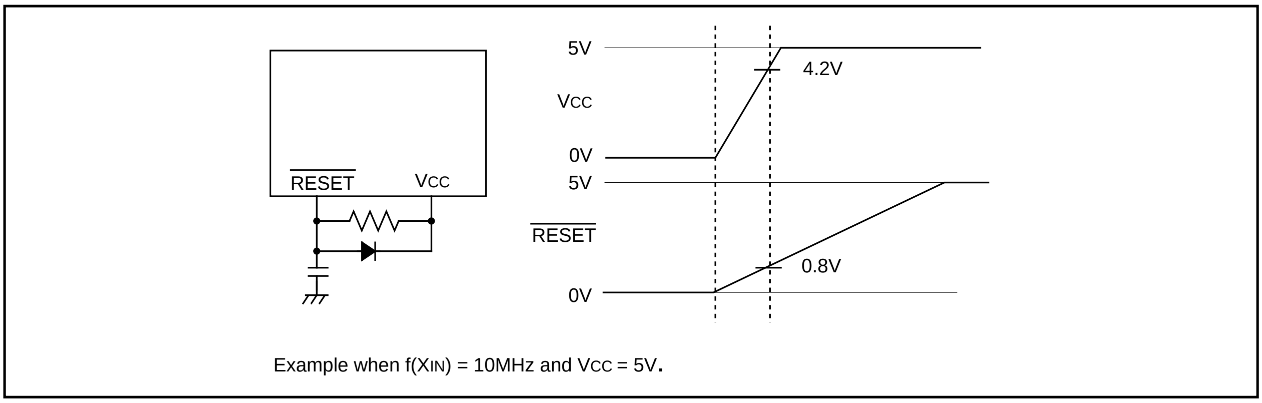


Figure 1.4.1. Example reset circuit

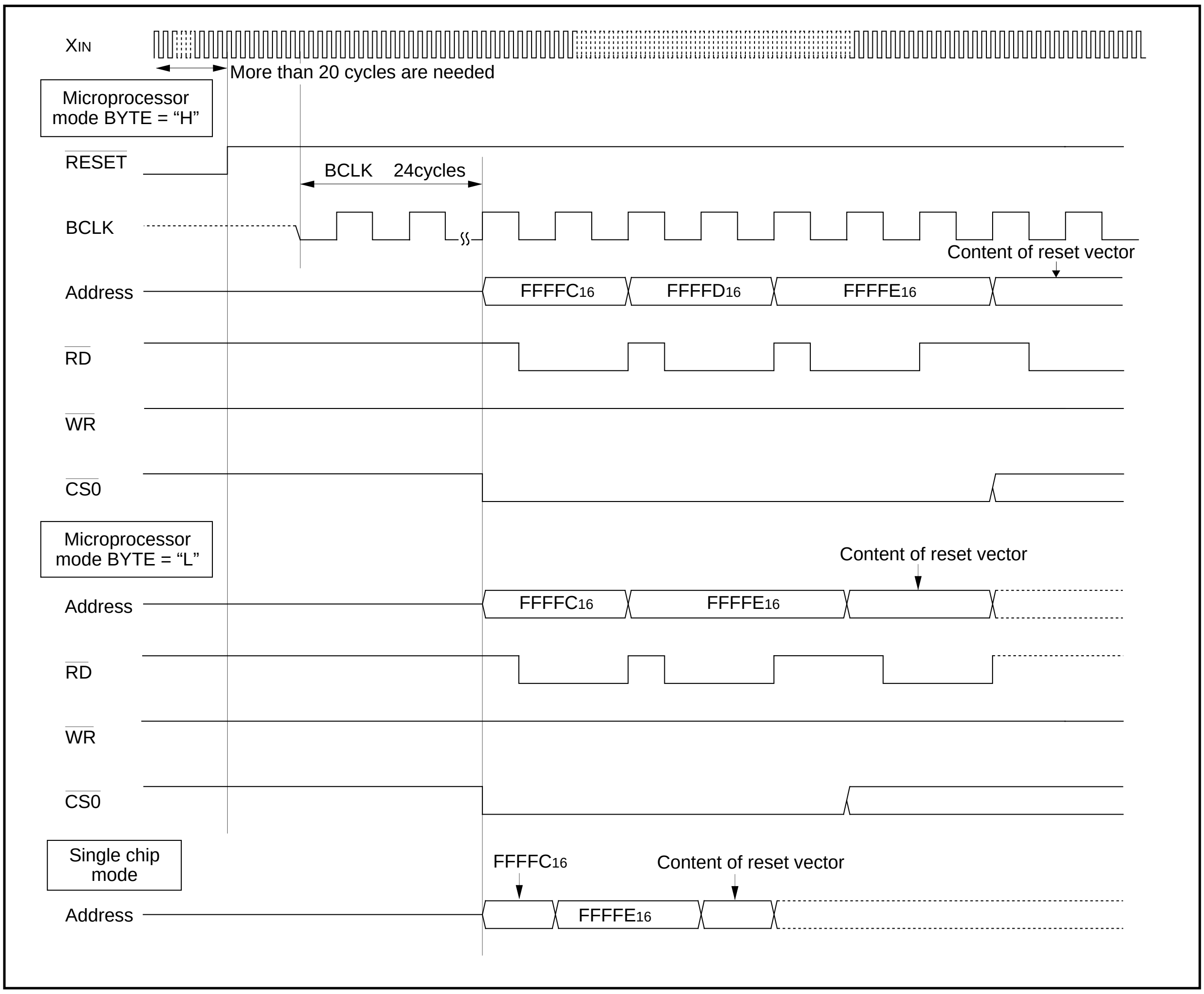


Figure 1.4.2. Reset sequence

Table 1.4.1 shows the statuses of the other pins while the $\overline{\text{RESET}}$ pin level is “L”. Figures 1.4.3 and 1.4.4 show the internal status of the microcomputer immediately after the reset is cancelled.

Table 1.4.1. Pin status when $\overline{\text{RESET}}$ pin level is “L”

Pin name	Status		
	CNVss = Vss	CNVss = Vcc	
		BYTE = Vss	BYTE = Vcc
P0	Input port (floating)	Data input (floating)	Data input (floating)
P1	Input port (floating)	Data input (floating)	Input port (floating)
P2, P3, P4	Input port (floating)	Address output (undefined)	Address output (undefined)
P5$_0$	Input port (floating)	$\overline{\text{WR}}$ output (“H” level is output)	$\overline{\text{WR}}$ output (“H” level is output)
P5$_1$	Input port (floating)	$\overline{\text{BHE}}$ output (undefined)	$\overline{\text{BHE}}$ output (undefined)
P5$_2$	Input port (floating)	$\overline{\text{RD}}$ output (“H” level is output)	$\overline{\text{RD}}$ output (“H” level is output)
P5$_3$	Input port (floating)	BCLK output	BCLK output
P5$_4$	Input port (floating)	$\overline{\text{HLDA}}$ output (The output value depends on the input to the $\overline{\text{HOLD}}$ pin)	$\overline{\text{HLDA}}$ output (The output value depends on the input to the $\overline{\text{HOLD}}$ pin)
P5$_5$	Input port (floating)	$\overline{\text{HOLD}}$ input (floating)	$\overline{\text{HOLD}}$ input (floating)
P5$_6$	Input port (floating)	$\overline{\text{RAS}}$ output	$\overline{\text{RAS}}$ output
P5$_7$	Input port (floating)	$\overline{\text{RDY}}$ input (floating)	$\overline{\text{RDY}}$ input (floating)
P6, P7, P8$_0$ to P8$_4$, P8$_6$, P8$_7$, P9, P10,	Input port (floating)	Input port (floating)	Input port (floating)
P11, P12, P13, P14, P15 (Note)	Input port (floating)	Input port (floating)	Input port (floating)

Note :Port P11 to P15 exist in 144-pin vrsion.

(1)	Processor mode register 0 (Note1)	(0004_{16})···	80_{16}
(2)	Processor mode register 1	(0005_{16})···	00_{16}
(3)	System clock control register 0	(0006_{16})···	08_{16}
(4)	System clock control register 1	(0007_{16})···	20_{16}
(5)	Wait control register	(0008_{16})···	FF_{16}
(6)	Address match interrupt enable register	(0009_{16})···	☒ ☒ ☒ ☒ 0 0 0 0
(7)	Protect register	($000A_{16}$)···	☒ ☒ ☒ ☒ ☒ 0 0 0
(8)	External data bus width control register (Note 2)	($000B_{16}$)···	☒ ☒ ☒ ☒ ? 0 0 0
(9)	Main clock divided register	($000C_{16}$)···	☒ ☒ ☒ 0 1 0 0 0
(10)	Watchdog timer control register	($000F_{16}$)···	0 0 0 ? ? ? ? ?
(11)	Address match interrupt register 0	(0010_{16})···	00_{16}
		(0011_{16})···	00_{16}
		(0012_{16})···	00_{16}
(12)	Address match interrupt register 1	(0014_{16})···	00_{16}
		(0015_{16})···	00_{16}
		(0016_{16})···	00_{16}
(13)	Address match interrupt register 2	(0018_{16})···	00_{16}
		(0019_{16})···	00_{16}
		($001A_{16}$)···	00_{16}
(14)	Address match interrupt register 3	($001C_{16}$)···	00_{16}
		($001D_{16}$)···	00_{16}
		($001E_{16}$)···	00_{16}
(15)	DMAM control register	(0040_{16})···	? ☒ ☒ ☒ ? ? ? ?
(16)	DMA0 interrupt control register	(0068_{16})···	☒ ☒ ☒ ☒ ? 0 0 0
(17)	Timer B5 interrupt control register	(0069_{16})···	☒ ☒ ☒ ☒ ? 0 0 0
(18)	DMA2 interrupt control register	($006A_{16}$)···	☒ ☒ ☒ ☒ ? 0 0 0
(19)	UART2 receive/ACK interrupt control register	($006B_{16}$)···	☒ ☒ ☒ ☒ ? 0 0 0
(20)	Timer A0 interrupt control register	($006C_{16}$)···	☒ ☒ ☒ ☒ ? 0 0 0
(21)	UART3 receive/ACK interrupt control register	($006D_{16}$)···	☒ ☒ ☒ ☒ ? 0 0 0
(22)	Timer A2 interrupt control register	($006E_{16}$)···	☒ ☒ ☒ ☒ ? 0 0 0
(23)	UART4 receive/ACK interrupt control register	($006F_{16}$)···	☒ ☒ ☒ ☒ ? 0 0 0
(24)	Timer A4 interrupt control register	(0070_{16})···	☒ ☒ ☒ ☒ ? 0 0 0
(25)	Bus collision detection(UART3) interrupt control register	(0071_{16})···	☒ ☒ ☒ ☒ ? 0 0 0
(26)	UART0 receive interrupt control register	(0072_{16})···	☒ ☒ ☒ ☒ ? 0 0 0
(27)	A-D conversion interrupt control register	(0073_{16})···	☒ ☒ ☒ ☒ ? 0 0 0
(28)	UART1 receive interrupt control register	(0074_{16})···	☒ ☒ ☒ ☒ ? 0 0 0
(29)	Timer B1 interrupt control register	(0076_{16})···	☒ ☒ ☒ ☒ ? 0 0 0
(30)	Timer B3 interrupt control register	(0078_{16})···	☒ ☒ ☒ ☒ ? 0 0 0
(31)	INT5 interrupt control register	($007A_{16}$)···	☒ ☒ 0 0 ? 0 0 0
(32)	INT3 interrupt control register	($007C_{16}$)···	☒ ☒ 0 0 ? 0 0 0
(33)	INT1 interrupt control register	($007E_{16}$)···	☒ ☒ 0 0 ? 0 0 0
(34)	DMA1 interrupt control register	(0088_{16})···	☒ ☒ ☒ ☒ ? 0 0 0
(35)	UART2 transmit/NACK interrupt control register	(0089_{16})···	☒ ☒ ☒ ☒ ? 0 0 0
(36)	DMA3 interrupt control register	($008A_{16}$)···	☒ ☒ ☒ ☒ ? 0 0 0
(37)	UART3 transmit/NACK interrupt control register	($008B_{16}$)···	☒ ☒ ☒ ☒ ? 0 0 0
(38)	Timer A1 interrupt control register	($008C_{16}$)···	☒ ☒ ☒ ☒ ? 0 0 0
(39)	UART4 receive/NACK interrupt control register	($008D_{16}$)···	☒ ☒ ☒ ☒ ? 0 0 0
(40)	Timer A3 interrupt control register	($008E_{16}$)···	☒ ☒ ☒ ☒ ? 0 0 0
(41)	Bus collision detection(UART2) interrupt control register	($008F_{16}$)···	☒ ☒ ☒ ☒ ? 0 0 0
(42)	UART0 transmit interrupt control register	(0090_{16})···	☒ ☒ ☒ ☒ ? 0 0 0
(43)	Bus collision detection(UART4) interrupt control register	(0091_{16})···	☒ ☒ ☒ ☒ ? 0 0 0
(44)	UART1 transmit interrupt control register	(0092_{16})···	☒ ☒ ☒ ☒ ? 0 0 0
(45)	Key input interrupt control register	(0093_{16})···	☒ ☒ ☒ ☒ ? 0 0 0
(46)	Timer B0 interrupt control register	(0094_{16})···	☒ ☒ ☒ ☒ ? 0 0 0
(47)	Timer B2 interrupt control register	(0096_{16})···	☒ ☒ ☒ ☒ ? 0 0 0
(48)	Timer B4 interrupt control register	(0098_{16})···	☒ ☒ ☒ ☒ ? 0 0 0
(49)	INT4 interrupt control register	($009A_{16}$)···	☒ ☒ 0 0 ? 0 0 0
(50)	INT2 interrupt control register	($009C_{16}$)···	☒ ☒ 0 0 ? 0 0 0
(51)	INT0 interrupt control register	($009E_{16}$)···	☒ ☒ 0 0 ? 0 0 0
(52)	Exit priority register	($009F_{16}$)···	☒ ☒ ☒ ☒ 0 0 0 0
(53)	XY control register	($02E0_{16}$)···	☒ ☒ ☒ ☒ ☒ ☒ 0 0
(54)	UART4 special mode register 3	($02F5_{16}$)···	00_{16}
(55)	UART4 special mode register 2	($02F6_{16}$)···	00_{16}
(56)	UART4 special mode register	($02F7_{16}$)···	00_{16}
(57)	UART4 transmit/receive mode register	($02F8_{16}$)···	00_{16}
(58)	UART4 transmit/receive control register 0	($02FC_{16}$)···	08_{16}
(59)	UART4 transmit/receive control register 1	($02FD_{16}$)···	02_{16}
(60)	Timer B3,4,5 count start flag	(0300_{16})···	0 0 0 ☒ ☒ ☒ ☒ ☒
(61)	Three-phase PWM control register 0	(0308_{16})···	00_{16}
(62)	Three-phase PWM control register 1	(0309_{16})···	0 0 0 0 ? 0 0 0
(63)	Three-phase output buffer register 0	($030A_{16}$)···	00_{16}
(64)	Three-phase output buffer register 1	($030B_{16}$)···	00_{16}
(65)	Timer B3 mode register	($031B_{16}$)···	0 0 ? ☒ 0 0 0 0
(66)	Timer B4 mode register	($031C_{16}$)···	0 0 ? ☒ 0 0 0 0
(67)	Timer B5 mode register	($031D_{16}$)···	0 0 ? ☒ 0 0 0 0

x : Nothing is mapped to this bit
? : Undefined

The content of other registers and RAM is undefined when the microcomputer is reset. The initial values must therefore be set.

Note 1: When the V_{CC} level is applied to the CNV_{SS} pin, it is 03_{16} at a reset.
Note 2: When the BYTE pin is "L", the third bit is "1". When the BYTE pin is "H", the third bit is "0".

Figure 1.4.3. Device's internal status after a reset is cleared

(68)	Interrupt cause select register	($031F_{16}$)···	x x 0 0 0 0 0 0
(69)	UART3 special mode register 3	(0325_{16})···	00_{16}
(70)	UART3 special mode register 2	(0326_{16})···	00_{16}
(71)	UART3 special mode register	(0327_{16})···	00_{16}
(72)	UART3 transmit/receive mode register	(0328_{16})···	00_{16}
(73)	UART3 transmit/receive control register 0	($032C_{16}$)···	08_{16}
(74)	UART3 transmit/receive control register 1	($032D_{16}$)···	02_{16}
(75)	UART2 special mode register 3	(0335_{16})···	0 0 0 x x x x x
(76)	UART2 special mode register 2	(0336_{16})···	00_{16}
(77)	UART2 special mode register	(0337_{16})···	00_{16}
(78)	UART2 transmit/receive mode register	(0338_{16})···	00_{16}
(79)	UART2 transmit/receive control register 0	($033C_{16}$)···	0 0 x 0 1 0 0 0
(80)	UART2 transmit/receive control register 1	($033D_{16}$)···	02_{16}
(81)	Count start flag	(0340_{16})···	00_{16}
(82)	Clock prescaler reset flag	(0341_{16})···	0 x x x x x x x
(83)	One-shot start flag	(0342_{16})···	00_{16}
(84)	Trigger select flag	(0343_{16})···	00_{16}
(85)	Up-down flag	(0344_{16})···	00_{16}
(86)	Timer A0 mode register	(0356_{16})···	0 0 0 0 0 ? 0 0
(87)	Timer A1 mode register	(0357_{16})···	0 0 0 0 0 ? 0 0
(88)	Timer A2 mode register	(0358_{16})···	0 0 0 0 0 ? 0 0
(89)	Timer A3 mode register	(0359_{16})···	0 0 0 0 0 ? 0 0
(90)	Timer A4 mode register	($035A_{16}$)···	0 0 0 0 0 ? 0 0
(91)	Timer B0 mode register	($035B_{16}$)···	0 0 ? x 0 0 0 0
(92)	Timer B1 mode register	($035C_{16}$)···	0 0 ? x 0 0 0 0
(93)	Timer B2 mode register	($035D_{16}$)···	0 0 ? x 0 0 0 0
(94)	UART0 transmit/receive mode register	(0360_{16})···	00_{16}
(95)	UART0 transmit/receive control register 0	(0364_{16})···	08_{16}
(96)	UART0 transmit/receive control register 1	(0365_{16})···	02_{16}
(97)	UART1 transmit/receive mode register	(0368_{16})···	00_{16}
(98)	UART1 transmit/receive control register 0	($036C_{16}$)···	08_{16}
(99)	UART1 transmit/receive control register 1	($036D_{16}$)···	02_{16}
(100)	UART transmit/receive control register 2	(0370_{16})···	x 0 0 0 0 0 0 0
(101)	Flash memory control register 1 (Note 1)	(0376_{16})···	? ? ? ? ? 0 ? ?
(102)	Flash memory control register 0 (Note 1)	(0377_{16})···	x x 0 0 0 0 0 1
(103)	DMA0 cause select register	(0378_{16})···	0 x 0 0 0 0 0 0
(104)	DMA1 cause select register	(0379_{16})···	0 x 0 0 0 0 0 0
(105)	DMA2 cause select register	($037A_{16}$)···	0 x 0 0 0 0 0 0
(106)	DMA3 cause select register	($037B_{16}$)···	0 x 0 0 0 0 0 0
(107)	A-D control register 2	(0394_{16})···	0 0 0 0 x x x 0
(108)	A-D control register 0	(0396_{16})···	0 0 0 0 0 ? ? ?
(109)	A-D control register 1	(0397_{16})···	00_{16}
(110)	D-A control register	($039C_{16}$)···	00_{16}
(111)	Function select register C	($03AF_{16}$)···	0 x x x x x x 0

(112)	Function select register A0	($03B0_{16}$)···	0 x 0 0 0 x 0 0
(113)	Function select register A1	($03B1_{16}$)···	x 0 0 0 0 0 0 0
(114)	Function select register B0	($03B2_{16}$)···	x x x 0 x x x x
(115)	Function select register B1	($03B3_{16}$)···	x x x 0 0 0 x 0
(116)	Function select register A2	($03B4_{16}$)···	x x x x x x 0 0
(117)	Function select register A3	($03B5_{16}$)···	00_{16}
(118)	Function select register B2	($03B6_{16}$)···	x x x x x x x 0
(119)	Function select register B3	($03B7_{16}$)···	0 0 0 0 0 x 0 x
(120)	Port P6 direction register	($03C2_{16}$)···	00_{16}
(121)	Port P7 direction register	($03C3_{16}$)···	00_{16}
(122)	Port P8 direction register	($03C6_{16}$)···	0 0 x 0 0 0 0 0
(123)	Port P9 direction register	($03C7_{16}$)···	00_{16}
(124)	Port P10 direction register	($03CA_{16}$)···	00_{16}
(125)	Port P11 direction register (Note 2)	($03CB_{16}$)···	x x x 0 0 0 0 0
(126)	Port P12 direction register (Note 2)	($03CE_{16}$)···	00_{16}
(127)	Port P13 direction register (Note 2)	($03CF_{16}$)···	00_{16}
(128)	Port P14 direction register (Note 2)	($03D2_{16}$)···	x 0 0 0 0 0 0 0
(129)	Port P15 direction register (Note 2)	($03D3_{16}$)···	00_{16}
(130)	Pull-up control register 2	($03DA_{16}$)···	00_{16}
(131)	Pull-up control register 3 (Note 2)	($03DB_{16}$)···	00_{16}
(132)	Pull-up control register 4 (Note 2)	($03DC_{16}$)···	$X0_{16}$
(133)	Port P0 direction register	($03E2_{16}$)···	00_{16}
(132)	Port P1 direction register	($03E3_{16}$)···	00_{16}
(135)	Port P2 direction register	($03E6_{16}$)···	00_{16}
(136)	Port P3 direction register	($03E7_{16}$)···	00_{16}
(137)	Port P4 direction register	($03EA_{16}$)···	00_{16}
(138)	Port P5 direction register	($03EB_{16}$)···	00_{16}
(139)	Pull-up control register 0	($03F0_{16}$)···	00_{16}
(140)	Pull-up control register 1	($03F1_{16}$)···	$X0_{16}$
(141)	Port control register	($03FF_{16}$)···	x x x x x x x 0
(142)	Data registers (R0/R1/R2/R3)		0000_{16}
(143)	Address registers (A0/A1)		000000_{16}
(144)	Static base register (SB)		000000_{16}
(145)	Frame base register (FB)		000000_{16}
(146)	Interrupt table register (INTB)		000000_{16}
(147)	User stack pointer (USP)		000000_{16}
(148)	Interrupt stack pointer (ISP)		000000_{16}
(149)	Flag register (FLG)		0000_{16}
(150)	DMA mode register (DMD0/DMD1)		00_{16}
(151)	DMA transfer count register (DCT0/DCT1)		??
(152)	DMA transfer count reload register (DRC0/DRC1)		??
(153)	DMA memory address register (DMA0/DMA1)		??
(154)	DMA SFR address register (DSA0/DSA1)		??
(155)	DMA memory address reload register (DRA0/DRA1)		??

The content of other registers and RAM is undefined when the microcomputer is reset. The initial values must therefore be set.

x : Nothing is mapped to this bit
? : Undefined

Note 1:This register exists in the flash memory version.
Note 2:This register exists in 144-pin version.

Figure 1.4.4. Device's internal status after a reset is cleared

0000_{16}		0060_{16}	
0001_{16}		0061_{16}	
0002_{16}		0062_{16}	
0003_{16}		0063_{16}	
0004_{16}	Processor mode register 0 (PM0)	0064_{16}	
0005_{16}	Processor mode register 1(PM1)	0065_{16}	
0006_{16}	System clock control register 0 (CM0)	0066_{16}	
0007_{16}	System clock control register 1 (CM1)	0067_{16}	
0008_{16}	Wait control register (WCR)	0068_{16}	DMA0 interrupt control register (DM0IC)
0009_{16}	Address match interrupt enable register (AIER)	0069_{16}	Timer B5 interrupt control register (TB5IC)
$000A_{16}$	Protect register (PRCR)	$006A_{16}$	DMA2 interrupt control register (DM1IC)
$000B_{16}$	External data bus width control register (DS)	$006B_{16}$	UART2 receive/ACK interrupt control register (S2RIC)
$000C_{16}$	Main clock division register (MCD)	$006C_{16}$	Timer A0 interrupt control register (TA0IC)
$000D_{16}$		$006D_{16}$	UART3 receive/ACK interrupt control register (S3RIC)
$000E_{16}$	Watchdog timer start register (WDTS)	$006E_{16}$	Timer A2 interrupt control register (TA2IC)
$000F_{16}$	Watchdog timer control register (WDC)	$006F_{16}$	UART4 receive/ACK interrupt control register (S4RIC)
0010_{16}	Address match interrupt register 0 (RMAD0)	0070_{16}	Timer A4 interrupt control register (TA4IC)
0011_{16}		0071_{16}	Bus collision detection(UART3) interrupt control register (BCN3IC)
0012_{16}		0072_{16}	UART0 receive interrupt control register (S0RIC)
0013_{16}		0073_{16}	A-D conversion interrupt control register (ADIC)
0014_{16}	Address match interrupt register 1 (RMAD1)	0074_{16}	UART1 receive interrupt control register (S1RIC)
0015_{16}		0075_{16}	
0016_{16}		0076_{16}	Timer B1 interrupt control register (TB1IC)
0017_{16}		0077_{16}	
0018_{16}	Address match interrupt register 2 (RMAD2)	0078_{16}	Timer B3 interrupt control register (TB3IC)
0019_{16}		0079_{16}	
$001A_{16}$		$007A_{16}$	INT5 interrupt control register (INT5IC)
$001B_{16}$		$007B_{16}$	
$001C_{16}$	Address match interrupt register 3 (RMAD3)	$007C_{16}$	INT3 interrupt control register (INT3IC)
$001D_{16}$		$007D_{16}$	
$001E_{16}$		$007E_{16}$	INT1 interrupt control register (INT1IC)
$001F_{16}$		$007F_{16}$	
0020_{16}	Emulator interrupt vector table register (EIAD) *	0080_{16}	
0021_{16}		0081_{16}	
0022_{16}		0082_{16}	
0023_{16}	Emulator interrupt detect register (EITD) *	0083_{16}	
0024_{16}	Emulator protect register (EPRR) *	0084_{16}	
0025_{16}		0085_{16}	
0026_{16}		0086_{16}	
0027_{16}		0087_{16}	
0028_{16}		0088_{16}	DMA1 interrupt control register (DM1IC)
0029_{16}		0089_{16}	UART2 transmit/NACK interrupt control register (S2TIC)
$002A_{16}$		$008A_{16}$	DMA3 interrupt control register (DM3IC)
$002B_{16}$		$008B_{16}$	UART3 transmit/NACK interrupt control register (S3TIC)
$002C_{16}$		$008C_{16}$	Timer A1 interrupt control register (TA1IC)
$002D_{16}$		$008D_{16}$	UART4 transmit/NACK interrupt control register (S4TIC)
$002E_{16}$		$008E_{16}$	Timer A3 interrupt control register (TA3IC)
$002F_{16}$		$008F_{16}$	Bus collision detection(UART2) interrupt control register (BCN2IC)
0030_{16}	ROM areaset register (ROA) *	0090_{16}	UART0 transmit interrupt control register (S0TIC)
0031_{16}	Debug monitor area set register (DBA) *	0091_{16}	Bus collision detection(UART4) interrupt control register (BCN4IC)
0032_{16}	Expansion area set register 0 (EXA0) *	0092_{16}	UART1 transmit interrupt control register (S1TIC)
0033_{16}	Expansion area set register 1 (EXA1) *	0093_{16}	Key input interrupt control register (KUPIC)
0034_{16}	Expansion area set register 2 (EXA2) *	0094_{16}	Timer B0 interrupt control register (TB0IC)
0035_{16}	Expansion area set register 3 (EXA3) *	0095_{16}	
0036_{16}		0096_{16}	Timer B2 interrupt control register (TB2IC)
0037_{16}		0097_{16}	
0038_{16}		0098_{16}	Timer B4 interrupt control register (TB4IC)
0039_{16}		0099_{16}	
$003A_{16}$		$009A_{16}$	INT4 interrupt control register (INT4IC)
$003B_{16}$		$009B_{16}$	
$003C_{16}$		$009C_{16}$	INT2 interrupt control register (INT2IC)
$003D_{16}$		$009D_{16}$	
$003E_{16}$		$009E_{16}$	INT0 interrupt control register (INT0IC)
$003F_{16}$		$009F_{16}$	Exit priority register (RLVL)
0040_{16}	DRAM control register (DRAMCONT)	$00A0_{16}$	
0041_{16}	DRAM refresh interval set register (REFCNT)	$00A1_{16}$	
0042_{16}		$00A2_{16}$	
0043_{16}		$00A3_{16}$	
0044_{16}		$00A4_{16}$	

*As this register is used exclusively for debugger purposes, user cannot use this. Do not access to the register.
(The blank area is reserved and cannot be used by user.)

Figure 1.5.1. Location of peripheral unit control registers (1)

Address	Register
02C0$_{16}$	X0 register (X0R) Y0 register (Y0R)
02C1$_{16}$	
02C2$_{16}$	X1 register (X1R) Y1 register (Y1R)
02C3$_{16}$	
02C4$_{16}$	X2 register (X2R) Y2 register (Y2R)
02C5$_{16}$	
02C6$_{16}$	X3 register (X3R) Y3 register (Y3R)
02C7$_{16}$	
02C8$_{16}$	X4 register (X4R) Y4 register (Y4R)
02C9$_{16}$	
02CA$_{16}$	X5 register (X5R) Y5 register (Y5R)
02CB$_{16}$	
02CC$_{16}$	X6 register (X6R) Y6 register (Y6R)
02CD$_{16}$	
02CE$_{16}$	X7 register (X7R) Y7 register (Y7R)
02CF$_{16}$	
02D0$_{16}$	X8 register (X8R) Y8 register (Y8R)
02D1$_{16}$	
02D2$_{16}$	X9 register (X9R) Y9 register (Y9R)
02D3$_{16}$	
02D4$_{16}$	X10 register (X10R) Y10 register (Y10R)
02D5$_{16}$	
02D6$_{16}$	X11 register (X11R) Y11 register (Y11R)
02D7$_{16}$	
02D8$_{16}$	X12 register (X12R) Y12 register (Y12R)
02D9$_{16}$	
02DA$_{16}$	X13 register (X13R) Y13 register (Y13R)
02DB$_{16}$	
02DC$_{16}$	X14 register (X14R) Y14 register (Y14R)
02DD$_{16}$	
02DE$_{16}$	X15 register (X15R) Y15 register (Y15R)
02DF$_{16}$	
02E0$_{16}$	XY control register (XYC)
02E1$_{16}$	
02E2$_{16}$	
02E3$_{16}$	
02E4$_{16}$	
02E5$_{16}$	
02E6$_{16}$	
02E7$_{16}$	
02E8$_{16}$	
02E9$_{16}$	
02EA$_{16}$	
02EB$_{16}$	
02EC$_{16}$	
02ED$_{16}$	
02EE$_{16}$	
02EF$_{16}$	
02F0$_{16}$	
02F1$_{16}$	
02F2$_{16}$	
02F3$_{16}$	
02F4$_{16}$	
02F5$_{16}$	UART4 special mode register 3 (U4SMR3)
02F6$_{16}$	UART4 special mode register 2 (U4SMR2)
02F7$_{16}$	UART4 special mode register (U4SMR)
02F8$_{16}$	UART4 transmit/receive mode register (U4MR)
02F9$_{16}$	UART4 bit rate generator (U4BRG)
02FA$_{16}$	UART4 transmit buffer register (U4TB)
02FB$_{16}$	
02FC$_{16}$	UART4 transmit/receive control register 0 (U4C0)
02FD$_{16}$	UART4 transmit/receive control register 1 (U4C1)
02FE$_{16}$	UART4 receive buffer register (U4RB)
02FF$_{16}$	

Address	Register
0300$_{16}$	Timer B3, 4, 5 count start flag (TBSR)
0301$_{16}$	
0302$_{16}$	Timer A1-1 register (TA11)
0303$_{16}$	
0304$_{16}$	Timer A2-1 register (TA21)
0305$_{16}$	
0306$_{16}$	Timer A4-1 register (TA41)
0307$_{16}$	
0308$_{16}$	Three-phase PWM control register 0(INVC0)
0309$_{16}$	Three-phase PWM control register 1(INVC1)
030A$_{16}$	Thrree-phase output buffer register 0(IDB0)
030B$_{16}$	Thrree-phase output buffer register 1(IDB1)
030C$_{16}$	Dead time timer(DTT)
030D$_{16}$	Timer B2 interrupt occurrence frequency set counter(ICTB2)
030E$_{16}$	
030F$_{16}$	
0310$_{16}$	Timer B3 register (TB3)
0311$_{16}$	
0312$_{16}$	Timer B4 register (TB4)
0313$_{16}$	
0314$_{16}$	Timer B5 register (TB5)
0315$_{16}$	
0316$_{16}$	
0317$_{16}$	
0318$_{16}$	
0319$_{16}$	
031A$_{16}$	
031B$_{16}$	Timer B3 mode register (TB3MR)
031C$_{16}$	Timer B4 mode register (TB4MR)
031D$_{16}$	Timer B5 mode register (TB5MR)
031E$_{16}$	
031F$_{16}$	Interrupt cause select register (IFSR)
0320$_{16}$	
0321$_{16}$	
0322$_{16}$	
0323$_{16}$	
0324$_{16}$	
0325$_{16}$	UART3 special mode register 3 (U3SMR3)
0326$_{16}$	UART3 special mode register 2 (U3SMR2)
0327$_{16}$	UART3 special mode register (U3SMR)
0328$_{16}$	UART3 transmit/receive mode register (U3MR)
0329$_{16}$	UART3 bit rate generator (U3BRG)
032A$_{16}$	UART3 transmit buffer register (U3TB)
032B$_{16}$	
032C$_{16}$	UART3 transmit/receive control register 0 (U3C0)
032D$_{16}$	UART3 transmit/receive control register 1 (U3C1)
032E$_{16}$	UART3 receive buffer register (U3RB)
032F$_{16}$	
0330$_{16}$	
0331$_{16}$	
0332$_{16}$	
0333$_{16}$	
0334$_{16}$	
0335$_{16}$	UART2 special mode register 3 (U2SMR3)
0336$_{16}$	UART2 special mode register 2 (U2SMR2)
0337$_{16}$	UART2 special mode register (U2SMR)
0338$_{16}$	UART2 transmit/receive mode register (U2MR)
0339$_{16}$	UART2 bit rate generator (U2BRG)
033A$_{16}$	UART2 transmit buffer register (U2TB)
033B$_{16}$	
033C$_{16}$	UART2 transmit/receive control register 0 (U2C0)
033D$_{16}$	UART2 transmit/receive control register 1 (U2C1)
033E$_{16}$	UART2 receive buffer register (U2RB)
033F$_{16}$	

(The blank area is reserved and cannot be used by user.)

Figure 1.5.2. Location of peripheral unit control registers (2)

0340_{16}	Count start flag (TABSR)	0380_{16}	A-D register 0 (AD0)
0341_{16}	Clock prescaler reset flag (CPSRF)	0381_{16}	
0342_{16}	One-shot start flag (ONSF)	0382_{16}	A-D register 1 (AD1)
0343_{16}	Trigger select register (TRGSR)	0383_{16}	
0344_{16}	Up-down flag (UDF)	0384_{16}	A-D register 2 (AD2)
0345_{16}		0385_{16}	
0346_{16}	Timer A0 register (TA0)	0386_{16}	A-D register 3 (AD3)
0347_{16}		0387_{16}	
0348_{16}	Timer A1 register (TA1)	0388_{16}	A-D register 4 (AD4)
0349_{16}		0389_{16}	
$034A_{16}$	Timer A2 register (TA2)	$038A_{16}$	A-D register 5 (AD5)
$034B_{16}$		$038B_{16}$	
$034C_{16}$	Timer A3 register (TA3)	$038C_{16}$	A-D register 6 (AD6)
$034D_{16}$		$038D_{16}$	
$034E_{16}$	Timer A4 register (TA4)	$038E_{16}$	A-D register 7 (AD7)
$034F_{16}$		$038F_{16}$	
0350_{16}	Timer B0 register (TB0)	0390_{16}	
0351_{16}		0391_{16}	
0352_{16}	Timer B1 register (TB1)	0392_{16}	
0353_{16}		0393_{16}	
0354_{16}	Timer B2 register (TB2)	0394_{16}	A-D control register 2 (ADCON2)
0355_{16}		0395_{16}	
0356_{16}	Timer A0 mode register (TA0MR)	0396_{16}	A-D control register 0 (ADCON0)
0357_{16}	Timer A1 mode register (TA1MR)	0397_{16}	A-D control register 1 (ADCON1)
0358_{16}	Timer A2 mode register (TA2MR)	0398_{16}	D-A register 0 (DA0)
0359_{16}	Timer A3 mode register (TA3MR)	0399_{16}	
$035A_{16}$	Timer A4 mode register (TA4MR)	$039A_{16}$	D-A register 1 (DA1)
$035B_{16}$	Timer B0 mode register (TB0MR)	$039B_{16}$	
$035C_{16}$	Timer B1 mode register (TB1MR)	$039C_{16}$	D-A control register (DACON)
$035D_{16}$	Timer B2 mode register (TB2MR)	$039D_{16}$	
$035E_{16}$		$039E_{16}$	
$035F_{16}$		$039F_{16}$	
0360_{16}	UART0 transmit/receive mode register (U0MR)	$03A0_{16}$	
0361_{16}	UART0 bit rate generator (U0BRG)	$03A1_{16}$	
0362_{16}	UART0 transmit buffer register (U0TB)	$03A2_{16}$	
0363_{16}		$03A3_{16}$	
0364_{16}	UART0 transmit/receive control register 0 (U0C0)	$03A4_{16}$	
0365_{16}	UART0 transmit/receive control register 1 (U0C1)	$03A5_{16}$	
0366_{16}	UART0 receive buffer register (U0RB)	$03A6_{16}$	
0367_{16}		$03A7_{16}$	
0368_{16}	UART1 transmit/receive mode register (U1MR)	$03A8_{16}$	
0369_{16}	UART1 bit rate generator (U1BRG)	$03A9_{16}$	
$036A_{16}$	UART1 transmit buffer register (U1TB)	$03AA_{16}$	
$036B_{16}$		$03AB_{16}$	
$036C_{16}$	UART1 transmit/receive control register 0 (U1C0)	$03AC_{16}$	
$036D_{16}$	UART1 transmit/receive control register 1 (U1C1)	$03AD_{16}$	
$036E_{16}$	UART1 receive buffer register (U1RB)	$03AE_{16}$	
$036F_{16}$		$03AF_{16}$	Function select register C(PSC)
0370_{16}	UART transmit/receive control register 2 (UCON2)	$03B0_{16}$	Function select register A0 (PS0)
0371_{16}		$03B1_{16}$	Function select register A1 (PS1)
0372_{16}		$03B2_{16}$	Function select register B0 (PSL0)
0373_{16}		$03B3_{16}$	Function select register B1 (PSL1)
0374_{16}		$03B4_{16}$	Function select register A2 (PS2)
0375_{16}		$03B5_{16}$	Function select register A3 (PS3)
0376_{16}	Flash memory control register 1 (FMR1) (Note)	$03B6_{16}$	Function select register B2 (PSL2)
0377_{16}	Flash memory control register 0 (FMR0) (Note)	$03B7_{16}$	Function select register B3 (PSL3)
0378_{16}	DMA0 request cause select register (DM0SL)	$03B8_{16}$	
0379_{16}	DMA1 request cause select register (DM1SL)	$03B9_{16}$	
$037A_{16}$	DMA2 request cause select register (DM2SL)	$03BA_{16}$	
$037B_{16}$	DMA3 request cause select register (DM3SL)	$03BB_{16}$	
$037C_{16}$	CRC data register (CRCD)	$03BC_{16}$	
$037D_{16}$		$03BD_{16}$	
$037E_{16}$	CRC input register (CRCIN)	$03BE_{16}$	
$037F_{16}$		$03BF_{16}$	

Note :This register exists in the flash memory version.

(The blank area is reserved and cannot be used by user.)

Figure 1.5.3. Location of peripheral unit control registers (3)

Address	<100-pin version>	<144-pin version>
$03C0_{16}$	Port P6 (P6)	Port P6 (P6)
$03C1_{16}$	Port P7 (P7)	Port P7 (P7)
$03C2_{16}$	Port P6 direction register (PD6)	Port P6 direction register (PD6)
$03C3_{16}$	Port P7 direction register (PD7)	Port P7 direction register (PD7)
$03C4_{16}$	Port P8 (P8)	Port P8 (P8)
$03C5_{16}$	Port P9 (P9)	Port P9 (P9)
$03C6_{16}$	Port P8 direction register (PD8)	Port P8 direction register (PD8)
$03C7_{16}$	Port P9 direction register (PD9)	Port P9 direction register (PD9)
$03C8_{16}$	Port P10 (P10)	Port P10 (P10)
$03C9_{16}$	▒	Port P11 (P11)
$03CA_{16}$	Port P10 direction register (PD10)	Port P10 direction register (PD10)
$03CB_{16}$	▒	Port P11 direction register (PD11)
$03CC_{16}$	▒	Port P12 (P12)
$03CD_{16}$	▒	Port P13 (P13)
$03CE_{16}$	▒	Port P12 direction register (PD12)
$03CF_{16}$	▒	Port P13 direction register (PD13)
$03D0_{16}$	▒	Port P14 (P14)
$03D1_{16}$	▒	Port P15 (P15)
$03D2_{16}$	▒	Port P14 direction register (PD14)
$03D3_{16}$	▒	Port P15 direction register (PD15)
$03D4_{16}$		
$03D5_{16}$		
$03D6_{16}$		
$03D7_{16}$		
$03D8_{16}$		
$03D9_{16}$		
$03DA_{16}$	Pull-up control register 2 (PUR2)	Pull-up control register 2 (PUR2)
$03DB_{16}$	Pull-up control register 3 (PUR3)	Pull-up control register 3 (PUR3)
$03DC_{16}$	▨	Pull-up control register 4 (PUR4)
$03DD_{16}$		
$03DE_{16}$		
$03DF_{16}$		
$03E0_{16}$	Port P0 (P0)	Port P0 (P0)
$03E1_{16}$	Port P1 (P1)	Port P1 (P1)
$03E2_{16}$	Port P0 direction register (PD0)	Port P0 direction register (PD0)
$03E3_{16}$	Port P1 direction register (PD1)	Port P1 direction register (PD1)
$03E4_{16}$	Port P2 (P2)	Port P2 (P2)
$03E5_{16}$	Port P3 (P3)	Port P3 (P3)
$03E6_{16}$	Port P2 direction register (PD2)	Port P2 direction register (PD2)
$03E7_{16}$	Port P3 direction register (PD3)	Port P3 direction register (PD3)
$03E8_{16}$	Port P4 (P4)	Port P4 (P4)
$03E9_{16}$	Port P5 (P5)	Port P5 (P5)
$03EA_{16}$	Port P4 direction register (PD4)	Port P4 direction register (PD4)
$03EB_{16}$	Port P5 direction register (PD5)	Port P5 direction register (PD5)
$03EC_{16}$		
$03ED_{16}$		
$03EE_{16}$		
$03EF_{16}$		
$03F0_{16}$	Pull-up control register 0 (PUR0)	Pull-up control register 0 (PUR0)
$03F1_{16}$	Pull-up control register 1 (PUR1)	Pull-up control register 1 (PUR1)
$03F2_{16}$		
$03F3_{16}$		
≈	≈	≈
$03FC_{16}$		
$03FD_{16}$		
$03FE_{16}$		
$03FF_{16}$	Port control register (PCR)	Port control register (PCR)

(The blank area is reserved and cannot be used by user.)

Note 1: ▒ Addresses $03C9_{16}$, $03CB_{16}$ to $03D3_{16}$ area is for future plan.
Must set "FF_{16}" to address $03CB_{16}$, $03CE_{16}$, $03CF_{16}$, $03D2_{16}$, $03D3_{16}$ at initial setting.
Note 2: ▨ Address $03DC_{16}$ area is for future plan. Must set "00_{16}" to address $03DC_{16}$ at initial setting.

Figure 1.5.4. Location of peripheral unit control registers (4)

Software Reset

Writing “1” to bit 3 of the processor mode register 0 (address 0004_{16}) applies a (software) reset to the microcomputer. A software reset has the same effect as a hardware reset. The contents of internal RAM are preserved.

Processor Mode

(1) Types of Processor Mode

One of three processor modes can be selected: single-chip mode, memory expansion mode, and microprocessor mode. The functions of some pins, the memory map, and the access space differ according to the selected processor mode.

- **Single-chip mode**

  In single-chip mode, only internal memory space (SFR, internal RAM, and internal ROM) can be accessed. Ports P0 to P10 can be used as programmable I/O ports or as I/O ports for the internal peripheral functions.

- **Memory expansion mode**

  In memory expansion mode, external memory can be accessed in addition to the internal memory space (SFR, internal RAM, and internal ROM).

  In this mode, some of the pins function as the address bus, the data bus, and as control signals. The number of pins assigned to these functions depends on the bus and register settings. (See “Bus Settings” for details.)

- **Microprocessor mode**

  In microprocessor mode, the SFR, internal RAM, and external memory space can be accessed. The internal ROM area cannot be accessed.

  In this mode, some of the pins function as the address bus, the data bus, and as control signals. The number of pins assigned to these functions depends on the bus and register settings. (See “Bus Settings” for details.)

(2) Setting Processor Modes

The processor mode is set using the CNV_{SS} pin and the processor mode bits (bits 1 and 0 at address 0004_{16}). Do not set the processor mode bits to “10_2”.

Regardless of the level of the CNV_{SS} pin, changing the processor mode bits selects the mode. Therefore, never change the processor mode bits when changing the contents of other bits. Also do not attempt to shift to or from the microprocessor mode within the program stored in the internal ROM area.

- **Applying V_{SS} to CNV_{SS} pin**

  The microcomputer begins operation in single-chip mode after being reset. Memory expansion mode is selected by writing “01_2” to the processor mode is selected bits.

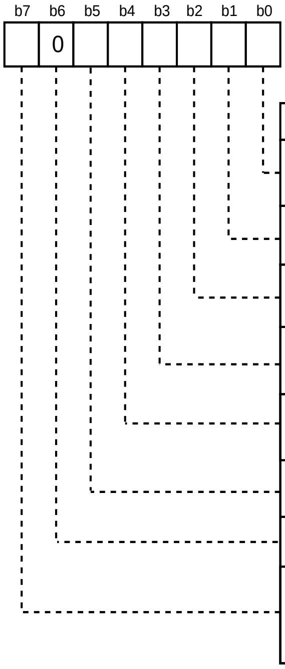
- **Applying V_{CC} to CNV_{SS} pin**

  The microcomputer starts to operate in microprocessor mode after being reset.

Figure 1.6.1 and 1.6.2 show the processor mode register 0 and 1.

Figure 1.6.3 shows the memory maps applicable for each processor modes.

Processor mode register 0 (Note 1)



Symbol	Address	When reset
PM0	0004_{16}	80_{16} (Note 2)

Bit symbol	Bit name	Function	R	W
PM00	Processor mode bit (Note 8)	b1 b0 0 0: Single-chip mode 0 1: Memory expansion mode 1 0: Inhibited 1 1: Microprocessor mode	○	○
PM01			○	○
PM02	R/W mode select bit (Note 7)	0 : $\overline{RD}$,$\overline{BHE}$,$\overline{WR}$ 1 : RD,WRH,WRL	○	○
PM03	Software reset bit	The device is reset when this bit is set to “1”. The value of this bit is “0” when read.	○	○
PM04	Multiplexed bus space select bit (Note 3)	b5 b4 0 0 : Multiplexed bus is not used 0 1 : Allocated to CS2 space 1 0 : Allocated to CS1 space 1 1 : Allocated to entire space (Note4)	○	○
PM05			○	○
Reserved bit		Must always be set to “0”	○	○
PM07	BCLK output disable bit (Note 5)	0 : BCLK is output (Note 6) 1 : Function set by bit 0,1 of system clock control register 0	○	○

Note 1: Set bit 1 of the protect register (address $000A_{16}$) to “1” when writing new values to this register.
Note 2: If the Vcc voltage is applied to the CNVss, the value of this register when reset is 03_{16}. (PM00 is set to “1” and PM07 is set to “0”.)
Note 3: Valid in microprocessor and memory expansion modes 1, 2 and 3. Do not use multiplex bus when mode 0 is selected. Do not set to allocated to CS2 space when mode 2 is selected.
Note 4: After the reset has been released, the M16C/80 group MCU operates using the separate bus. As a result, in microprocessor mode, you cannot select the full $\overline{CS}$ space multiplex bus.
When you select the full $\overline{CS}$ space multiplex bus in memory expansion mode, the address bus operates with 64 Kbytes boundaries for each chip select.
Mode 0: Multiplexed bus cannot be used.
Mode 1: $\overline{CS0}$ to $\overline{CS2}$ when you select full $\overline{CS}$ space.
Mode 2: $\overline{CS0}$ to $\overline{CS1}$ when you select full $\overline{CS}$ space.
Mode 3: $\overline{CS0}$ to $\overline{CS3}$ when you select full $\overline{CS}$ space.
Note 5: No BCLK is output in single chip mode even when "0" is set in PM07. When stopping clock output in microprocessor or memory expansion mode, make the following settings: PM07="1", bit 0 (CM00) and bit 1 (CM01) of system clock control register 0 (address 0006_{16}) = "0". "L" is now output from $P5_3$.
Note 6: When selecting BCLK, set bits 0 and 1 of system clock control register 0 (CM00, CM01) to "0".
Note 7: When using 16-bit bus width in DRAM controller, set this bit to "1".
Note 8: Do not set the processor mode bits and other bits simultaneously when setting the processor mode bits to “01_2” or “11_2”. Set the other bits first, and then change the processor mode bits.

Figure 1.6.1. Processor mode register 0

Processor mode register 1 (Note 1) :Mask ROM version ROMless version (144-pin version)

b7	b6	b5	b4	b3	b2	b1	b0
				0			

Symbol	Address	When reset
PM1	0005_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
PM10	External memory area mode bit (Note 3)	b1 b0 0 0 : Mode 0 ($P4_4$ to $P4_7$: A_{20} to $\overline{A_{23}}$) 0 1 : Mode 1 ($P4_4$: A_{20}, $P4_5$ to $P4_7$: $\overline{CS2}$ to $\overline{CS0}$) 1 0 : Mode 2 ($P4_4$, $P4_5$: A_{20}, A_{21}, $P4_6$, $P4_7$: $\overline{CS1}$, $\overline{CS0}$) 1 1 : Mode 3 (Note 2) ($P4_4$ to $P4_7$: $\overline{CS3}$ to $\overline{CS0}$)	○	○
PM11				
PM12	Internal memory wait bit	0 : No wait state 1 : Wait state inserted	—	—
Reserved bit		Must always be set to “0”	—	○
PM14	ALE pin select bit (Note 3)	b5 b4 0 0 : No ALE 0 1 : $P5_3$/BCLK (Note 4) 1 0 : $P5_6$/$\overline{RAS}$ 1 1 : $P5_4$/HLDA	○	○
PM15			○	○
Nothing is assinged. When read, the content is indeterminate.			—	—

Note 1: Set bit 1 of the protect register (address $000A_{16}$) to “1” when writing new values to this register.
Note 2: When mode 3 is selected, DRAMC is not used.
Note 3: Valid in memory expansion mode or in microprocessor mode.
Note 4: When selecting $P5_3$/BCLK, set bits 0 and 1 of system clock control register 0 (CM00, CM01) to "0".

Processor mode register 1 (Note 1) :Flash memory version

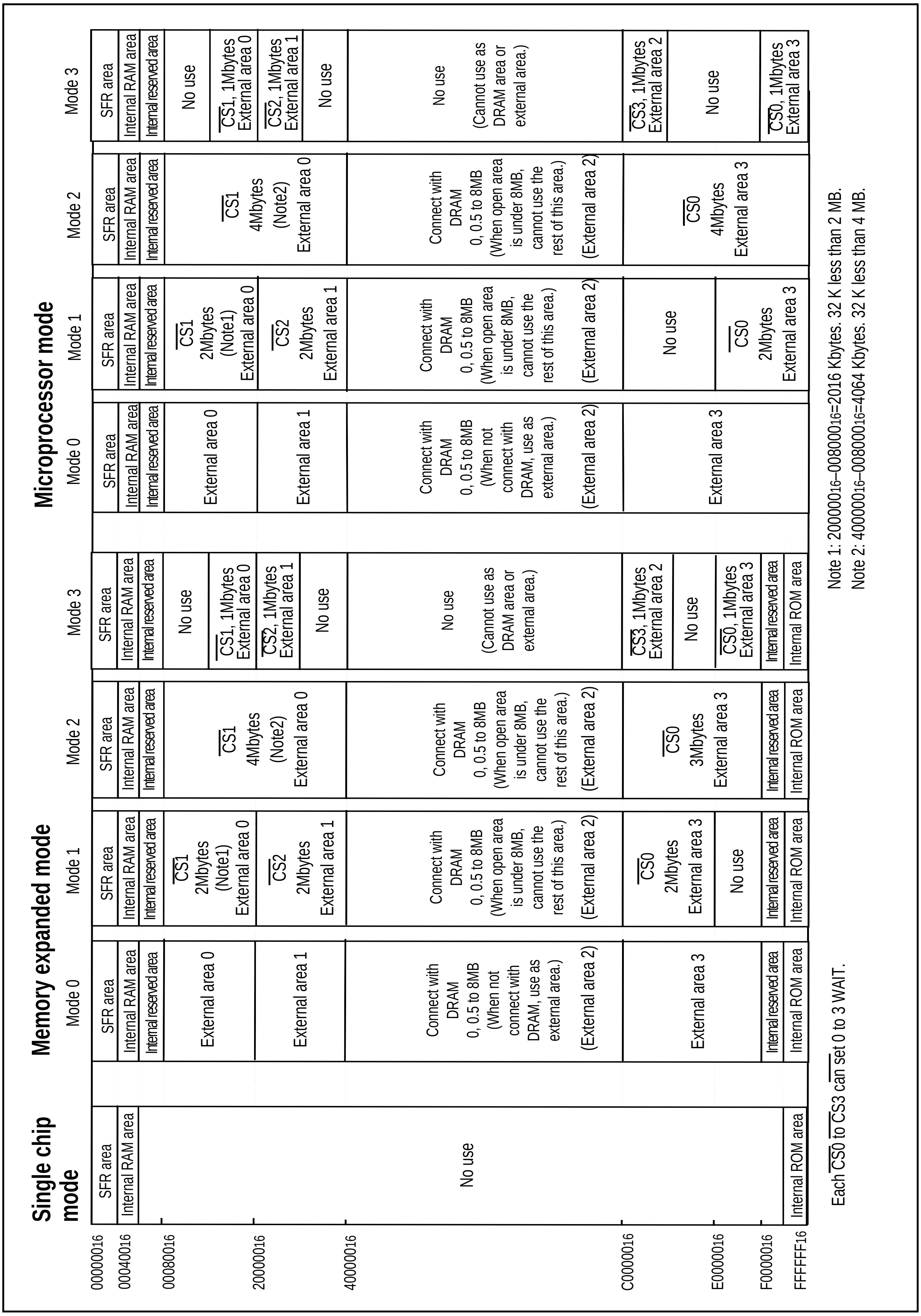
b7	b6	b5	b4	b3	b2	b1	b0
				0			

Symbol	Address	When reset
PM1	0005_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
PM10	External memory area mode bit (Note 3)	b1 b0 0 0 : Mode 0 ($P4_4$ to $P4_7$: A_{20} to $\overline{A_{23}}$) 0 1 : Mode 1 ($P4_4$: A_{20}, $P4_5$ to $P4_7$: $\overline{CS2}$ to $\overline{CS0}$) 1 0 : Mode 2 ($P4_4$, $P4_5$: A_{20}, A_{21}, $P4_6$, $P4_7$: $\overline{CS1}$, $\overline{CS0}$) 1 1 : Mode 3 (Note 2) ($P4_4$ to $P4_7$: $\overline{CS3}$ to $\overline{CS0}$)	○	○
PM11				
PM12	Internal memory wait bit	0 : No wait state 1 : Wait state inserted	—	—
Reserved bit		Must always be set to “0”	—	○
PM14	ALE pin select bit (Note 3)	b5 b4 0 0 : No ALE 0 1 : $P5_3$/BCLK (Note 4) 1 0 : $P5_6$/$\overline{RAS}$ 1 1 : $P5_4$/HLDA	○	○
PM15			○	○
Reserved bit		Must always be set to “1” (Note 5)	○	○

Note 1: Set bit 1 of the protect register (address $000A_{16}$) to “1” when writing new values to this register.
Note 2: When mode 3 is selected, DRAMC is not used.
Note 3: Valid in memory expansion mode or in microprocessor mode.
Note 4: When selecting $P5_3$/BCLK, set bits 0 and 1 of system clock control register 0 (CM00, CM01) to "0".
Note 5: Rewrite this bit when the main clock is in division by 8 mode.

Figure 1.6.2. Processor mode register 1


Single chip mode
Memory expanded mode
Microprocessor mode
Mode 0
Mode 1
Mode 2
Mode 3
00000016
00040016
00080016
20000016
40000016
C0000016
E0000016
F0000016
FFFFFF16
SFR area
Internal RAM area
Internal reserved area
Internal ROM area
No use
External area 0
External area 1
(External area 2)
External area 3
Connect with DRAM 0, 0.5 to 8MB (When not connect with DRAM, use as external area.)
Connect with DRAM 0, 0.5 to 8MB (When open area is under 8MB, cannot use the rest of this area.)
CS1 2Mbytes (Note1) External area 0
CS2 2Mbytes External area 1
CS0 2Mbytes External area 3
CS1 4Mbytes (Note2) External area 0
CS0 3Mbytes External area 3
CS0 4Mbytes External area 3
CS1, 1Mbytes External area 0
CS2, 1Mbytes External area 1
No use (Cannot use as DRAM area or external area.)
CS3, 1Mbytes External area 2
CS0, 1Mbytes External area 3

Each CS0 to CS3 can set 0 to 3 WAIT.

Note 1: 20000016–0080016=2016 Kbytes. 32 K less than 2 MB.
Note 2: 40000016–0080016=4064 Kbytes. 32 K less than 4 MB.

Figure 1.6.3. Memory maps in each processor mode

Bus Settings

The BYTE pin, bit 0 to 3 of the external data bus width control register (address $000B_{16}$), bits 4 and 5 of the processor mode register 0 (address 0004_{16}) and bit 0 and 1 of the processor mode register 1 (address 0005_{16}) are used to change the bus settings.
Table 1.7.1 shows the factors used to change the bus settings, figure 1.7.1 shows external data bus width control register and table 1.7.2 shows external area 0 to 3 and external area mode.

Table 1.7.1. Factors for switching bus settings

Bus setting	Switching factor
Switching external address bus width	External data bus width control register
Switching external data bus width	BYTE pin (external area 3 only)
Switching between separate and multiplex bus	Bits 4 and 5 of processor mode register 0

(1) Selecting external address bus width

You can select the width of the address bus output externally from the 16 Mbytes address space, the number of chip select signals, and the address area of the chip select signals. (Note, however, that when you select “Full $\overline{CS}$ space multiplex bus”, addresses A_0 to A_{15} are output.) The combination of bits 0 and 1 of the processor mode register 1 allow you to set the external area mode.
When using DRAM controller, the DRAM area is output by multiplexing of the time splitting of the row and column addresses.

(2) Selecting external data bus width

You can select 8-bit or 16-bit for the width of the external data bus for external areas 0, 1, 2, and 3. When the data bus width bit of the external data bus width control register is “0”, the data bus width is 8 bits; when “1”, it is 16 bits. The width can be set for each of the external areas. The default bus width for external area 3 is 16 bits when the BYTE pin is “L” after a reset, or 8 bits when the BYTE pin is “H” after a reset. The bus width selection is valid only for the external bus (the internal bus width is always 16 bits). During operation, fix the level of the BYTE pin to “H” or “L”.

(3) Selecting separate/multiplex bus

The bus format can be set to multiplex or separate bus using bits 4 and 5 of the processor mode register 0.

- **Separate bus**

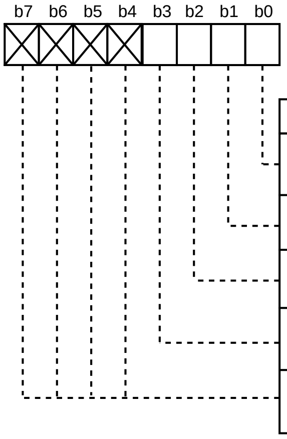
  In this bus configuration, input and output is performed on separate data and address buses. The data bus width can be set to 8 bits or 16 bits using the external data bus width control register. For all programmable external areas, P0 is the data bus when the external data bus is set to 8 bits, and P1 is a programmable IO port. When the external data bus width is set to 16 bits for any of the external areas, P0 and P1 (although P1 is undefined for any 8-bit bus areas) are the data bus.
  When accessing memory using the separate bus configuration, you can select a software wait using the wait control register.

- **Multiplex bus**

  In this bus configuration, data and addresses are input and output on a time-sharing basis. For areas for which 8-bit has been selected using the external data bus width control register, the 8 bits D_0 to D_7 are multiplexed with the 8 bits A_0 to A_7. For areas for which 16-bit has been selected using the external data bus width control register, the 16 bits D_0 to D_{15} are multiplexed with the 16 bits A_0 to A_{15}. When accessing memory using the multiplex bus configuration, two waits are inserted regardless of whether you select “No wait” or “1 wait’ in the appropriate bit of the wait control register.

The default after a reset is the separate bus configuration, and the full $\overline{CS}$ space multiplex bus configuration cannot be selected in microprocessor mode. If you select “Full $\overline{CS}$ space multiplex bus”, the 16 bits from A0 to A15 are output for the address

External data bus width control register



Symbol	Address	When reset
DS	000B$_{16}$	XXXXX000$_2$

Bit symbol	Bit name	Function	R	W
DS0	External area 0 data bus width bit	0 : 8 bits data bus width 1 : 16 bits data bus width	○	○
DS1	External area 1 data bus width bit	0 : 8 bits data bus width 1 : 16 bits data bus width	○	○
DS2	External area 2 data bus width bit	0 : 8 bits data bus width 1 : 16 bits data bus width	○	○
DS3	External area 3 data bus width bit (Note)	0 : 8 bits data bus width 1 : 16 bits data bus width	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—

Note: The value after a reset is determined by the input via the BYTE pin.

Figure 1.7.1. External data bus width control register

Table 1.7.2. External area 0 to 3 and external area mode

	External area mode (Note 2)	Mode 0	Mode 1	Mode 2	Mode 3
External area 0	Memory expansion mode, Microprocessor mode	008000$_{16}$ to 1FFFFF$_{16}$	<CS1 area> 008000$_{16}$ to 1FFFFF$_{16}$	<CS1 area> 008000$_{16}$ to 1FFFFF$_{16}$	<CS1 area> 100000$_{16}$ to 1FFFFF$_{16}$
External area 1	Memory expansion mode, Microprocessor mode	200000$_{16}$ to 3FFFFF$_{16}$	<CS2 area> 200000$_{16}$ to 3FFFFF$_{16}$	No area is selected.	<CS2 area> 200000$_{16}$ to 2FFFFF$_{16}$
External area 2	Memory expansion mode, Microprocessor mode	400000$_{16}$ to BFFFFF$_{16}$ (Note 1)	<DRAMC area> 400000$_{16}$ to BFFFFF$_{16}$	<DRAMC area> 400000$_{16}$ to BFFFFF$_{16}$	<CS3 area> C00000$_{16}$ to CFFFFF$_{16}$
External area 3	Memory expansion mode	C00000$_{16}$ to EFFFFF$_{16}$	<CS0 area> C00000$_{16}$ to EFFFFF$_{16}$	<CS0 area> C00000$_{16}$ to EFFFFF$_{16}$	<CS0 area> E00000$_{16}$ to EFFFFF$_{16}$
	Microprocessor mode	C00000$_{16}$ to FFFFFF$_{16}$	<CS0 area> E00000$_{16}$ to FFFFFF$_{16}$	<CS0 area> C00000$_{16}$ to FFFFFF$_{16}$	<CS0 area> F00000$_{16}$ to FFFFFF$_{16}$

Note 1: DRAMC area when using DRAMC.

Note 2: Set the external area mode (modes 0, 1, 2, and 3) using bits 0 and 1 of the processor mode register 1 (address 0005$_{16}$).

Table 1.7.3. Each processor mode and port function

Processor mode	Single-chip mode	Memory expansion mode/microprocessor modes				Memory expansion mode	
Multiplexed bus space select bit		"01", "10" CS1 or CS2 : multiplexed bus, and the other : separate bus		"00" Separate bus		"11" (Note 1) All space multiplexed bus	
Data bus width BYTE pin level		All external area is 8 bits	Some external area is 16 bits	All external area is 8 bits	Some external area is 16 bits	All external area is 8 bits	Some external area is 16 bits
$P0_0$ to $P0_7$	I/O port	Data bus	Data bus	Data bus	Data bus	I/O port	I/O port
$P1_0$ to $P1_7$	I/O port	I/O port	Data bus	I/O port	Data bus	I/O port	I/O port
$P2_0$ to $P2_7$	I/O port	Address bus /data bus (Note 2)	Address bus /data bus (Note 2)	Address bus	Address bus	Address bus /data bus	Address bus /data bus
$P3_0$ to $P3_7$	I/O port	Address bus	Address bus /data bus (Note 2)	Address bus	Address bus	Address bus	Address bus /data bus
$P4_0$ to $P4_3$	I/O port	Address bus	Address bus	Address bus	Address bus	I/O port	I/O port
$P4_4$ to $P4_6$	I/O port	$\overline{CS}$ (chip select) or address bus (A_{20} to A_{22}) (For details, refer to "Bus control") (Note 5)					
$P4_7$	I/O port	$\overline{CS}$ (chip select) or address bus ($\overline{A}_{23}$) (For details, refer to "Bus control") (Note 5)					
$P5_0$ to $P5_3$	I/O port	Outputs $\overline{RD}$, $\overline{WRL}$, $\overline{WRH}$ and BCLK, or $\overline{RD}$, $\overline{BHE}$, $\overline{WR}$ and BCLK (For details, refer to "Bus control") (Note 3,4)					
$P5_4$	I/O port	$\overline{HLDA}$(Note 3)	$\overline{HLDA}$(Note 3)	$\overline{HLDA}$(Note 3)	$\overline{HLDA}$(Note 3)	$\overline{HLDA}$(Note 3)	$\overline{HLDA}$(Note 3)
$P5_5$	I/O port	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$
$P5_6$	I/O port	RAS (Note 3)	RAS (Note 3)	RAS (Note 3)	RAS (Note 3)	RAS (Note 3)	RAS (Note 3)
$P5_7$	I/O port	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$

Note 1:The default after a reset is the separate bus configuration, and "Full $\overline{CS}$ space multiplex bus" cannot be selected in microprocessor mode. When you select "Full $\overline{CS}$ space multiplex bus" in extended memory mode, the address bus operates with 64 Kbytes boundaries for each chip select.
Note 2: Address bus in separate bus configuration.
Note 3: The ALE output pin is selected using bits 4 and 5 of the processor mode register 1.
Note 4: When you have selected use of the DRAM controller and you access the DRAM area, these are $\overline{CASL}$, $\overline{CASH}$, $\overline{DW}$, and BCLK outputs.
Note 5: The $\overline{CS}$ signal and address bus selection are set by the external area mode.

Bus Control

The following explains the signals required for accessing external devices and software waits. The signals required for accessing the external devices are valid when the processor mode is set to memory expansion mode and microprocessor mode.

(1) Address bus/data bus

There are 24 pins, A_0 to A_{22} and $\overline{A_{23}}$ for the address bus for accessing the 16 Mbytes address space. $\overline{A_{23}}$ is an inverted output of the MSB of the address.

The data bus consists of pins for data IO. The external data bus control register (address 000B$_{16}$) selects the 8-bit data bus, D_0 to D_7 for each external area, or the 16-bit data bus, D_0 to D_{15}. After a reset, there is by default an 8-bit data bus for the external area 3 when the BYTE pin is “H”, or a 16-bit data bus when the BYTE pin is “L”.

When shifting from single-chip mode to extended memory mode, the value on the address bus is undefined until an external area is accessed.

When accessing a DRAM area with DRAM control in use, a multiplexed signal consisting of row address and column address is output to A_8 to A_{20}.

(2) Chip select signals

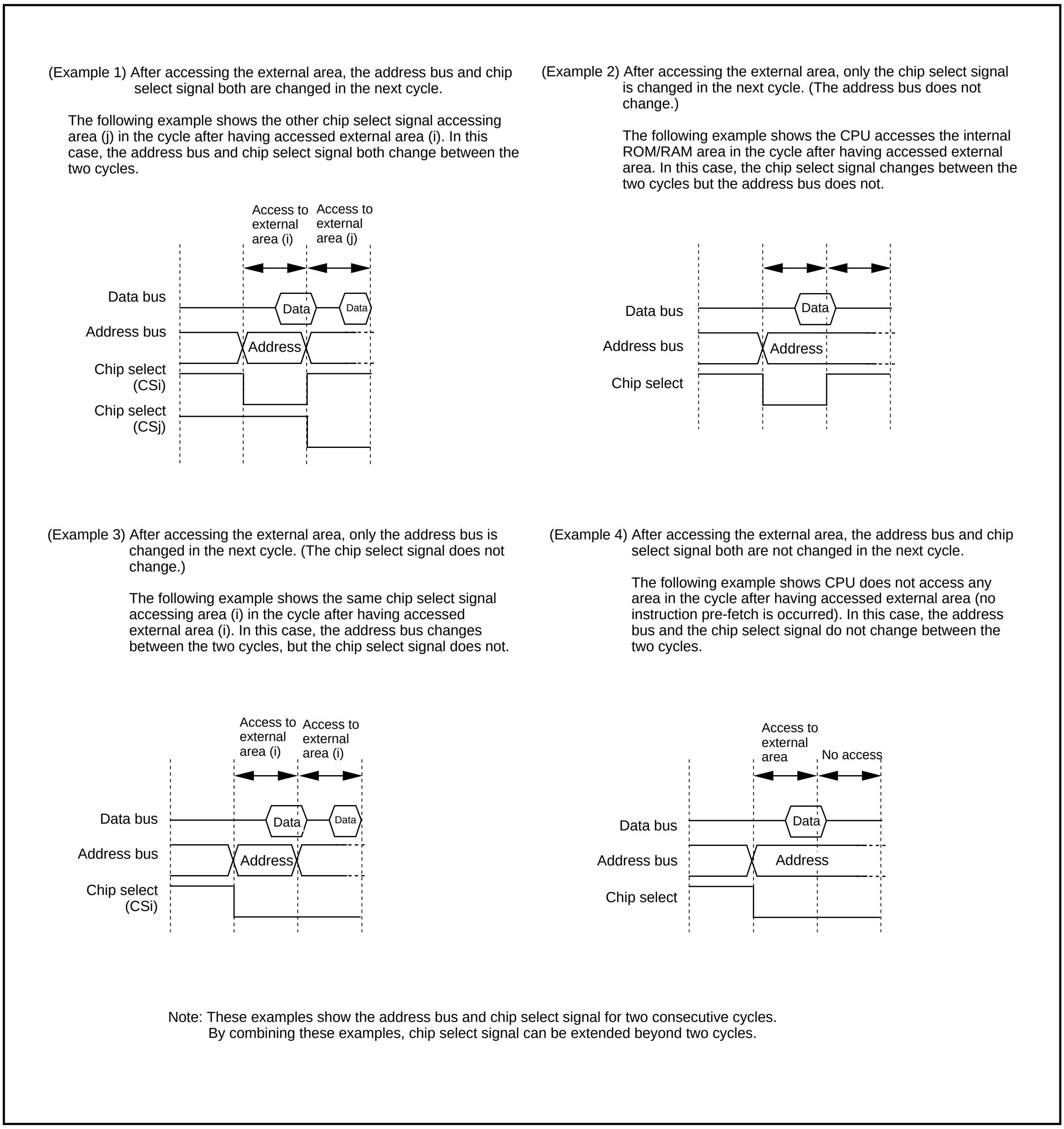
The chip select signals share A_0 to A_{22} and $\overline{A_{23}}$. You can use bits 0 and 1 of the processor mode register 1 (address 0005$_{16}$) to set the external area mode, then select the chip select area and number of address outputs.

In microprocessor mode, external area mode 0 is selected after a reset. The external area can be split into a maximum of four using the chip select signals. Table 1.7.4 shows the external areas specified by the chip select signals.

Table 1.7.4. External areas specified by the chip select signals

Memory space expansion mode		Processor mode	Chip select signal			
			$\overline{CS0}$	$\overline{CS1}$	$\overline{CS2}$	$\overline{CS3}$
Specified address range	Mode 0		—— ($\overline{A23}$)	—— (A22)	—— (A21)	—— (A20)
	Mode 1	Memory expansion mode	C00000$_{16}$ to DFFFFF$_{16}$ (2 Mbytes)	008000$_{16}$ to 1FFFFF$_{16}$ (2016 Kbytes)	200000$_{16}$ to 3FFFFF$_{16}$ (2 Mbytes)	—— (A20)
		Microprocessor mode	E00000$_{16}$ to FFFFFF$_{16}$ (2 Mbytes)			
	Mode 2	Memory expansion mode	C00000$_{16}$ to EFFFFF$_{16}$ (3 Mbytes)	008000$_{16}$ to 3FFFFF$_{16}$ (4064 Kbytes)	—— (A21)	—— (A20)
		Microprocessor mode	C00000$_{16}$ to FFFFFF$_{16}$ (4 Mbytes)			
	Mode 3	Memory expansion mode	E00000$_{16}$ to EFFFFF$_{16}$ (1 Mbytes)	100000$_{16}$ to 1FFFFF$_{16}$ (1 Mbytes)	200000$_{16}$ to 2FFFFF$_{16}$ (1 Mbytes)	C00000$_{16}$ to CFFFFF$_{16}$ (1 Mbytes)
		Microprocessor mode	F00000$_{16}$ to FFFFFF$_{16}$ (1 Mbytes)			

The chip select signal turns “L” (active) in synchronize with the address bus. However, its turning “H” depends on the area accessed in the next cycle. Figure 1.7.2 shows the output examples of the address bus and chip select signals.



(Example 1) After accessing the external area, the address bus and chip select signal both are changed in the next cycle.

The following example shows the other chip select signal accessing area (j) in the cycle after having accessed external area (i). In this case, the address bus and chip select signal both change between the two cycles.

(Example 2) After accessing the external area, only the chip select signal is changed in the next cycle. (The address bus does not change.)

The following example shows the CPU accesses the internal ROM/RAM area in the cycle after having accessed external area. In this case, the chip select signal changes between the two cycles but the address bus does not.

(Example 3) After accessing the external area, only the address bus is changed in the next cycle. (The chip select signal does not change.)

The following example shows the same chip select signal accessing area (i) in the cycle after having accessed external area (i). In this case, the address bus changes between the two cycles, but the chip select signal does not.

(Example 4) After accessing the external area, the address bus and chip select signal both are not changed in the next cycle.

The following example shows CPU does not access any area in the cycle after having accessed external area (no instruction pre-fetch is occurred). In this case, the address bus and the chip select signal do not change between the two cycles.

Note: These examples show the address bus and chip select signal for two consecutive cycles.
By combining these examples, chip select signal can be extended beyond two cycles.

Figure 1.7.2. Example of address bus and chip select signal outputs (Separate bus)

(3) Read/write signals

With a 16-bit data bus, bit 2 of the processor mode register 0 (address 0004_{16}) select the combinations of $\overline{\text{RD}}$, $\overline{\text{BHE}}$, and $\overline{\text{WR}}$ signals or $\overline{\text{RD}}$, $\overline{\text{WRL}}$, and $\overline{\text{WRH}}$ signals. With a 8-bit full space data bus, use the combination of $\overline{\text{RD}}$, $\overline{\text{WR}}$, and $\overline{\text{BHE}}$ signals as read/write signals. (Set "0" to bit 2 of the processor mode register 0 (address 0004_{16}).) When using both 8-bit and 16-bit data bus widths and you access an 8-bit data bus area, the $\overline{\text{RD}}$, $\overline{\text{WR}}$ and $\overline{\text{BHE}}$ signals combination is selected regardless of the value of bit 2 of the processor mode register 0 (address 0004_{16}).

Tables 1.7.5 and 1.7.6 show the operation of these signals.

After a reset has been cancelled, the combination of $\overline{\text{RD}}$, $\overline{\text{WR}}$, and $\overline{\text{BHE}}$ signals is automatically selected.

When switching to the $\overline{\text{RD}}$, $\overline{\text{WRL}}$, and $\overline{\text{WRH}}$ combination, do not write to external memory until bit 2 of the processor mode register 0 (address 0004_{16}) has been set (Note).

Note 1: Before attempting to change the contents of the processor mode register 0, set bit 1 of the protect register (address $000A_{16}$) to “1”.

Note 2: When using 16-bit data bus width for DRAM controller, select $\overline{\text{RD}}$, $\overline{\text{WRL}}$, and $\overline{\text{WRH}}$ signals.

Table 1.7.5. Operation of $\overline{\text{RD}}$, $\overline{\text{WRL}}$, and $\overline{\text{WRH}}$ signals

Data bus width	$\overline{\text{RD}}$	$\overline{\text{WRL}}$	$\overline{\text{WRH}}$	Status of external data bus
16-bit	L	H	H	Read data
	H	L	H	Write 1 byte of data to even address
	H	H	L	Write 1 byte of data to odd address
	H	L	L	Write data to both even and odd addresses
8-bit	H	L (Note)	Not used	Write 1 byte of data
	L	H (Note)	Not used	Read 1 byte of data

Note: It becomes $\overline{\text{WR}}$ signal.

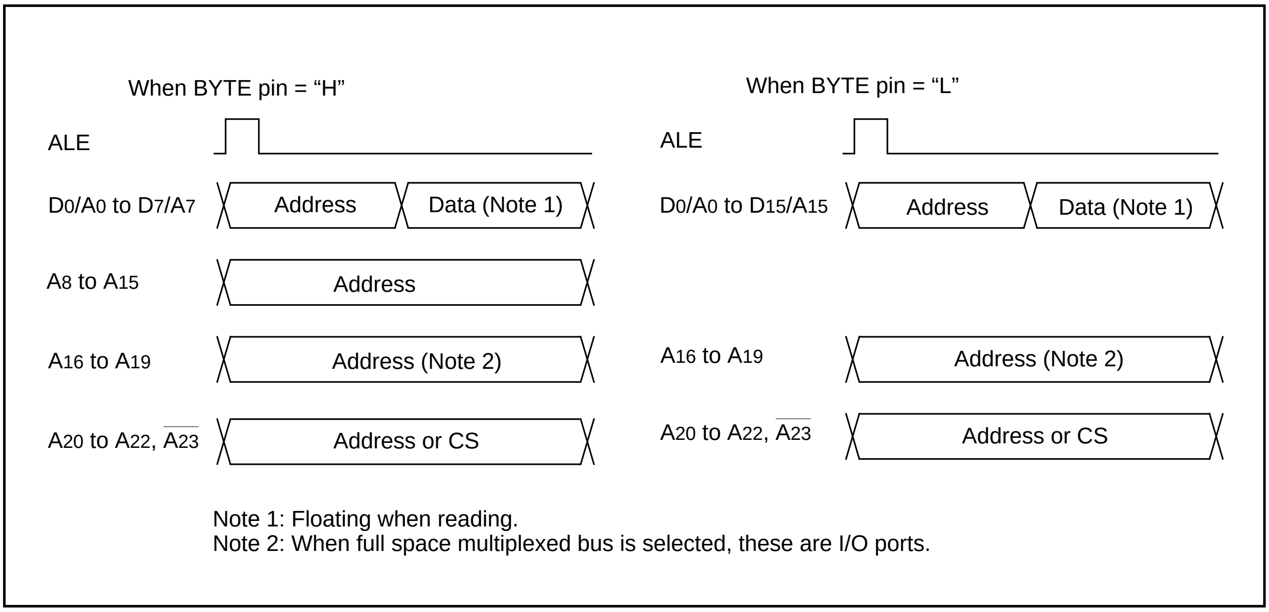
Table 1.7.6. Operation of $\overline{\text{RD}}$, $\overline{\text{WR}}$, and $\overline{\text{BHE}}$ signals

Data bus width	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{BHE}}$	A0	Status of external data bus
16-bit	H	L	L	H	Write 1 byte of data to odd address
	L	H	L	H	Read 1 byte of data from odd address
	H	L	H	L	Write 1 byte of data to even address
	L	H	H	L	Read 1 byte of data from even address
	H	L	L	L	Write data to both even and odd addresses
	L	H	L	L	Read data from both even and odd addresses
8-bit	H	L	Not used	H / L	Write 1 byte of data
	L	H	Not used	H / L	Read 1 byte of data

(4) ALE signal

The ALE signal latches the address when accessing the multiplex bus space. Latch the address when the ALE signal falls. The ALE output pin is selected using bits 4 and 5 of the processor mode register 1 (address 0005_{16}).

The ALE signal is occurred regardless of internal area and external area.



Note 1: Floating when reading.
Note 2: When full space multiplexed bus is selected, these are I/O ports.

Figure 1.7.3. ALE signal and address/data bus

(5) Ready signal

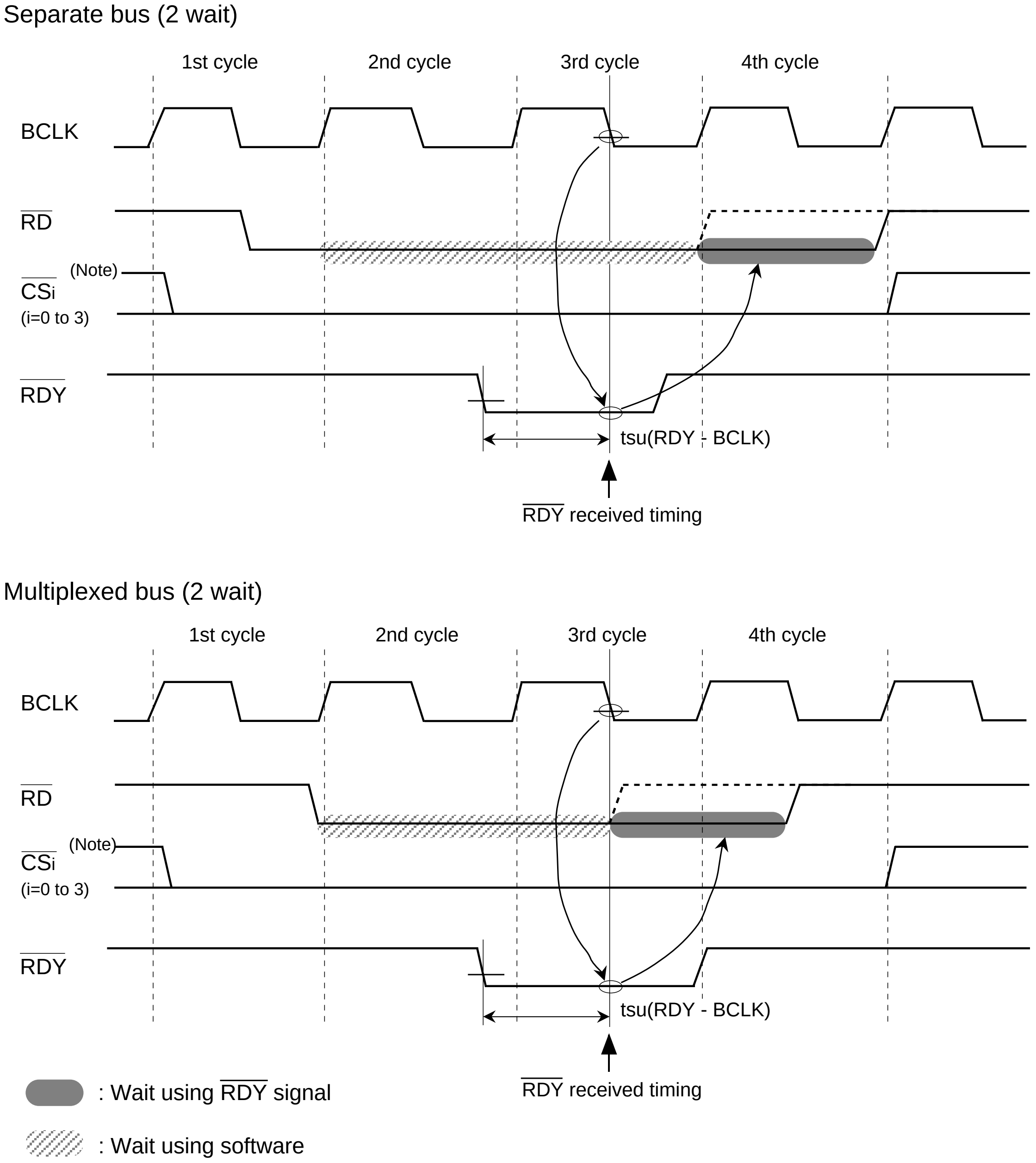
The ready signal facilitates access of external devices that require a long time for access. As shown in Figure 1.7.2, inputting “L” to the $\overline{\text{RDY}}$ pin at the falling edge of BCLK causes the microcomputer to enter the ready state. Inputting “H” to the $\overline{\text{RDY}}$ pin at the falling edge of BCLK cancels the ready state. Table 1.7.7 shows the microcomputer status in the ready state. Figure 1.7.4 shows the example of the $\overline{\text{RD}}$ signal being extended using the $\overline{\text{RDY}}$ signal.

Ready is valid when accessing the external area during the bus cycle in which the software wait is applied. When no software wait is operating, the $\overline{\text{RDY}}$ signal is ignored, but even in this case, unused pins must be pulled up.

Table 1.7.7. Microcomputer status in ready state (Note)

Item	Status
Oscillation	On
$\overline{\text{RD}}$/$\overline{\text{WR}}$ signal, address bus, data bus, $\overline{\text{CS}}$ ALE signal, $\overline{\text{HLDA}}$, programmable I/O ports	Maintain status when ready signal received
Internal peripheral circuits	On

Note: The ready signal cannot be received immediately prior to a software wait.



$\overline{\text{RDY}}$ signal received timing for i wait(s): i + 1 cycles
(i = 1 to 3)

Note: Chip select may get longer by a state of CPU such as an instruction queue buffer.

Figure 1.7.4. Example of $\overline{\text{RD}}$ signal extended by $\overline{\text{RDY}}$ signal

(6) Hold signal

The hold signal is used to transfer the bus privileges from the CPU to the external circuits. Inputting “L” to the $\overline{\text{HOLD}}$ pin places the microcomputer in the hold state at the end of the current bus access. This status is maintained and “L” is output from the $\overline{\text{HLDA}}$ pin as long as “L” is input to the $\overline{\text{HOLD}}$ pin. Table 1.7.8 shows the microcomputer status in the hold state. The bus is used in the following descending order of priority: $\overline{\text{HOLD}}$, DMAC, CPU.

$\overline{\text{HOLD}}$ > DMAC > CPU

Figure 1.7.5. Example of $\overline{\text{RD}}$ signal extended by $\overline{\text{RDY}}$ signal

Table 1.7.8. Microcomputer status in hold state

Item		Status
Oscillation		ON
$\overline{\text{RD}}$/$\overline{\text{WR}}$ signal, address bus, data bus, $\overline{\text{CS}}$, $\overline{\text{BHE}}$		Floating
Programmable I/O ports	P0, P1, P2, P3, P4, P5 P6, P7, P8, P9, P10 P11, P12, P13, P14, P15 (Note)	Maintains status when hold signal is received
$\overline{\text{HLDA}}$		Output “L”
Internal peripheral circuits		ON (but watchdog timer stops)
ALE signal		Undefined

Note: Ports P11 to P15 exist in 144-pin version.

(7) External bus status when accessing to internal area

Table 1.7.9 shows external bus status when accessing to internal area

Table 1.7.9. External bus status when accessing to internal area

Item		SFR accessing status	Internal ROM/RAM accessing status
Address bus		Remain address of external area accessed immediately before	
Data bus	When read	Floating	
	When write	Floating	
$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$		Output "H"	
$\overline{\text{BHE}}$		Remain external area status accessed immediately before	
$\overline{\text{CS}}$		Output "H"	
ALE		ALE output	

(8) BCLK output

BCLK output can be selected by bit 7 of the processor mode register 0 (address 0004_{16} :PM07) and bit 1 and bit 0 of the system clock select register 0 (address 0006_{16} :CM01, CM00). Setting PM07 to “0” and CM01 and CM00 to “00” outputs the BCLK signal from $P5_3$. However, in single chip mode, BCLK signal is not output. When setting PM07 to “1”, the function is as set by CM01 and CM00.

(9) DRAM controller signals ($\overline{\text{RAS}}$, $\overline{\text{CASL}}$, $\overline{\text{CASH}}$, and $\overline{\text{DW}}$)

Bits 1, 2, and 3 of the DRAM control register (address 0004_{16}) select the DRAM space and enable the DRAM controller. The DRAM controller signals are then output when the DRAM area is accessed. Table 1.7.10 shows the operation of the respective signals.

Table 1.7.10. Operation of $\overline{\text{RAS}}$, $\overline{\text{CASL}}$, $\overline{\text{CASH}}$, and $\overline{\text{DW}}$ signals

Data bus width	$\overline{\text{RAS}}$	$\overline{\text{CASL}}$	$\overline{\text{CASH}}$	$\overline{\text{DW}}$	Status of external data bus
16-bit	L	L	L	H	Read data from both even and odd addresses
	L	L	L	H	Read 1 byte of data from even address
	L	H	H	H	Read 1 byte of data from odd address
	L	L	L	L	Write data to both even and odd addresses
	L	L	H	L	Write 1 byte of data to even address
	L	H	L	L	Write 1 byte of data to odd address
8-bit	L	L	Not used	H	Read 1 byte of data
	L	L	Not used	L	Write 1 byte of data

(10) Software wait

A software wait can be inserted by setting the wait control register (address 0008_{16}). Figure 1.7.6 shows wait control register

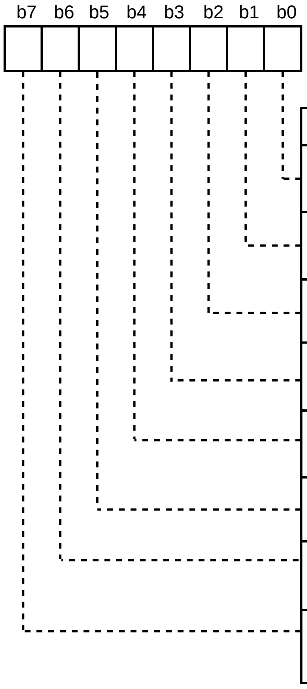
You can use the external area I wait bits (where I = 0 to 3) of the wait control register to specify from “No wait” to “3 waits” for the external memory area. When you select “No wait”, the read cycle is executed in the BCLK1 cycle. The write cycle is executed in the BCLK2 cycle (which has 1 wait). When accessing external memory using the multiplex bus, access has two waits regardless of whether you specify “No wait” or “1 wait” in the appropriate external area i wait bits in the wait control register.

Software waits in the internal memory (internal RAM and internal ROM) can be set using the internal memory wait bits of the processor mode register 1 (address 0005_{16}). Setting the internal memory wait bit = “0” sets “No wait”. Setting the internal memory wait bit = “1” specifies a wait.

The SFR area is not affected by the setting of the internal memory wait bit and is always accessed in the BCLK2 cycle.

Table 1.7.11 shows the software waits and bus cycles. Figures 1.7.7 and 1.7.8 show example bus timings when using software waits.

Wait control register



Symbol	Address	When reset
WCR	0008$_{16}$	FF$_{16}$

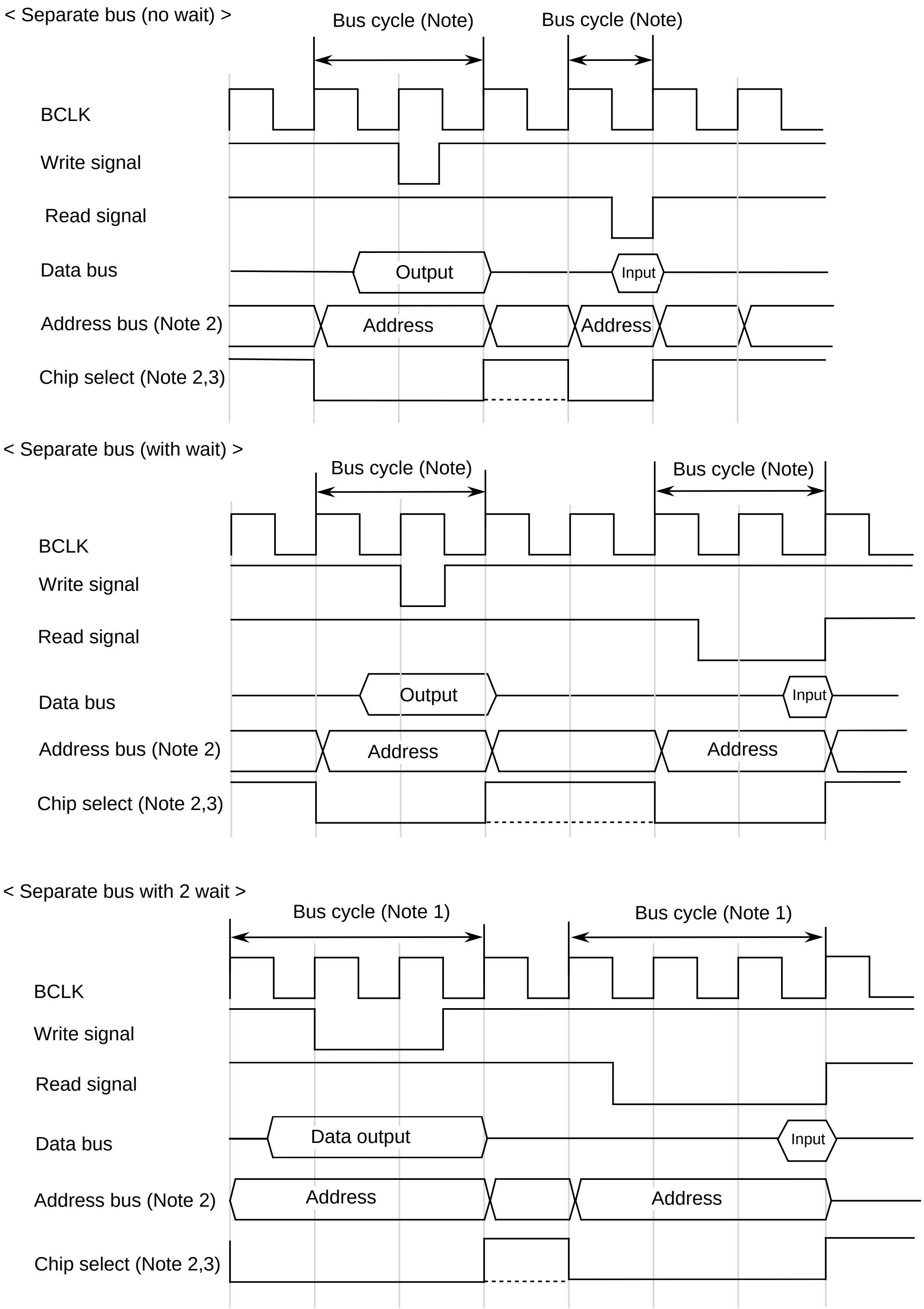
Bit symbol	Bit name	Function	R	W
WCR0	External area 0 wait bit	b1 b0 0 0: Without wait 0 1: With 1 wait 1 0: With 2 wait 1 1: With 3 wait	○	○
WCR1			○	○
WCR2	External area 1 wait bit	b3 b2 0 0: Without wait 0 1: With 1 wait 1 0: With 2 wait 1 1: With 3 wait	○	○
WCR			○	○
WCR4	External area 2 wait bit	b5 b4 0 0: Without wait 0 1: With 1 wait 1 0: With 2 wait 1 1: With 3 wait	○	○
WCR5			○	○
WCR6	External area 3 wait bit	b7 b6 0 0: Without wait 0 1: With 1 wait 1 0: With 2 wait 1 1: With 3 wait	○	○
WCR7			○	○

Note 1: When using the multiplex bus configuration, there are two waits regardless of whether you have specified "No wait" or "1 wait". However, you can specify "2 wait" or "3 wait".
Note 2: When using the separate bus configuration, the read bus cycle is executed in the BCLK1 cycle, and the write cycle is executed in the BCLK2 cycle (with 1 wait).

Figure 1.7.6. Wait control register

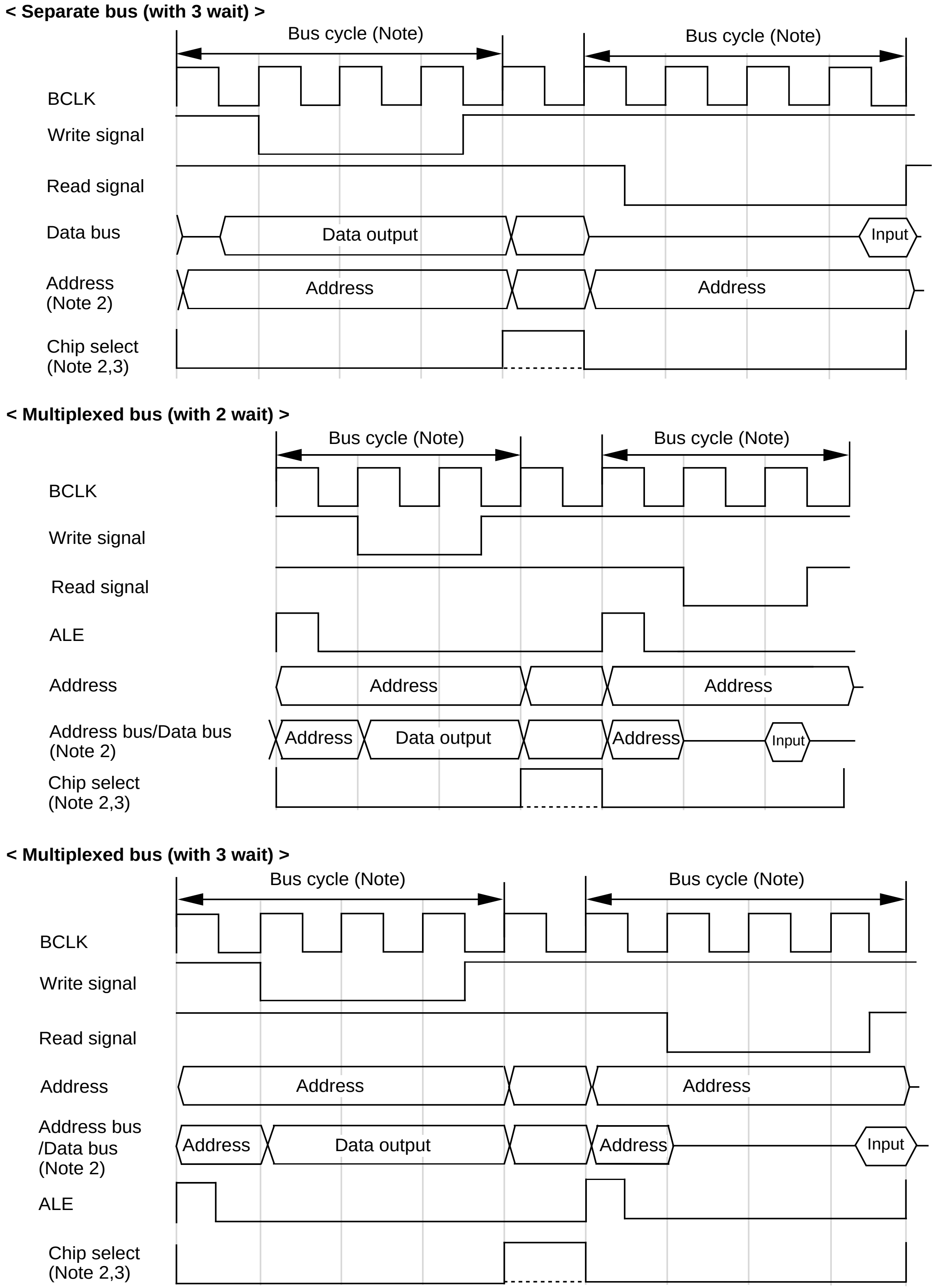
Table 1.7.11. Software waits and bus cycles

Area	Bus status	Internal memory wait bit	External memory area i wait bit	Bus cycle
SFR	———	———	———	2 BCLK cycles
Internal ROM/RAM	———	0	———	1 BCLK cycle
	———	1	———	2 BCLK cycles
External memory area	Separate bus	———	00$_2$	Read :1 BCLK cycle Write : 2 BCLK cycles
			01$_2$	2 BCLK cycles
			10$_2$	3 BCLK cycles
			11$_2$	4 BCLK cycles
	Multiplex bus	———	00$_2$	3 BCLK cycle
			01$_2$	3 BCLK cycles
			10$_2$	3 BCLK cycles
			11$_2$	4 BCLK cycles



Note 1: This timing example shows bus cycle length. Read cycle and write cycle may be continued after this bus cycle.
Note 2: Address bus and chip select may get longer by a state of CPU such as an instruction queue buffer.
Note 3: When accessing same external area (same $\overline{CS}$ area) continuously, chip select may output continuously.

Figure 1.7.7. Typical bus timings using software wait



Note 1: This timing example shows bus cycle length. Read cycle and write cycle may be continued after this bus cycle.
Note 2: Address bus and chip select may get longer by a state of CPU such as an instruction queue buffer.
Note 3: When accessing same external area (same $\overline{\mathrm{CS}}$ area) continuously, chip select may output continuously.

Figure 1.7.8. Typical bus timings using software wait

Clock Generating Circuit

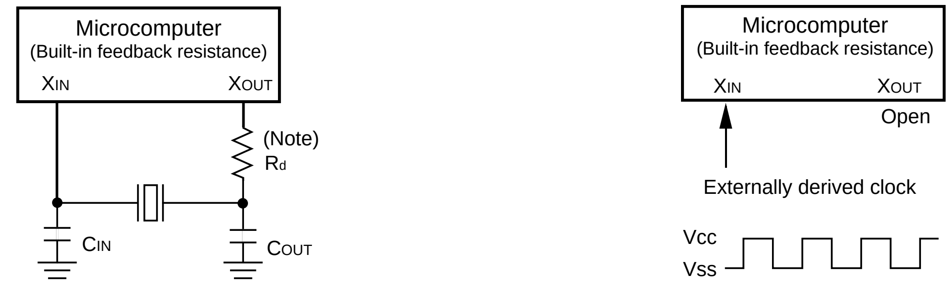
The clock generating circuit contains two oscillator circuits that supply the operating clock sources to the CPU and internal peripheral units.

Table 1.8.1. Main clock and sub clock generating circuits

	Main clock generating circuit	Sub clock generating circuit
Use of clock	• CPU's operating clock source • Internal peripheral units' operating clock source	• CPU's operating clock source • Timer A/B's count clock source
Usable oscillator	Ceramic or crystal oscillator	Crystal oscillator
Pins to connect oscillator	XIN, XOUT	XCIN, XCOUT
Oscillation stop/restart function	Available	Available
Oscillator status immediately after reset	Oscillating	Stopped
Other	Externally derived clock can be input	

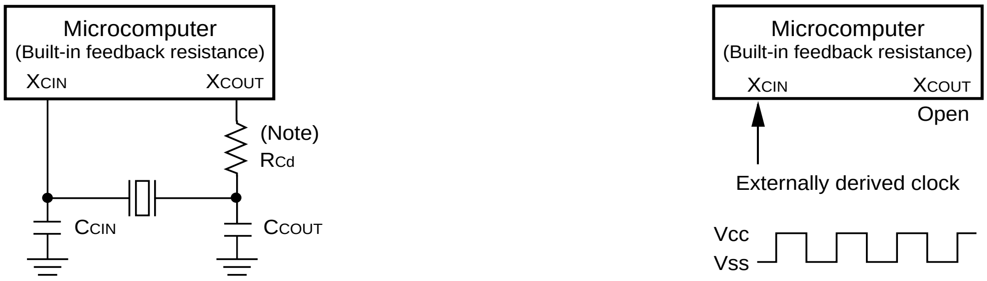
Example of oscillator circuit

Figure 1.8.1 shows some examples of the main clock circuit, one using an oscillator connected to the circuit, and the other one using an externally derived clock for input. Figure 1.8.2 shows some examples of sub clock circuits, one using an oscillator connected to the circuit, and the other one using an externally derived clock for input. Circuit constants in Figures 1.8.1 and 1.8.2 vary with each oscillator used. Use the values recommended by the manufacturer of your oscillator.



Note: Insert a damping resistance if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by the maker of the oscillator.
When the oscillation drive capacity is set to low, check that oscillation is stable.
Insert a feedback resistance between XIN and XOUT when an oscillation manufacture required.

Figure 1.8.1. Examples of main clock



Note: Insert a damping resistance if required. The resistance will vary depending on the oscillator and the oscillation drive capacity setting. Use the value recommended by the maker of the oscillator.
When the oscillation drive capacity is set to low, check that oscillation is stable.
Insert a feedback resistance between XCIN and XCOUT when an oscillation manufacture required.

Figure 1.8.2. Examples of sub clock

Clock Control

Figure 1.8.3 shows the block diagram of the clock generating circuit.

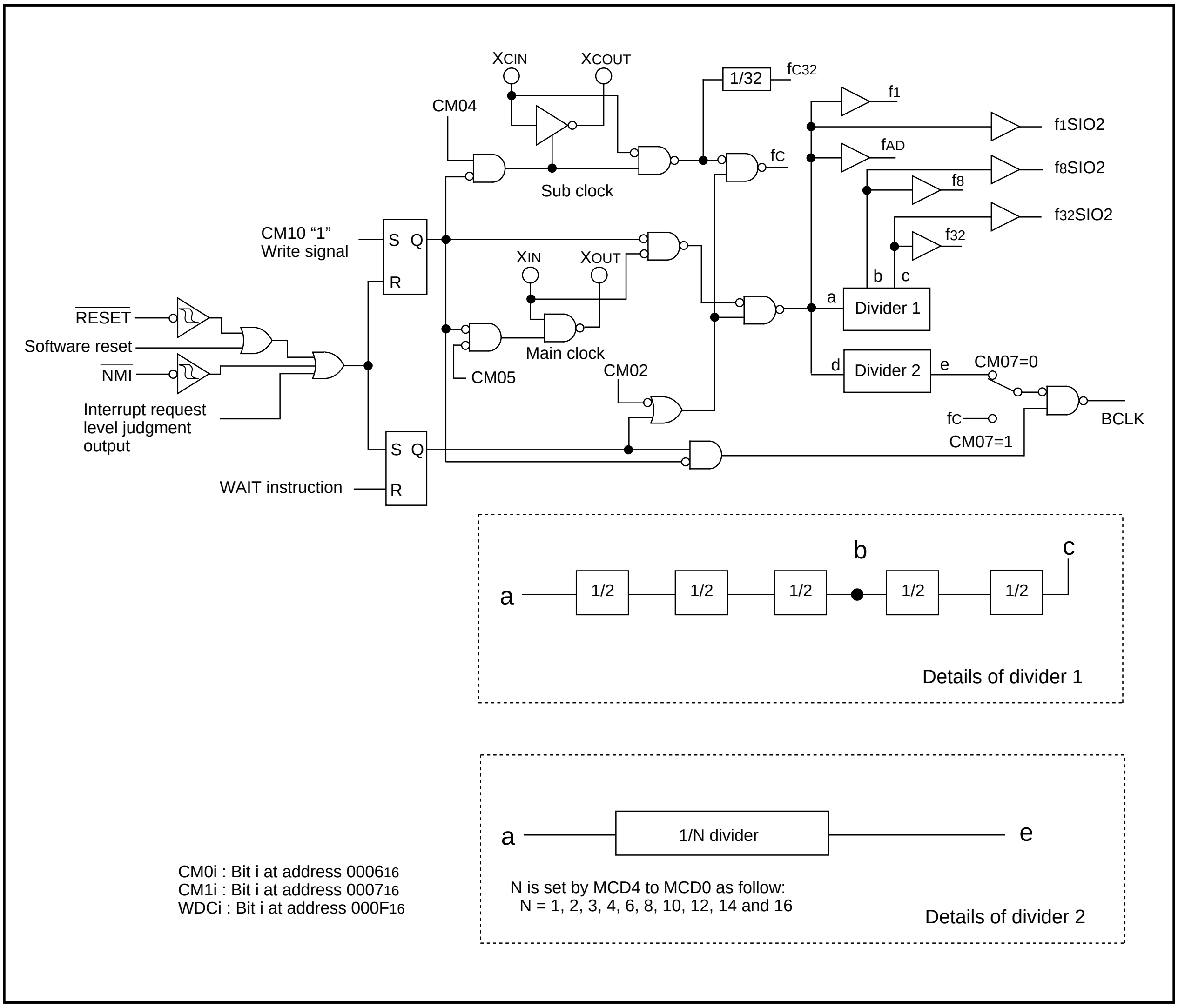


Figure 1.8.3. Clock generating circuit

The following paragraphs describes the clocks generated by the clock generating circuit.

(1) Main clock

The main clock is generated by the main clock oscillation circuit. After a reset, the clock is divided by 8 to the BCLK. The clock can be stopped using the main clock stop bit (bit 5 at address 0006_{16}). Switching to the sub clock oscillation as CPU operating clock source before stopping the clock reduces the power dissipation.

When the main clock is stoped (bit 5 at address 0006_{16} =1) or the mode is shifted to stop mode (bit 0 at address 0007_{16} =1), the main clock division register (address $000C_{16}$) is set to the division by 8 ("08_{16}").

After the oscillation of the main clock oscillation circuit has stabilized, the drive capacity of the main clock oscillation circuit can be reduced using the XIN-XOUT drive capacity select bit (bit 5 at address 0007_{16}). Reducing the drive capacity of the main clock oscillation circuit reduces the power dissipation. This bit defaults to “1” when shifting from high-speed or middle-speed mode to stop mode and after a reset. This bit remains in low-speed and low power dissipation mode.

(2) Sub clock

The sub clock is generated by the sub clock oscillation circuit. No sub clock is generated after a reset. After oscillation is started using the port Xc select bit (bit 4 at address 0006_{16}), the sub clock can be selected as the BCLK by using the system clock select bit (bit 7 at address 0006_{16}). However, be sure that the sub clock oscillation has fully stabilized before switching.

After the oscillation of the sub clock oscillation circuit has stabilized, the drive capacity of the sub clock oscillation circuit can be reduced using the XCIN-XCOUT drive capacity select bit (bit 3 at address 0006_{16}). Reducing the drive capacity of the sub clock oscillation circuit reduces the power dissipation. This bit changes to “1” when shifting to stop mode and at a reset.

When the sub clock is used, set ports $P8_6$ and $P8_7$ to no pull-up resistance with the input port.

(3) BCLK

The BCLK is the clock that drives the CPU, and is either fc or is derived by dividing the main clock by 1, 2, 3, 4, 6, 8, 10, 12, 14 or 16. The BCLK is derived by dividing the main clock by 8 after a reset.

This signal is output from BCLK pin using CM01, CM00 and PM07 in memory expansion mode and microprocessor mode.

When main clock is stoped or shifting to stop mode, the main clock division register (address $000C_{16}$) is set to the division by 8 ("08_{16}").

(4) Peripheral function clock

- f_1, f_8, f_{32}, f_{1SIO2}, f_{8SIO2}, f_{32SIO2}

The clock for the peripheral devices is derived from the main clock or by dividing it by 8 or 32. The peripheral function clock is stopped by stopping the main clock or by setting the WAIT peripheral function clock stop bit (bit 2 at 0006_{16}) to “1” and then executing a WAIT instruction.

- f_{AD}

This clock has the same frequency as the main clock and is used for A-D conversion.

(5) f_{C32}

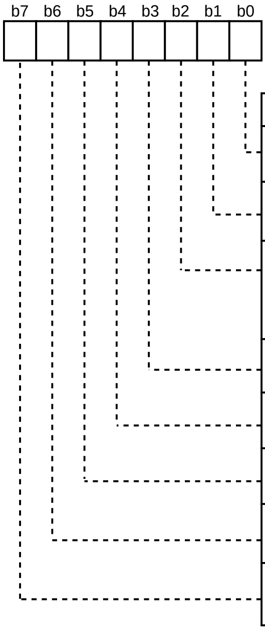
This clock is derived by dividing the sub clock by 32. It is used for the timer A and timer B counts.

(6) f_C

This clock has the same frequency as the sub clock. It is used for BCLK and for the watchdog timer.

Figure 1.8.4 shows the system clock control registers 0 and 1 and figure 1.8.5 shows main clock division register.

System clock control register 0 (Note 1)

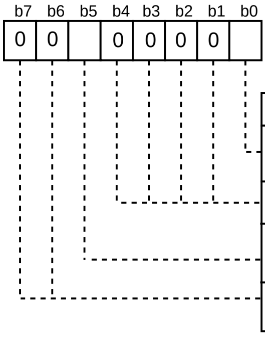


Symbol	Address	When reset
CM0	0006_{16}	08_{16}

Bit symbol	Bit	Function	R	W
CM00	Clock output function select bit (Note 2)	b1 b0 0 0 : I/O port $P5_3$ 0 1 : f_C output (Note 3) 1 0 : f_8 output (Note 3) 1 1 : f_{32} output (Note 3)	○	○
CM01			○	○
CM02	WAIT peripheral function clock stop bit	0 : Do not stop peripheral clock in wait mode 1 : Stop peripheral clock in wait mode (Note 10)	○	○
CM03	XCIN-XCOUT drive capacity select bit (Note 4)	0 : LOW 1 : HIGH	○	○
CM04	Port XC select bit	0 : I/O port 1 : XCIN-XCOUT generation (Note 11)	○	○
CM05	Main clock (XIN-XOUT) stop bit (Note 5, 6)	0 : On 1 : Off (Note 7)	○	○
CM06	Watchdog timer function select bit	0 : Watchdog timer interrupt 1 : Reset (Note 8)	○	○
CM07	System clock select bit (Note 9)	0 : XIN, XOUT 1 : XCIN, XCOUT	○	○

Note 1: Set bit 0 of the protect register (address $000A_{16}$) to “1” before writing to this register.
Note 2: When outputting BCLK (bit 7 of processor mode register 0 is "0"), set these bits to "00". When outputting ALE to $P5_3$ (bit 5 and 4 of processor mode register 0 is "01"), set these bits to "00". The port $P5_3$ function is not selected even when you set "00" in microprocessor or memory expansion mode and bit 7 of the processor mode register 0 is "1".
Note 3: When selecting f_C, f_8 or f_{32} in single chip mode, must use $P5_7$ as input port.
Note 4: Changes to “1” when shifting to stop mode or reset.
Note 5: When entering the power saving mode, the main clock is stopped using this bit. To stop the main clock, set system clock stop bit (CM07) to "1" while an oscillation of sub clock is stable. Then set this bit to "1".
Note 6: When this bit is "1", XOUT is "H". Also, the internal feedback resistance remains ON, so XIN is pulled up to XOUT ("H" level) via the feedback resistance.
Note 7: When the main clock is stopped, the main clock division register (address $000C_{16}$) is set to the division by 8 mode.
Note 8: When "1" has been set once, "0" cannot be written by software.
Note 9: To set CM07 "1" from "0", first set CM04 to "1", and an oscillation of sub clock is stable. Then set CM07. Do not set CM04 and CM07 simultaneously. Also, to set CM07 "0" from "1", first set CM05 to "1", and an oscillation of main clock is stable. Then set CM07.
Note 10: f_{C32} is not included.
Note 11: When XCIN-XCOUT is used, set port $P8_6$ and $P8_7$ to no pull-up resistance with the input port.

System clock control register 1 (Note 1)



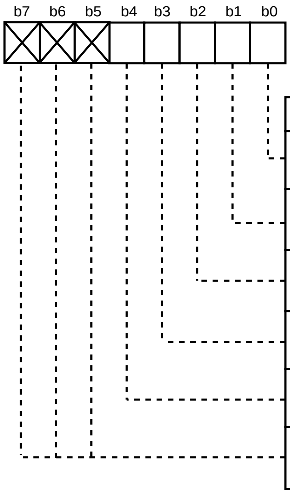
Symbol	Address	When reset
CM1	0007_{16}	20_{16}

Bit symbol	Bit	Function	R	W
CM10	All clock stop control bit (Note 3)	0 : Clock on 1 : All clocks off (stop mode) (Note 4)	○	○
Reserved bit		Always set to “0”	○	○
CM15	XIN-XOUT drive capacity select bit (Note 2)	0 : LOW 1 : HIGH	○	○
Reserved bit		Always set to “0”	○	○

Note 1: Set bit 0 of the protect register (address $000A_{16}$) to “1” before writing to this register.
Note 2: Changes to “1” when shifting from high-speed or middle-speed mode to stop mode or reset. This bit is remained in low speed or low power dissipation mode.
Note 3: When this bit is "1", XOUT is "H", and the internal feedback resistance is disabled. XCIN and XCOUT are high-inpedance.
Note 4: When the main clock is stopped, the main clock division register (address $000C_{16}$) is set to the division by 8 mode.

Figure 1.8.4. System clock control registers 0 and 1

Main clock division register (Note 1)



Symbol	Address	When reset
MCD	000C$_{16}$	XXX01000$_2$

Bit symbol	Bit name	Function	R	W
MCD0	Main clock division select bit (Note 2)	b4 b3 b2 b1 b0 1 0 0 1 0 : No division mode 0 0 0 1 0 : Division by 2 mode 0 0 0 1 1 : Division by 3 mode 0 0 1 0 0 : Division by 4 mode 0 0 1 1 0 : Division by 6 mode 0 1 0 0 0 : Division by 8 mode 0 1 0 1 0 : Division by 10 mode 0 1 1 0 0 : Division by 12 mode 0 1 1 1 0 : Division by 14 mode 0 0 0 0 0 : Division by 16 mode	○	○
MCD1			○	○
MCD2			○	○
MCD3			○	○
MCD4			○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—

Note 1: Set bit 0 of the protect register (address 000A$_{16}$) to “1” before writing to this register.
Note 2: These bits are "01000$_2$" (8-division mode) when main clock is stopped or you shift to stop mode.
Note 3: Do not attempt to set combinations of values other than those shown in this figure.

Figure 1.8.5. Main clock division register

Clock Output

In single chip mode, when the BCLK output function select bit (bit 7 at address 0004$_{16}$:PM07) is “1”, you can output f$_8$, f$_{32}$, or fc from the P5$_3$/BCLK/ALE/CLK$_{OUT}$ pins by setting the clock output function select bits (bits 1 and 0 at address 0006$_{16}$:CM01, CM00).(Note)

Even when you set PM07 to “0” and CM01 and CM00 to “00$_2$”, no BCLK is output.

In memory expansion mode or microprocessor mode, when the ALE pin select bits (bits 5 and 4 at address 0005$_{16}$:PM15, PM14) are other than “01$_2$(P5$_3$/BCLK)” and PM07 is “1”, you can output f$_8$, f$_{32}$, or fc from the P5$_3$/BCLK/ALE/CLK$_{OUT}$ pins by setting CM01 and CM00.

In memory expansion mode or microprocessor mode, when PM15 and PM14 are other than “01$_2$(P5$_3$/BCLK)” and PM07 is “0” and CM01 and CM00 to “00$_2$”, BCLK is output from the P5$_3$/BCLK/ALE/CLK$_{OUT}$ pins.

When stopping clock output in memory expansion mode or microprocessor mode, set PM07 to “1” and CM01 and CM00 to “00$_2$” (IO port P5$_3$). The P5$_3$ function is not selected. When PM15 and PM14 are “01$_2$ (P5$_3$/BCLK)” and CM01 and CM00 are “00$_2$”, PM07 is ignored and the P5$_3$ pin is set for ALE output.

When the WAIT peripheral function clock stop bit (bit 2 at address 0006$_{16}$) is set to “1”, f$_8$ or f$_{32}$ clock output is stopped when a WAIT command is executed.

Table 1.8.2 shows clock output setting (single chip mode) and Table 1.8.3 shows clock output setting (memory expansion/microprocessor mode).

Note :When outputting the f$_8$, f$_{32}$ or fc from port P5$_3$/BCLK/ALE/CLK$_{OUT}$ pin in single chip mode, use port P5$_7$/$\overline{\text{RDY}}$ as an input only port.

Table 1.8.2. Clock output setting (single chip mode)

BCLK output function select bit	Clock output function select bit		ALE pin select bit		P5$_3$/BCLK/ALE/CLK$_{OUT}$ pin function
PM07	CM01	CM00	PM15	PM14	
0/1	0	0	Ignored	Ignored	P5$_3$ I/O port
1	0	1	Ignored	Ignored	fc output (Note)
1	1	0	Ignored	Ignored	f$_8$ output (Note)
1	1	1	Ignored	Ignored	f$_{32}$ output (Note)

Note :Must use P5$_7$ as input port.

Table 1.8.3. Clock output setting (memory expansion/microprocessor mode)

BCLK output function select bit	Clock output function select bit		ALE pin select bit		P5$_3$/BCLK/ALE/CLK$_{OUT}$ pin function
PM07	CM01	CM00	PM15	PM14	
0	0	0	0 1 1	0 0 1	BCLK output
1	0	0			"L" output (not P5$_3$)
1	0	1			fc output
1	1	0			f$_8$ output
1	1	1			f$_{32}$ output
Ignored	0	0	0	1	ALE output

Stop Mode

Writing “1” to the all-clock stop control bit (bit 0 at address 0007$_{16}$) stops all oscillation and the microcomputer enters stop mode. In stop mode, the content of the internal RAM is retained provided that V$_{CC}$ remains above 2V.

Because the oscillation of BCLK, f$_1$ to f$_{32}$, f$_{1SIO2}$ to f$_{32SIO2}$, fc, fc$_{32}$, and f$_{AD}$ stops in stop mode, peripheral functions such as the A-D converter and watchdog timer do not function. However, timer A and timer B operate provided that the event counter mode is set to an external pulse, and UARTi(i = 0 to 2) functions provided an external clock is selected. Table 1.8.4 shows the status of the ports in stop mode.

Stop mode is cancelled by a hardware reset or interrupt.

When using an interrupt to exit stop mode, the relevant interrupt must have been enabled and set to a priority level above the level set by the interrupt priority set bits (bits 2, 1, and 0 at address 009F$_{16}$) for exiting a stop/wait state. Set the interrupt priority set bits for the exit from a stop/wait state to the same level as the flag register (FLG) processor interrupt level (IPL). Figure 1.8.6 shows the exit priority register.

When exiting stop mode using an interrupt, the relevant interrupt routine is executed.

Although stop mode is cancelled by hardware reset only, the interrupt enable flag (I flag) must be set to "1".

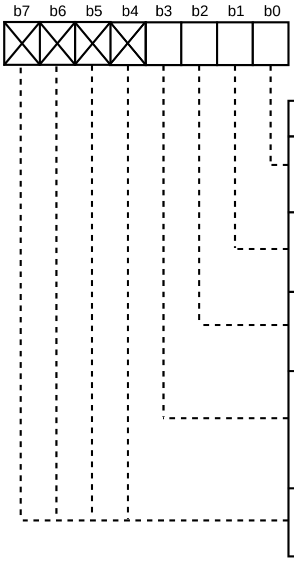
When shifting to stop mode and reset, the main clock division register (000C$_{16}$) is set to “08$_{16}$”.

Table 1.8.4. Port status during stop mode

Pin		Memory expansion mode Microprocessor mode	Single-chip mode
Address bus, data bus, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{BHE}}$		Retains status before stop mode	
$\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\text{DW}}$, $\overline{\text{CASL}}$, $\overline{\text{CASH}}$		“H” (Note)	
$\overline{\text{RAS}}$		“H” (Note)	
$\overline{\text{HLDA}}$, BCLK		“H”	
ALE		“H”	
Port		Retains status before stop mode	Retains status before stop mode
CLKOUT	When f_c selected	“H”	“H”
	When f_8, f_{32} selected	Retains status before stop mode	Retains status before stop mode

Note :When self-refresh is done in operating DRAM control, $\overline{\text{CAS}}$ and $\overline{\text{RAS}}$ becomes “L”.

Exit priority register



Symbol RLVL   Address 009F$_{16}$ When reset XXXX0000$_2$

Bit symbol	Bit	Function	R	W
RLVL0	Interrupt priority set bit for exiting Stop/Wait state (Note 1,2)	b2 b1 b0 0 0 0 : Level 0 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	○	○
RLVL1			○	○
RLVL2			○	○
FSIT	High-speed interrupt set bit (Note 3)	0: Interrupt priority level 7 = normal interrupt 1: Interrupt priority level 7 = high-speed interrupt	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			–	–

Note 1: Exits the Stop or Wait mode when the requested interrupt priority level is higher than that set in the exit priority register.
Note 2: Set to the same value as the processor interrupt priority level (IPL) set in the flag register (FLG).
Note 3: The high-speed interrupt can only be specified for interrupts with interrupt priority level 7. Specify interrupt priority level 7 for only one interrupt.

Figure 1.8.6. Exit priority register

Wait Mode

When a WAIT instruction is executed, the BCLK stops and the microcomputer enters the wait mode. In this mode, oscillation continues but the BCLK and watchdog timer stop. Writing “1” to the WAIT peripheral function clock stop bit and executing a WAIT instruction stops the clock being supplied to the internal peripheral functions, allowing power dissipation to be reduced. Table 1.8.5 shows the status of the ports in wait mode.

Wait mode is cancelled by a hardware reset or interrupt. If an interrupt is used to cancel wait mode, the microcomputer restarts using as BCLK the clock that had been selected when the WAIT instruction was executed.

When using an interrupt to exit Wait mode, the relevant interrupt must have been enabled and set to a priority level above the level set by the interrupt priority set bits for exiting a stop/wait state (bits 2, 1, and 0 at address 009F$_{16}$). Set the interrupt priority set bits for the exit from a stop/wait state to the same level as the flag register (FLG) processor interrupt level (IPL).

When using an interrupt to exit Wait mode, the microcomputer resumes operating the clock that was operating when the WAIT command was executed as BCLK from the interrupt routine.

Table 1.8.5. Port status during wait mode

Pin		Memory expansion mode Microprocessor mode	Single-chip mode
Address bus, data bus, $\overline{CS0}$ to $\overline{CS3}$, $\overline{BHE}$		Retains status before wait mode	
$\overline{RD}$, $\overline{WR}$, $\overline{WRL}$, $\overline{WRH}$, $\overline{DW}$, $\overline{CASL}$, $\overline{CASH}$		“H” (Note)	
$\overline{RAS}$		“H” (Note)	
$\overline{HLDA}$,BCLK		“H”	
ALE		“L”	
Port		Retains status before wait mode	Retains status before wait mode
CLK$_{OUT}$	When f$_C$ selected	Does not stop	
	When f_8, f_{32} selected	Does not stop when the WAIT peripheral function clock stop bit is “0”. When the WAIT peripheral function clock stop bit is “1”, the status immediately prior to entering wait mode is maintained.	

Note :When self-refresh is done in operating DRAM control, $\overline{CAS}$ and $\overline{RAS}$ becomes “L”.

Status Transition of BCLK

Power dissipation can be reduced and low-voltage operation achieved by changing the count source for BCLK. Table 1.8.6 shows the operating modes corresponding to the settings of system clock control registers 0 and main clock division register.

After a reset, operation defaults to division by 8 mode. When shifting to stop mode, reset or stopping main clock, the main clock division register (address 000C$_{16}$) is set to “08$_{16}$”.

(1) Division by 2 mode

The main clock is divided by 2 to obtain the BCLK.

(2) Division by 3 mode

The main clock is divided by 3 to obtain the BCLK.

(3) Division by 4 mode

The main clock is divided by 4 to obtain the BCLK.

(4) Division by 6 mode

The main clock is divided by 6 to obtain the BCLK.

(5) Division by 8 mode

The main clock is divided by 8 to obtain the BCLK. After reset, this mode is executed. Note that oscillation of the main clock must have stabilized before transferring from this mode to no-division, division by 2, 6, 10, 12, 14 and 16 mode.

Oscillation of the sub clock must have stabilized before transferring to low-speed and low power dissipation mode.

(6) Division by 10 mode

The main clock is divided by 10 to obtain the BCLK.

(7) Division by 12 mode

The main clock is divided by 12 to obtain the BCLK.

(8) Division by 14 mode

The main clock is divided by 14 to obtain the BCLK.

(9) Division by 16 mode

The main clock is divided by 16 to obtain the BCLK.

(10) No-division mode

The main clock is divided by 1 to obtain the BCLK.

(11) Low-speed mode

f$_C$ is used as BCLK. Note that oscillation of both the main and sub clocks must have stabilized before transferring from this mode to another or vice versa. At least 2 to 3 seconds are required after the sub clock starts. Therefore, the program must be written to wait until this clock has stabilized immediately after powering up and after stop mode is cancelled.

(12) Low power dissipation mode

f$_C$ is the BCLK and the main clock is stopped.

When the main clock is stoped, the main clock division register (address 000C$_{16}$) is set to the division by 8 mode.

Note: When count source of BCLK is changed from clock A to clock B (X_{IN} to X_{CIN} or X_{CIN} to X_{IN}), clock B needs to be stable before changing. Please wait to change modes until after oscillation has stabilized.

Table 1.8.6. Operating modes dictated by settings of system clock control register 0 and main clock division register

CM07	CM05	CM04	MCD4	MCD3	MCD2	MCD1	MCD0	Operating mode of BCLK
0	0	Invalid	1	0	0	1	0	No division
0	0	Invalid	0	0	0	1	0	Division by 2 mode
0	0	Invalid	0	0	0	1	1	Division by 3 mode
0	0	Invalid	0	0	1	0	0	Division by 4 mode
0	0	Invalid	0	0	1	1	0	Division by 6 mode
0	0	Invalid	0	1	0	0	0	Division by 8 mode
0	0	Invalid	0	1	0	1	0	Division by 10 mode
0	0	Invalid	0	1	1	0	0	Division by 12 mode
0	0	Invalid	0	1	1	1	0	Division by 14 mode
0	0	Invalid	0	0	0	0	0	Division by 16 mode
1	0	1	Invalid	Invalid	Invalid	Invalid	Invalid	Low-speed mode
1	1	1	Invalid	Invalid	Invalid	Invalid	Invalid	Low power dissipation mode

CM0i: Clock control register 0 (address 0006_{16}) bit i

MCDi: Main clock division register (address $000C_{16}$) bit i

Power Saving

In Power Save modes, the CPU and oscillator stop and the operating clock is slowed to minimize power dissipation by the CPU. The following outlines the Power Save modes.
There are three power save modes.

(1) Normal operating mode

- **High-speed mode**

  In this mode, one main clock cycle forms BCLK. The CPU operates on the selected internal clock. The peripheral functions operate on the clocks specified for each respective function.

- **Medium-speed mode**

  In this mode, the main clock is divided into 2, 3, 4, 6, 8, 10, 12, 14, or 16 to form BCLK. The CPU operates on the selected internal clock. The peripheral functions operated on the clocks specified for each respective function.

- **Low-speed mode**

  In this mode, fc forms BCLK. The CPU operates on the fc clock. fc is the clock supplied by the subclock. The peripheral functions operate on the clocks specified for each respective function.

- **Low power-dissipation mode**

  This mode is selected when the main clock is stopped from low-speed mode. The CPU operates on the fc clock. fc is the clock supplied by the subclock. Only the peripheral functions for which the subclock was selected as the count source continue to run.

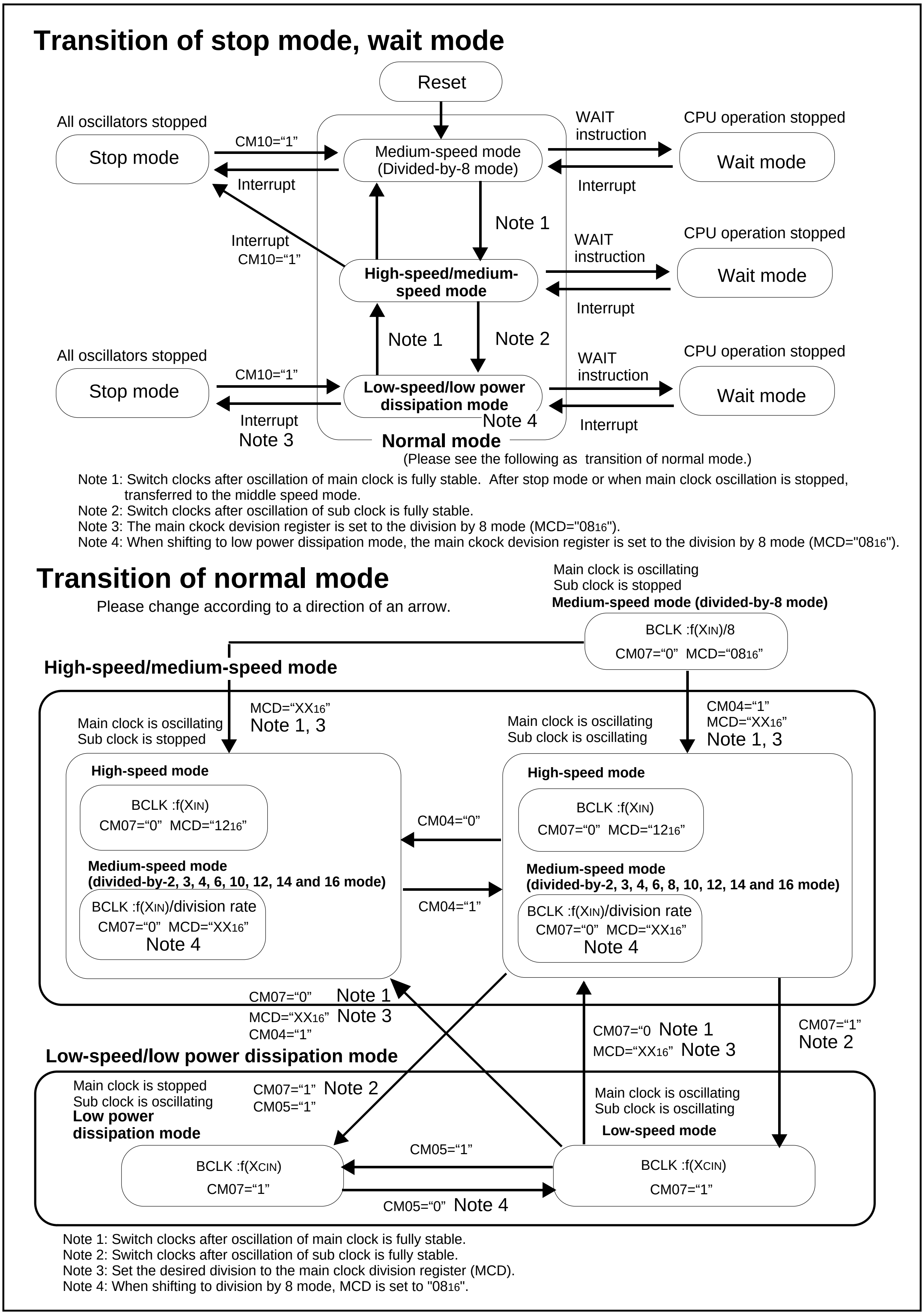
(2) Wait mode

CPU operation is halted in this mode. The oscillator continues to run.

(3) Stop mode

All oscillators stop in this mode. The CPU and internal peripheral functions all stop. Of all 3 power saving modes, power savings are greatest in this mode.

Figure 1.8.7 shows the clock transition between each of the three modes, (1), (2), and (3).



Note 1: Switch clocks after oscillation of main clock is fully stable. After stop mode or when main clock oscillation is stopped, transferred to the middle speed mode.
Note 2: Switch clocks after oscillation of sub clock is fully stable.
Note 3: The main ckock devision register is set to the division by 8 mode (MCD="08_{16}").
Note 4: When shifting to low power dissipation mode, the main ckock devision register is set to the division by 8 mode (MCD="08_{16}").

Note 1: Switch clocks after oscillation of main clock is fully stable.
Note 2: Switch clocks after oscillation of sub clock is fully stable.
Note 3: Set the desired division to the main clock division register (MCD).
Note 4: When shifting to division by 8 mode, MCD is set to "08_{16}".

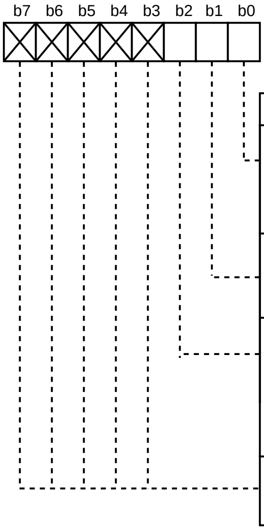
Figure 1.8.7. Clock transition

Protection

The protection function is provided so that the values in important registers cannot be changed in the event that the program runs out of control. Figure 1.8.8 shows the protect register. The values in the processor mode register 0 (address 0004_{16}), processor mode register 1 (address 0005_{16}), system clock control register 0 (address 0006_{16}), system clock control register 1 (address 0007_{16}), main clock division register (address $000C_{16}$), port P9 direction register (address $03C7_{16}$) and function select register A3 (address $03B5_{16}$) can only be changed when the respective bit in the protect register is set to “1”. Therefore, important outputs can be allocated to port P9.

If, after “1” (write-enabled) has been written to the PRC2 (bit 2 at address $000A_{16}$), a value is written to any address, the bit automatically reverts to “0” (write-inhibited). Change port P9 input/output and function select register A3 immediately after setting "1" to PRC2. Interrupt and DMA transfer should not be inserted between instructions. However, the PRC0 (bit 0 at address $000A_{16}$) and PRC1 (bit 1 at address $000A_{16}$) do not automatically return to “0” after a value has been written to an address. The program must therefore be written to return these bits to “0”.

Protect register



Symbol	Address	When reset
PRCR	$000A_{16}$	$XXXXX000_2$

Bit symbol	Bit name	Function	R	W
PRC0	Enables writing to system clock control registers 0 and 1 (addresses 0006_{16} and 0007_{16}) and main clock division register (address $000C_{16}$)	0 : Write-inhibited 1 : Write-enabled	○	○
PRC1	Enables writing to processor mode registers 0 and 1 (addresses 0004_{16} and 0005_{16})	0 : Write-inhibited 1 : Write-enabled	○	○
PRC2	Enables writing to port P9 direction register (address $03C7_{16}$) and function select register A3 (address $03B5_{16}$) (Note)	0 : Write-inhibited 1 : Write-enabled	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			–	–

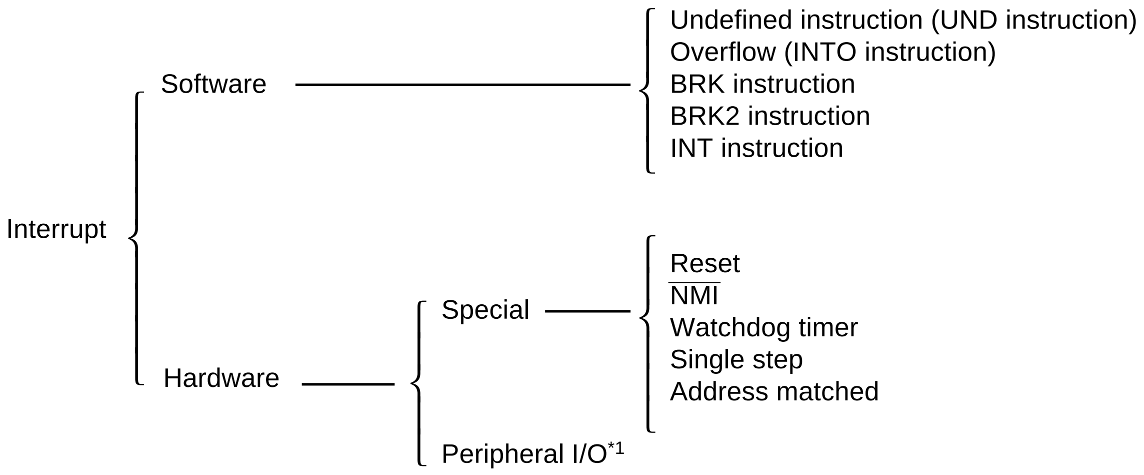
Note: Writing a value to an address after “1” is written to this bit returns the bit to “0”. Other bits do not automatically return to “0” and they must therefore be reset by the program.

Figure 1.8.8. Protect register

Interrupt Outline

Types of Interrupts

Figure 1.9.1 lists the types of interrupts.



*1 Peripheral I/O interrupts are generated by the peripheral functions built into the microcomputer system. High-speed interrupt can be used as highest priority in peripheral I/O interrupts.

Figure 1.9.1. Classification of interrupts

- Maskable interrupt : An interrupt which can be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority **can be changed** by priority level.
- Non-maskable interrupt : An interrupt which cannot be enabled (disabled) by the interrupt enable flag (I flag) or whose interrupt priority **cannot be changed** by priority level.

Software Interrupts

Software interrupts are generated by some instruction that generates an interrupt request when executed. Software interrupts are nonmaskable interrupts.

(1) Undefined-instruction interrupt

This interrupt occurs when the UND instruction is executed.

(2) Overflow interrupt

This interrupt occurs if the INTO instruction is executed when the O flag is 1.

The following lists the instructions that cause the O flag to change:

ABS, ADC, ADCF, ADD, ADDX, CMP, CMPX, DIV, DIVU, DIVX, NEG, RMPA, SBB, SCMPU, SHA, SUB, SUBX

(3) BRK interrupt

This interrupt occurs when the BRK instruction is executed.

(4) BRK2 interrupt

This interrupt occurs when the BRK2 instruction is executed. This interrupt is used exclusively for debugger purposes. You normally do not need to use this interrupt.

(5) INT instruction interrupt

This interrupt occurs when the INT instruction is executed after specifying a software interrupt number from 0 to 63. Note that software interrupt numbers 0 to 43 are assigned to peripheral I/O interrupts. This means that by executing the INT instruction, you can execute the same interrupt routine as used in peripheral I/O interrupts.

The stack pointer used in INT instruction interrupt varies depending on the software interrupt number. For software interrupt numbers 0 to 31, the U flag is saved when an interrupt occurs and the U flag is cleared to 0 to choose the interrupt stack pointer (ISP) before executing the interrupt sequence. The previous U flag before the interrupt occurred is restored when control returns from the interrupt routine. For software interrupt numbers 32 to 63, such stack pointer switchover does not occur.

However, in peripheral I/O interrupts, the U flag is saved when an interrupt occurs and the U flag is cleared to 0 to choose ISP.

Therefore movement of U flag is different by peripheral I/O interrupt or INT instruction in software interrupt number 32 to 43.

Hardware Interrupts

There are Two types in hardware Interrupts; special interrupts and Peripheral I/O interrupts.

(1) Special interrupts

Special interrupts are nonmaskable interrupts.

- Reset

  A reset occurs when the $\overline{\text{RESET}}$ pin is pulled low.

- **$\overline{\text{NMI}}$ interrupt**

  This interrupt occurs when the $\overline{\text{NMI}}$ pin is pulled low.

- **Watchdog timer interrupt**

  This interrupt is caused by the watchdog timer.

- **Address-match interrupt**

  This interrupt occurs immediately before the instruction at the address indicated by the address match interrupt register is executed while the address match interrupt enable bit is set to “1”.

  This interrupt does not occur if any address other than the start address of an instruction is set in the address match register.

- **Single-step interrupt**

  This interrupt is used exclusively for debugger purposes, do not use it in other circumstances. A single-step interrupt occurs when the D flag is set (= 1); in this case, an interrupt is generated after one instruction is executed.

(2) Peripheral I/O interrupts

A peripheral I/O interrupt is generated by one of built-in peripheral functions. Built-in peripheral functions are dependent on classes of products, so the interrupt factors too are dependent on classes of products. The interrupt vector table is the same as the one for software interrupt numbers 0 through 43 the INT instruction uses. Peripheral I/O interrupts are maskable interrupts.

- **Bus collision detection, start/stop condition detection interrupts (UART2, UART3, UART4), fault error interrupts (UART3, 4)**

  This is an interrupt that the serial I/O bus collision detection generates. When I^{2}C mode is selected, start, stop condition interrupt is selected. When $\overline{\text{SS}}$ pin is selected, fault error interrupt is selected.

- **DMA0 through DMA3 interrupts**

  These are interrupts that DMA generates.

- **Key-input interrupt**

  A key-input interrupt occurs if an “L” is input to the $\overline{\text{KI}}$ pin.

- **A-D conversion interrupt**

  This is an interrupt that the A-D converter generates.

- **UART0, UART1, UART2/NACK, UART3/NACK and UART4/NACK transmission interrupt**

  These are interrupts that the serial I/O transmission generates.

- **UART0, UART1, UART2/ACK, UART3/ACK and UART4/ACK reception interrupt**

  These are interrupts that the serial I/O reception generates.

- **Timer A0 interrupt through timer A4 interrupt**

  These are interrupts that timer A generates

- **Timer B0 interrupt through timer B5 interrupt**

  These are interrupts that timer B generates.

- **$\overline{\text{INT0}}$ interrupt through $\overline{\text{INT5}}$ interrupt**

  An $\overline{\text{INT}}$ interrupt selects a edge sense or a level sense. In edge sense, an $\overline{\text{INT}}$ interrupt occurs if either a rising edge or a falling edge or a both edge is input to the $\overline{\text{INT}}$ pin. In level sense, an $\overline{\text{INT}}$ interrupt occurs if either an "H" level or an "L" level is input to the $\overline{\text{INT}}$ pin.

High-speed interrupts

High-speed interrupts are interrupts in which the response is executed at 5 cycles and the return is 3 cycles.

When a high-speed interrupt is received, the flag register (FLG) and program counter (PC) are saved to the save flag register (SVF) and save PC register (SVP) and the program is executed from the address shown in the vector register (VCT).

Execute a FREIT instruction to return from the high-speed interrupt routine.

High-speed interrupts can be set by setting “1” in the high-speed interrupt specification bit allocated to bit 3 of the exit priority register. Setting “1” in the high-speed interrupt specification bit makes the interrupt set to level 7 in the interrupt control register into a high-speed interrupt.

You can only set one interrupt as a high-speed interrupt. When using a high-speed interrupt, do not set multiple interrupts as level 7 interrupts.

The interrupt vector for a high-speed interrupt must be set in the vector register (VCT).

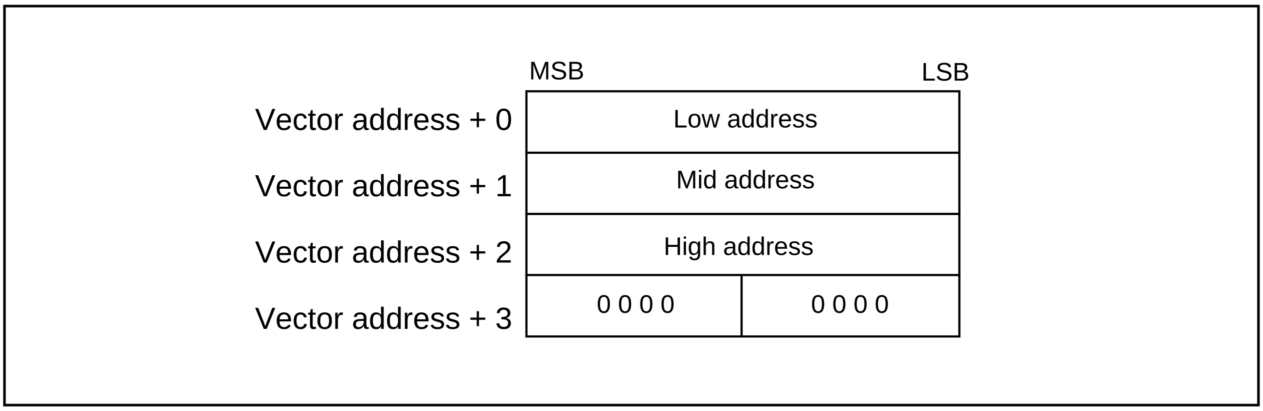
When using a high-speed interrupt, you can use a maximum of two DMAC channels.

The execution speed is improved when register bank 1 is used with high speed interrupt register selected by not saving registers to the stack but to the switching register bank. In this case, switch register bank mode for high-speed interrupt routine.

Interrupts and Interrupt Vector Tables

If an interrupt request is accepted, a program branches to the interrupt routine set in the interrupt vector table. Set the first address of the interrupt routine in each vector table. Figure 1.9.2 shows the format for specifying the address.

Two types of interrupt vector tables are available — fixed vector table in which addresses are fixed and variable vector table in which addresses can be varied by the setting.



	MSB	LSB
Vector address + 0	Low address	
Vector address + 1	Mid address	
Vector address + 2	High address	
Vector address + 3	0 0 0 0	0 0 0 0

Figure 1.9.2. Format for specifying interrupt vector addresses

- **Fixed vector tables**

  The fixed vector table is a table in which addresses are fixed. The vector tables are located in an area extending from FFFFDC$_{16}$ to FFFFFF$_{16}$. One vector table comprises four bytes. Set the first address of interrupt routine in each vector table. Table 1.9.1 shows the interrupts assigned to the fixed vector tables and addresses of vector tables.

Table 1.9.1. Interrupt factors (fixed interrupt vector addresses)

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks
Undefined instruction	FFFFDC$_{16}$ to FFFFDF$_{16}$	Interrupt on UND instruction
Overflow	FFFFE0$_{16}$ to FFFFE3$_{16}$	Interrupt on INTO instruction
BRK instruction execution	FFFFE4$_{16}$ to FFFFE7$_{16}$	If content of FFFFE7$_{16}$ is filled with FF$_{16}$, program starts from the address shown by the vector in the variable vector table
Address match	FFFFE8$_{16}$ to FFFFEB$_{16}$	There is an address-matching interrupt enable bit
Watchdog timer	FFFFF0$_{16}$ to FFFFF3$_{16}$	
$\overline{\text{NMI}}$	FFFFF8$_{16}$ to FFFFFB$_{16}$	External interrupt by input to $\overline{\text{NMI}}$ pin
Reset	FFFFFC$_{16}$ to FFFFFF$_{16}$	

- **Vector table dedicated for emulator**

  Table 1.9.2 shows interrupt vector address which is vector table register dedicated for emulator (address 000020$_{16}$ to 000022$_{16}$). These instructions are not effected with interrupt enable flag (I flag) (non maskable interrupt).

  This interrupt is used exclusively for debugger purposes. You normally do not need to use this interrupt. Do not access to the interrupt vector table register dedicated for emulator (address 000020$_{16}$ to 000022$_{16}$).

Table 1.9.2. Interrupt vector table register for emulator

Interrupt source	Vector table addresses Address (L) to address (H)	Remarks
BRK2 instruction	Interrupt vector table register for emulator 000020$_{16}$ to 000022$_{16}$	Interrupt for debugger
Single step	Interrupt vector table register for emulator 000020$_{16}$ to 000022$_{16}$	Interrupt for debugger

- **Variable vector tables**

  The addresses in the variable vector table can be modified, according to the user’s settings. Indicate the first address using the interrupt table register (INTB). The 256-byte area subsequent to the address the INTB indicates becomes the area for the variable vector tables. One vector table comprises four bytes. Set the first address of the interrupt routine in each vector table. Table 1.9.3 shows the interrupts assigned to the variable vector tables and addresses of vector tables.

  Set an even address to the start address of vector table setting in INTB so that operating efficiency is increased.

Table 1.9.3. Interrupt causes (variable interrupt vector addresses)

Software interrupt number	Vector table address Address (L) to address (H)	Interrupt source	Remarks
Software interrupt number 0	+0 to +3 (Note 1)	BRK instruction	Cannot be masked I flag
Software interrupt number 8	+32 to +35 (Note 1)	DMA0	
Software interrupt number 9	+36 to +39 (Note 1)	DMA1	
Software interrupt number 10	+40 to +43 (Note 1)	DMA2	
Software interrupt number 11	+44 to +47 (Note 1)	DMA3	
Software interrupt number 12	+48 to +51 (Note 1)	Timer A0	
Software interrupt number 13	+52 to +55 (Note 1)	Timer A1	
Software interrupt number 14	+56 to +59 (Note 1)	Timer A2	
Software interrupt number 15	+60 to +63 (Note 1)	Timer A3	
Software interrupt number 16	+64 to +67 (Note 1)	Timer A4	
Software interrupt number 17	+68 to +71 (Note 1)	UART0 transmit	
Software interrupt number 18	+72 to +75 (Note 1)	UART0 receive	
Software interrupt number 19	+76 to +79 (Note 1)	UART1 transmit	
Software interrupt number 20	+80 to +83 (Note 1)	UART1 receive	
Software interrupt number 21	+84 to +87 (Note 1)	Timer B0	
Software interrupt number 22	+88 to +91 (Note 1)	Timer B1	
Software interrupt number 23	+92 to +95 (Note 1)	Timer B2	
Software interrupt number 24	+96 to +99 (Note 1)	Timer B3	
Software interrupt number 25	+100 to +103 (Note 1)	Timer B4	
Software interrupt number 26	+104 to +107 (Note 1)	$\overline{\text{INT5}}$	
Software interrupt number 27	+108 to +111 (Note 1)	$\overline{\text{INT4}}$	
Software interrupt number 28	+112 to +115 (Note 1)	$\overline{\text{INT3}}$	
Software interrupt number 29	+116 to +119 (Note 1)	$\overline{\text{INT2}}$	
Software interrupt number 30	+120 to +123 (Note 1)	$\overline{\text{INT1}}$	
Software interrupt number 31	+124 to +127 (Note 1)	$\overline{\text{INT0}}$	
Software interrupt number 32	+128 to +131 (Note 1)	Timer B5	
Software interrupt number 33	+132 to +135 (Note 1)	UART2 transmit/NACK (Note 2)	
Software interrupt number 34	+136 to +139 (Note 1)	UART2 receive/ACK (Note 2)	
Software interrupt number 35	+140 to +143 (Note 1)	UART3 transmit/NACK (Note 2)	
Software interrupt number 36	+144 to +147 (Note 1)	UART3 receive/ACK (Note 2)	
Software interrupt number 37	+148 to +151 (Note 1)	UART4 transmit/NACK (Note 2)	
Software interrupt number 38	+152 to +155 (Note 1)	UART4 receive/ACK (Note 2)	
Software interrupt number 39	+156 to +159 (Note 1)	Bus collision detection, start/stop condition detection (UART2) (Note 2)	
Software interrupt number 40	+160 to +163 (Note 1)	Bus collision detection, start/stop condition detection, fault error (UART3) (Note 2, 3)	
Software interrupt number 41	+164 to +167 (Note 1)	Bus collision detection, start/stop condition detection, fault error (UART4) (Note 2, 3)	
Software interrupt number 42	+168 to +171 (Note 1)	A-D	
Software interrupt number 43	+172 to +175 (Note 1)	Key input interrupt	
Software interrupt number 44 to Software interrupt number 63	+176 to +179 (Note 1) to +252 to +255 (Note 1)	Software interrupt	Cannot be masked I flag

Note 1: Address relative to address in interrupt table register (INTB).
Note 2: When I^{2}C mode is selected, NACK/ACK, start/stop condition detection interrupts are selected.
Note 3: The fault error interrupt is selected when $\overline{\text{SS}}$ pin is selected.

Interrupt control registers

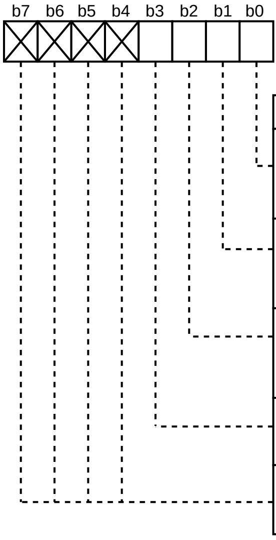
Peripheral I/O interrupts have their own interrupt control registers. Figure 1.9.3 shows the interrupt control registers.

When using an interrupt to exit Stop mode or Wait mode, the relevant interrupt must have been enabled and set to a priority level above the level set by the interrupt priority set bits for exit a stop/wait state (bits 2, 1, and 0 at address $009F_{16}$). Set the interrupt priority set bits for the exit from a stop/wait state to the same level as the flag register (FLG) processor interrupt level (IPL).

Figure 1.9.4 shows the exit priority register.

Interrupt control register

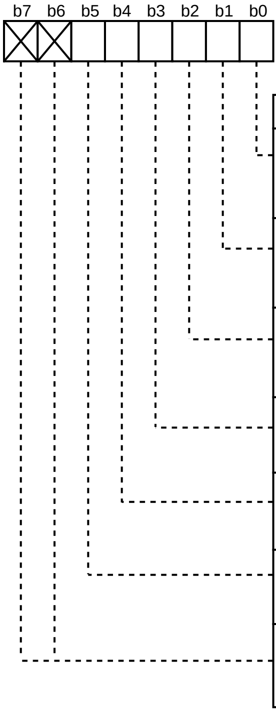
Symbol	Address	When reset
ADIC	0073$_{16}$	XXXXX000$_2$
BCNiIC(i=2 to 4)	008F$_{16}$, 0071$_{16}$, 0091$_{16}$	XXXXX000$_2$
DMiIC(i=0 to 3)	0068$_{16}$, 0088$_{16}$, 006A$_{16}$, 008A$_{16}$	XXXXX000$_2$
KUPIC	0093$_{16}$	XXXXX000$_2$
TAiIC(i=0 to 4)	006C$_{16}$, 008C$_{16}$, 006E$_{16}$, 008E$_{16}$, 0070$_{16}$	XXXXX000$_2$
TBiIC(i=0 to 5)	0094$_{16}$, 0076$_{16}$, 0096$_{16}$, 0078$_{16}$, 0098$_{16}$, 0069$_{16}$	XXXXX000$_2$
SiTIC(i=0 to 4)	0090$_{16}$, 0092$_{16}$, 0089$_{16}$, 008B$_{16}$, 008D$_{16}$	XXXXX000$_2$
SiRIC(i=0 to 4)	0072$_{16}$, 0074$_{16}$, 006B$_{16}$, 006D$_{16}$, 006F$_{16}$	XXXXX000$_2$



Bit symbol	Bit name	Function	R	W
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0 : Interrupt not requested 1 : Interrupt requested	○	○ (Note)
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			–	–

Note: This bit can only be accessed for reset (= 0), but cannot be accessed for set (= 1).

Symbol	Address	When reset
INTiIC(i=0 to 5)	009E$_{16}$, 007E$_{16}$, 009C$_{16}$, 007C$_{16}$, 009A$_{16}$, 007A$_{16}$	XX00X000$_2$

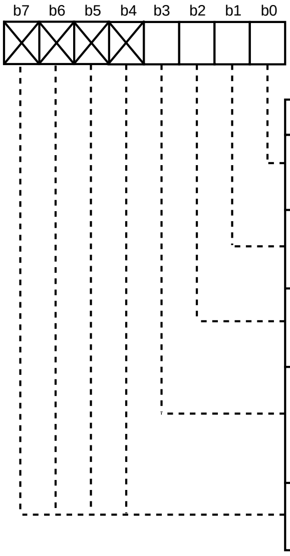


Bit symbol	Bit name	Function	R	W
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 (Note 2) 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	○	○
ILVL1			○	○
ILVL2			○	○
IR	Interrupt request bit	0: Interrupt not requested 1: Interrupt requested	○	○ (Note 1)
POL	Polarity select bit	0 : Selects falling edge or L level 1 : Selects rising edge or H level	○	○
LVS	Level sense/edge sense select bit	0 : Edge sense 1 : Level sense (Note 3)	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			–	–

Note 1: This bit can only be accessed for reset (= 0), but cannot be accessed for set (= 1).
Note 2: When INT3 to INT5 are used for data bus in microprocessor mode or memory expansion mode, set the interrupt disabled to INT3IC, INT4IC and INT5IC.
Note 3: When level sense is selected, set related bit of interrupt cause select register (address 031F$_{16}$) to one edge.

Figure 1.9.3. Interrupt control register

Exit priority register



Symbol	Address	When reset
RLVL	009F16	XXXX00002

Bit symbol	Bit	Function	R	W
RLVL0	Interrupt priority set bit for exiting Stop/Wait state (Note 1,2)	b2 b1 b0 0 0 0 : Level 0 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	○	○
RLVL1			○	○
RLVL2			○	○
FSIT	High-speed interrupt set bit (Note 3)	0: Interrupt priority level 7 = normal interrupt 1: Interrupt priority level 7 = high-speed interrupt	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			–	–

Note 1: Exits the Stop or Wait mode when the requested interrupt priority level is higher than that set in the exit priority register.
Note 2: Set to the same value as the processor interrupt priority level (IPL) set in the flag register (FLG).
Note 3: The high-speed interrupt can only be specified for interrupts with interrupt priority level 7. Specify interrupt priority level 7 for only one interrupt.

Figure 1.9.4. Exit priority register

Interrupt Enable Flag (I Flag)

The interrupt enable flag (I flag) is used to disable/enable maskable interrupts. When this flag is set (= 1), all maskable interrupts are enabled; when the flag is cleared to 0, they are disabled. This flag is automatically cleared to 0 after a reset is cleared.

Interrupt Request Bit

This bit is set (= 1) by hardware when an interrupt request is generated. The bit is cleared to 0 by hardware when the interrupt request is acknowledged and jump to the interrupt vector.
This bit can be cleared to 0 (but cannot be set to 1) in software.

Interrupt Priority Level Select Bit and Processor Interrupt Priority Level (IPL)

Interrupt priority levels are set by the interrupt priority select bit in an interrupt control register. When an interrupt request is generated, the interrupt priority level of this interrupt is compared with the processor interrupt priority level (IPL). This interrupt is enabled only when its interrupt priority level is greater than the processor interrupt priority level (IPL). This means that you can disable any particular interrupt by setting its interrupt priority level to 0.

Table 1.9.4 shows how interrupt priority levels are set. Table 1.9.5 shows interrupt enable levels in relation to the processor interrupt priority level (IPL).

The following lists the conditions under which an interrupt request is acknowledged:

- Interrupt enable flag (I flag) = 1
- Interrupt request bit = 1
- Interrupt priority level > Processor interrupt priority level (IPL)

The interrupt enable flag (I flag), interrupt request bit, interrupt priority level select bit, and the processor interrupt priority level (IPL) all are independent of each other, so they do not affect any other bit.

Table 1.9.4 Interrupt Priority Levels

Interrupt priority level select bit			Interrupt priority level	Priority order
b2	b1	b0		
0	0	0	Level 0 (interrupt disabled)	———
0	0	1	Level 1	Low
0	1	0	Level 2	
0	1	1	Level 3	
1	0	0	Level 4	
1	0	1	Level 5	
1	1	0	Level 6	
1	1	1	Level 7	High

Table 1.9.5 IPL and Interrupt Enable Levels

Processor interrupt priority level (IPL)			Enabled interrupt priority levels
IPL$_2$	IPL$_1$	IPL$_0$	
0	0	0	Interrupt levels 1 and above are enabled.
0	0	1	Interrupt levels 2 and above are enabled.
0	1	0	Interrupt levels 3 and above are enabled.
0	1	1	Interrupt levels 4 and above are enabled.
1	0	0	Interrupt levels 5 and above are enabled.
1	0	1	Interrupt levels 6 and above are enabled.
1	1	0	Interrupt levels 7 and above are enabled.
1	1	1	All maskable interrupts are disabled.

Rewrite the interrupt control register

When a instruction to rewrite the interrupt control register is executed but the interrupt is disabled, the interrupt request bit is not set sometimes even if the interrupt request for that register has been generated. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

Instructions : AND, OR, BCLR, BSET

Interrupt Sequence

An interrupt sequence — what are performed over a period from the instant an interrupt is accepted to the instant the interrupt routine is executed — is described here.

If an interrupt occurs during execution of an instruction, the processor determines its priority when the execution of the instruction is completed, and transfers control to the interrupt sequence from the next cycle. If an interrupt occurs during execution of either the SCMPU, SIN, SMOVB, SMOVF, SMOVU, SSTR, SOUT or RMPA instruction, the processor temporarily suspends the instruction being executed, and transfers control to the interrupt sequence.

In the interrupt sequence, the processor carries out the following in sequence given:

(1) CPU gets the interrupt information (the interrupt number and interrupt request level) by reading address 000000_{16} (address 000002_{16} when high-speed interrupt). After this, the related interrupt request bit is "0".

(2) Saves the content of the flag register (FLG) as it was immediately before the start of interrupt sequence in the temporary register (Note) within the CPU.

(3) Sets the interrupt enable flag (I flag), the debug flag (D flag), and the stack pointer select flag (U flag) to “0” (the U flag, however does not change if the INT instruction, in software interrupt numbers 32 through 63, is executed)

(4) Saves the content of the temporary register (Note 1) within the CPU in the stack area. Saves in the flag save register (SVF) in high-speed interrupt.

(5) Saves the content of the program counter (PC) in the stack area. Saves in the PC save register (SVP) in high-speed interrupt.

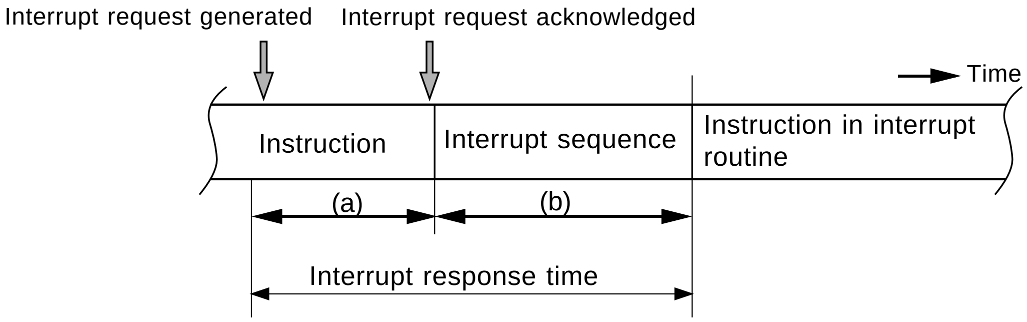
(6) Sets the interrupt priority level of the accepted instruction in the IPL.

After the interrupt sequence is completed, the processor resumes executing instructions from the first address of the interrupt routine.

Note: This register cannot be utilized by the user.

Interrupt Response Time

'Interrupt response time' is the period between the instant an interrupt occurs and the instant the first instruction within the interrupt routine has been executed. This time comprises the period from the occurrence of an interrupt to the completion of the instruction under execution at that moment (a) and the time required for executing the interrupt sequence (b). Figure 1.9.5 shows the interrupt response time.


Interrupt request generated
Interrupt request acknowledged
Time
Instruction
Interrupt sequence
Instruction in interrupt routine
(a)
(b)
Interrupt response time

(a) The period from the occurrence of an interrupt to the completion of the instruction under execution.

(b) The time required for executing the interrupt sequence.

Figure 1.9.5 Interrupt response time

Time (a) varies with each instruction being executed. The DIVX instruction requires a maximum time that consists of 24* cycles.
Time (b) is shown in table 1.9.6.

* It is when the divisor is immediate or register. When the divisor is memory, the following value is added.

• Normal addressing	: 2 + X
• Index addressing	: 3 + X
• Indirect addressing	: 5 + X + 2Y
• Indirect index addressing	: 6 + X + 2Y

X is number of wait of the divisor area. Y is number of wait of the indirect address stored area. When X and Y are in odd address or in 8 bits bus area, double the value of X and Y.

Table 1.9.6 Interrupt Sequence Execution Time

Interrupt	Interrupt vector address	16 bits data bus	8 bits data bus
Peripheral I/O	Even address Odd address (Note 1)	14 cycles 16 cycles	16 cycles 16 cycles
INT instruction	Even address Odd address (Note 1)	12 cycles 14 cycles	14 cycles 14 cycles
$\overline{\text{NMI}}$ Watchdog timer Undefined instruction Address match	Even address (Note 2)	13 cycles	15 cycles
Overflow	Even address (Note 2)	14 cycles	16 cycles
BRK instruction (Variable vector table)	Even address Odd address (Note 1)	17 cycles 19 cycles	19 cycles 19 cycles
Single step BRK2 instruction BRK instruction (Fixed vector table)	Even address (Note 2)	19 cycles	21 cycles
High-speed interrupt (Note 3)	Vector table is internal register	5 cycles	

Note 1: Allocate interrupt vector addresses in even addresses, if possible.
Note 2: The vector table is fixed to even address.
Note 3: The high-speed interrupt is independent of these conditions.

Changes of IPL When Interrupt Request Acknowledged

When an interrupt request is acknowledged, the interrupt priority level of the acknowledged interrupt is set to the processor interrupt priority level (IPL).

If an interrupt request is acknowledged that does not have an interrupt priority level, the value shown in Table 1.9.7 is set to the IPL.

Table 1.9.7 Relationship between Interrupts without Interrupt Priority Levels and IPL

Interrupt sources without interrupt priority levels	Value that is set to IPL
Watchdog timer, $\overline{\text{NMI}}$	7
Reset	0
Other	Not changed

Saving Registers

In an interrupt sequence, only the contents of the flag register (FLG) and program counter (PC) are saved to the stack area.

The order in which these contents are saved is as follows: First, the FLG register is saved to the stack area. Next, the 16 high-order bits and 16 low-order bits of the program counter expanded to 32-bit are saved. Figure 1.9.6 shows the stack status before an interrupt request is acknowledged and the stack status after an interrupt request is acknowledged.

In a high-speed interrupt sequence, the contents of the flag register (FLG) is saved to the flag save register (SVF) and program counter (PC) is saved to PC save register (SVP).

If there are any other registers you want to be saved, save them in software at the beginning of the interrupt routine. The PUSHM instruction allows you to save all registers except the stack pointer (SP) by a single instruction.

The execution speed is improved when register bank 1 is used with high speed interrupt register selected by not saving registers to the stack but to the switching register bank. In this case, switch register bank mode for high-speed interrupt routine.

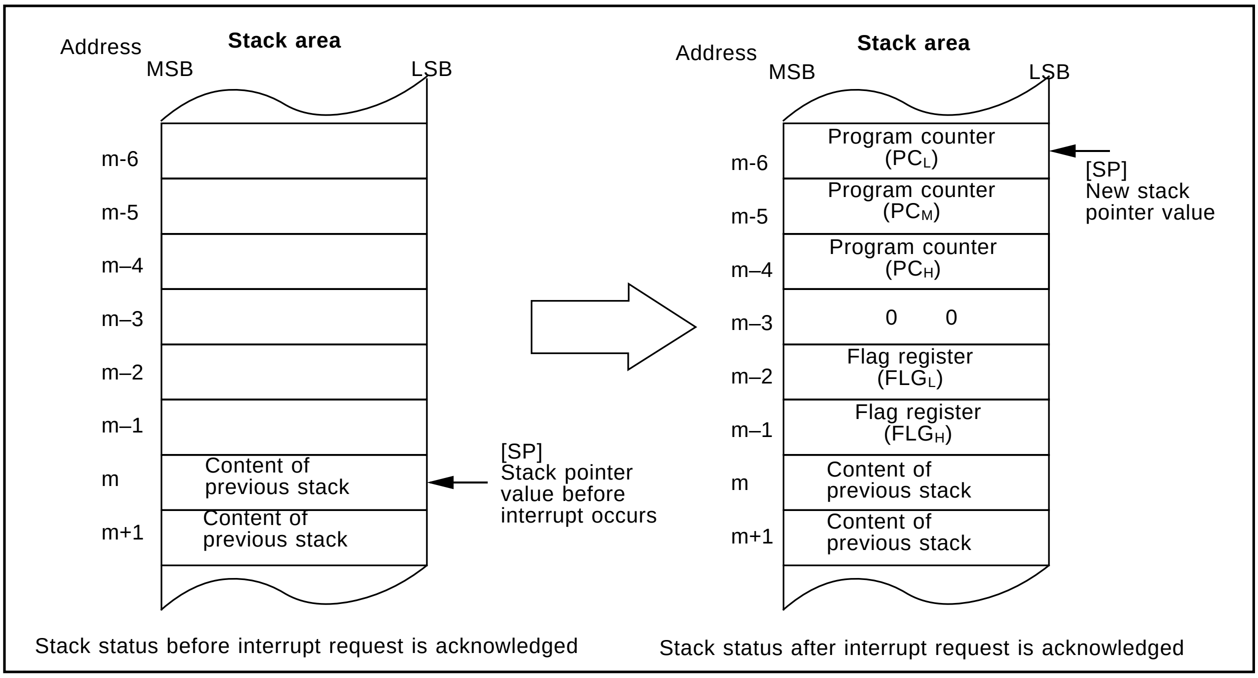

Address
Stack area
MSB
LSB
m-6
m-5
m–4
m–3
m–2
m–1
m
m+1
Content of previous stack
Content of previous stack
[SP] Stack pointer value before interrupt occurs
Address
Stack area
MSB
LSB
m-6
m-5
m–4
m–3
m–2
m–1
m
m+1
Program counter (PC_L)
Program counter (PC_M)
Program counter (PC_H)
0 0
Flag register (FLG_L)
Flag register (FLG_H)
Content of previous stack
Content of previous stack
[SP] New stack pointer value
Stack status before interrupt request is acknowledged
Stack status after interrupt request is acknowledged

Figure 1.9.6 Stack status before and after an interrupt request is acknowledged

Return from Interrupt Routine

As you execute the REIT instruction at the end of the interrupt routine, the contents of the flag register (FLG) and program counter (PC) that have been saved to the stack area immediately preceding the interrupt sequence are automatically restored. In high-speed interrupt, as you execute the FREIT instruction at the end of the interrupt routine, the contents of the flag register (FLG) and program counter (PC) that have been saved to the save registers immediately preceding the interrupt sequence are automatically restored.

Then control returns to the routine that was under execution before the interrupt request was acknowledged, and processing is resumed from where control left off.

If there are any registers you saved via software in the interrupt routine, be sure to restore them using an instruction (e.g., POPM instruction) before executing the REIT or FREIT instruction.

When switching the register bank before executing REIT and FREIT instruction, switched to the register bank immediately before the interrupt sequence.

Interrupt Priority

If two or more interrupt requests are sampled active at the same time, whichever interrupt request is acknowledged that has the highest priority.

Maskable interrupts (Peripheral I/O interrupts) can be assigned any desired priority by setting the interrupt priority level select bit accordingly. If some maskable interrupts are assigned the same priority level, the priority between these interrupts is resolved by the priority that is set in hardware.

Certain nonmaskable interrupts such as a reset (reset is given the highest priority) and watchdog timer interrupt have their priority levels set in hardware. Figure 1.9.7 lists the hardware priority levels of these interrupts.

Software interrupts are not subjected to interrupt priority. They always cause control to branch to an interrupt routine whenever the relevant instruction is executed.

Interrupt Resolution Circuit

Interrupt resolution circuit selects the highest priority interrupt when two or more interrupt requests are sampled active at the same time.

Figure 1.9.8 shows the interrupt resolution circuit.

Reset > $\overline{\text{NMI}}$ > Watchdog > Peripheral I/O > Single step > Address match

Figure 1.9.7. Interrupt priority that is set in hardware

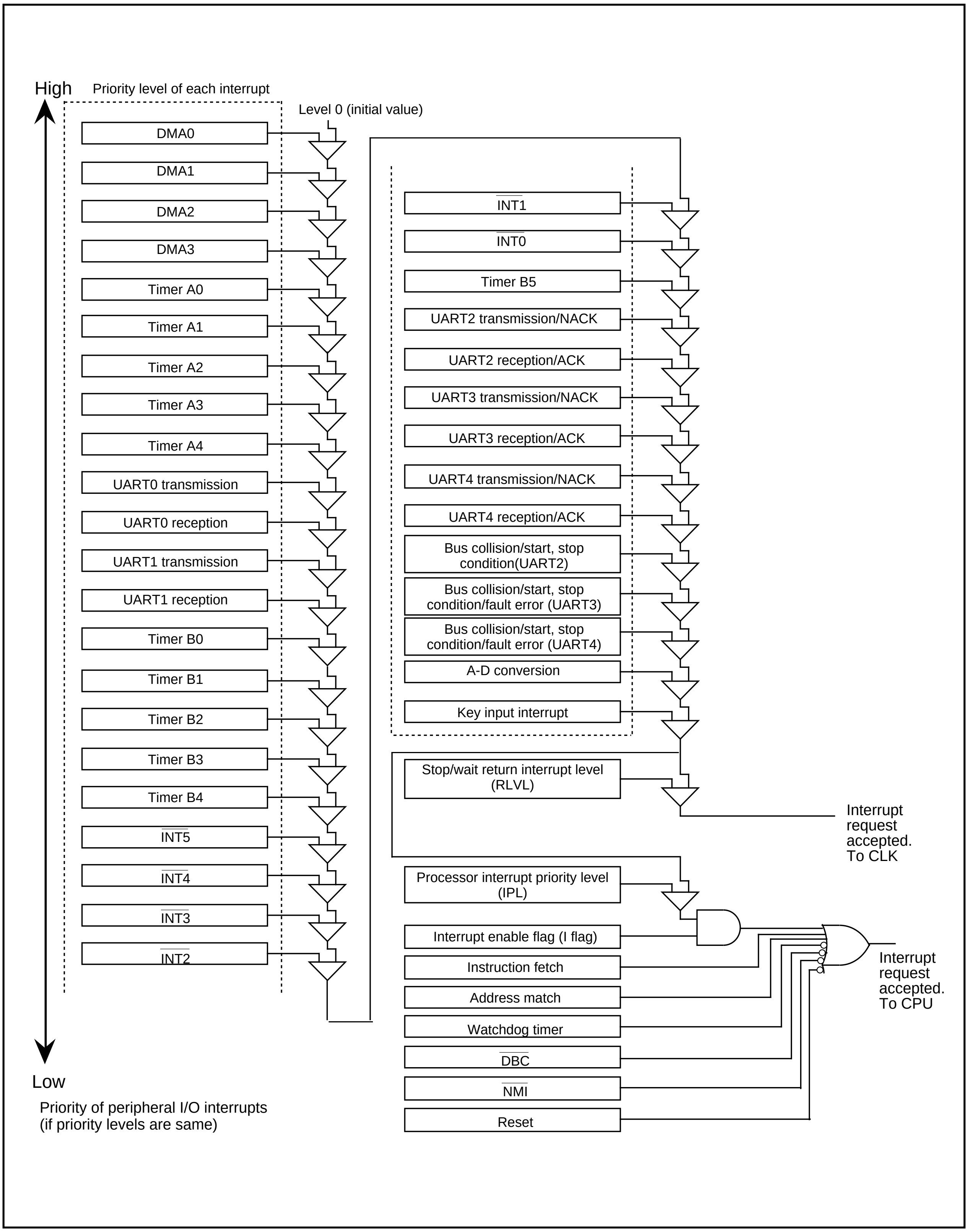


Figure 1.9.8. Interrupt resolution circuit

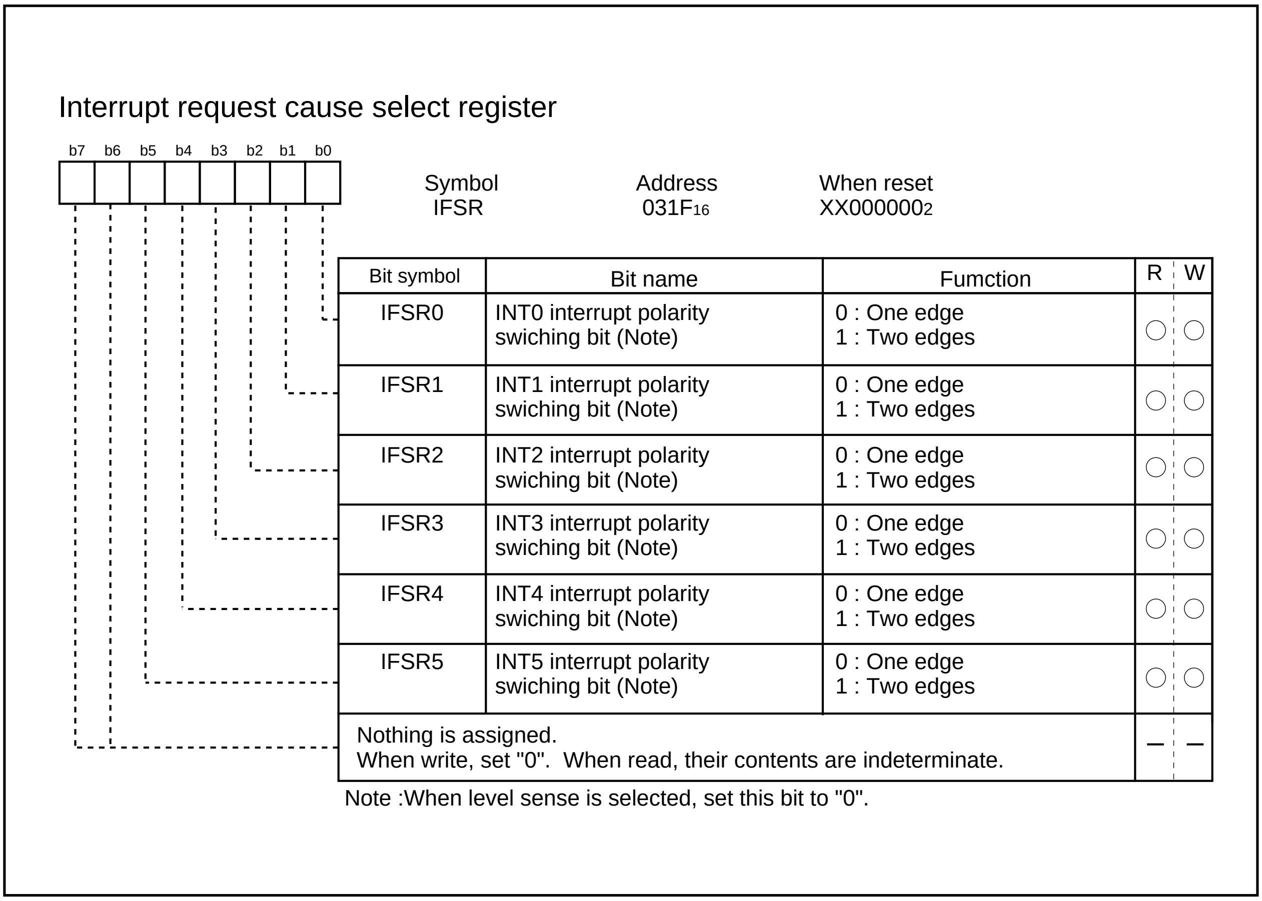
$\overline{\text{INT}}$ Interrupts

$\overline{\text{INT0}}$ to $\overline{\text{INT5}}$ are external input interrupts. The level sense/edge sense switching bits of the interrupt control register select the input signal level and edge at which the interrupt can be set to occur on input signal level and input signal edge. The polarity bit selects the polarity.

With the external interrupt input edge sense, the interrupt can be set to occur on both rising and falling edges by setting the INTi interrupt polarity switch bit of the interrupt request select register (address 031F$_{16}$) to “1”. When you select both edges, set the polarity switch bit of the corresponding interrupt control register to the falling edge (“0”).

When you select level sense, the INTi interrupt polarity switch bit of the interrupt request select register (address 031F$_{16}$) to “0”.

Figure 1.9.9 shows the interrupt request select register.



Interrupt request cause select register

Symbol	Address	When reset
IFSR	031F$_{16}$	XX000000$_2$

Bit symbol	Bit name	Fumction	R	W
IFSR0	INT0 interrupt polarity swiching bit (Note)	0 : One edge 1 : Two edges	○	○
IFSR1	INT1 interrupt polarity swiching bit (Note)	0 : One edge 1 : Two edges	○	○
IFSR2	INT2 interrupt polarity swiching bit (Note)	0 : One edge 1 : Two edges	○	○
IFSR3	INT3 interrupt polarity swiching bit (Note)	0 : One edge 1 : Two edges	○	○
IFSR4	INT4 interrupt polarity swiching bit (Note)	0 : One edge 1 : Two edges	○	○
IFSR5	INT5 interrupt polarity swiching bit (Note)	0 : One edge 1 : Two edges	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			–	–

Note :When level sense is selected, set this bit to "0".

Figure 1.9.9 Interrupt request cause select register

$\overline{\text{NMI}}$ Interrupt

An $\overline{\text{NMI}}$ interrupt is generated when the input to the P85/$\overline{\text{NMI}}$ pin changes from “H” to “L”. The $\overline{\text{NMI}}$ interrupt is a non-maskable external interrupt. The pin level can be checked in the port $P8_5$ register (bit 5 at address $03C4_{16}$).

This pin cannot be used as a normal port input.

Notes:

When not intending to use the $\overline{\text{NMI}}$ function, be sure to connect the $\overline{\text{NMI}}$ pin to V_{CC} (pulled-up). The $\overline{\text{NMI}}$ interrupt is non-maskable. Because it cannot be disabled, the pin must be pulled up.

Key Input Interrupt

If the direction register of any of $P10_4$ to $P10_7$ is set for input and a falling edge is input to that port, a key input interrupt is generated. A key input interrupt can also be used as a key-on wakeup function for cancelling the wait mode or stop mode. However, if you intend to use the key input interrupt, do not use $P10_4$ to $P10_7$ as A-D input ports. Figure 1.9.10 shows the block diagram of the key input interrupt. Note that if an “L” level is input to any pin that has not been disabled for input, inputs to the other pins are not detected as an interrupt.

Setting the key input interrupt disable bit (bit 7 at address $03AF_{16}$) to “1” disables key input interrupts from occurring regardless of the setting in the interrupt control register. When “1” is set in the key input interrupt disable register, there is no input via the port pin even when the direction register is set to input.

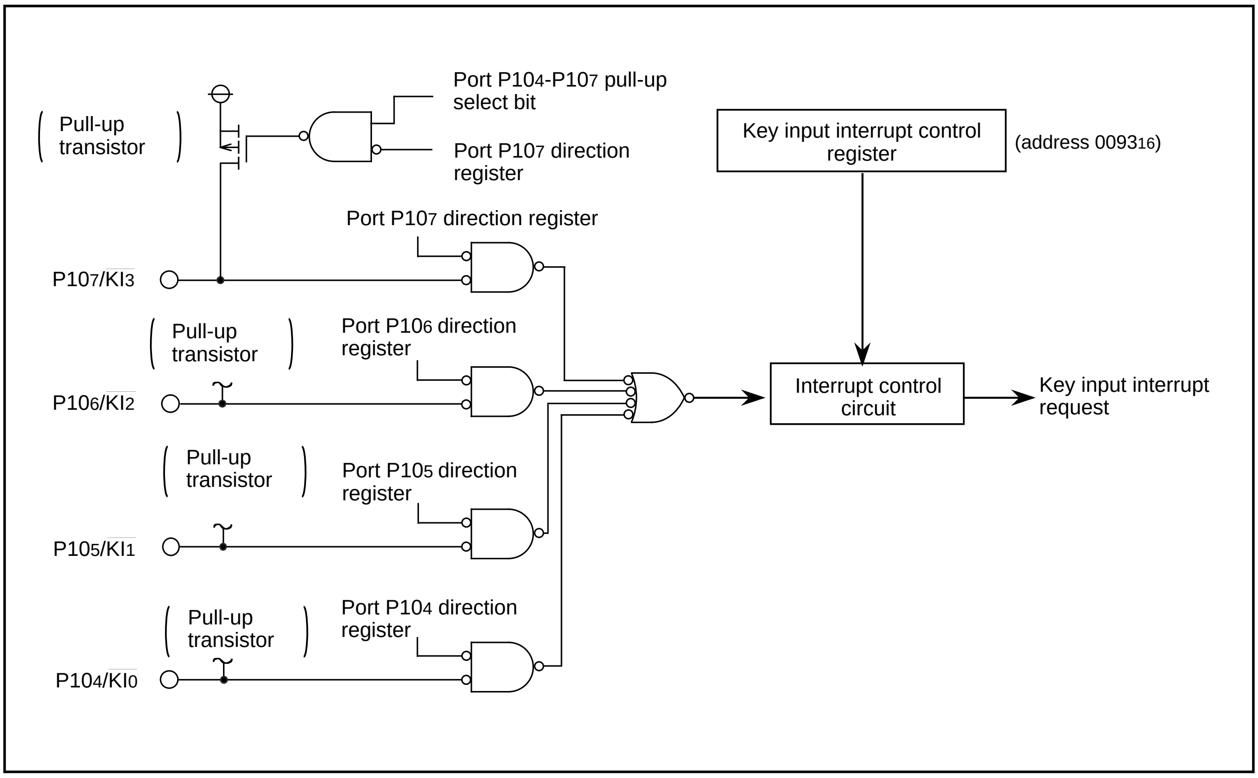

Port P104-P107 pull-up select bit
Pull-up transistor
Port P107 direction register
Key input interrupt control register
(address 009316)
Port P107 direction register
P107/KI3
Pull-up transistor
Port P106 direction register
P106/KI2
Interrupt control circuit
Key input interrupt request
Pull-up transistor
Port P105 direction register
P105/KI1
Pull-up transistor
Port P104 direction register
P104/KI0

Figure 1.9.10. Block diagram of key input interrupt

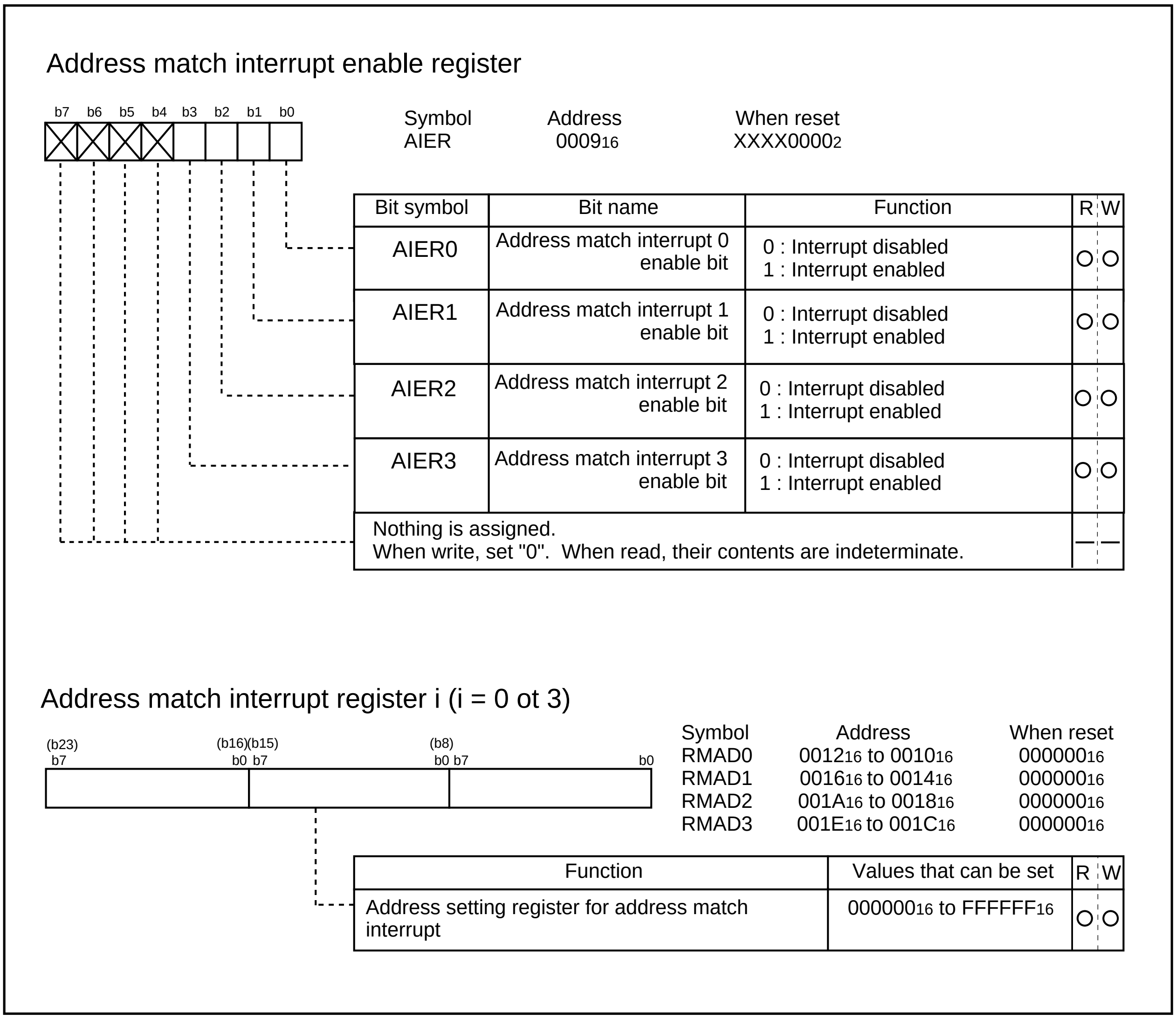
Address Match Interrupt

An address match interrupt is generated when the address match interrupt address register contents match the program counter value. Four address match interrupts can be set, each of which can be enabled and disabled by an address match interrupt enable bit. Address match interrupts are not affected by the interrupt enable flag (I flag) and processor interrupt priority level (IPL).

Figure 1.9.11 shows the address match interrupt-related registers.

Set the start address of an instruction to the address match interrupt register.

Address match interrupt is not generated when address such as the middle of instruction or table data is set.



Address match interrupt enable register

Symbol	Address	When reset
AIER	0009_{16}	$XXXX0000_2$

Bit symbol	Bit name	Function	R	W
AIER0	Address match interrupt 0 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	○	○
AIER1	Address match interrupt 1 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	○	○
AIER2	Address match interrupt 2 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	○	○
AIER3	Address match interrupt 3 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—

Address match interrupt register i (i = 0 ot 3)

Symbol	Address	When reset
RMAD0	0012_{16} to 0010_{16}	000000_{16}
RMAD1	0016_{16} to 0014_{16}	000000_{16}
RMAD2	$001A_{16}$ to 0018_{16}	000000_{16}
RMAD3	$001E_{16}$ to $001C_{16}$	000000_{16}

Function	Values that can be set	R	W
Address setting register for address match interrupt	000000_{16} to $FFFFFF_{16}$	○	○

Figure 1.9.11. Address match interrupt-related registers

Precautions for Interrupts

(1) Reading addresses 000000_{16} and 000002_{16}

- When maskable interrupt is occurred, CPU read the interrupt information (the interrupt number and interrupt request level) in the interrupt sequence from address 000000_{16}. When high-speed interrupt is occurred, CPU read from address 000002_{16}.
  The interrupt request bit of the certain interrupt will then be set to “0”.
  However, reading addresses 000000_{16} and 000002_{16} by software does not set request bit to “0”.

(2) Setting the stack pointer

- The value of the stack pointer immediately after reset is initialized to 000000_{16}. Accepting an interrupt before setting a value in the stack pointer may become a factor of runaway. Be sure to set a value in the stack pointer before accepting an interrupt. When using the $\overline{\text{NMI}}$ interrupt, initialize the stack point at the beginning of a program. Any interrupt including the $\overline{\text{NMI}}$ interrupt is generated immediately after executing the first instruction after reset. Set an even address to the stack pointer so that the operating efficiency of accessign memory is increased.

(3) The $\overline{\text{NMI}}$ interrupt

- As for the $\overline{\text{NMI}}$ interrupt pin, an interrupt cannot be disabled. Connect it to the Vcc pin via a resistance (pull-up) if unused. Be sure to work on it.
- The $\overline{\text{NMI}}$ pin also serves as P8$_5$, which is exclusively input. Reading the contents of the P8 register allows reading the pin value. Use the reading of this pin only for establishing the pin level at the time when the $\overline{\text{NMI}}$ interrupt is input.
- Signal of "L" level width more than 1 clock of CPU operation clock (BCLK) is necessary for $\overline{\text{NMI}}$ pin.

(4) External interrupt

- Edge sense
  Either an “L” level or an “H” level of at least 250 ns width is necessary for the signal input to pins INT$_0$ to INT$_5$ regardless of the CPU operation clock.
- Level sense
  Either an “L” level or an “H” level of 1 cycle of BCLK + at least 200 ns width is necessary for the signal input to pins INT$_0$ to INT$_5$ regardless of the CPU operation clock. (When X$_{IN}$=20MHz and no division mode, at least 250 ns width is necessary.)
- When the polarity of the INT$_0$ to INT$_5$ pins is changed, the interrupt request bit is sometimes set to "1". After changing the polarity, set the interrupt request bit to "0". Figure 1.9.12 shows the procedure for changing the $\overline{\text{INT}}$ interrupt generate factor.

(5) Rewrite the interrupt control register

- When a instruction to rewrite the interrupt control register is executed but the interrupt is disabled, the interrupt request bit is not set sometimes even if the interrupt request for that register has been generated. This will depend on the instruction. If this creates problems, use the below instructions to change the register.
  Instructions : AND, OR, BCLR, BSET

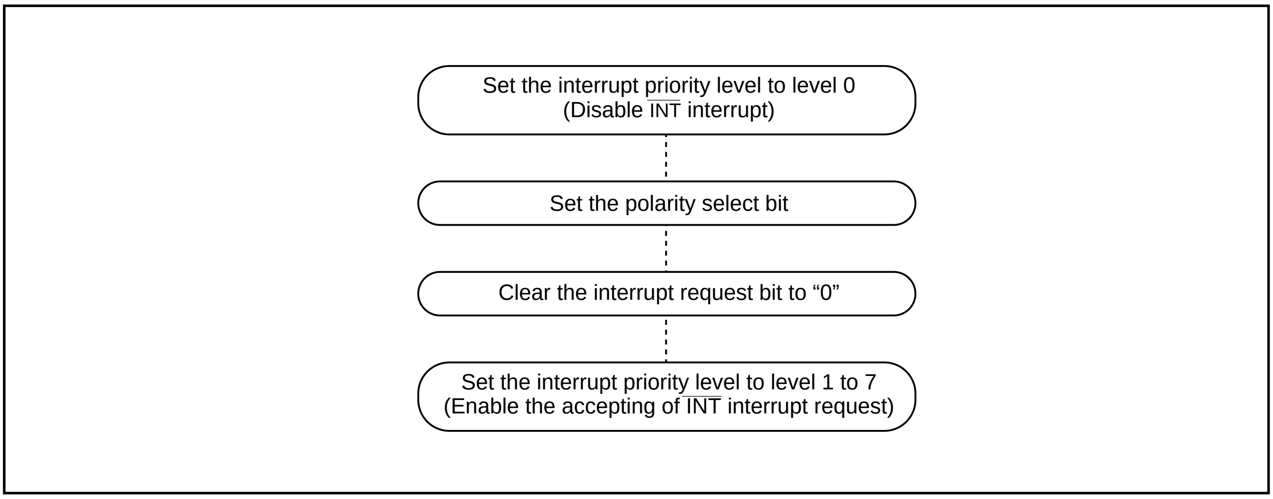

Set the interrupt priority level to level 0
(Disable $\overline{\text{INT}}$ interrupt)
Set the polarity select bit
Clear the interrupt request bit to “0”
Set the interrupt priority level to level 1 to 7
(Enable the accepting of $\overline{\text{INT}}$ interrupt request)

Figure 1.9.12. Switching condition of $\overline{\text{INT}}$ interrupt request

(6) Address match interrupt

- Do not set the following addresses to the address match interrupt register.
  1. The address of the starting instruction in an interrupt routine.
  2. Any of the next 7 instructions addresses immediately after an instruction to clear an interrupt request bit of an interrupt control register or an instruction to rewrite an interrupt priority level to a smaller value.
  3. Any of the next 3 instructions addresses immediately after an instruction to set the interrupt enable flag (I flag).
  4. Any of the next 3 instructions addresses immediately after an instruction to rewrite a processor interrupt priority level (IPL) to a smaller value.

```
Example 1)
      Interrupt_A:                         ; Interrupt A routine
              pushm   R0,R1,R2,R3,A0,A1    ; <----  Do not set address match interrupt to the
              ••••                         ;        start address of an interrupt instruction
Example 2)
      mov.b     #0,TA0IC             ;Change TA0 interrupt priority level to a smaller value
      nop                            ; 1st instruction  ┐
      nop                            ; 2nd instruction  │
      nop                            ; 3rd instruction  │  Do not set address match interrupt
      nop                            ; 4th instruction  ┤  during this period
      nop                            ; 5th instruction  │
      nop                            ; 6th instruction  │
      nop                            ; 7th instruction  ┘
```

Example 3)

```
        fset    I               ; Set I flag ( interrupt enabled)
        nop                     ; 1st instruction  ⎫
        nop                     ; 2nd instruction  ⎬ Do not set address match interrupt
        nop                     ; 3rd instruction  ⎭ during this period
```

Example 4)

```
        ldipl   #0              ; Rewrite IPL to a smaller value
        nop                     ; 1st instruction  ⎫
        nop                     ; 2nd instruction  ⎬ Do not set address match interrupt
        nop                     ; 3rd instruction  ⎭ during this period
```

- To return from an interrupt to the address set in an address match interrupt register using return instruction (reit or freit)

  To rewrite the interrupt control register within the interrupt routine, add the below processing to the end of the routine (immediately before the reit or freit instruction). Also, if multiple interrupts are enabled with other interrupts, add the below processing to the end of the interrupt that enables the multiple interrupts.

If the interrupt control register is being rewritten within the non-maskable interrupt routine, add the below processing to the end of all interrupts.

Additional process

```
                                ; Execute after the register reset instruction (popm instruction)
        fclr    U               ; Select ISP (Unnecessary if the ISP has been selected)
        pushm   R0              ; Store R0 register
        mov.w   6[SP],R0        ; Read FLG on stack (use "stc SVF,R0" when high-speed
                                ;                              interrupt)
        ldc     R0,FLG          ; Set in FLG
        popm    R0              ; Restore R0 register
        nop                     ; Dummy
        reit                    ; Interrupt completed (use freit when high-speed interrupt)
```

Example 5)

If rewriting the interrupt control register for interrupt B with the interrupt A routine and enabling multiple interrupts with interrupt C, the above processing is required at the end of the interrupt A and interrupt C routines.

```
            Interrupt A routine
Interrupt_A:
    pushm R0,R1,R2,R3,A0,A1      ; Store registers
    ••••
    bclr   3,TA0IC               ; Rewrite interrupt control register of interrupt B
    ••••
    popm   R0,R1,R2,R3,A0,A1     ; Restore registers
    fclr   U                     ; Select ISP (Unnecessary if the ISP has been selected)
    pushm R0                     ; Store R0 register
    mov.w 6[SP],R0               ; Read FLG on stack
    ldc    R0,FLG                ; Set in FLG
    popm   R0                    ; Restore R0 register
    nop                          ; Dummy
    reit                         ; Interrupt completed


            Interrupt C routine
Interrupt_C:
    pushm R0,R1,R2,R3,A0,A1      ; Store registers
    fset   I                     ; Multiple interrupt enabled
    ••••
    ••••
    popm   R0,R1,R2,R3,A0,A1     ;Restore registers
    fclr   U                     ; Select ISP (Unnecessary if the ISP has been selected)
    pushm R0                     ; Store R0 register
    mov.w 6[SP],R0               ; Read FLG on stack
    ldc    R0,FLG                ; Set in FLG
    popm   R0                    ; Restore R0 register
    nop                          ; Dummy
    reit                         ; Interrupt completed
```

Watchdog Timer

The watchdog timer has the function of detecting when the program is out of control. The watchdog timer is a 15-bit counter which down-counts the clock derived by dividing the BCLK using the prescaler. Whether a watchdog timer interrupt is generated or reset is selected when an underflow occurs in the watchdog timer. Watchdog timer interrupt is selected when bit 6 of the system control register 0 (address 0008_{16} :CM06) is "0" and reset is selected when CM06 is "1". No value other than "1" can be written in CM06. Once when reset is selected (CM06="1"), watchdog timer interrupt cannot be selected by software.

When X_{IN} is selected for the BCLK, bit 7 of the watchdog timer control register (address $000F_{16}$) selects the prescaler division ratio (by 16 or by 128). When X_{CIN} is selected as the BCLK, the prescaler is set for division by 2 regardless of bit 7 of the watchdog timer control register (address $000F_{16}$). Therefore, the watchdog timer cycle can be calculated as follows. However, errors can arise in the watchdog timer cycle due to the prescaler.

When X_{IN} is selected in BCLK

$$\text{Watchdog timer cycle} = \frac{\text{Prescaler division ratio (16 or 128) x watchdog timer count (32768)}}{\text{BCLK}}$$

When X_{CIN} is selected in BCLK

$$\text{Watchdog timer cycle} = \frac{\text{Prescaler division ratio (2) x watchdog timer count (32768)}}{\text{BCLK}}$$

For example, when BCLK is 20MHz and the prescaler division ratio is set to 16, the monitor timer cycle is approximately 26.2 ms.

The watchdog timer is initialized by writing to the watchdog timer start register (address $000E_{16}$) and when a watchdog timer interrupt request is generated. The prescaler is initialized only when the microcomputer is reset. After a reset is cancelled, the watchdog timer and prescaler are both stopped. The count is started by writing to the watchdog timer start register (address $000E_{16}$). CM06 is initialized only at reset. After reset, watchdog timer interrupt is selected.

The watchdog timer and the prescaler stop in stop mode, wait mode and hold status. After exiting these modes and status, counting starts from the value remained before.

In the stop mode, wait mode and hold state, the watchdog timer and prescaler are stopped. Counting is resumed from the held value when the modes or state are released. Figure 1.10.1 shows the block diagram of the watchdog timer. Figure 1.10.2 shows the watchdog timer-related registers.

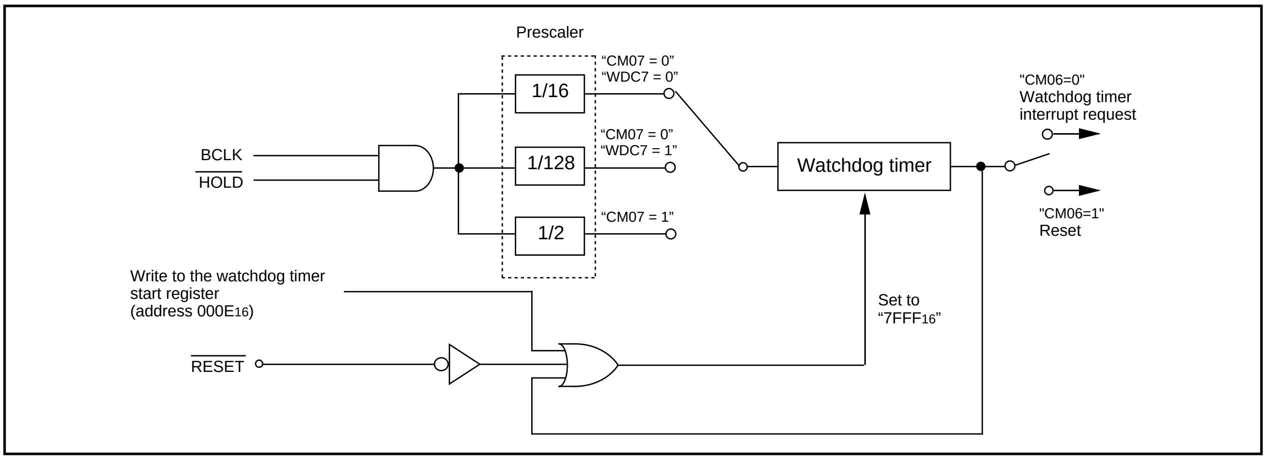


Figure 1.10.1. Block diagram of watchdog timer

Watchdog timer control register

b7	b6	b5	b4	b3	b2	b1	b0
	0	0					

Symbol WDC   Address 000F16   When reset 000XXXXX2

Bit symbol	Bit name	Function	R	W
High-order bit of watchdog timer			○	×
Reserved bit		Must always be set to “0”	○	○
WDC7	Prescaler select bit	0 : Divided by 16 1 : Divided by 128	○	○

Watchdog timer start register

b7 … b0

Symbol WDTS   Address 000E16   When reset Indeterminate

Function	R	W
The watchdog timer is initialized and starts counting after a write instruction to this register. The watchdog timer value is always initialized to “7FFF16” regardless of whatever value is written.	×	○

System clock control register 0 (Note 1)

b7	b6	b5	b4	b3	b2	b1	b0

Symbol CM0   Address 000616   When reset 0816

Bit symbol	Bit	Function	R	W
CM00	Clock output function select bit (Note 2)	b1 b0 0 0 : I/O port P53 0 1 : fC output (Note 3) 1 0 : f8 output (Note 3) 1 1 : f32 output (Note 3)	○	○
CM01			○	○
CM02	WAIT peripheral function clock stop bit	0 : Do not stop peripheral clock in wait mode 1 : Stop peripheral clock in wait mode (Note 10)	○	○
CM03	XCIN-XCOUT drive capacity select bit (Note 4)	0 : LOW 1 : HIGH	○	○
CM04	Port XC select bit	0 : I/O port 1 : XCIN-XCOUT generation (Note 11)	○	○
CM05	Main clock (XIN-XOUT) stop bit (Note 5, 6)	0 : On 1 : Off (Note 7)	○	○
CM06	Watchdog timer function select bit	0 : Watchdog timer interrupt 1 : Reset (Note 8)	○	○
CM07	System clock select bit (Note 9)	0 : XIN, XOUT 1 : XCIN, XCOUT	○	○

Note 1: Set bit 0 of the protect register (address 000A16) to “1” before writing to this register.
Note 2: When outputting BCLK (bit 7 of processor mode register 0 is "0"), set these bits to "00". When outputting ALE to P53 (bit 5 and 4 of processor mode register 0 is "01"), set these bits to "00". The port P53 function is not selected even when you set "00" in microprocessor or memory expansion mode and bit 7 of the processor mode register 0 is "1".
Note 3: When selecting fC, f8 or f32 in single chip mode, must use P57 as input port.
Note 4: Changes to “1” when shifting to stop mode or reset.
Note 5: When entering the power saving mode, the main clock is stopped using this bit. To stop the main clock, set system clock stop bit (CM07) to "1" while an oscillation of sub clock is stable. Then set this bit to "1".
Note 6: When this bit is "1", XOUT is "H". Also, the internal feedback resistance remains ON, so XIN is pulled up to XOUT ("H" level) via the feedback resistance.
Note 7: When the main clock is stopped, the main clock division register (address 000C16) is set to the division by 8 mode.
Note 8: When "1" has been set once, "0" cannot be written by software.
Note 9: To set CM07 "1" from "0", first set CM04 to "1", and an oscillation of sub clock is stable. Then set CM07. Do not set CM04 and CM07 simultaneously. Also, to set CM07 "0" from "1", first set CM05 to "1", and an oscillation of main clock is stable. Then set CM07.
Note 10: fc32 is not included.
Note 11: When XCIN-XCOUT is used, set port P86 and P87 to no pull-up resistance with the input port.

Figure 1.10.2. Watchdog timer control and start registers

DMAC

This microcomputer has four DMAC (direct memory access controller) channels that allow data to be sent to memory without using the CPU. DMAC is a function that to transmit 1 data of a source address (8 bits / 16 bits) to a destination address when transmission request occurs. When using three or more DMAC channels, the register bank 1 register and high-speed interrupt register are used as DMAC registers. If you are using three or more DMAC channels, you cannot, therefore, use high-speed interrupts. The CPU and DMAC use the same data bus, but the DMAC has a higher bus access privilege than the CPU, and because of the use of cycle-steeling, operations are performed at high-speed from the occurrence of a transfer request until one word (16 bits) or 1 byte (8 bits) of data have been sent. Figure 1.11.1 shows the mapping of registers used by the DMAC. Table 1.11.1 shows DMAC specifications. Figures 1.11.2 to 1.11.5 show the structures of the registers used.

As the registers shown in Figure 1.11.1 is allocated in CPU, use LDC instruction when writing. When writing to DCT2, DCT3, DRC2, DRC3, DMA2 and DMA3, set register bank select flag (B flag) to "1" and use MOV instruction to set R0 to R3, A0 and A1 registers. When writing to DSA2 and DSA3, set register bank select flag (B flag) to "1" and use LDC instruction to set SB and FB registers.

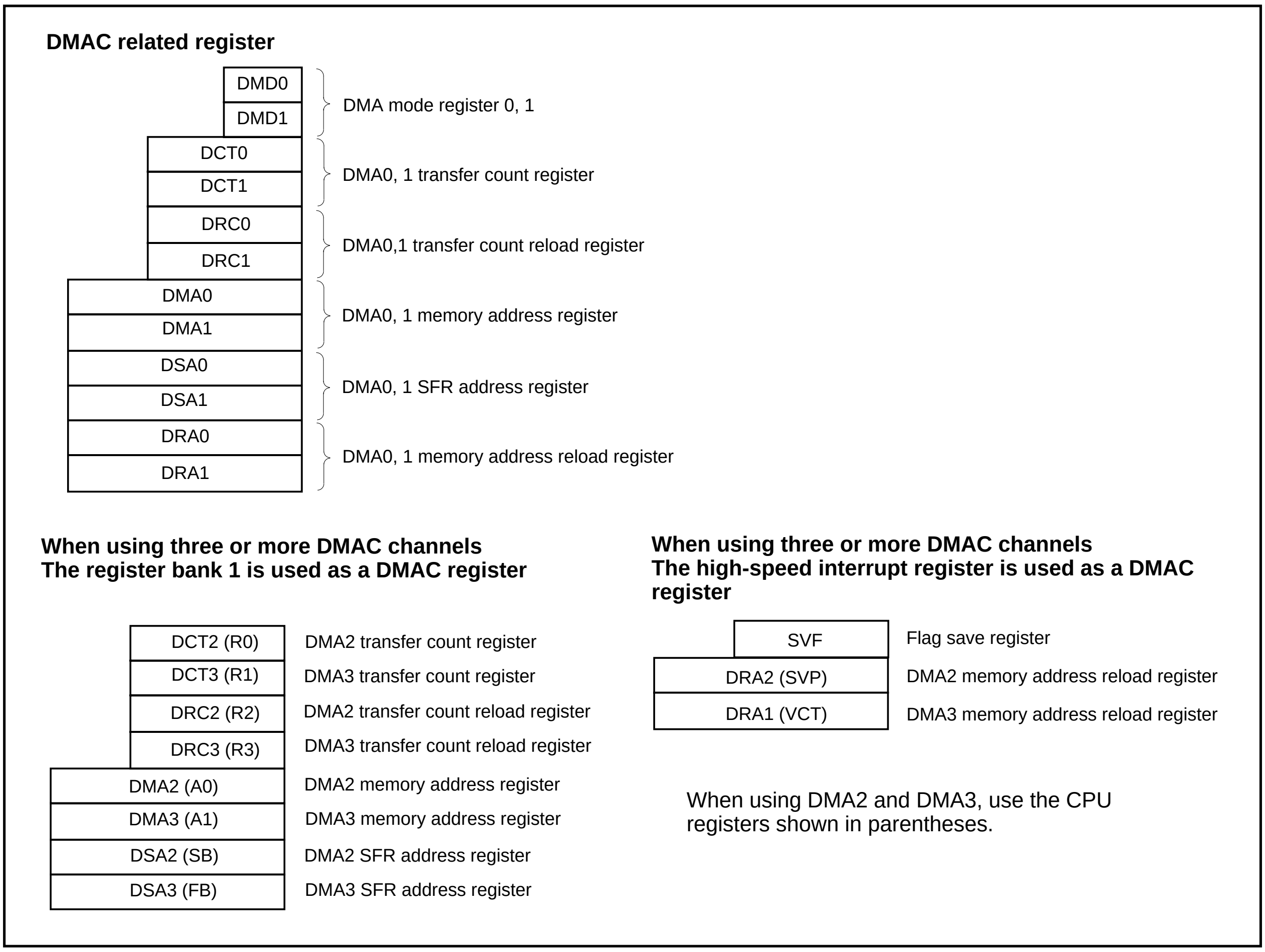

DMAC related register
DMD0
DMD1
DMA mode register 0, 1
DCT0
DCT1
DMA0, 1 transfer count register
DRC0
DRC1
DMA0,1 transfer count reload register
DMA0
DMA1
DMA0, 1 memory address register
DSA0
DSA1
DMA0, 1 SFR address register
DRA0
DRA1
DMA0, 1 memory address reload register
When using three or more DMAC channels
The register bank 1 is used as a DMAC register
DCT2 (R0)
DMA2 transfer count register
DCT3 (R1)
DMA3 transfer count register
DRC2 (R2)
DMA2 transfer count reload register
DRC3 (R3)
DMA3 transfer count reload register
DMA2 (A0)
DMA2 memory address register
DMA3 (A1)
DMA3 memory address register
DSA2 (SB)
DMA2 SFR address register
DSA3 (FB)
DMA3 SFR address register
When using three or more DMAC channels
The high-speed interrupt register is used as a DMAC register
SVF
Flag save register
DRA2 (SVP)
DMA2 memory address reload register
DRA1 (VCT)
DMA3 memory address reload register
When using DMA2 and DMA3, use the CPU registers shown in parentheses.

Figure 1.11.1. Register map using DMAC

In addition to writing to the software DMA request bit to start DMAC transfer, the interrupt request signals output from the functions specified in the DMA request factor select bits are also used. However, in contrast to the interrupt requests, repeated DMA requests can be received, regardless of the interrupt flag.

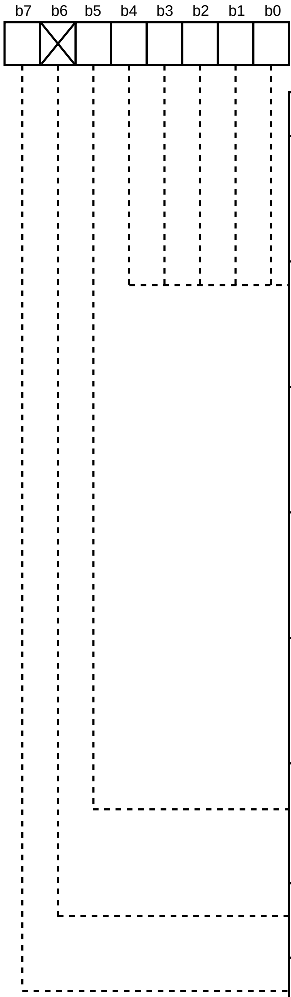
(Note, however, that the number of actual transfers may not match the number of transfer requests if the DMA request cycle is shorter than the DMR transfer cycle. For details, see the description of the DMAC request bit.)

Table 1.11.1. DMAC specifications

Item	Specification
No. of channels	4 (cycle steal method)
Transfer memory space	• From any address in the 16 Mbytes space to a fixed address (16 Mbytes space) • From a fixed address (16 Mbytes space) to any address in the 16 M bytes space
Maximum No. of bytes transferred	128 Kbytes (with 16-bit transfers) or 64 Kbytes (with 8-bit transfers)
DMA request factors (Note)	Falling edge of $\overline{\mathrm{INT0}}$ to $\overline{\mathrm{INT3}}$ or both edge Timer A0 to timer A4 interrupt requests Timer B0 to timer B5 interrupt requests UART0 to UART4 transmission and reception interrupt requests A-D conversion interrupt requests Software triggers
Channel priority	DMA0 > DMA1 > DMA2 > DMA3 (DMA0 is the first priority)
Transfer unit	8 bits or 16 bits
Transfer address direction	forward/fixed (forward direction cannot be specified for both source and destination simultaneously)
Transfer mode	• Single transfer Transfer ends when the transfer count register is "0000_{16}". • Repeat transfer When the transfer counter is "0000_{16}", the value in the transfer counter reload register is reloaded into the transfer counter and the DMA transfer is continued
DMA interrupt request generation timing	When the transfer counter register changes from "0001_{16}" to "0000_{16}".
DMA startup	• Single transfer Transfer starts when DMA transfer count register is more than "0001_{16}" and the DMA is requested after “01_2” is written to the channel i transfer mode select bits • Repeat transfer Transfer starts when the DMA is requested after “11_2” is written to the channel i transfer mode select bits
DMA shutdown	• Single transfer When “00_2” is written to the channel i transfer mode select bits and DMA transfer count register becomes "0000_{16}" by DMA transfer or write • Repeat transfer When “00_2” is written to the channel i transfer mode select bits
Reload timing	When the transfer counter register changes from "0001_{16}" to "0000_{16}" in repeat transfer mode.
Reading / writing the register	Registers are always read/write enabled.
Number of DMA transfer cycles	Between SFR and internal RAM : 3 cycles Between external I/O and external memory : minimum 3 cycles

Note: DMA transfer is not effective to any interrupt.

DMAi request cause select register (i = 0 to 3)(Note 1)



Symbol	Address	When reset
DMiSL	0378_{16} to $037B_{16}$	$0X000000_2$

Bit symbol	Bit name	Function	R	W
DSEL0	DMA request cause select bit (Note 2)	b4 b3 b2 b1 b0 0 0 0 0 0 : Software trigger 0 0 0 0 1 : Falling edge of INTi pin (Note 3) 0 0 0 1 0 : Two edges of INTi pin (Note 3) 0 0 0 1 1 : Timer A0 0 0 1 0 0 : Timer A1 0 0 1 0 1 : Timer A2 0 0 1 1 0 : Timer A3 0 0 1 1 1 : Timer A4 0 1 0 0 0 : Timer B0 0 1 0 0 1 : Timer B1 0 1 0 1 0 : Timer B2 0 1 0 1 1 : Timer B3 0 1 1 0 0 : Timer B4 0 1 1 0 1 : Timer B5 0 1 1 1 0 : UART0 transmit 0 1 1 1 1 : UART0 receive 1 0 0 0 0 : UART1 transmit 1 0 0 0 1 : UART1 receive 1 0 0 1 0 : UART2 transmit 1 0 0 1 1 : UART2 receive/ACK (Note 4) 1 0 1 0 0 : UART3 transmit 1 0 1 0 1 : UART3 receive/ACK (Note 4) 1 0 1 1 0 : UART4 transmit 1 0 1 1 1 : UART4 receive/ACK (Note 4) 1 1 0 0 0 : A-D conversion 1 1 0 0 1 to 1 1 1 1 1 : Inhibit	○	○
DSEL1			○	○
DSEL2			○	○
DSEL3			○	○
DSEL4			○	○
DSR	Software DMA request bit (Note 5)	If software trigger is selected, a DMA request is generated by setting this bit to "1" (When read, the value of this bit is always "0")	○	○
Nothing is assigned. When write, set "0". When read, its content is indeterminate.			–	–
DRQ	DMA request bit (Note 5,6)	0 : Not requested 1 : Requested	○	○

Note 1: Please refer to DMAC precautions.
Note 2: Set DMA inhibit before changing the DMA request cause. Set DRQ to "1" simultaneously.
e.g.) MOV.B #083h, DMiSL ; Set timer A0
Note 3: DMA0-INT0, DMA1-INT1, DMA2-INT2, and DMA3-INT3 correspond to DMAi and INTi. However, when INT3 pin becomes data bus in microprocessor mode, DMA3-INT3 cannot be used.
Note 4: UARTi reception and ACK switching are effected using the UARTi special mode register and UARTi special mode register 2.
Note 5: When setting DSR to "1", set DRQ to "1" using OR instruction etc. simultaneously.
e.g.) OR.B #0A0h, DMiSL
Note 6: Do not write "0" to this bit. There is no need to clear the DMA request bit.

Figure 1.11.2. DMAC register (1)

DMA mode register 0
(CPU internal register)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol DMD0 | When reset 00₁₆

Bit symbol	Bit name	Function	R	W
MD00	Channel 0 transfer mode select bit	b1 b0 0 0 : DMA inhibit 0 1 : Single transfer 1 0 : Reserved 1 1 : Repeat transfer	○	○
MD01			○	○
BW0	Channel 0 transfer unit select bit	0 : 8 bits 1 : 16 bits	○	○
RW0	Channel 0 transfer direction select bit	0 : Fixed address to memory (forward direction) 1 : Memory (forward direction) to fixed address	○	○
MD10	Channel 1 transfer mode select bit	b5 b4 0 0 : DMA inhibit 0 1 : Single transfer 1 0 : Reserved 1 1 : Repeat transfer	○	○
MD11			○	○
BW1	Channel 1 transfer unit select bit	0 : 8 bits 1 : 16 bits	○	○
RW1	Channel 1 transfer direction select bit	0 : Fixed address to memory (forward direction) 1 : Memory (forward direction) to fixed address	○	○

DMA mode register 1
(CPU internal register)

b7 b6 b5 b4 b3 b2 b1 b0

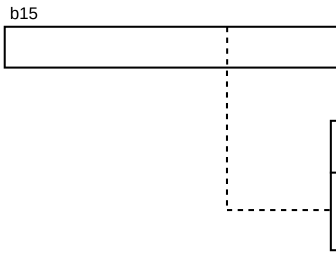
Symbol DMD1 | When reset 00₁₆

Bit symbol	Bit name	Function	R	W
MD20	Channel 2 transfer mode select bit	b1 b0 0 0 : DMA inhibit 0 1 : Single transfer 1 0 : Reserved 1 1 : Repeat transfer	○	○
MD21			○	○
BW2	Channel 2 transfer unit select bit	0 : 8 bits 1 : 16 bits	○	○
RW2	Channel 2 transfer direction select bit	0 : Fixed address to memory (forward direction) 1 : Memory (forward direction) to fixed address	○	○
MD30	Channel 3 transfer mode select bit	b5 b4 0 0 : DMA inhibit 0 1 : Single transfer 1 0 : Reserved 1 1 : Repeat transfer	○	○
MD31			○	○
BW3	Channel 3 transfer unit select bit	0 : 8 bits 1 : 16 bits	○	○
RW3	Channel 3 transfer direction select bit	0 : Fixed address to memory (forward direction) 1 : Memory (forward direction) to fixed address	○	○

Figure 1.11.3. DMAC register (2)

DMAi transfer count register (i = 0 to 3)
(CPU internal register)

Symbol	When reset
DCT0	XXXX$_{16}$
DCT1	XXXX$_{16}$
DCT2 (bank 1;R0) (Note 1)	0000$_{16}$
DCT3 (bank 1;R1) (Note 1)	0000$_{16}$

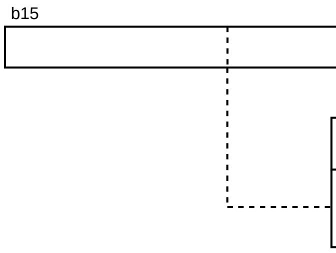


Function	Transfer count specification	R	W
• Transfer counter Set transfer number	0000$_{16}$ to FFFF$_{16}$	○	○

Note 1: When setting DCT2 and DCT3, set "1" to the register bank select flag (B flag) of flag register (FLG), and then set desired value to R0 and R1 of register bank 1.
Note 2: When "0" is set to this register, data transfer is not done even if DMA is requested.

DMAi transfer count reload register (i = 0 to 3)
(CPU internal register)

Symbol	When reset
DRC0	XXXX$_{16}$
DRC1	XXXX$_{16}$
DRC2 (bank 1;R2) (Note 1)	0000$_{16}$
DRC3 (bank 1;R3) (Note 1)	0000$_{16}$



Function	Transfer count specification	R	W
• Transfer count register reload value Set transfer number	0000$_{16}$ to FFFF$_{16}$	○	○

Note: When setting DRC2 and DRC3, set "1" to the register bank select flag (B flag) of flag register (FLG), and then set desired value to R2 and R3 of register bank 1.

Figure 1.11.4. DMAC register (3)

DMAi memory address register (i = 0 to 3)
(CPU internal register)

b23 b0

Symbol	When reset
DMA0	XXXXXX16
DMA1	XXXXXX16
DMA2 (bank 1;A0) (Note 1)	00000016
DMA3 (bank 1;A1) (Note 1)	00000016

Function	Transfer address specification area	R	W
• Memory address (Note 2) Set source or destination memory address	00000016 to FFFFFF16 (16 Mbytes area)	○	○

Note 1: When setting DMA2 and DMA3, set "1" to the register bank select flag (B flag) of flag register (FLG), and set desired value to A0 and A1 of register bank 1.
Note 2: When the transfer direction select bit is "0" (fixed address to memory), this register is destination memory address.
When the transfer direction select bit is "1" (memory to fixed address), this register is source memory address.

DMAi SFR address register (i = 0 to 3)
(CPU internal register)

b23 b0

Symbol	When reset
DSA0	XXXXXX16
DSA1	XXXXXX16
DSA2 (bank 1;SB) (Note 1)	00000016
DSA3 (bank 1;FB) (Note 1)	00000016

Function	Transfer address specification area	R	W
• SFR address (Note 2) Set source or destination fixed address	00000016 to FFFFFF16 (16 Mbytes area)	○	○

Note 1: When setting DSA2, set "1" to the register bank select flag (B flag) of flag register (FLG), and set desired value to SB of register bank 1.
When setting DSA3, set "1" to the register bank select flag (B flag) of flag register (FLG), and set desired value to FB of register bank 1.
Note 2: When the transfer direction select bit is "0" (fixed address to memory), this register is destination fixed address.
When the transfer direction select bit is "1" (memory to fixed address), this register is source fixed address.

DMAi memory address reload register (i = 0 to 3)
(CPU internal register)

b23 b0

Symbol	When reset
DRA0	XXXXXX16
DRA1	XXXXXX16
DRA2 (SVP) (Note)	XXXXXX16
DRA3 (VCT) (Note)	XXXXXX16

Function	Transfer address specification area	R	W
• Memory address register reload value Set source or destination memory address	00000016 to FFFFFF16 (16 Mbytes area)	○	○

Note: When setting DRA2, set desired value to save PC register (SVP).
When setting DRA3, set desired value to vector register (VCT).

Figure 1.11.5. DMAC register (4)

(1) Transfer cycle

The transfer cycle consists of the bus cycle in which data is read from memory or from the SFR area (source read) and the bus cycle in which the data is written to memory or to the SFR area (destination write). The number of read and write bus cycles depends on the source and destination addresses. In memory expansion mode and microprocessor mode, the number of read and write bus cycles also depends on the level of the BYTE pin. Also, the bus cycle itself is longer when software waits are inserted.

(a) Effect of source and destination addresses

When 16-bit data is transferred on a 16-bit data bus, and the source and destination both start at odd addresses, there are one more source read cycle and destination write cycle than when the source and destination both start at even addresses.

(b) Effect of external data bus width control register

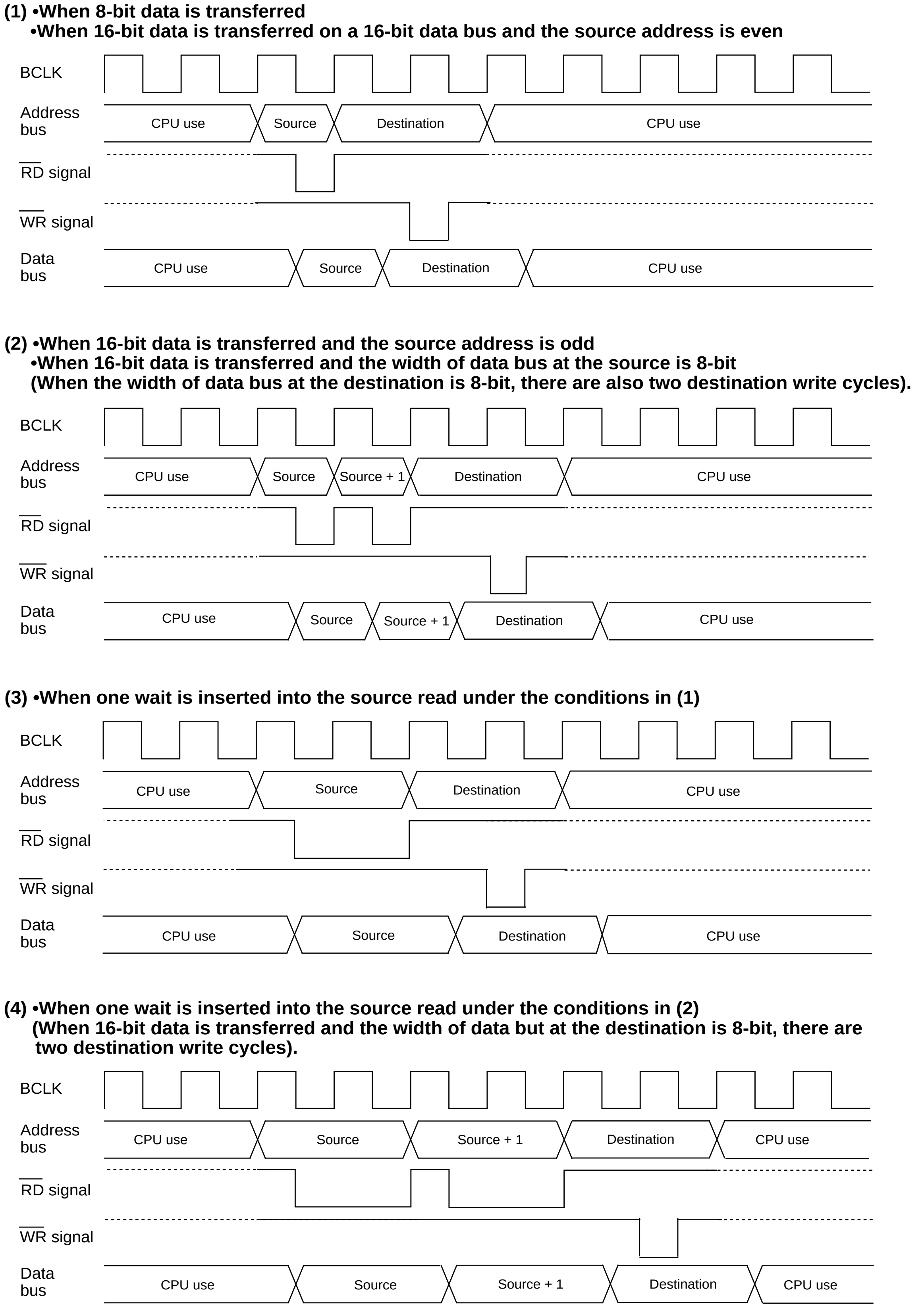
When in memory expansion mode or microprocessor mode, the transfer cycle changes according to the data bus width at the source and destination.

1. When transferring 16 bits of data and the data bus width at the source and at the destination is 8 bits (data bus width bit = “0”), there are two 8-bit data transfers. Therefore, two bus cycles are required for reading and two cycles for writing.
2. When transferring 16 bits of data and the data bus width at the source is 8 bits (data bus width bit = “0”) and the data bus width at the destination is 16 bits (data bus width bit = “1”), the data is read in two 8-bit blocks and written as 16-bit data. Therefore, two bus cycles are required for reading and one cycle for writing.
3. When transferring 16 bits of data and the data bus width at the source is 16 bits (data bus width bit = “1”) and the data bus width at the destination is 8 bits (data bus width bit = “0”), 16 bits of data are read and written as two 8-bit blocks. Therefore, one bus cycle is required for reading and two cycles for writing.

(c) Effect of software wait

When the SFR area or a memory area with a software wait is accessed, the number of cycles is increased for the wait by 1 bus cycle. The length of the cycle is determined by BCLK.

Figure 1.11.6 shows the example of the transfer cycles for a source read. Figure 1.11.6 shows the destination is external area, the destination write cycle is shown as two cycle (one bus cycle) and the source read cycles for the different conditions. In reality, the destination write cycle is subject to the same conditions as the source read cycle, with the transfer cycle changing accordingly. When calculating the transfer cycle, remember to apply the respective conditions to both the destination write cycle and the source read cycle. For example (2) in Figure 1.11.6, if data is being transferred in 16-bit units on an 8-bit bus, two bus cycles are required for both the source read cycle and the destination write cycle.



(1) •When 8-bit data is transferred
•When 16-bit data is transferred on a 16-bit data bus and the source address is even

(2) •When 16-bit data is transferred and the source address is odd
•When 16-bit data is transferred and the width of data bus at the source is 8-bit
(When the width of data bus at the destination is 8-bit, there are also two destination write cycles).

(3) •When one wait is inserted into the source read under the conditions in (1)

(4) •When one wait is inserted into the source read under the conditions in (2)
(When 16-bit data is transferred and the width of data but at the destination is 8-bit, there are two destination write cycles).

Note: The same timing changes occur with the respective conditions at the destination as at the source.

Figure 1.11.6. Example of the transfer cycles for a source read

(2) DMAC transfer cycles

Any combination of even or odd transfer read and write addresses is possible. Table 1.11.2 shows the number of DMAC transfer cycles.

The number of DMAC transfer cycles can be calculated as follows:

No. of transfer cycles per transfer unit = No. of read cycles x j + No. of write cycles x k

Table 1.11.2. No. of DMAC transfer cycles

Transfer unit	Bus width	Access address	Single-chip mode		Memory expansion mode Microprocessor mode	
			No. of read cycles	No. of write cycles	No. of read cycles	No. of write cycles
8-bit transfers (BWi = “0”)	16-bit (DSi = “1”)	Even	1	1	1	1
		Odd	1	1	1	1
	8-bit (DSi = “0”)	Even	—	—	1	1
		Odd	—	—	1	1
16-bit transfers (BWi = “1”)	16-bit (DSi = “1”)	Even	1	1	1	1
		Odd	2	2	2	2
	8-bit (DSi = “0”)	Even	—	—	2	2
		Odd	—	—	2	2

Coefficient j, k

Internal Memory			External Memory					
Internal ROM/RAM		SFR area	Separate bus				Multiplex bus	
No wait	With wait		No wait	One wait	Two waits	Three waits	Two waits	Three waits
j=1 k=1	j=2 k=2	j=2 k=2	j=1 k=2	j=2 k=2	j=3 k=3	j=4 k=4	j=3 k=3	j=4 k=4

DMA Request Bit

The DMAC can issue DMA requests using preselected DMA request factors for each channel as triggers.

The DMA transfer request factors include the reception of DMA request signals from the internal peripheral functions, software DMA factors generated by the program, and external factors using input from external interrupt signals.

See the description of the DMAi factor selection register for details of how to select DMA request factors.

DMA requests are received as DMA requests when the DMAi request bit is set to “1” and the channel i transfer mode select bits are “01” or “11”. Therefore, even if the DMAi request bit is “1”, no DMA request is received if the channel i transfer mode select bit is “00”. In this case, DMAi request bit is cleared.

Because the channel i transfer mode select bits default to “00” after a reset, remember to set the channel i transfer mode select bit for the channel to be activated after setting the DMAC related registers. This enables receipt of the DMA requests for that channel, and DMA transfers are then performed when the DMAi request bit is set.

The following describes when the DMAi request bit is set and cleared.

(1) Internal factors

The DMAi request flag is set to “1” in response to internal factors at the same time as the interrupt request bit of the interrupt control register for each factor is set. This is because, except for software trigger DMA factors, they use the interrupt request signals output by each function.

The DMAi request bit is cleared to "0" when the DMA transfer starts or the DMA transfer is in disable state (channel i transfer mode select bits are "00" and the DMAi transfer count register is "0").

(2) External factors

These are DMA request factors that are generated by the input edge from the $\overline{\text{INTi}}$ pin (where i indicates the DMAC channel). When the $\overline{\text{INTi}}$ pin is selected by the DMAi request factor select bit as an external factor, the inputs from these pins become the DMA request signals.

When an external factor is selected, the DMAi request bit is set, according to the function specified in the DMA request factor select bit, on either the falling edge of the signal input via the $\overline{\text{INTi}}$ pins, or both edges.

When an external factor is selected, the DMAi request bit is cleared, in the same way as the DMAi request bit is cleared for internal factors, when the DMA transfer starts or the DMA transfer is in disable state.

(3) Relationship between external factor request input and DMAi request flag, and DMA transfer timing

When the request inputs to DMAi occur in the same sampling cycle (between the falling edge of BCLK and the next falling edge), the DMAi request bits are set simultaneously, but if the DMAi enable bits are all set, DMA0 takes priority and the transfer starts. When one transfer unit is complete, the bus privilege is returned to the CPU. When the CPU has completed one bus access, DMA1 transfer starts, and, when one transfer unit is complete, the privilege is again returned to the CPU.

The priority is as follows: DMA0 > DMA1 > DMA2 > DMA3.

Figure 1.11.7. DMA transfer example by external factors shows what happens when DMA0 and DMA1 requests occur in the same sampling cycle.

In this example, DMA transfer request signals are input simultaneously from external factors and the DMA transfers are executed in the minimum cycles.

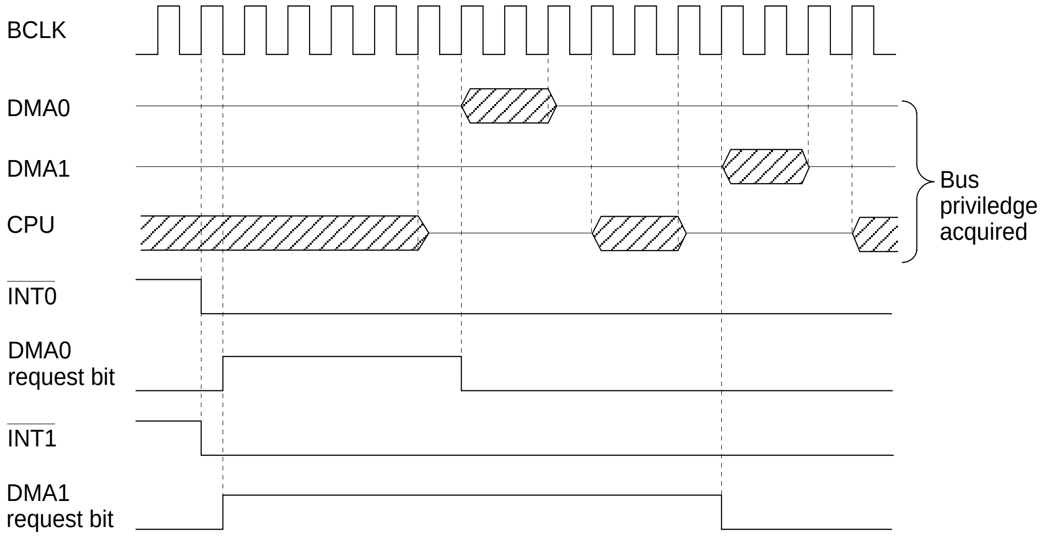


Figure 1.11.7. DMA transfer example by external factors

Precautions for DMAC

(1) Do not clear the DMA request bit of the DMAi request cause select register.

In M16C/80, when a DMA request is generated while the channel is disabled (Note), the DMA transfer is not executed and the DMA request bit is cleared automatically.

Note :The DMA is disabled or the transfer count register is "0".

(2) When DMA transfer is done by a software trigger, set DSR and DRQ of the DMAi request cause select register to "1" simultaneously using the OR instruction.

```
e.g.) OR.B   #0A0h, DMiSL       ; DMiSL is DMAi request cause select register
```

(3) When changing the DMAi request cause select bit of the DMAi request cause select register, set "1" to the DMA request bit, simultaneously. In this case, set the corresponding DMA channel to disabled before changing the DMAi request cause select bit. At least 2 instructions are needed from the instruction to write to the DMAi request cause select register to enable DMA.

Example) When DMA request cause is changed to timer A0 and using DMA0 in single transfer after DMA initial setting

```
push.w    R0                    ; Store R0 register
stc       DMD0, R0              ; Read DMA mode register 0
and.b     #11111100b, R0L       ; Clear DMA0 transfer mode select bit to "00"
ldc       R0, DMD0              ; DMA0 disabled
mov.b     #10000011b, DM0SL     ; Select timer A0
                                ; (Write "1" to DMA request bit simultaneously)
mov.b     R0L, R0L              ; Dummy cycle             }  At least 2 instructions are
or.b      #00000001b, R0L       ; Set DMA0 single transfer }  needed until DMA enabled.
ldc       R0, DMD0              ; DMA0 enabled
pop.w     R0                    ; Restore R0 register
```

Timer

There are eleven 16-bit timers. These timers can be classified by function into timers A (five) and timers B (six). All these timers function independently. Figures 1.12.1 and 1.12.2 show the block diagram of timers.

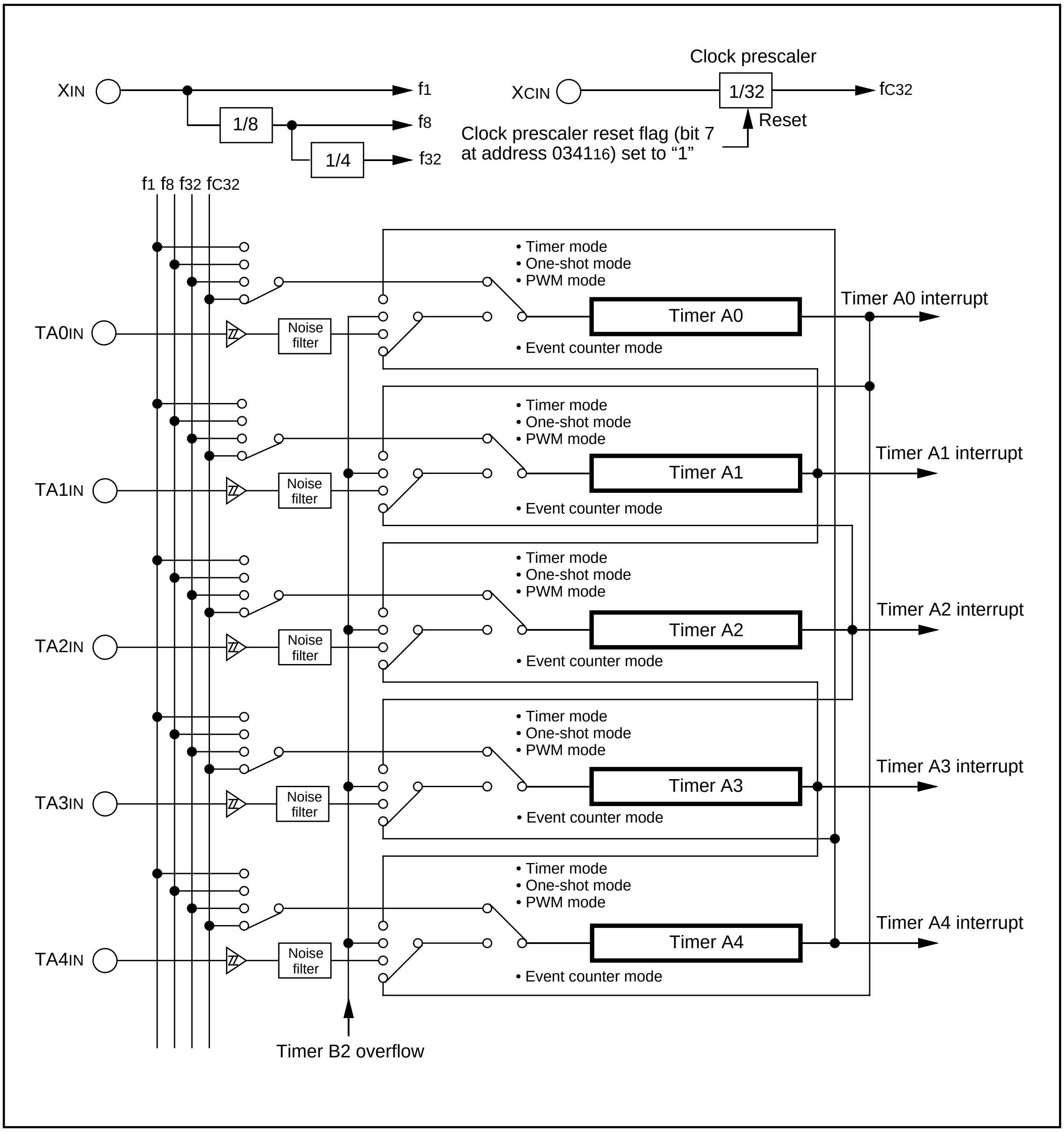


Figure 1.12.1. Timer A block diagram

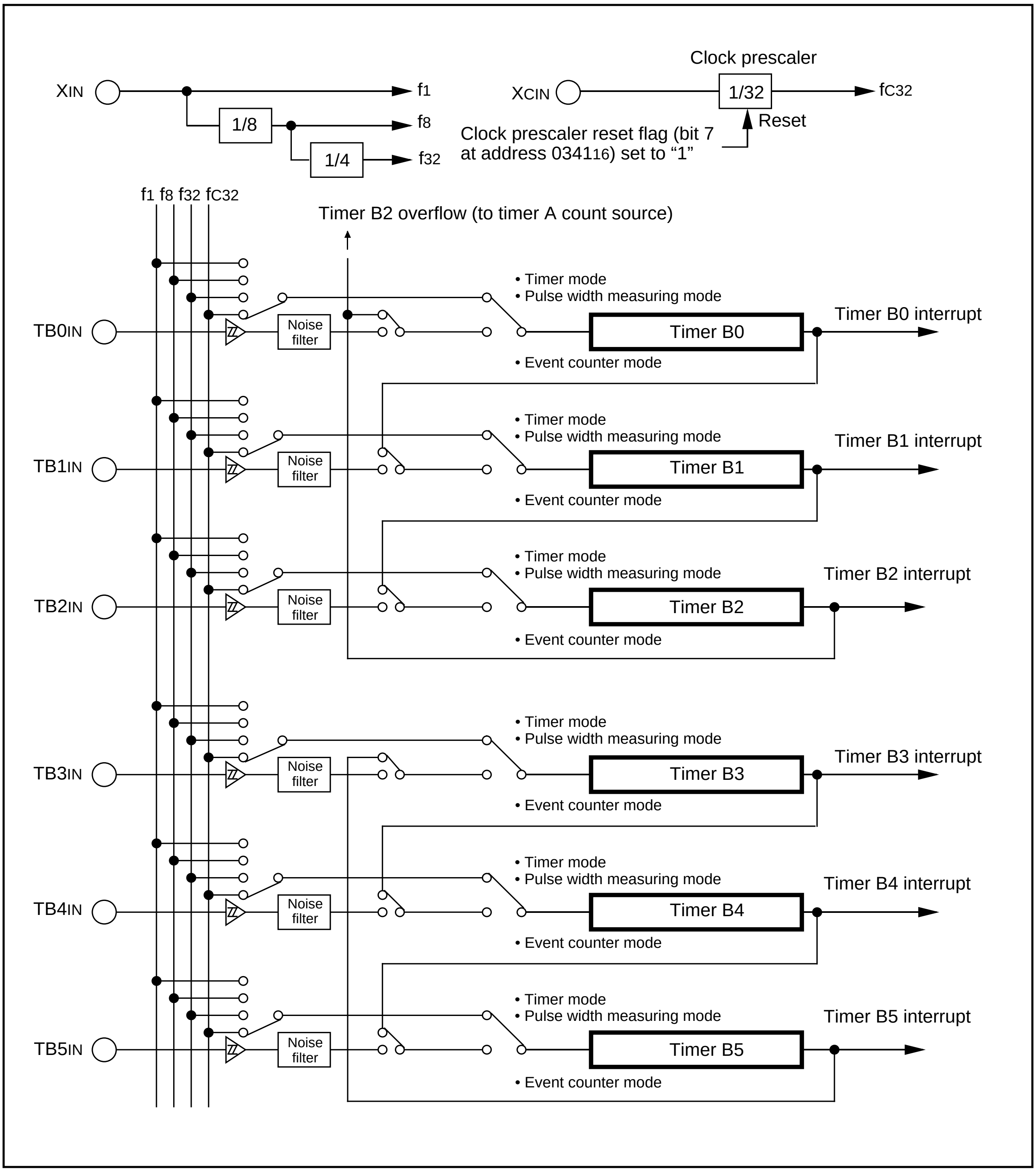


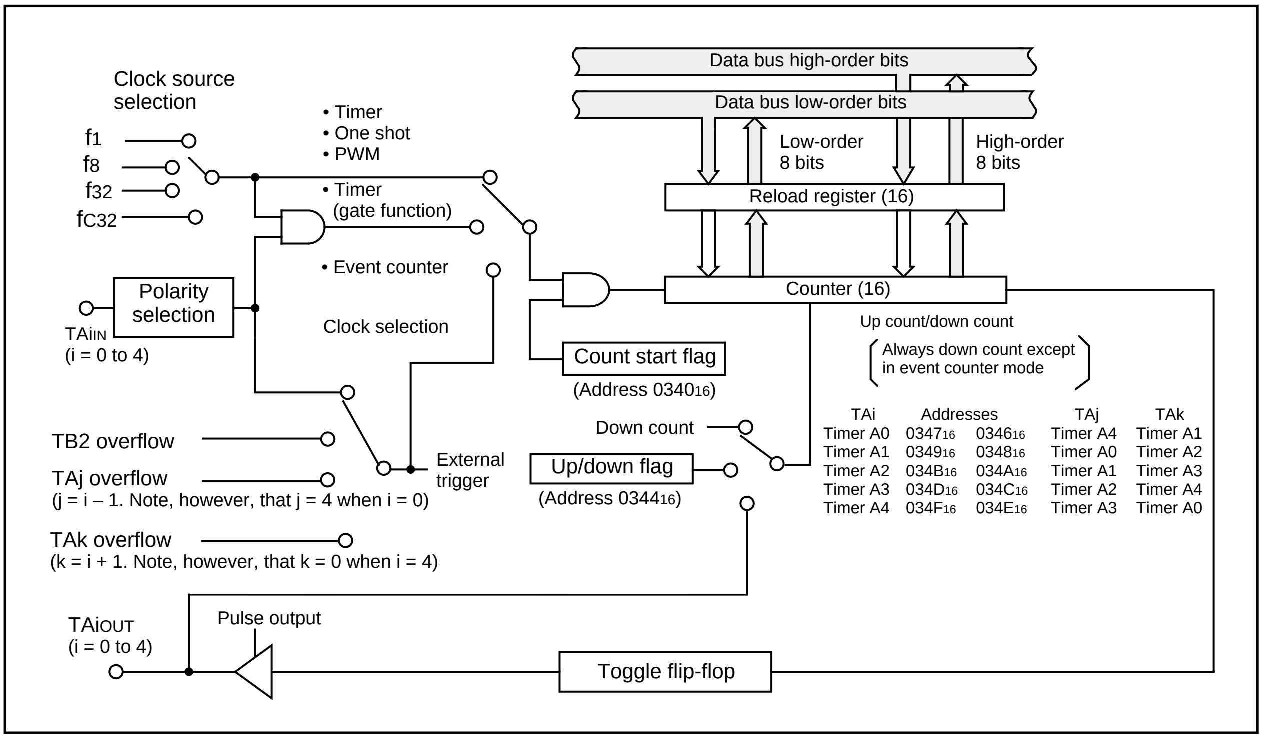
Figure 1.12.2. Timer B block diagram

Timer A

Figure 1.13.1 shows the block diagram of timer A. Figures 1.13.2 to 1.13.4 show the timer A-related registers. Except in event counter mode, timers A0 through A4 all have the same function. Use the timer Ai mode register (i = 0 to 4) bits 0 and 1 to choose the desired mode.

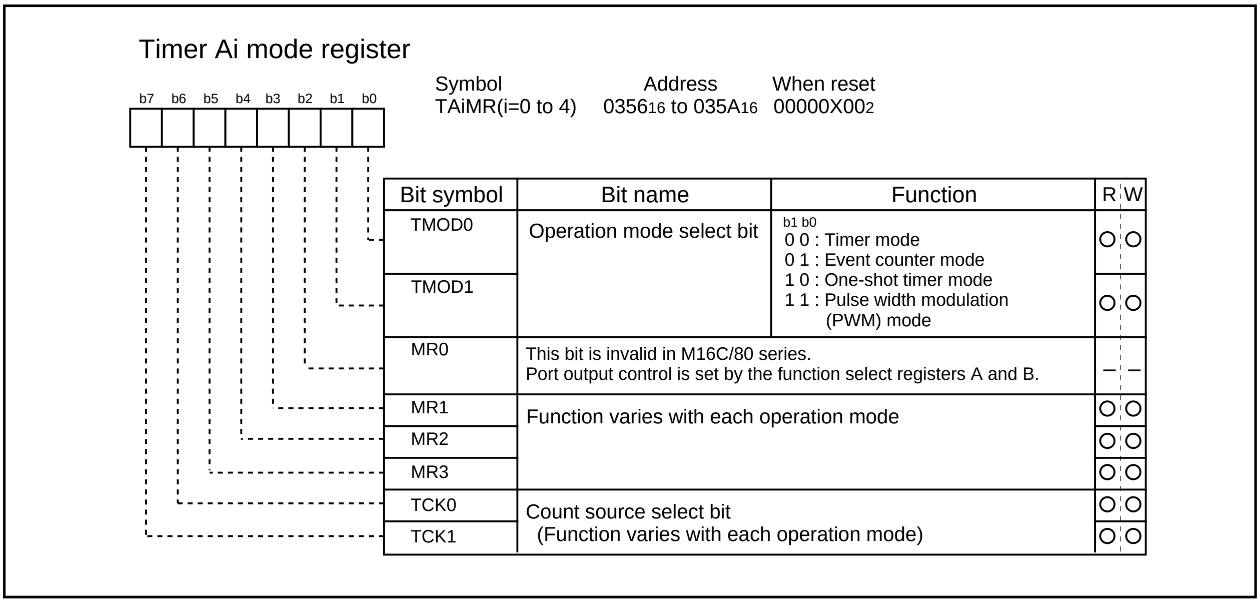
Timer A has the four operation modes listed as follows:

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external source or a timer over flow.
- One-shot timer mode: The timer stops counting when the count reaches “0000$_{16}$”.
- Pulse width modulation (PWM) mode: The timer outputs pulses of a given width.



TAi	Addresses		TAj	TAk
Timer A0	0347$_{16}$	0346$_{16}$	Timer A4	Timer A1
Timer A1	0349$_{16}$	0348$_{16}$	Timer A0	Timer A2
Timer A2	034B$_{16}$	034A$_{16}$	Timer A1	Timer A3
Timer A3	034D$_{16}$	034C$_{16}$	Timer A2	Timer A4
Timer A4	034F$_{16}$	034E$_{16}$	Timer A3	Timer A0

Figure 1.13.1. Block diagram of timer A



Timer Ai mode register

Symbol	Address	When reset
TAiMR(i=0 to 4)	0356$_{16}$ to 035A$_{16}$	00000X00$_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot timer mode 1 1 : Pulse width modulation (PWM) mode	○	○
TMOD1			○	○
MR0	This bit is invalid in M16C/80 series. Port output control is set by the function select registers A and B.		–	–
MR1	Function varies with each operation mode		○	○
MR2			○	○
MR3			○	○
TCK0	Count source select bit (Function varies with each operation mode)		○	○
TCK1			○	○

Figure 1.13.2. Timer A-related registers (1)

Timer Ai register (Note 1)

(b15) b7 (b8) b0 b7 b0

Symbol	Address	When reset
TA0	0347$_{16}$,0346$_{16}$	Indeterminate
TA1	0349$_{16}$,0348$_{16}$	Indeterminate
TA2	034B$_{16}$,034A$_{16}$	Indeterminate
TA3	034D$_{16}$,034C$_{16}$	Indeterminate
TA4	034F$_{16}$,034E$_{16}$	Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts an internal count source	0000$_{16}$ to FFFF$_{16}$	○	○
• Event counter mode Counts pulses from an external source or timer overflow	0000$_{16}$ to FFFF$_{16}$	○	○
• One-shot timer mode (Note 2, 3) Counts a one shot width	0000$_{16}$ to FFFF$_{16}$	×	○
• Pulse width modulation mode (16-bit PWM) (Note 2, 4) Functions as a 16-bit pulse width modulator	0000$_{16}$ to FFFE$_{16}$	×	○
• Pulse width modulation mode (8-bit PWM) (Note 2, 4) Timer low-order address functions as an 8-bit prescaler and high-order address functions as an 8-bit pulse width modulator	00$_{16}$ to FE$_{16}$ (High-order address) 00$_{16}$ to FF$_{16}$ (Low-order address)	×	○

Note 1: Read and write data in 16-bit units.
Note 2: Use MOV instruction to write to this register.
Note 3: When the timer Ai register is set to "0000$_{16}$", the counter does not operate and the timer Ai interrupt request is not generated. When the pulse is set to output, the pulse does not output from the TAiOUT pin.
Note 4: When the timer Ai register is set to "0000$_{16}$", the pulse width modulator does not operate and the output level of the TAiOUT pin remains "L" level, therefore the timer Ai interrupt request is not generated. This also occurs in the 8-bit pulse width modulator mode when the significant 8 high-order bits in the timer Ai register are set to "00$_{16}$".

Count start flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TABSR	0340$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TA1S	Timer A1 count start flag		○	○
TA2S	Timer A2 count start flag		○	○
TA3S	Timer A3 count start flag		○	○
TA4S	Timer A4 count start flag		○	○
TB0S	Timer B0 count start flag		○	○
TB1S	Timer B1 count start flag		○	○
TB2S	Timer B2 count start flag		○	○

Up/down flag (Note)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UDF	0344$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
TA0UD	Timer A0 up/down flag	0 : Down count 1 : Up count This specification becomes valid when the up/down flag content is selected for up/down switching cause	○	○
TA1UD	Timer A1 up/down flag		○	○
TA2UD	Timer A2 up/down flag		○	○
TA3UD	Timer A3 up/down flag		○	○
TA4UD	Timer A4 up/down flag		○	○
TA2P	Timer A2 two-phase pulse signal processing select bit	0 : two-phase pulse signal processing disabled 1 : two-phase pulse signal processing enabled When not using the two-phase pulse signal processing function, set the select bit to "0"	×	○
TA3P	Timer A3 two-phase pulse signal processing select bit		×	○
TA4P	Timer A4 two-phase pulse signal processing select bit		×	○

Note: Use MOV instruction to write to this register.

Figure 1.13.3. Timer A-related registers (2)

One-shot start flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
ONSF	0342_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
TA0OS	Timer A0 one-shot start flag	1 : Timer start When read, the value is “0”	○	○
TA1OS	Timer A1 one-shot start flag		○	○
TA2OS	Timer A2 one-shot start flag		○	○
TA3OS	Timer A3 one-shot start flag		○	○
TA4OS	Timer A4 one-shot start flag		○	○
TAZIE	Z phase input enable bit	0 : Invalid 1 : Valid	○	○
TA0TGL	Timer A0 event/trigger select bit	b7 b6 0 0 : Input on $TA0_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA4 overflow is selected 1 1 : TA1 overflow is selected	○	○
TA0TGH			○	○

Note: Set the corresponding function select register A to I/O port, and port direction register to “0”.

Trigger select register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TRGSR	0343_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
TA1TGL	Timer A1 event/trigger select bit	b1 b0 0 0 : Input on $TA1_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA0 overflow is selected 1 1 : TA2 overflow is selected	○	○
TA1TGH			○	○
TA2TGL	Timer A2 event/trigger select bit	b3 b2 0 0 : Input on $TA2_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA1 overflow is selected 1 1 : TA3 overflow is selected	○	○
TA2TGH			○	○
TA3TGL	Timer A3 event/trigger select bit	b5 b4 0 0 : Input on $TA3_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA2 overflow is selected 1 1 : TA4 overflow is selected	○	○
TA3TGH			○	○
TA4TGL	Timer A4 event/trigger select bit	b7 b6 0 0 : Input on $TA4_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA3 overflow is selected 1 1 : TA0 overflow is selected	○	○
TA4TGH			○	○

Note: Set the corresponding function select register A to I/O port, and port direction register to “0”.

Clock prescaler reset flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
CPSRF	0341_{16}	$0XXXXXXX_2$

Bit symbol	Bit name	Function	R	W
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			–	–
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is “0”)	○	○

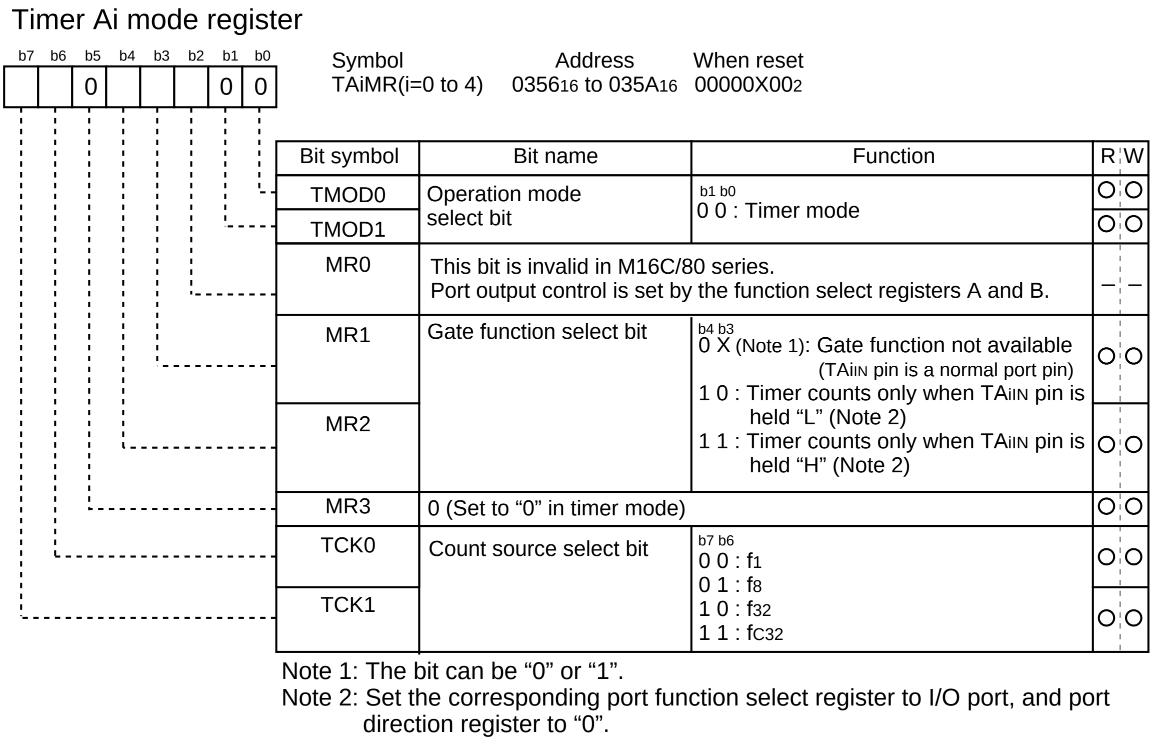
Figure 1.13.4. Timer A-related registers (3)

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 1.13.1.) Figure 1.13.5 shows the timer Ai mode register in timer mode.

Table 1.13.1. Specifications of timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, fc_{32}
Count operation	• Down count • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	When the timer underflows
TAiIN pin function	Programmable I/O port or gate input
TAiOUT pin function	Programmable I/O port or pulse output (Setting by the corresponding function select registers A and B)
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	• When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter • When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)
Select function	• Gate function Counting can be started and stopped by the TAiIN pin's input signal • Pulse output function Each time the timer underflows, the TAiOUT pin's polarity is reversed



Symbol	Address	When reset
TAiMR(i=0 to 4)	0356_{16} to $035A_{16}$	$00000X00_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	○	○
TMOD1			○	○
MR0	This bit is invalid in M16C/80 series. Port output control is set by the function select registers A and B.		–	–
MR1	Gate function select bit	b4 b3 0 X (Note 1): Gate function not available (TAiIN pin is a normal port pin) 1 0 : Timer counts only when TAiIN pin is held "L" (Note 2) 1 1 : Timer counts only when TAiIN pin is held "H" (Note 2)	○	○
MR2			○	○
MR3	0 (Set to "0" in timer mode)		○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : fc_{32}	○	○
TCK1			○	○

Note 1: The bit can be "0" or "1".
Note 2: Set the corresponding port function select register to I/O port, and port direction register to "0".

Figure 1.13.5. Timer Ai mode register in timer mode

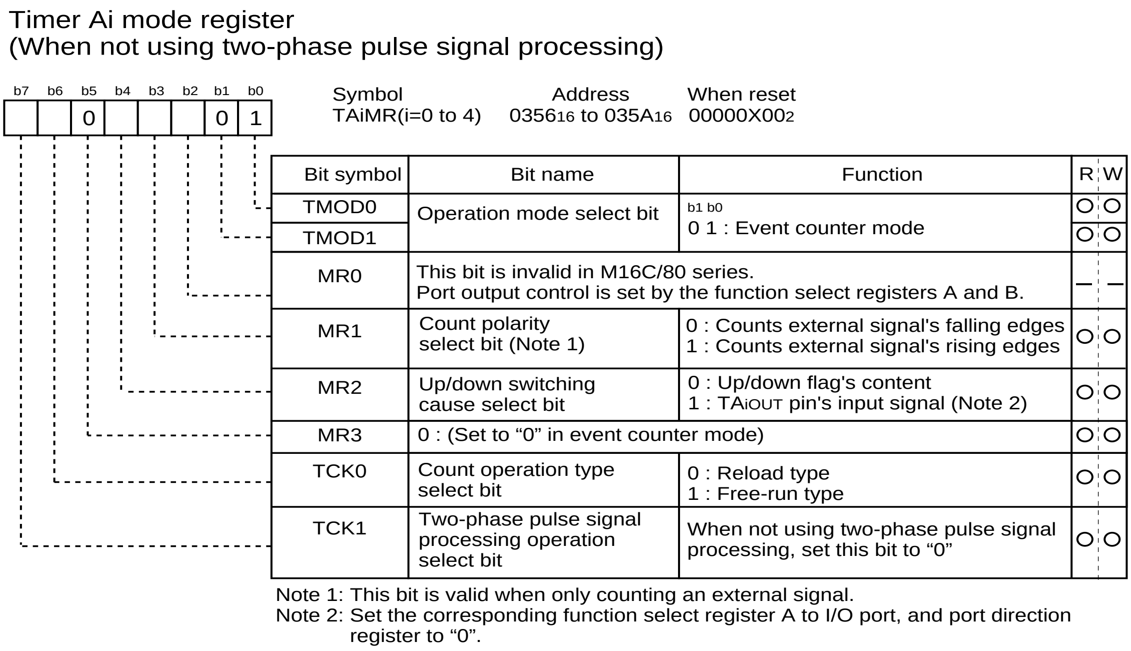
(2) Event counter mode

In this mode, the timer counts an external signal or an internal timer's overflow. Timers A0 and A1 can count a single-phase external signal. Timers A2, A3, and A4 can count a single-phase and a two-phase external signal. Table 1.13.2 lists timer specifications when counting a single-phase external signal. Figure 1.13.6 shows the timer Ai mode register in event counter mode. Table 1.13.3 lists timer specifications when counting a two-phase external signal. Figure 1.13.7 shows the timer Ai mode register in event counter mode.

Table 1.13.2. Timer specifications in event counter mode (when not processing two-phase pulse signal)

Item	Specification
Count source	• External signals input to TAi_{IN} pin (effective edge can be selected by software) • TB2 overflows or underflows, TAj overflows or underflows
Count operation	• Up count or down count can be selected by external signal or software • When the timer overflows or underflows, it reloads the reload register con tents before continuing counting (Note)
Divide ratio	• 1/ ($FFFF_{16}$ - n + 1) for up count • 1/ (n + 1) for down count n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer overflows or underflows
TAi_{IN} pin function	Programmable I/O port or count source input
TAi_{OUT} pin function	Programmable I/O port, pulse output, or up/down count select input (Setting by the corresponding function select registers A and B)
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	• When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter • When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)
Select function	• Free-run count function Even when the timer overflows or underflows, the reload register content is not reloaded to it • Pulse output function Each time the timer overflows or underflows, the TAi_{OUT} pin's polarity is reversed

Note: This does not apply when the free-run function is selected.



Timer Ai mode register
(When not using two-phase pulse signal processing)

b7	b6	b5	b4	b3	b2	b1	b0
		0				0	1

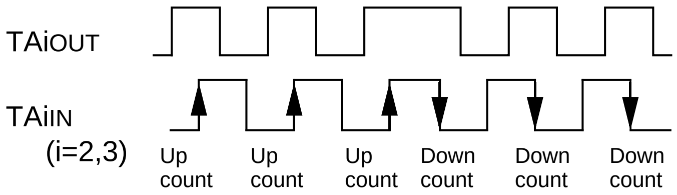
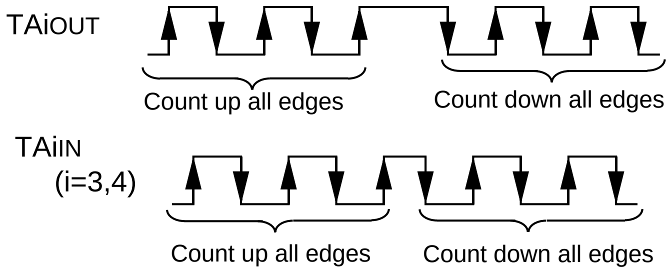
Symbol	Address	When reset
TAiMR(i=0 to 4)	0356_{16} to $035A_{16}$	$00000X00_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	○	○
TMOD1			○	○
MR0	This bit is invalid in M16C/80 series. Port output control is set by the function select registers A and B.		–	–
MR1	Count polarity select bit (Note 1)	0 : Counts external signal's falling edges 1 : Counts external signal's rising edges	○	○
MR2	Up/down switching cause select bit	0 : Up/down flag's content 1 : TAi_{OUT} pin's input signal (Note 2)	○	○
MR3	0 : (Set to "0" in event counter mode)		○	○
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	○	○
TCK1	Two-phase pulse signal processing operation select bit	When not using two-phase pulse signal processing, set this bit to "0"	○	○

Note 1: This bit is valid when only counting an external signal.
Note 2: Set the corresponding function select register A to I/O port, and port direction register to "0".

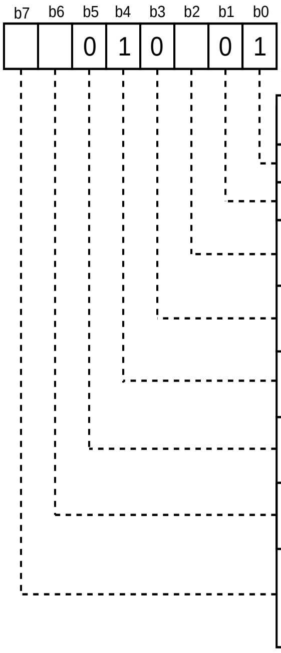
Figure 1.13.6. Timer Ai mode register in event counter mode

Table 1.13.3. Timer specifications in event counter mode

Item	Specification
Count source	• Two-phase pulse signals input to TAiIN or TAiOUT pin
Count operation	• Up count or down count can be selected by two-phase pulse signal • When the timer overflows or underflows, the reload register content is reloaded and the timer starts over again (Note)
Divide ratio	• 1/ (FFFF$_{16}$ - n + 1) for up count • 1/ (n + 1) for down count n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	Timer overflows or underflows
TAiIN pin function	Two-phase pulse input
TAiOUT pin function	Two-phase pulse input (Set the corresponding function select registers A to I/O port)
Read from timer	Count value can be read out by reading timer A2, A3, or A4 register
Write to timer	• When counting stopped When a value is written to timer A2, A3, or A4 register, it is written to both reload register and counter • When counting in progress When a value is written to timer A2, A3, or A4 register, it is written to only reload register. (Transferred to counter at next reload time.)
Select function (Note 2)	• Normal processing operation (TimerA2 and timer A3) The timer counts up rising edges or counts down falling edges on the TAiIN pin when input signal on the TAiOUT pin is “H”  TAiOUT TAiIN (i=2,3) Up count Up count Up count Down count Down count Down count • Multiply-by-4 processing operation (TimerA3 and timer A4) If the phase relationship is such that the TAiIN pin goes “H” when the input signal on the TAiOUT pin is “H”, the timer counts up rising and falling edges on the TAiOUT and TAiIN pins. If the phase relationship is such that the TAiIN pin goes “L” when the input signal on the TAiOUT pin is “H”, the timer counts down rising and falling edges on the TAiOUT and TAiIN pins.  TAiOUT Count up all edges Count down all edges TAiIN (i=3,4) Count up all edges Count down all edges

(when processing two-phase pulse signal with timers A2, A3, and A4)
Note 1: This does not apply when the free-run function is selected.
Note 2: Timer A3 is selectable. Timer A2 is fixed to normal processing operation and timer A4 is fixed to multiply-by-4 operation.

Timer Ai mode register
(When using two-phase pulse signal processing)



Symbol	Address	When reset
TAiMR(i=2 to 4)	0358_{16} to $035A_{16}$	$00000X00_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	○	○
TMOD1			○	○
MR0	This bit is invalid in M16C/80 series. (Note 1) Port output control is set by the function select registers A and B.		–	–
MR1	0 (Set to “0” when using two-phase pulse signal processing)		○	○
MR2	1 (Set to “1” when using two-phase pulse signal processing)		○	○
MR3	0 (Set to “0” when using two-phase pulse signal processing)		○	○
TCK0	Count operation type select bit	0 : Reload type 1 : Free-run type	○	○
TCK1	Two-phase pulse processing operation select bit (Note 2)(Note 3)	0 : Normal processing operation 1 : Multiply-by-4 processing operation	○	○

Note 1: Set the corresponding function select register A to I/O port.
Note 2: This bit is valid for timer A3 mode register.
Timer A2 is fixed to normal processing operation and timer A4 is fixed to multiply-by-4 processing operation.
Note 3: When performing two-phase pulse signal processing, make sure the two-phase pulse signal processing operation select bit (address 0344_{16}) is set to “1”. Also, always be sure to set the event/trigger select bit (addresses 0343_{16}) to “00”.

Figure 1.13.7. Timer Ai mode register in event counter mode

• Counter Resetting by Two-Phase Pulse Signal Processing

This function resets the timer counter to “0” when the Z-phase (counter reset) is input during two-phase pulse signal processing.

This function can only be used in timer A3 event counter mode, two-phase pulse signal processing, free-run type, and multiply-by-4 processing. The Z phase is input to the INT2 pin.

When the Z-phase input enable bit (bit 5 at address 0342_{16}) is set to “1”, the counter can be reset by Z-phase input. For the counter to be reset to “0” by Z-phase input, you must first write “0000_{16}” to the timer A3 register (address $034D_{16}$ and $034C_{16}$).

The Z-phase is input when the INT2 input edge is detected. The edge polarity is selected by the INT2 polarity switch bit (bit 4 at address $009C_{16}$). The Z-phase must have a pulse width greater than 1 cycle of the timer A3 count source. Figure 1.13.8 shows the relationship between the two-phase pulse (A phase and B phase) and the Z phase.

The counter is reset at the count source following Z-phase input. Figure 1.13.9 shows the timing at which the counter is reset to “0”.

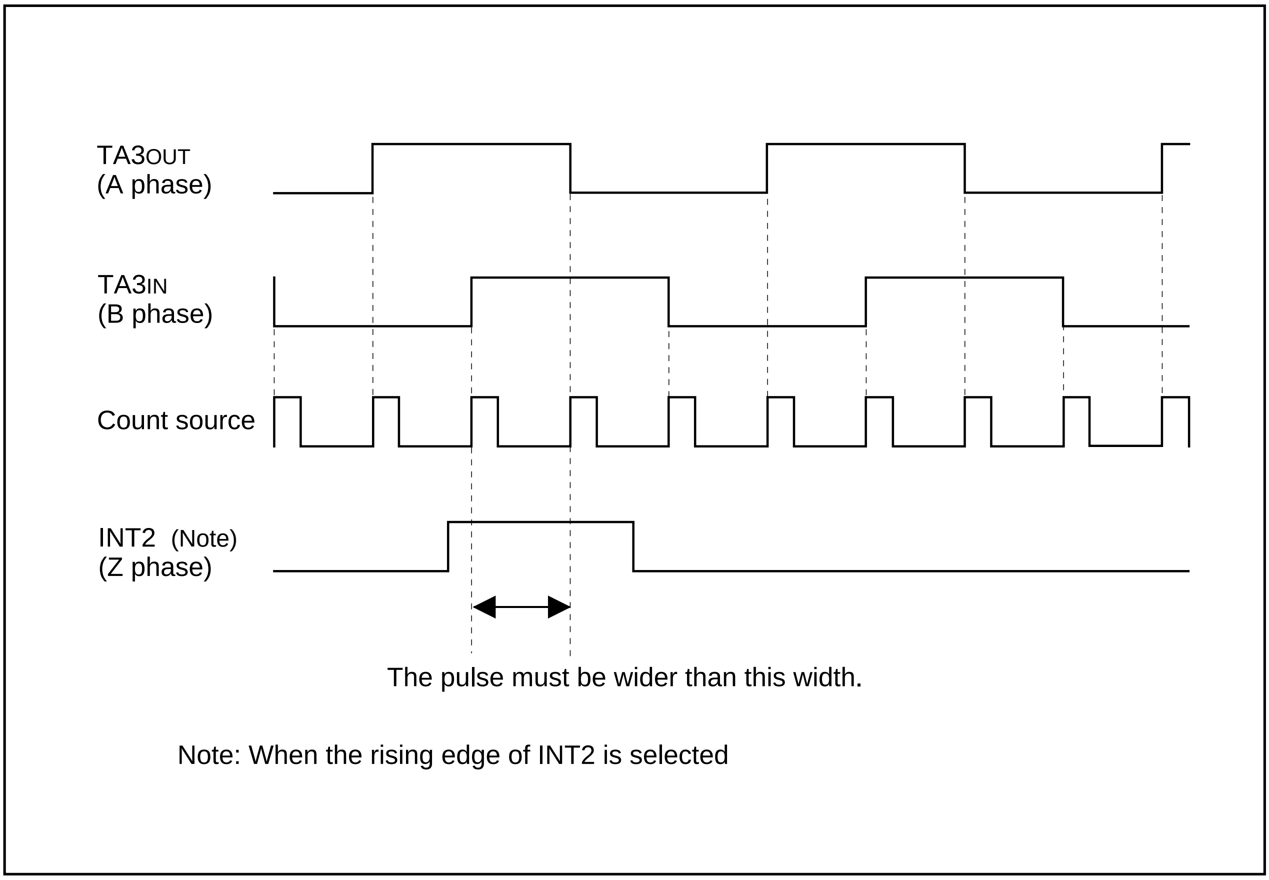

TA3OUT
(A phase)
TA3IN
(B phase)
Count source
INT2 (Note)
(Z phase)
The pulse must be wider than this width.
Note: When the rising edge of INT2 is selected

Figure 1.13.8. The relationship between the two-phase pulse (A phase and B phase) and the Z phase

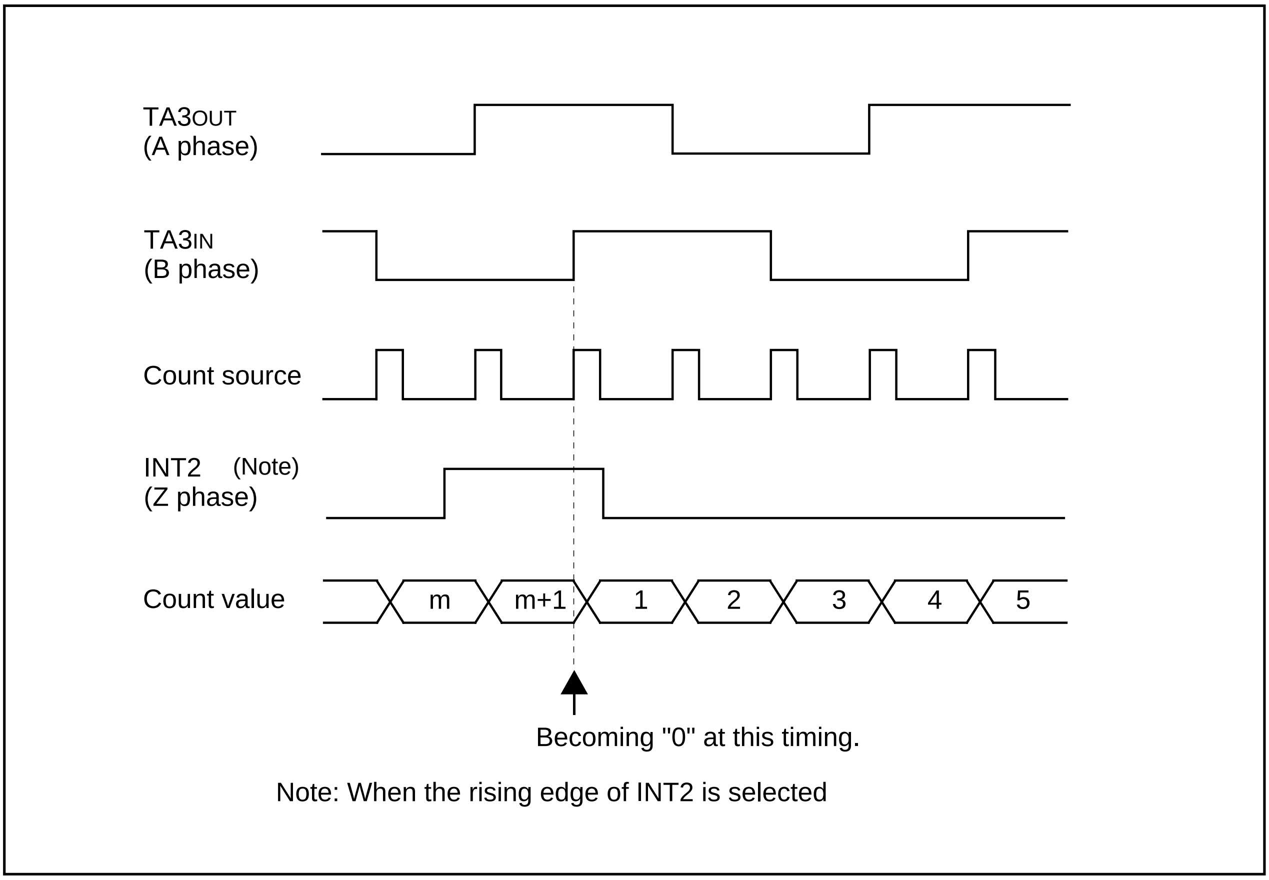


Figure 1.13.9. The counter reset timing

Note that timer A3 interrupt requests occur successively two times when timer A3 underflow and INT2 input reload are happened at the same timing.

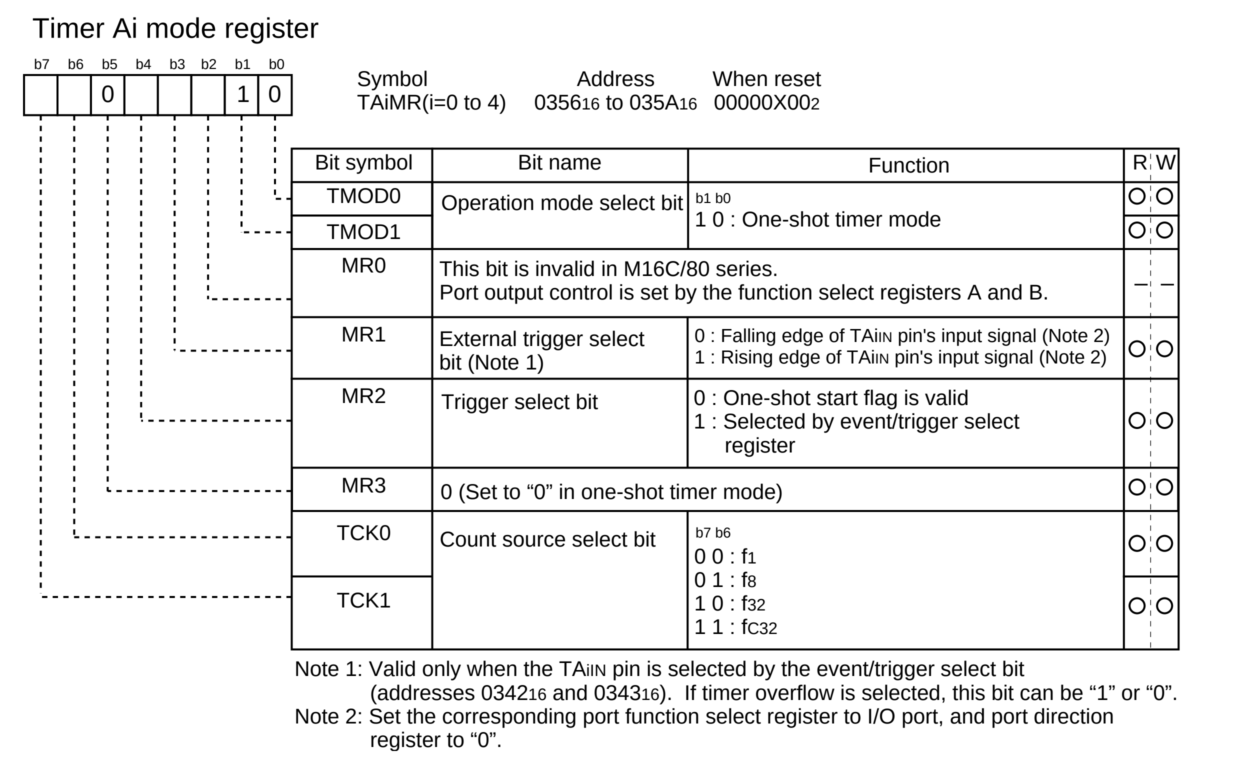
Do not use timer A3 interrupt request when this function is used.

(3) One-shot timer mode

In this mode, the timer operates only once. (See Table 1.13.4.) When a trigger occurs, the timer starts up and continues operating for a given period. Figure 1.13.10 shows the timer Ai mode register in one-shot timer mode.

Table 1.13.4. Timer specifications in one-shot timer mode

Item	Specification
Count source	f_1, f_8, f_{32}, fc_{32}
Count operation	• The timer counts down • When the count reaches 0000_{16}, the timer stops counting after reloading a new count • If a trigger occurs when counting, the timer reloads a new count and restarts counting
Divide ratio	1/n n : Set value
Count start condition	• An external trigger is input • The timer overflows • The one-shot start flag is set (= 1)
Count stop condition	• A new count is reloaded after the count has reached 0000_{16} • The count start flag is reset (= 0)
Interrupt request generation timing	The count reaches 0000_{16}
TAi_{IN} pin function	Programmable I/O port or trigger input
TAi_{OUT} pin function	Programmable I/O port or pulse output (Setting by the corresponding function select registers A and B)
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	• When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter • When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)



Timer Ai mode register

Symbol	Address	When reset
TAiMR(i=0 to 4)	0356_{16} to $035A_{16}$	$00000X00_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 0 : One-shot timer mode	○	○
TMOD1			○	○
MR0	This bit is invalid in M16C/80 series. Port output control is set by the function select registers A and B.		–	–
MR1	External trigger select bit (Note 1)	0 : Falling edge of TAi_{IN} pin's input signal (Note 2) 1 : Rising edge of TAi_{IN} pin's input signal (Note 2)	○	○
MR2	Trigger select bit	0 : One-shot start flag is valid 1 : Selected by event/trigger select register	○	○
MR3	0 (Set to “0” in one-shot timer mode)		○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : fc_{32}	○	○
TCK1			○	○

Note 1: Valid only when the TAi_{IN} pin is selected by the event/trigger select bit (addresses 0342_{16} and 0343_{16}). If timer overflow is selected, this bit can be “1” or “0”.
Note 2: Set the corresponding port function select register to I/O port, and port direction register to “0”.

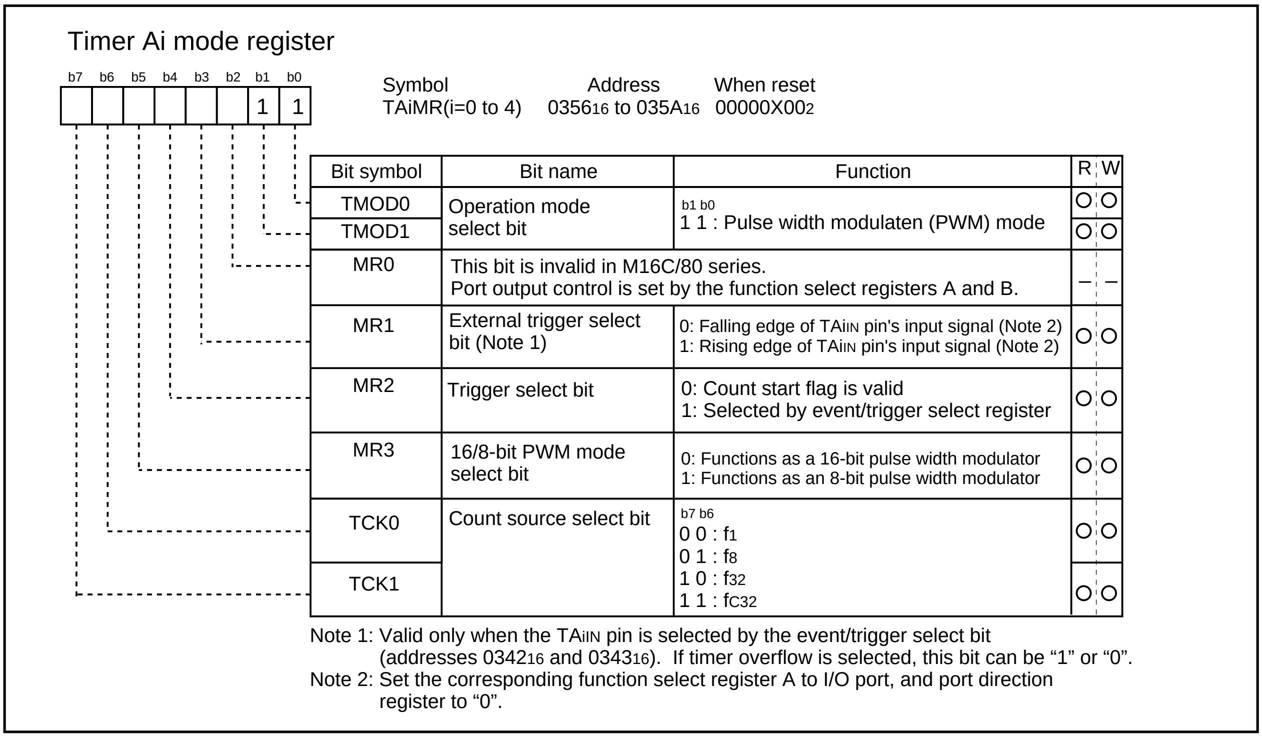
Figure 1.13.10. Timer Ai mode register in one-shot timer mode

(4) Pulse width modulation (PWM) mode

In this mode, the timer outputs pulses of a given width in succession. (See Table 1.13.5.) In this mode, the counter functions as either a 16-bit pulse width modulator or an 8-bit pulse width modulator. Figure 1.13.11 shows the timer Ai mode register in pulse width modulation mode. Figure 1.13.12 shows the example of how a 16-bit pulse width modulator operates. Figure 1.13.13 shows the example of how an 8-bit pulse width modulator operates.

Table 1.13.5. Timer specifications in pulse width modulation mode

Item	Specification
Count source	f_1, f_8, f_{32}, f_{C32}
Count operation	• The timer counts down (operating as an 8-bit or a 16-bit pulse width modulator) • The timer reloads a new count at a rising edge of PWM pulse and continues counting • The timer is not affected by a trigger that occurs when counting
16-bit PWM	• High level width n / fi n : Set value • Cycle time $(2^{16}-1)$ / fi fixed
8-bit PWM	• High level width n×(m+1) / fi n : values set to timer Ai register's high-order address • Cycle time $(2^{8}-1)$×(m+1) / fi m:values set to timer Ai register's low-order address
Count start condition	• External trigger is input • The timer overflows • The count start flag is set (= 1)
Count stop condition	• The count start flag is reset (= 0)
Interrupt request generation timing	PWM pulse goes "L"
TAiIN pin function	Programmable I/O port or trigger input
TAiOUT pin function	Pulse output (TAiOUT output is selected by the corresponding function select registers A and B)
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	• When counting stopped When a value is written to timer Ai register, it is written to both reload register and counter • When counting in progress When a value is written to timer Ai register, it is written to only reload register (Transferred to counter at next reload time)



Timer Ai mode register

Symbol	Address	When reset
TAiMR(i=0 to 4)	0356_{16} to $035A_{16}$	$00000X00_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 1 : Pulse width modulaten (PWM) mode	○	○
TMOD1			○	○
MR0	This bit is invalid in M16C/80 series. Port output control is set by the function select registers A and B.		–	–
MR1	External trigger select bit (Note 1)	0: Falling edge of TAiIN pin's input signal (Note 2) 1: Rising edge of TAiIN pin's input signal (Note 2)	○	○
MR2	Trigger select bit	0: Count start flag is valid 1: Selected by event/trigger select register	○	○
MR3	16/8-bit PWM mode select bit	0: Functions as a 16-bit pulse width modulator 1: Functions as an 8-bit pulse width modulator	○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: Valid only when the TAiIN pin is selected by the event/trigger select bit (addresses 0342_{16} and 0343_{16}). If timer overflow is selected, this bit can be "1" or "0".
Note 2: Set the corresponding function select register A to I/O port, and port direction register to "0".

Figure 1.13.11. Timer Ai mode register in pulse width modulation mode

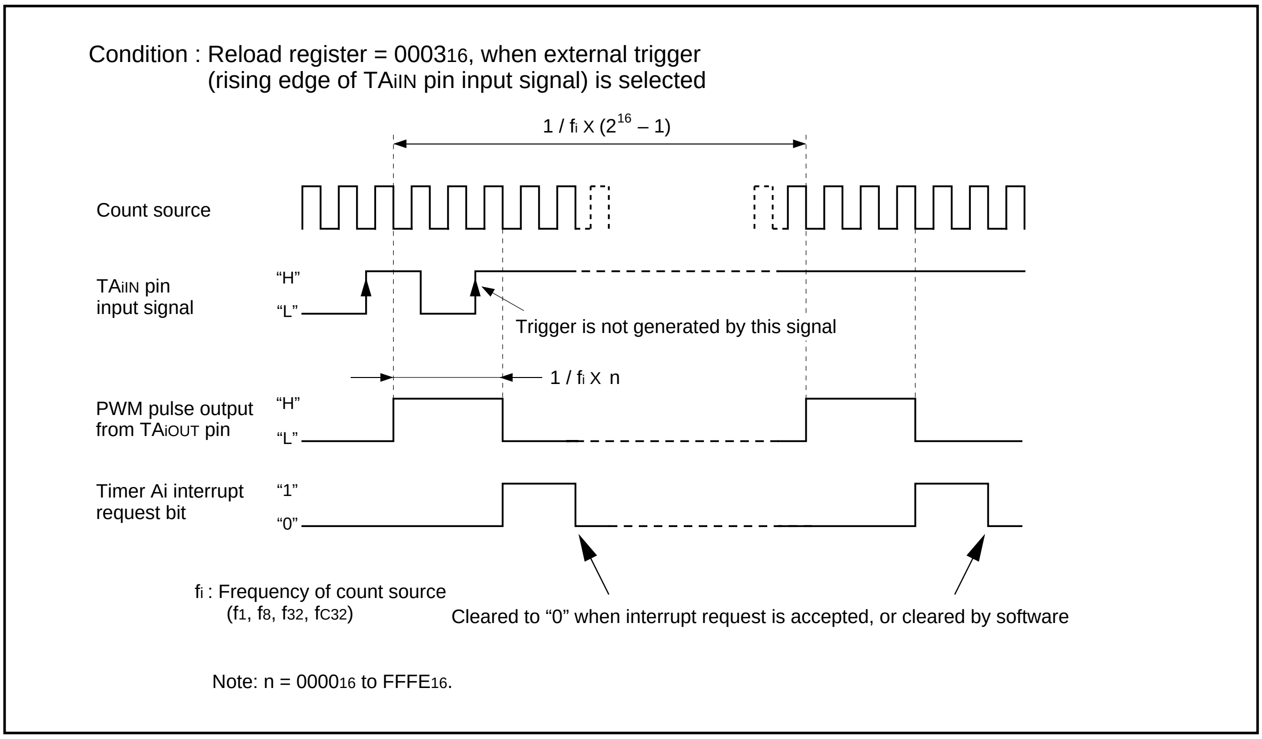

Condition : Reload register = 0003₁₆, when external trigger
(rising edge of TAiIN pin input signal) is selected
1 / fi X (2^16 – 1)
Count source
TAiIN pin
input signal
"H"
"L"
Trigger is not generated by this signal
1 / fi X n
PWM pulse output
from TAiOUT pin
"H"
"L"
Timer Ai interrupt
request bit
"1"
"0"
fi : Frequency of count source
(f1, f8, f32, fC32)
Cleared to "0" when interrupt request is accepted, or cleared by software
Note: n = 0000₁₆ to FFFE₁₆.

Figure 1.13.12. Example of how a 16-bit pulse width modulator operates

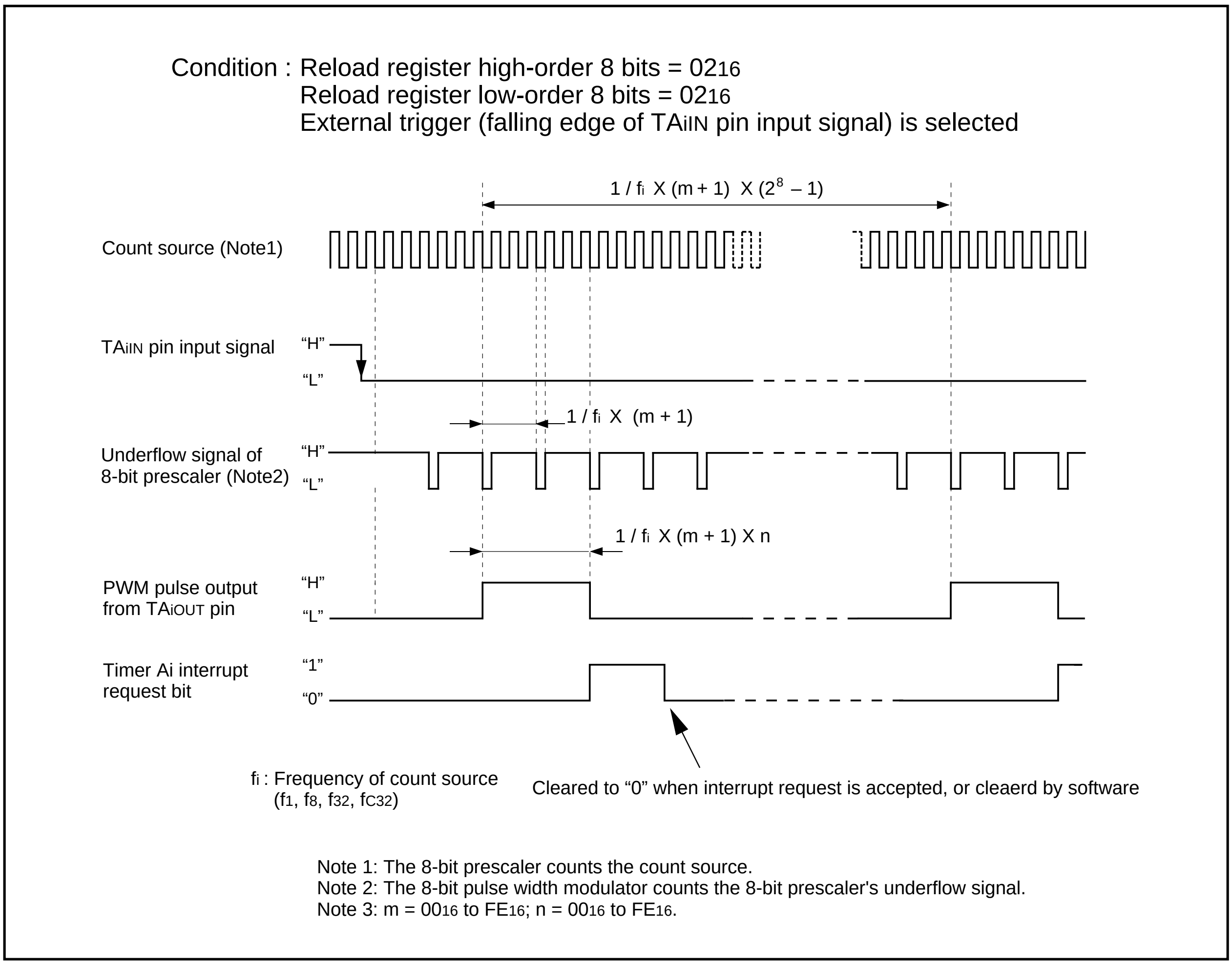
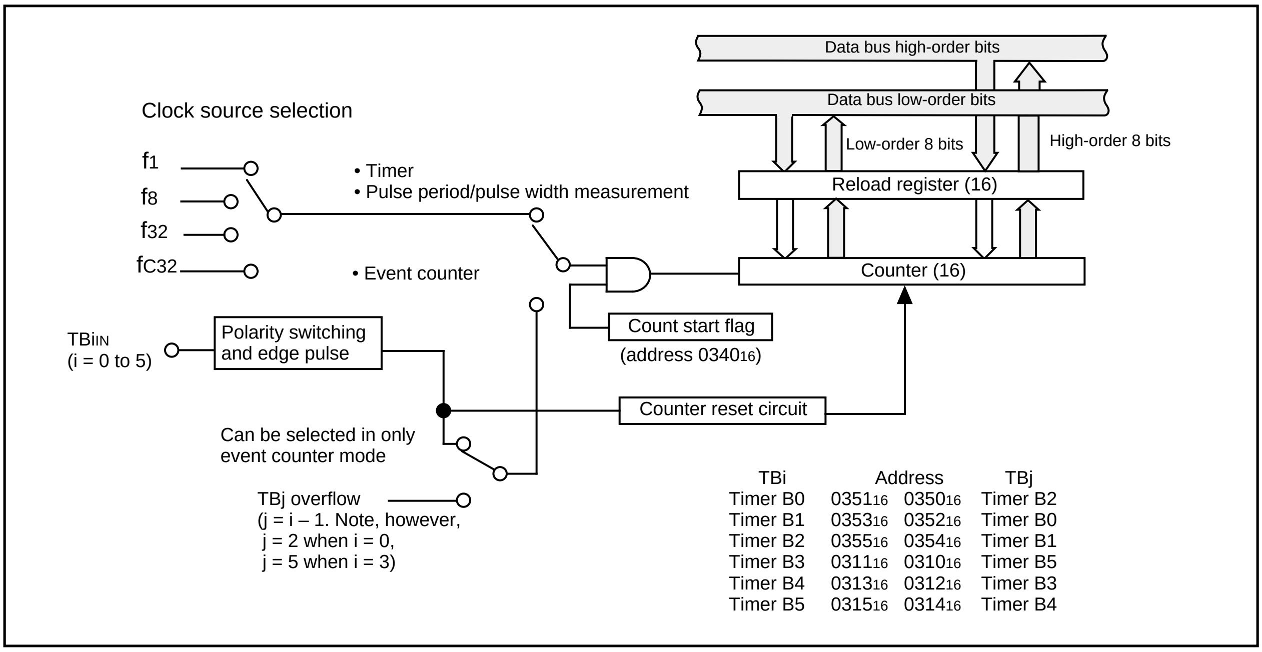

Condition : Reload register high-order 8 bits = 02₁₆
Reload register low-order 8 bits = 02₁₆
External trigger (falling edge of TAiIN pin input signal) is selected
1 / fi X (m + 1) X (2^8 – 1)
Count source (Note1)
TAiIN pin input signal
"H"
"L"
1 / fi X (m + 1)
Underflow signal of
8-bit prescaler (Note2)
"H"
"L"
1 / fi X (m + 1) X n
PWM pulse output
from TAiOUT pin
"H"
"L"
Timer Ai interrupt
request bit
"1"
"0"
fi : Frequency of count source
(f1, f8, f32, fC32)
Cleared to "0" when interrupt request is accepted, or cleaerd by software
Note 1: The 8-bit prescaler counts the count source.
Note 2: The 8-bit pulse width modulator counts the 8-bit prescaler's underflow signal.
Note 3: m = 00₁₆ to FE₁₆; n = 00₁₆ to FE₁₆.

Figure 1.13.13. Example of how an 8-bit pulse width modulator operates

Timer B

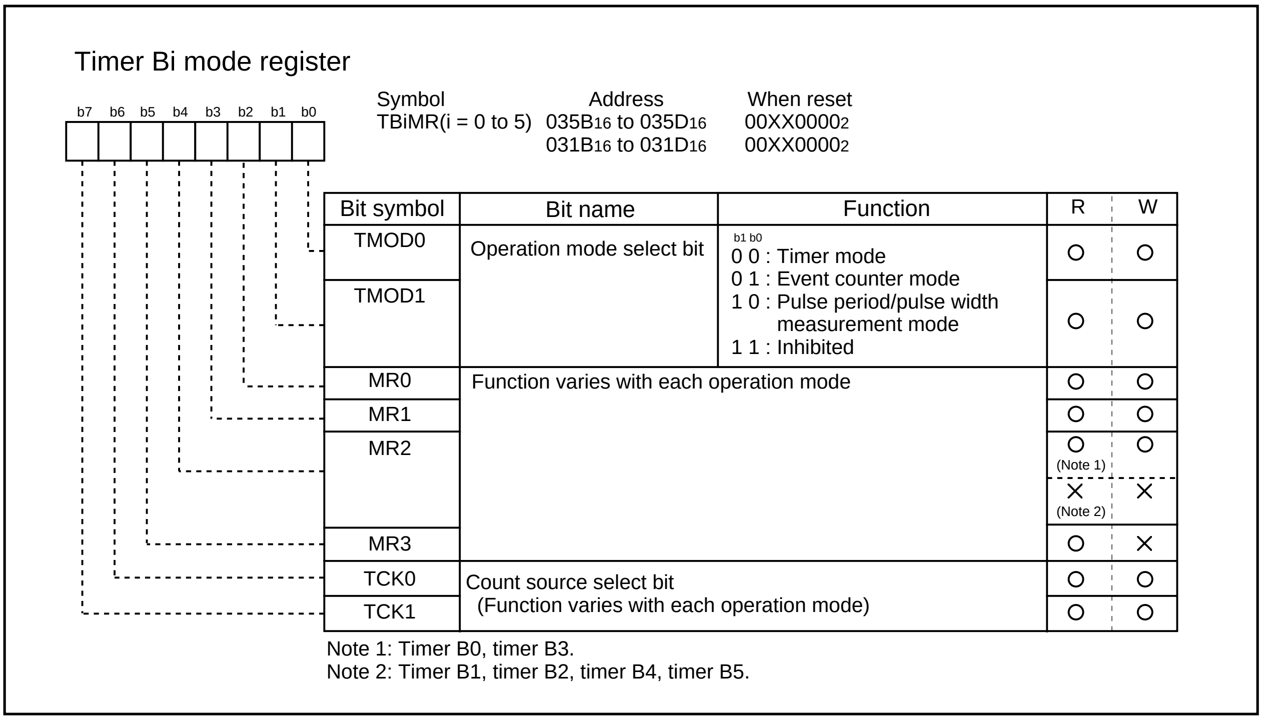
Figure 1.14.1 shows the block diagram of timer B. Figures 1.14.2 and 1.14.3 show the timer B-related registers. Use the timer Bi mode register (i = 0 to 5) bits 0 and 1 to choose the desired mode.
Timer B has three operation modes listed as follows:

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts pulses from an external source or a timer overflow.
- Pulse period/pulse width measuring mode: The timer measures an external signal's pulse period or pulse width.



TBi	Address		TBj
Timer B0	0351_{16}	0350_{16}	Timer B2
Timer B1	0353_{16}	0352_{16}	Timer B0
Timer B2	0355_{16}	0354_{16}	Timer B1
Timer B3	0311_{16}	0310_{16}	Timer B5
Timer B4	0313_{16}	0312_{16}	Timer B3
Timer B5	0315_{16}	0314_{16}	Timer B4

Figure 1.14.1. Block diagram of timer B


b7 b6 b5 b4 b3 b2 b1 b0

Timer Bi mode register

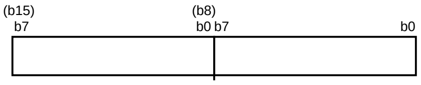
Symbol	Address	When reset
TBiMR(i = 0 to 5)	$035B_{16}$ to $035D_{16}$	$00XX0000_2$
	$031B_{16}$ to $031D_{16}$	$00XX0000_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period/pulse width measurement mode 1 1 : Inhibited	○	○
TMOD1			○	○
MR0	Function varies with each operation mode		○	○
MR1			○	○
MR2			○ (Note 1) × (Note 2)	○ ×
MR3			○	×
TCK0	Count source select bit (Function varies with each operation mode)		○	○
TCK1			○	○

Note 1: Timer B0, timer B3.
Note 2: Timer B1, timer B2, timer B4, timer B5.

Figure 1.14.2. Timer B-related registers (1)

Timer Bi register (Note)

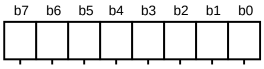


Symbol	Address	When reset
TB0	0351$_{16}$, 0350$_{16}$	Indeterminate
TB1	0353$_{16}$, 0352$_{16}$	Indeterminate
TB2	0355$_{16}$, 0354$_{16}$	Indeterminate
TB3	0311$_{16}$, 0310$_{16}$	Indeterminate
TB4	0313$_{16}$, 0312$_{16}$	Indeterminate
TB5	0315$_{16}$, 0314$_{16}$	Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts the timer's period	0000$_{16}$ to FFFF$_{16}$	○	○
• Event counter mode Counts external pulses input or a timer overflow	0000$_{16}$ to FFFF$_{16}$	○	○
• Pulse period / pulse width measurement mode Measures a pulse period or width	——	○	×

Note: Read and write data in 16-bit units.

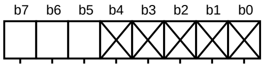
Count start flag



Symbol	Address	When reset
TABSR	0340$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TA1S	Timer A1 count start flag		○	○
TA2S	Timer A2 count start flag		○	○
TA3S	Timer A3 count start flag		○	○
TA4S	Timer A4 count start flag		○	○
TB0S	Timer B0 count start flag		○	○
TB1S	Timer B1 count start flag		○	○
TB2S	Timer B2 count start flag		○	○

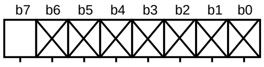
Timer B3, 4, 5 count start flag



Symbol	Address	When reset
TBSR	0300$_{16}$	000XXXXX$_2$

Bit symbol	Bit name	Function	R	W
Nothing is assigned. When write, set "0". When read, the value of these bits is indeterminate.			—	—
TB3S	Timer B3 count start flag	0 : Stops counting 1 : Starts counting	○	○
TB4S	Timer B4 count start flag		○	○
TB5S	Timer B5 count start flag		○	○

Clock prescaler reset flag



Symbol	Address	When reset
CPSRF	0341$_{16}$	0XXXXXXX$_2$

Bit symbol	Bit name	Function	R	W
Nothing is assigned. When write, set "0". When read, the value of these bits is indeterminate.			—	—
CPSR	Clock prescaler reset flag	0 : No effect 1 : Prescaler is reset (When read, the value is “0”)	○	○

Figure 1.14.3. Timer B-related registers (2)

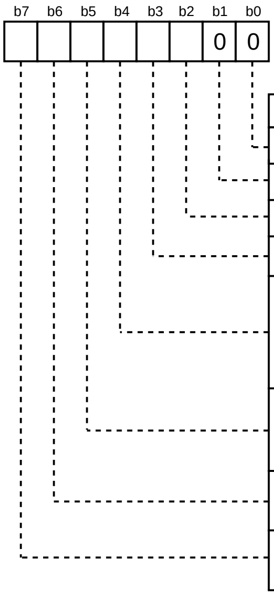
(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 1.14.1.) Figure 1.14.4 shows the timer Bi mode register in timer mode.

Table 1.14.1. Timer specifications in timer mode

Item	Specification
Count source	f1, f8, f32, fC32
Count operation	• Counts down • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBiIN pin function	Programmable I/O port
Read from timer	Count value is read out by reading timer Bi register
Write to timer	• When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter • When counting in progress When a value is written to timer Bi register, it is written to only reload register (Transferred to counter at next reload time)

Timer Bi mode register



Symbol	Address	When reset
TBiMR(i = 0 to 5)	$035B_{16}$ to $035D_{16}$	$00XX0000_2$
	$031B_{16}$ to $031D_{16}$	$00XX0000_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	○	○
TMOD1			○	○
MR0	Invalid in timer mode Can be “0” or “1”		○	○
MR1			○	○
MR2	0 (Set to “0” in timer mode ; i = 0, 3)		○ (Note 1)	○
	Nothing is assiigned (i = 1, 2, 4, 5). When write, set "0". When read, its content is indeterminate.		× (Note 2)	×
MR3	Invalid in timer mode. When write, set "0". When read in timer mode, its content is indeterminate.		○	×
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Note 1: Timer B0, timer B3.
Note 2: Timer B1, timer B2, timer B4, timer B5.

Figure 1.14.4. Timer Bi mode register in timer mode

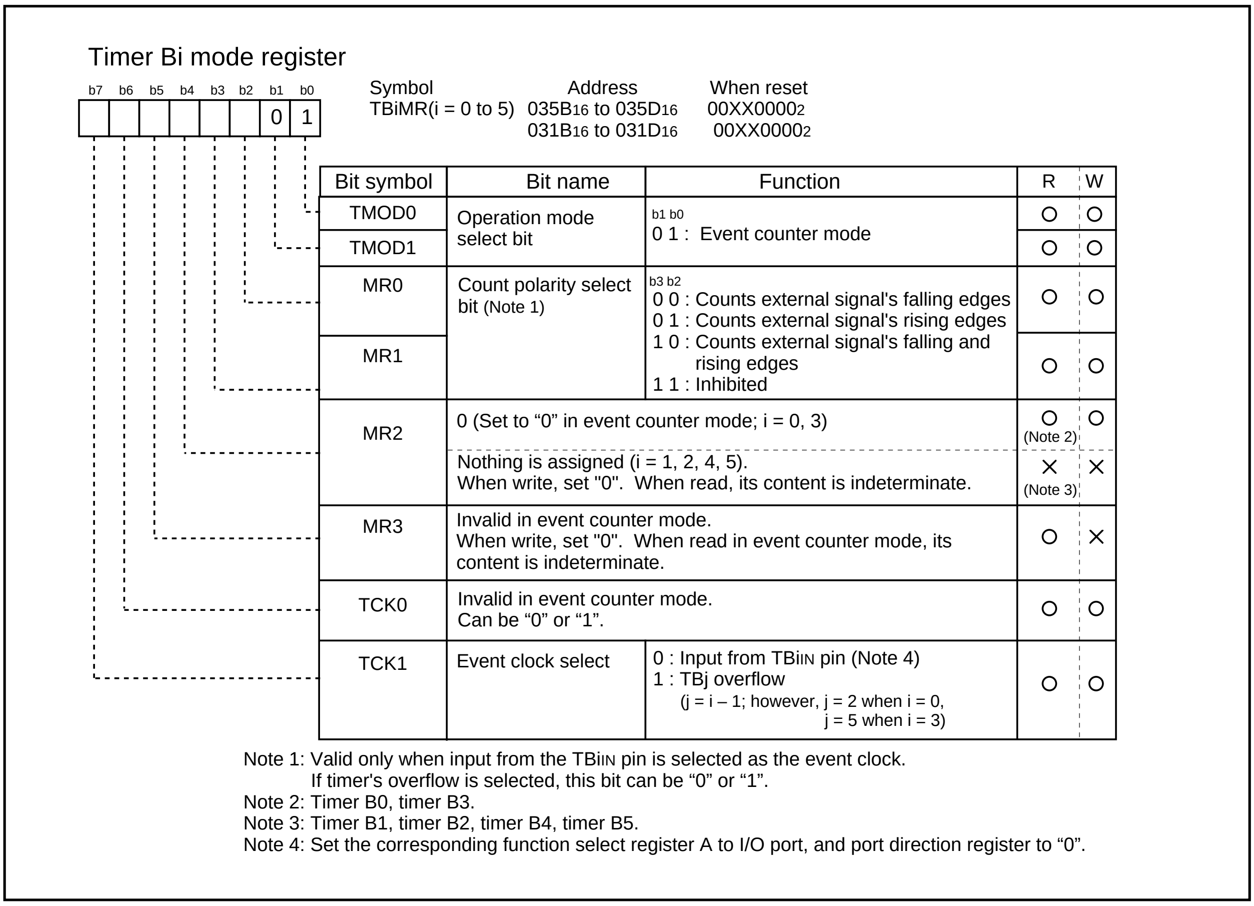
(2) Event counter mode

In this mode, the timer counts an external signal or an internal timer's overflow. (See Table 1.14.2.)

Figure 1.14.5 shows the timer Bi mode register in event counter mode.

Table 1.14.2. Timer specifications in event counter mode

Item	Specification
Count source	• External signals input to TBiIN pin Effective edge of count source can be a rising edge, a falling edge, or falling and rising edges as selected by software • TBi overflows or underflows
Count operation	• Counts down • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(n+1) n : Set value
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	The timer underflows
TBiIN pin function	Count source input (Set the corresponding function select register A to I/O port.)
Read from timer	Count value can be read out by reading timer Bi register
Write to timer	• When counting stopped When a value is written to timer Bi register, it is written to both reload register and counter • When counting in progress When a value is written to timer Bi register, it is written to only reload register (Transferred to counter at next reload time)



Timer Bi mode register

b7	b6	b5	b4	b3	b2	b1	b0
						0	1

Symbol	Address	When reset
TBiMR(i = 0 to 5)	$035B_{16}$ to $035D_{16}$	$00XX0000_2$
	$031B_{16}$ to $031D_{16}$	$00XX0000_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	○	○
TMOD1			○	○
MR0	Count polarity select bit (Note 1)	b3 b2 0 0 : Counts external signal's falling edges 0 1 : Counts external signal's rising edges 1 0 : Counts external signal's falling and rising edges 1 1 : Inhibited	○	○
MR1			○	○
MR2	0 (Set to “0” in event counter mode; i = 0, 3)		○ (Note 2)	○
	Nothing is assigned (i = 1, 2, 4, 5). When write, set "0". When read, its content is indeterminate.		× (Note 3)	×
MR3	Invalid in event counter mode. When write, set "0". When read in event counter mode, its content is indeterminate.		○	×
TCK0	Invalid in event counter mode. Can be “0” or “1”.		○	○
TCK1	Event clock select	0 : Input from TBiIN pin (Note 4) 1 : TBj overflow (j = i – 1; however, j = 2 when i = 0, j = 5 when i = 3)	○	○

Note 1: Valid only when input from the TBiIN pin is selected as the event clock.
If timer's overflow is selected, this bit can be “0” or “1”.
Note 2: Timer B0, timer B3.
Note 3: Timer B1, timer B2, timer B4, timer B5.
Note 4: Set the corresponding function select register A to I/O port, and port direction register to “0”.

Figure 1.14.5. Timer Bi mode register in event counter mode

(3) Pulse period/pulse width measurement mode

In this mode, the timer measures the pulse period or pulse width of an external signal. (See Table 1.14.3.) Figure 1.14.6 shows the timer Bi mode register in pulse period/pulse width measurement mode. Figure 1.14.7 shows the operation timing when measuring a pulse period. Figure 1.14.8 shows the operation timing when measuring a pulse width.

Table 1.14.3. Timer specifications in pulse period/pulse width measurement mode

Item	Specification
Count source	f1, f8, f32, fc32
Count operation	• Up count • Counter value “0000$_{16}$” is transferred to reload register at measurement pulse's effective edge and the timer continues counting
Count start condition	Count start flag is set (= 1)
Count stop condition	Count start flag is reset (= 0)
Interrupt request generation timing	• When measurement pulse's effective edge is input (Note 1) • When an overflow occurs. (Simultaneously, the timer Bi overflow flag changes to “1”. The timer Bi overflow flag changes to “0” when the count start flag is “1” and a value is written to the timer Bi mode register.)
TBi$_{IN}$ pin function	Measurement pulse input (Set the corresponding function select register A to I/O port.)
Read from timer	When timer Bi register is read, it indicates the reload register’s content (measurement result) (Note 2)
Write to timer	Cannot be written to

Note 1: An interrupt request is not generated when the first effective edge is input after the timer has started counting.
Note 2: The value read out from the timer Bi register is indeterminate until the second effective edge is input after the timer.

Timer Bi mode register

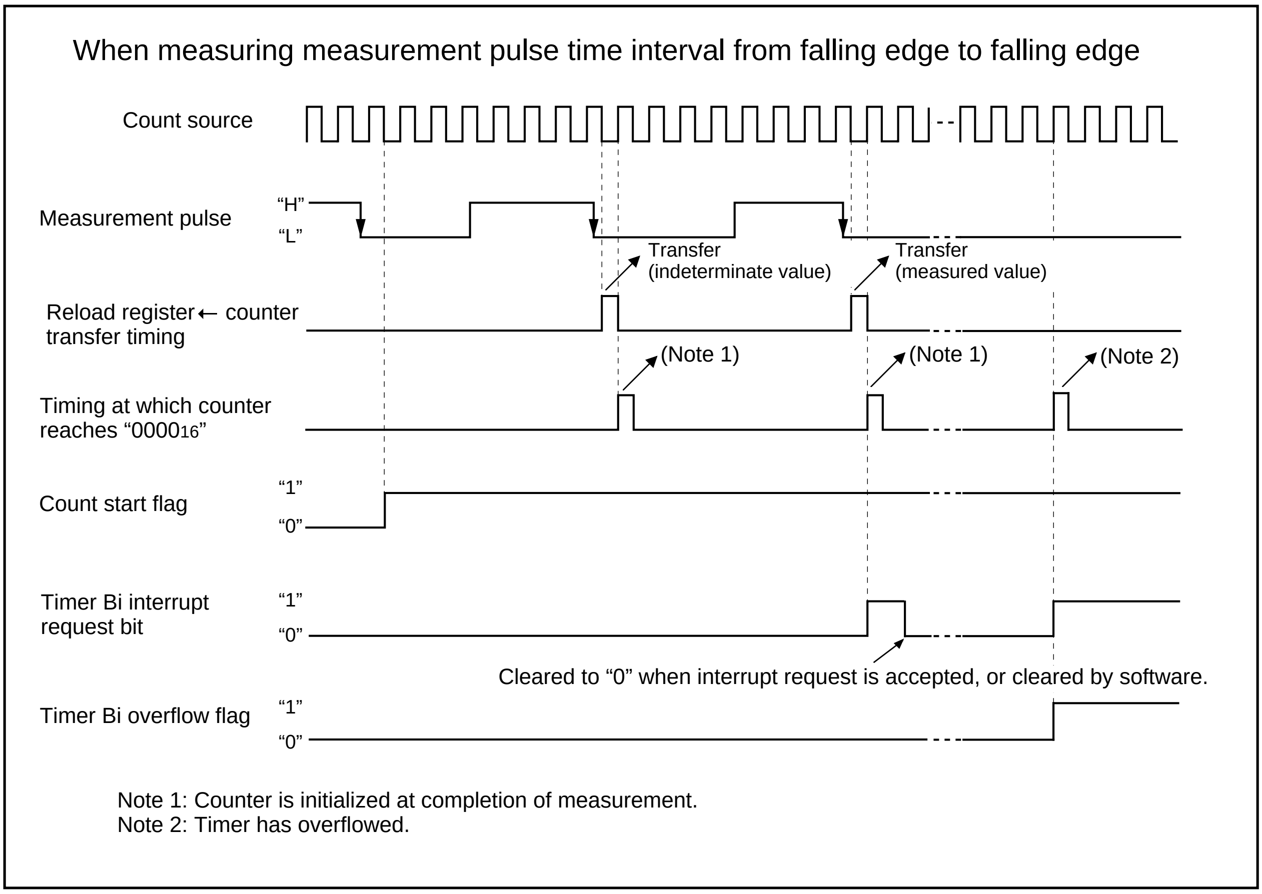
b7	b6	b5	b4	b3	b2	b1	b0
						1	0

Symbol	Address	When reset
TBiMR(i = 0 to 5)	035B$_{16}$ to 035D$_{16}$	00XX0000$_2$
	031B$_{16}$ to 031D$_{16}$	00XX0000$_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 0 : Pulse period / pulse width measurement mode	○	○
TMOD1			○	○
MR0	Measurement mode select bit	b3 b2 0 0 : Pulse period measurement (Interval between measurement pulse's falling edge to falling edge) 0 1 : Pulse period measurement (Interval between measurement pulse's rising edge to rising edge) 1 0 : Pulse width measurement (Interval between measurement pulse's falling edge to rising edge, and between rising edge to falling edge) 1 1 : Inhibited	○	○
MR1			○	○
MR2	0 (Set to “0” in pulse period/pulse width measurement mode; i = 0, 3)		○ (Note 2)	○
	Nothing is assigned (i = 1, 2, 4, 5). When write, set "0". When read, its content is indeterminate.		× (Note 3)	×
MR3	Timer Bi overflow flag (Note 1)	0 : Timer did not overflow 1 : Timer has overflowed	○	×
TCK0	Count source select bit	b7 b6 0 0 : f$_1$ 0 1 : f$_8$ 1 0 : f$_{32}$ 1 1 : fc$_{32}$	○	○
TCK1			○	○

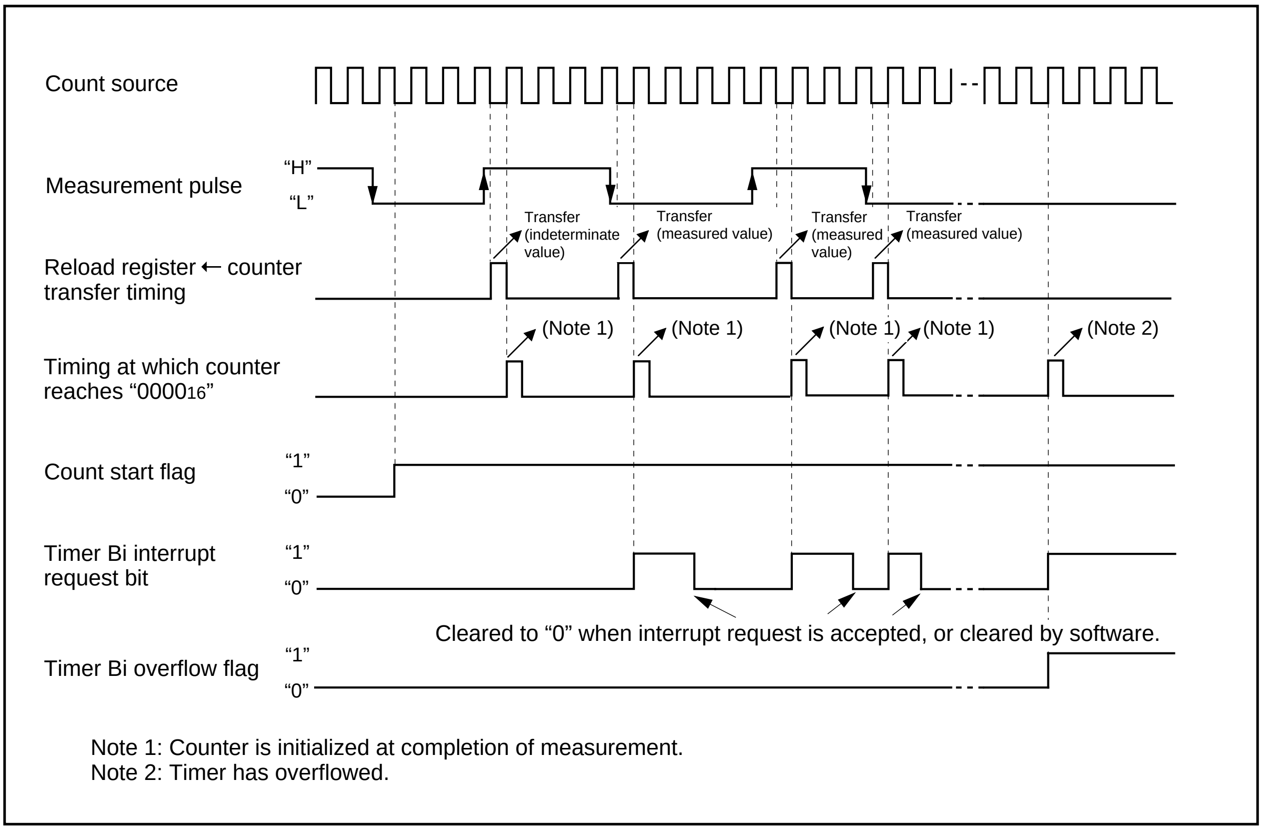
Note 1: It is indeterminate when reset.
The timer Bi overflow flag changes to “0” when the count start flag is “1” and a value is written to the timer Bi mode register. This flag cannot be set to “1” by software.
Note 2: Timer B0, timer B3.
Note 3: Timer B1, timer B2, timer B4, timer B5.

Figure 1.14.6. Timer Bi mode register in pulse period/pulse width measurement mode


When measuring measurement pulse time interval from falling edge to falling edge
Count source
Measurement pulse
"H"
"L"
Transfer (indeterminate value)
Transfer (measured value)
Reload register ← counter transfer timing
(Note 1)
(Note 1)
(Note 2)
Timing at which counter reaches "0000$_{16}$"
Count start flag
"1"
"0"
Timer Bi interrupt request bit
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software.
Timer Bi overflow flag
"1"
"0"

Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.

Figure 1.14.7. Operation timing when measuring a pulse period


Count source
Measurement pulse
"H"
"L"
Transfer (indeterminate value)
Transfer (measured value)
Transfer (measured value)
Transfer (measured value)
Reload register ← counter transfer timing
(Note 1)
(Note 1)
(Note 1)
(Note 1)
(Note 2)
Timing at which counter reaches "0000$_{16}$"
Count start flag
"1"
"0"
Timer Bi interrupt request bit
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software.
Timer Bi overflow flag
"1"
"0"

Note 1: Counter is initialized at completion of measurement.
Note 2: Timer has overflowed.

Figure 1.14.8. Operation timing when measuring a pulse width

Three-phase motor control timers’ functions

Use of more than one built-in timer A and timer B provides the means of outputting three-phase motor driving waveforms.

Figures 1.15.1 through 1.15.3 show registers related to timers for three-phase motor control.

Three-phase PWM control register 0

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
INVC0	0308$_{16}$	00$_{16}$

Bit symbol	Bit name	Description	R	W
INV0$_0$	Effective interrupt output polarity select bit	0: A timer B2 interrupt occurs when the timer A1 reload control signal is “0”. 1: A timer B2 interrupt occurs when the timer A1 reload control signal is “1”. **Effective only in three-phase mode 1**	○	○
INV0$_1$	Effective interrupt output specification bit (Note 4)	0: Not specified. 1: Selected by the effective interrupt output polarity selection bit. **Effective only in three-phase mode 1**	○	○
INV0$_2$	Mode select bit (Note 2)	0: Normal mode 1: Three-phase PWM output mode	○	○
INV0$_3$	Output control bit	0: Output disabled 1: Output enabled	○	○
INV0$_4$	Positive and negative phases concurrent L output disable function enable bit	0: Feature disabled 1: Feature enabled	○	○
INV0$_5$	Positive and negative phases concurrent L output detect flag	0: Not detected yet 1: Already detected	○	○ (Note 1)
INV0$_6$	Modulation mode select bit (Note 3)	0: Triangular wave modulation mode 1: Sawtooth wave modulation mode	○	○
INV0$_7$	Software trigger bit	1: Trigger generated The value, when read, is “0”.	○	○

Note 1: No value other than “0” can be written.

Note 2: Selecting three-phase PWM output mode causes the dead time timer, the U, V, W phase output control circuits, and the timer B2 interrupt occurrences frequency set circuit works.
For U, $\overline{U}$, V, $\overline{V}$, W and $\overline{W}$ output from P8$_0$, P8$_1$, and P7$_2$ through P7$_5$, setting of function select registers A, B and C is required.

Note 3: **In triangular wave modulation mode:** The dead time timer starts in synchronization with the falling edge of timer Ai output. The data transfer from the three-phase buffer register to the three-phase output shift register is made only once in synchronization with the transfer trigger signal after writing to the three-phase output buffer register.
In sawtooth wave modulation mode: The dead time timer starts in synchronization with the falling edge of timer A output and with the transfer trigger signal. The data transfer from the three-phase output buffer register to the three-phase output shift register is made with respect to every transfer trigger.

Note 4: Set bit 1 of this register to "1" after setting timer B2 interrupt frequency set counter.

Three-phase PWM control register 1

b7 b6 b5 b4 b3 b2 b1 b0

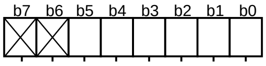
Symbol	Address	When reset
INVC1	0309$_{16}$	XXX0X000$_2$

Bit symbol	Bit name	Description	R	W
INV1$_0$	Timer Ai start trigger signal select bit	0: Timer B2 overflow signal 1: Timer B2 overflow signal, signal for writing to timer B2	○	○
INV1$_1$	Timer A1-1, A2-1, A4-1 control bit	0: Three-phase mode 0 1: Three-phase mode 1	○	○
INV1$_2$	Dead time timer count source select bit	0 : f$_1$ 1 : f$_1$/2	○	○
INV1$_3$	Carrier wave detect flag (Note)	0: Rising edge of triangular waveform 1: Falling edge of triangular waveform	○	X
INV1$_4$	Output porality control bit	0 : Low active 1 : High active	○	○
Noting is assigned. When write, set "0". When read, their contents are "0".			–	–

Note : INV1$_3$ is valid when INV0$_6$ = 0 and INV1$_1$ = 1.

Figure 1.15.1. Registers related to timers for three-phase motor control

Three-phase output buffer register 0 (Note)

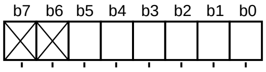


Symbol	Address	When reset
IDB0	030A$_{16}$	00$_{16}$

Bit Symbol	Bit name	Function	R	W
DU0	U phase output buffer 0	Setting in U phase output buffer 0	○	○
DUB0	$\overline{U}$ phase output buffer 0	Setting in $\overline{U}$ phase output buffer 0	○	○
DV0	V phase output buffer 0	Setting in V phase output buffer 0	○	○
DVB0	$\overline{V}$ phase output buffer 0	Setting in $\overline{V}$ phase output buffer 0	○	○
DW0	W phase output buffer 0	Setting in W phase output buffer 0	○	○
DWB0	$\overline{W}$ phase output buffer 0	Setting in $\overline{W}$ phase output buffer 0	○	○
Nothing is assigned. When write, set "0". When read, its content is "0".			—	—

Note: When executing read instruction of this register, the contents of three-phase shift register is read out.

Three-phase output buffer register 1 (Note)



Symbol	Address	When reset
IDB1	030B$_{16}$	00$_{16}$

Bit Symbol	Bit name	Function	R	W
DU1	U phase output buffer 1	Setting in U phase output buffer 1	○	○
DUB1	$\overline{U}$ phase output buffer 1	Setting in $\overline{U}$ phase output buffer 1	○	○
DV1	V phase output buffer 1	Setting in V phase output buffer 1	○	○
DVB1	$\overline{V}$ phase output buffer 1	Setting in $\overline{V}$ phase output buffer 1	○	○
DW1	W phase output buffer 1	Setting in W phase output buffer 1	○	○
DWB1	$\overline{W}$ phase output buffer 1	Setting in $\overline{W}$ phase output buffer 1	○	○
Nothing is assigned. When write, set "0". When read, its content is "0".			—	—

Note: When executing read instruction of this register, the contents of three-phase shift register is read out.

Dead time timer (Note)

Symbol	Address	When reset
DTT	030C$_{16}$	Indeterminate

Function	Values that can be set	R	W
Set dead time timer	1 to 255	—	○

Note: Use MOV instruction to write to this register.

Timer B2 interrupt occurrences frequency set counter (Note 1 to 3)

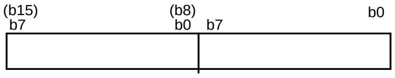
Symbol	Address	When reset
ICTB2	030D$_{16}$	Indeterminate

Function	Values that can be set	R	W
Set occurrence frequency of timer B2 interrupt request	1 to 15	—	○

Note 1: When the effective interrupt output specification bit (INV01: bit 1 at 0308$_{16}$) is set to "1" and three-phase motor control timer is operating, do not rewrite to this register.
Note 2: Do not write to this register at the timing of timer B2 overflow.
Note 3: Use MOV instruction to write to this register.

Figure 1.15.2. Registers related to timers for three-phase motor control

Timer Ai register (Note)

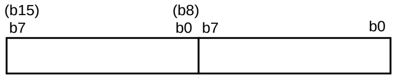


Symbol	Address	When reset
TA1	0349_{16},0348_{16}	Indeterminate
TA2	$034B_{16}$,$034A_{16}$	Indeterminate
TA4	$034F_{16}$,$034E_{16}$	Indeterminate
TB2	0355_{16},0354_{16}	Indeterminate

Function	Values that can be set	R	W
• Timer mode Counts an internal count source	0000_{16} to $FFFF_{16}$	○	○
• One-shot timer mode Counts a one shot width	0000_{16} to $FFFF_{16}$ (Note 2, 3)	×	○

Note 1: Read and write data in 16-bit units.
Note 2: When the timer Ai register is set to "0000_{16}", the counter does not operate and a timer Ai interrupt does not occur.
Note 3: When writing to this register, use MOV instruction.

Timer Ai-1 register (Note)

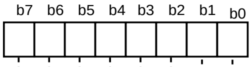


Symbol	Address	When reset
TA11	0303_{16},0302_{16}	Indeterminate
TA21	0305_{16},0304_{16}	Indeterminate
TA41	0307_{16},0306_{16}	Indeterminate

Function	Values that can be set	R	W
Counts an internal count source	0000_{16} to $FFFF_{16}$	○	○

Note: Read and write data in 16-bit units.

Trigger select register

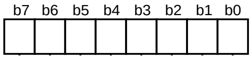


Symbol	Address	When reset
TRGSR	0343_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
TA1TGL	Timer A1 event/trigger select bit	b1 b0 0 0 : Input on TA1$_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA0 overflow is selected 1 1 : TA2 overflow is selected	○	○
TA1TGH			○	○
TA2TGL	Timer A2 event/trigger select bit	b3 b2 0 0 : Input on TA2$_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA1 overflow is selected 1 1 : TA3 overflow is selected	○	○
TA2TGH			○	○
TA3TGL	Timer A3 event/trigger select bit	b5 b4 0 0 : Input on TA3$_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA2 overflow is selected 1 1 : TA4 overflow is selected	○	○
TA3TGH			○	○
TA4TGL	Timer A4 event/trigger select bit	b7 b6 0 0 : Input on TA4$_{IN}$ is selected (Note) 0 1 : TB2 overflow is selected 1 0 : TA3 overflow is selected 1 1 : TA0 overflow is selected	○	○
TA4TGH			○	○

Note: Set the corresponding port function select register to I/O port, and port direction register to "0".

Count start flag



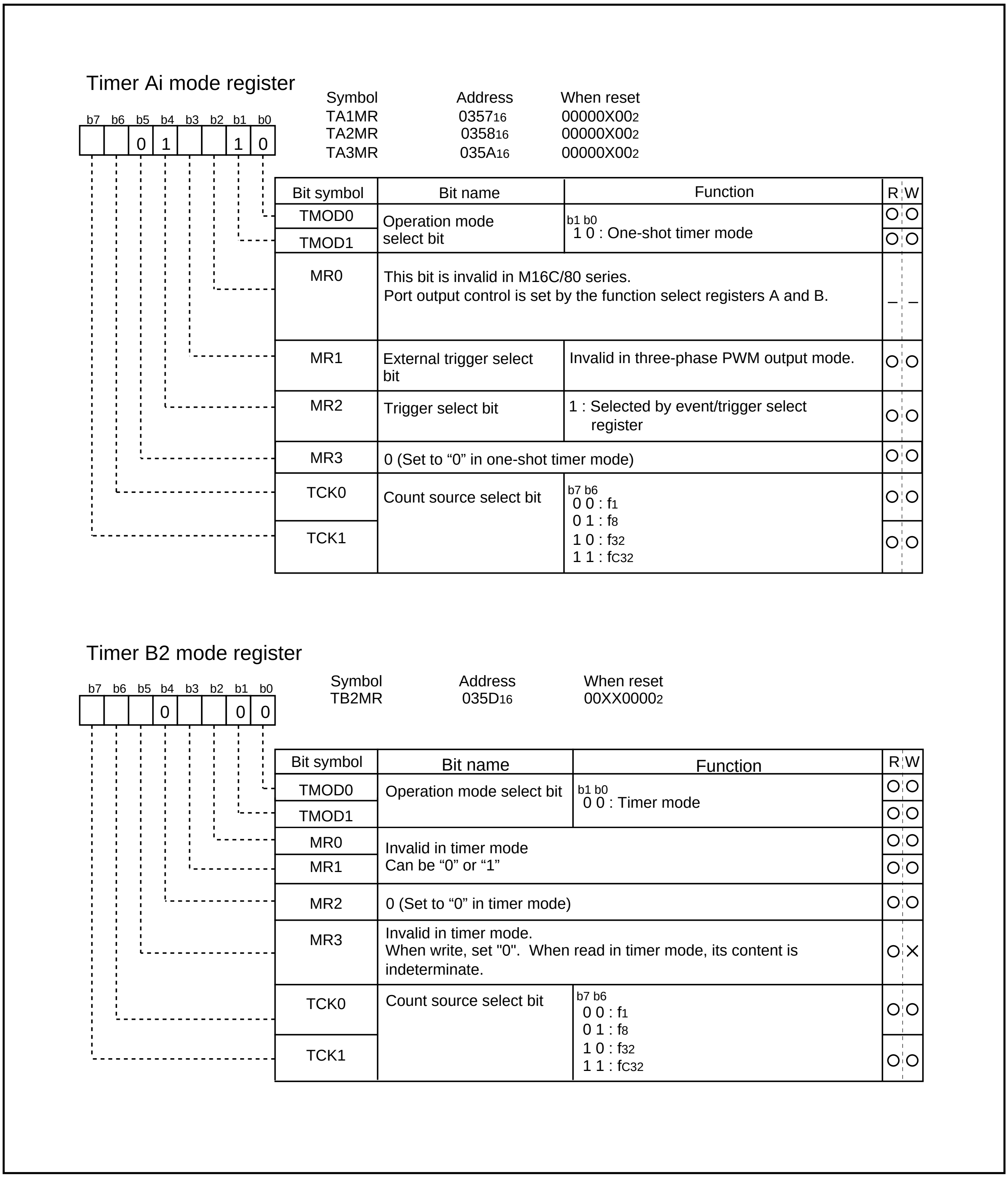
Symbol	Address	When reset
TABSR	0340_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
TA0S	Timer A0 count start flag	0 : Stops counting 1 : Starts counting	○	○
TA1S	Timer A1 count start flag		○	○
TA2S	Timer A2 count start flag		○	○
TA3S	Timer A3 count start flag		○	○
TA4S	Timer A4 count start flag		○	○
TB0S	Timer B0 count start flag		○	○
TB1S	Timer B1 count start flag		○	○
TB2S	Timer B2 count start flag		○	○

Figure 1.15.3. Registers related to timers for three-phase motor control

Three-phase motor driving waveform output mode (three-phase PWM output mode)

Setting “1” in the mode select bit (bit 2 at 0308_{16}) shown in Figure 1.15.1 causes three-phase PWM output mode that uses four timers A1, A2, A4, and B2 to be selected. As shown in Figure 1.15.4, set timers A1, A2, and A4 in one-shot timer mode, set the trigger in timer B2, and set timer B2 in timer mode using the respective timer mode registers.



Timer Ai mode register

b7	b6	b5	b4	b3	b2	b1	b0
		0	1			1	0

Symbol	Address	When reset
TA1MR	0357_{16}	$00000X00_2$
TA2MR	0358_{16}	$00000X00_2$
TA3MR	$035A_{16}$	$00000X00_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 1 0 : One-shot timer mode	○	○
TMOD1			○	○
MR0	This bit is invalid in M16C/80 series. Port output control is set by the function select registers A and B.		–	–
MR1	External trigger select bit	Invalid in three-phase PWM output mode.	○	○
MR2	Trigger select bit	1 : Selected by event/trigger select register	○	○
MR3	0 (Set to “0” in one-shot timer mode)		○	○
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Timer B2 mode register

b7	b6	b5	b4	b3	b2	b1	b0
			0			0	0

Symbol	Address	When reset
TB2MR	$035D_{16}$	$00XX0000_2$

Bit symbol	Bit name	Function	R	W
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	○	○
TMOD1			○	○
MR0	Invalid in timer mode Can be “0” or “1”		○	○
MR1			○	○
MR2	0 (Set to “0” in timer mode)		○	○
MR3	Invalid in timer mode. When write, set "0". When read in timer mode, its content is indeterminate.		○	×
TCK0	Count source select bit	b7 b6 0 0 : f_1 0 1 : f_8 1 0 : f_{32} 1 1 : f_{C32}	○	○
TCK1			○	○

Figure 1.15.4. Timer mode registers in three-phase PWM output mode

Figure 1.15.5 shows the block diagram for three-phase waveform mode. In "L" active output polarity in three-phase waveform mode, the positive-phase waveforms (U phase, V phase, and W phase) and negative waveforms ($\overline{U}$ phase, $\overline{V}$ phase, and $\overline{W}$ phase), six waveforms in total, are output from P8$_0$, P8$_1$, P7$_2$, P7$_3$, P7$_4$, and P7$_5$ as active on the "L" level. Of the timers used in this mode, timer A4 controls the U phase and $\overline{U}$ phase, timer A1 controls the V phase and $\overline{V}$ phase, and timer A2 controls the W phase and $\overline{W}$ phase respectively; timer B2 controls the periods of one-shot pulse output from timers A4, A1, and A2.
In outputting a waveform, dead time can be set so as to cause the "L" level of the positive waveform output (U phase, V phase, and W phase) not to lap over the "L" level of the negative waveform output ($\overline{U}$ phase, $\overline{V}$ phase, and $\overline{W}$ phase).
To set short circuit time, use three 8-bit timers sharing the reload register for setting dead time. A value from 1 through 255 can be set as the count of the timer for setting dead time. The timer for setting dead time works as a one-shot timer. If a value is written to the dead timer (030C$_{16}$), the value is written to the reload register shared by the three timers for setting dead time.
Any of the timers for setting dead time takes the value of the reload register into its counter, if a start trigger comes from its corresponding timer, and performs a down count in line with the clock source selected by the dead time timer count source select bit (bit 2 at 0309$_{16}$). The timer can receive another trigger again before the workings due to the previous trigger are completed. In this instance, the timer performs a down count from the reload register's content after its transfer, provoked by the trigger, to the timer for setting dead time.
Since the timer for setting dead time works as a one-shot timer, it starts outputting pulses if a trigger comes; it stops outputting pulses as soon as its content becomes 00$_{16}$, and waits for the next trigger to come.
The positive waveforms (U phase, V phase, and W phase) and the negative waveforms ($\overline{U}$ phase, $\overline{V}$ phase, and $\overline{W}$ phase) in three-phase waveform mode are output from respective ports by means of setting "1" in the output control bit (bit 3 at 0308$_{16}$). Setting "0" in this bit causes the ports to be the high-impedance state. This bit can be set to "0" not only by use of the applicable instruction, but by entering a falling edge in the $\overline{NMI}$ terminal or by resetting. Also, if "1" is set in the positive and negative phases concurrent L output disable function enable bit (bit 4 at 0308$_{16}$) causes one of the pairs of U phase and $\overline{U}$ phase, V phase and $\overline{V}$ phase, and W phase and $\overline{W}$ phase concurrently go to "L", as a result, the output control bit becomes the high-impedance state.

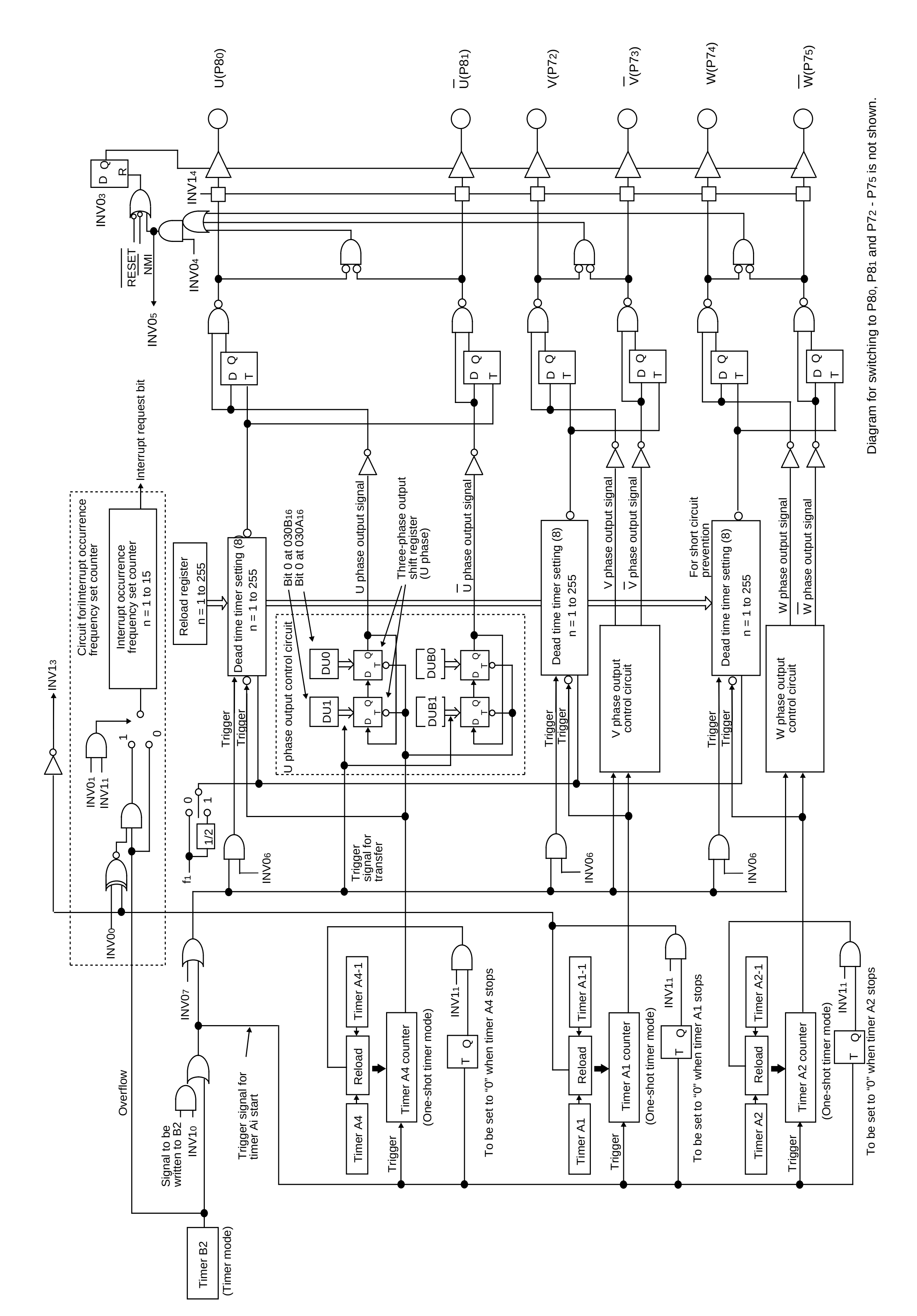
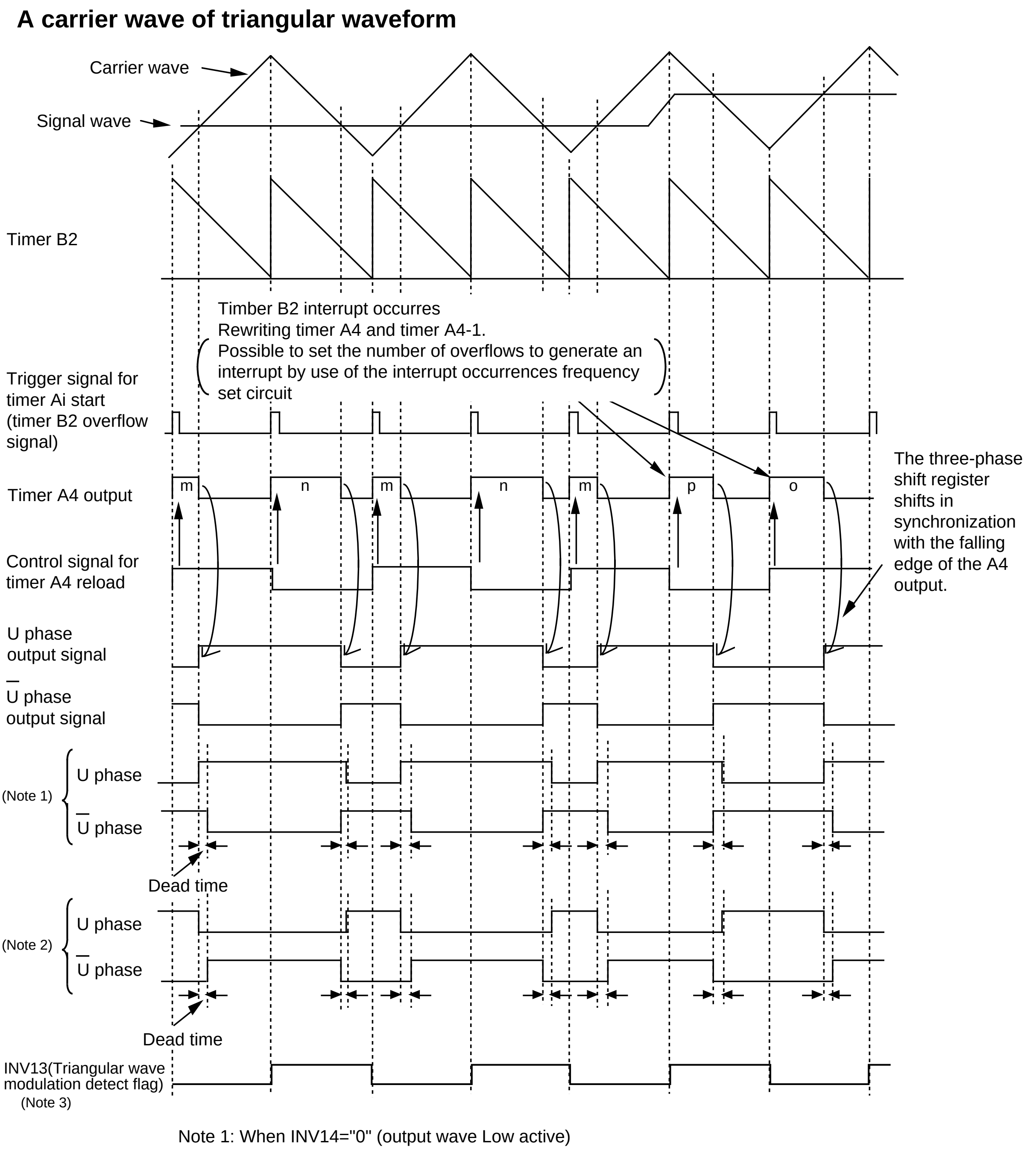


Figure 1.15.5. Block diagram for three-phase waveform mode

Triangular wave modulation

To generate a PWM waveform of triangular wave modulation, set "0" in the modulation mode select bit (bit 6 at 0308$_{16}$). Also, set "1" in the timers A4-1, A1-1, A2-1 control bit (bit 1 at 0309$_{16}$). In this mode, each of timers A4, A1, and A2 has two timer registers, and alternately reloads the timer register's content to the counter every time timer B2 counter's content becomes 0000$_{16}$. If "0" is set to the effective interrupt output specification bit (bit 1 at 0308$_{16}$), the frequency of interrupt requests that occur every time the timer B2 counter's value becomes 0000$_{16}$ can be set by use of the timer B2 counter (030D$_{16}$) for setting the frequency of interrupt occurrences. The frequency of occurrences is given by (setting; setting π 0).
Setting "1" in the effective interrupt output specification bit (bit 1 at 0308$_{16}$) provides the means to choose which value of the timer A1 reload control signal to use, "0" or "1", to cause timer B2's interrupt request to occur. To make this selection, use the effective interrupt output polarity selection bit (bit 0 at 0308$_{16}$).
An example of U phase waveform is shown in Figure 1.15.6, and the description of waveform output workings is given below. Set "1" in DU0 (bit 0 at 030A$_{16}$). And set "0" in DUB0 (bit 1 at 030A$_{16}$). In addition, set "0" in DU1 (bit 0 at 030B$_{16}$) and set "1" in DUB1 (bit 1 at 030B$_{16}$). Also, set "0" in the effective interrupt output specification bit (bit 1 at 0308$_{16}$) to set a value in the timer B2 interrupt occurrence frequency set counter. By this setting, a timer B2 interrupt occurs when the timer B2 counter's content becomes 0000$_{16}$ as many as (setting) times. Furthermore, set "1" in the effective interrupt output specification bit (bit 1 at 0308$_{16}$), set in the effective interrupt polarity select bit (bit 0 at 0308$_{16}$) and set "1" in the interrupt occurrence frequency set counter (030D$_{16}$). These settings cause a timer B2 interrupt to occur every other interval when the U phase output goes to "H".
When the timer B2 counter's content becomes 0000$_{16}$, timer A4 starts outputting one-shot pulses. In this instance, the content of DU1 (bit 0 at 030B$_{16}$) and that of DU0 (bit 0 at 030A$_{16}$) are set in the three-phase output shift register (U phase), the content of DUB1 (bit 1 at 030B$_{16}$) and that of DUB0 (bit 1 at 030A$_{16}$) are set in the three-phase shift register ($\overline{U}$ phase). After triangular wave modulation mode is selected, however, no setting is made in the shift register even though the timer B2 counter's content becomes 0000$_{16}$.
The value of DU0 and that of DUB0 are output to the U terminal (P8$_0$) and to the $\overline{U}$ terminal (P8$_1$) respectively. When the timer A4 counter counts the value written to timer A4 (034F$_{16}$, 034E$_{16}$) and when timer A4 finishes outputting one-shot pulses, the three-phase shift register's content is shifted one position, and the value of DU1 and that of DUB1 are output to the U phase output signal and to $\overline{U}$ phase output signal respectively. At this time, one-shot pulses are output from the timer for setting dead time used for setting the time over which the "L" level of the U phase waveform doesn't lap over the "L" level of the $\overline{U}$ phase waveform, which has the opposite phase of the former. The U phase waveform output that started from the "H" level keeps its level until the timer for setting dead time finishes outputting one-shot pulses even though the three-phase output shift register's content changes from "1" to "0" by the effect of the one-shot pulses. When the timer for setting dead time finishes outputting one-shot pulses, "0" already shifted in the three-phase shift register goes effective, and the U phase waveform changes to the "L" level. When the timer B2 counter's content becomes 0000$_{16}$, the timer A4 counter starts counting the value written to timer A4-1 (0307$_{16}$, 0306$_{16}$), and starts outputting one-shot pulses. When timer A4 finishes outputting one-shot pulses, the three-phase shift register's content is shifted one position, but if the three-phase output shift register's content changes from "0" to "1" as a result of the shift, the output level changes from "L" to "H" without waiting for the timer for setting dead time to finish outputting one-shot pulses. A U phase waveform is generated by these workings repeatedly. With the exception that the three-phase output shift register on the U phase side is used, the workings in generating a U phase waveform, which has the opposite phase of the U phase waveform, are the same as in generating a U

phase waveform. In this way, a waveform can be picked up from the applicable terminal in a manner in which the "L" level of the U phase waveform doesn't lap over that of the U phase waveform, which has the opposite phase of the U phase waveform. The width of the "L" level too can be adjusted by varying the values of timer B2, timer A4, and timer A4-1. In dealing with the V and W phases, and $\overline{V}$ and $\overline{W}$ phases, the latter are of opposite phase of the former, have the corresponding timers work similarly to dealing with the U and $\overline{U}$ phases to generate an intended waveform.


A carrier wave of triangular waveform
Carrier wave
Signal wave
Timer B2
Timber B2 interrupt occurres
Rewriting timer A4 and timer A4-1.
Possible to set the number of overflows to generate an interrupt by use of the interrupt occurrences frequency set circuit
Trigger signal for timer Ai start (timer B2 overflow signal)
Timer A4 output
m
n
m
n
m
p
o
Control signal for timer A4 reload
U phase output signal
U phase output signal
The three-phase shift register shifts in synchronization with the falling edge of the A4 output.
(Note 1)
U phase
U phase
Dead time
(Note 2)
U phase
U phase
Dead time
INV13(Triangular wave modulation detect flag) (Note 3)

Note 1: When INV14="0" (output wave Low active)
Note 2: When INV14="1" (output wave High active)
Note 3: Set to triangular wave modulation mode and to three-phase mode 1.

Figure 1.15.6. Timing chart of operation (1)

Assigning certain values to DU0 (bit 0 at $030A_{16}$) and DUB0 (bit 1 at $030A_{16}$), and to DU1 (bit 0 at $030B_{16}$) and DUB1 (bit 1 at $030B_{16}$) allows you to output the waveforms as shown in Figure 1.15.7, that is, to output the U phase alone, to fix $\overline{U}$ phase to "H", to fix the U phase to "H," or to output the $\overline{U}$ phase alone.

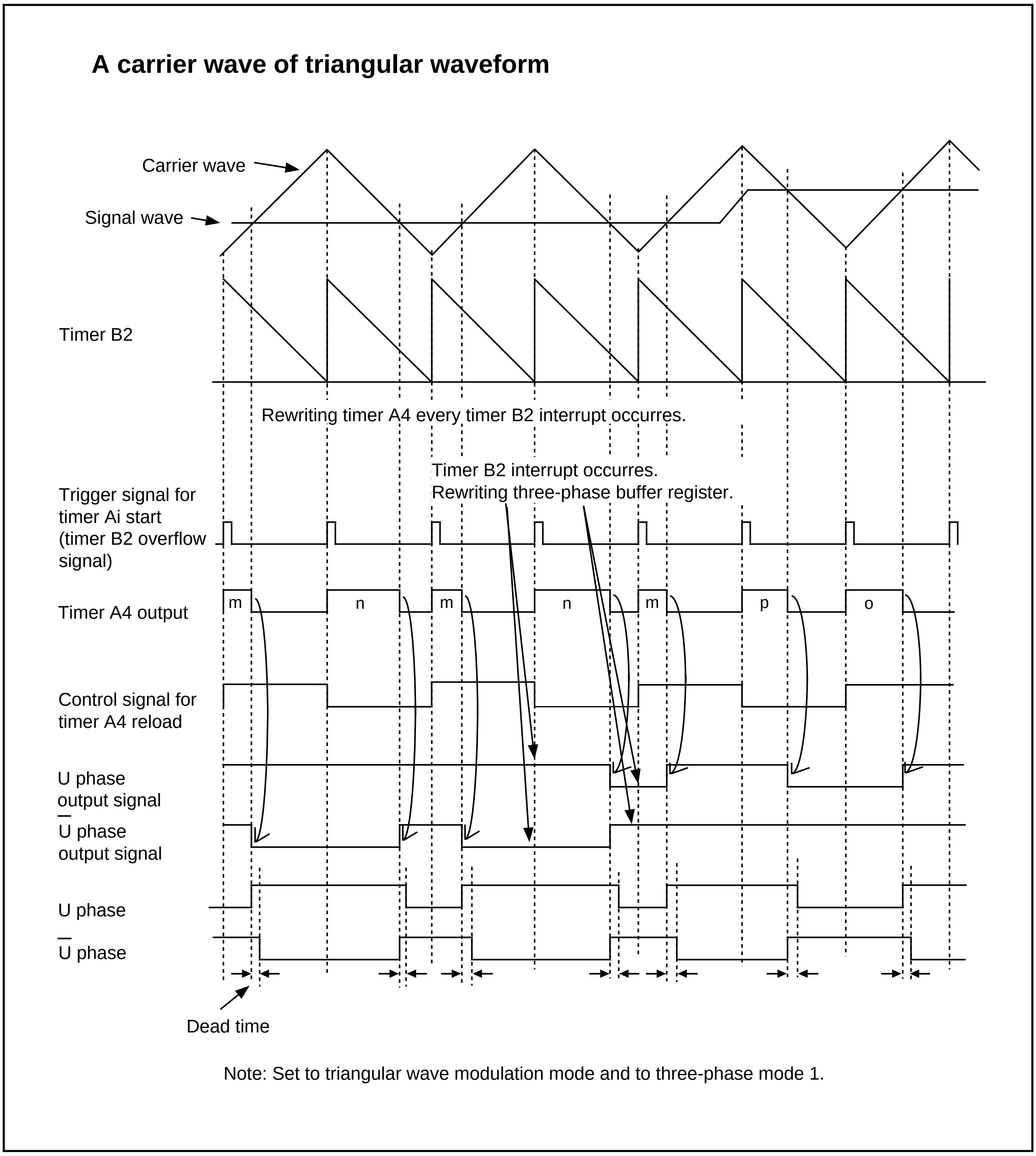

A carrier wave of triangular waveform
Carrier wave
Signal wave
Timer B2
Rewriting timer A4 every timer B2 interrupt occurres.
Timer B2 interrupt occurres.
Rewriting three-phase buffer register.
Trigger signal for timer Ai start (timer B2 overflow signal)
Timer A4 output
m
n
m
n
m
p
o
Control signal for timer A4 reload
U phase output signal
$\overline{U}$ phase output signal
U phase
$\overline{U}$ phase
Dead time
Note: Set to triangular wave modulation mode and to three-phase mode 1.

Figure 1.15.7. Timing chart of operation (2)

Sawtooth modulation

To generate a PWM waveform of sawtooth wave modulation, set “1” in the modulation mode select bit (bit 6 at 0308$_{16}$). Also, set “0” in the timers A4, A1, and A2-1 control bit (bit 1 at 0309$_{16}$). In this mode, the timer registers of timers A4, A1, and of A2 comprise conventional timers A4, A1, and A2 alone, and reload the corresponding timer register's content to the counter every time the timer B2 counter's content becomes 0000$_{16}$. The effective interrupt output specification bit (bit 1 at 0308$_{16}$) and the effective interrupt output polarity select bit (bit 0 at 0308$_{16}$) go nullified.

An example of U phase waveform is shown in Figure 1.15.8, and the description of waveform output workings is given below. Set “1” in DU0 (bit 0 at 030A$_{16}$), and set “0” in DUB0 (bit 1 at 030A$_{16}$). In addition, set “0” in DU1 (bit 0 at 030B$_{16}$) and set “1” in DUB1 (bit 1 at 030B$_{16}$).

When the timber B2 counter's content becomes 0000$_{16}$, timer B2 generates an interrupt, and timer A4 starts outputting one-shot pulses at the same time. In this instance, the contents of the three-phase buffer registers DU1 and DU0 are set in the three-phase output shift register (U phase), and the contents of DUB1 and DUB0 are set in the three-phase output register (U phase). After this, the three-phase buffer register's content is set in the three-phase shift register every time the timer B2 counter's content becomes 0000$_{16}$.

The value of DU0 and that of DUB0 are output to the U terminal (P8$_0$) and to the $\overline{U}$ terminal (P8$_1$) respectively. When the timer A4 counter counts the value written to timer A4 (034F$_{16}$, 034E$_{16}$) and when timer A4 finishes outputting one-shot pulses, the three-phase output shift register's content is shifted one position, and the value of DU1 and that of DUB1 are output to the U phase output signal and to the $\overline{U}$ output signal respectively. At this time, one-shot pulses are output from the timer for setting dead time used for setting the time over which the “L” level of the U phase waveform doesn't lap over the “L” level of the $\overline{U}$ phase waveform, which has the opposite phase of the former. The U phase waveform output that started from the “H” level keeps its level until the timer for setting dead time finishes outputting one-shot pulses even though the three-phase output shift register's content changes from “1” to “0 ”by the effect of the one-shot pulses. When the timer for setting dead time finishes outputting one-shot pulses, 0 already shifted in the three-phase shift register goes effective, and the U phase waveform changes to the “L” level. When the timer B2 counter's content becomes 0000$_{16}$, the contents of the three-phase buffer registers DU1 and DU0 are set in the three-phase shift register (U phase), and the contents of DUB1 and DUB0 are set in the three-phase shift register ($\overline{U}$ phase) again.

A U phase waveform is generated by these workings repeatedly. With the exception that the three-phase output shift register on the $\overline{U}$ phase side is used, the workings in generating a $\overline{U}$ phase waveform, which has the opposite phase of the U phase waveform, are the same as in generating a U phase waveform. In this way, a waveform can be picked up from the applicable terminal in a manner in which the “L” level of the U phase waveform doesn't lap over that of the U phase waveform, which has the opposite phase of the U phase waveform. The width of the “L” level too can be adjusted by varying the values of timer B2 and timer A4. In dealing with the V and W phases, and $\overline{V}$ and $\overline{W}$ phases, the latter are of opposite phase of the former, have the corresponding timers work similarly to dealing with the U and $\overline{U}$ phases to generate an intended waveform.

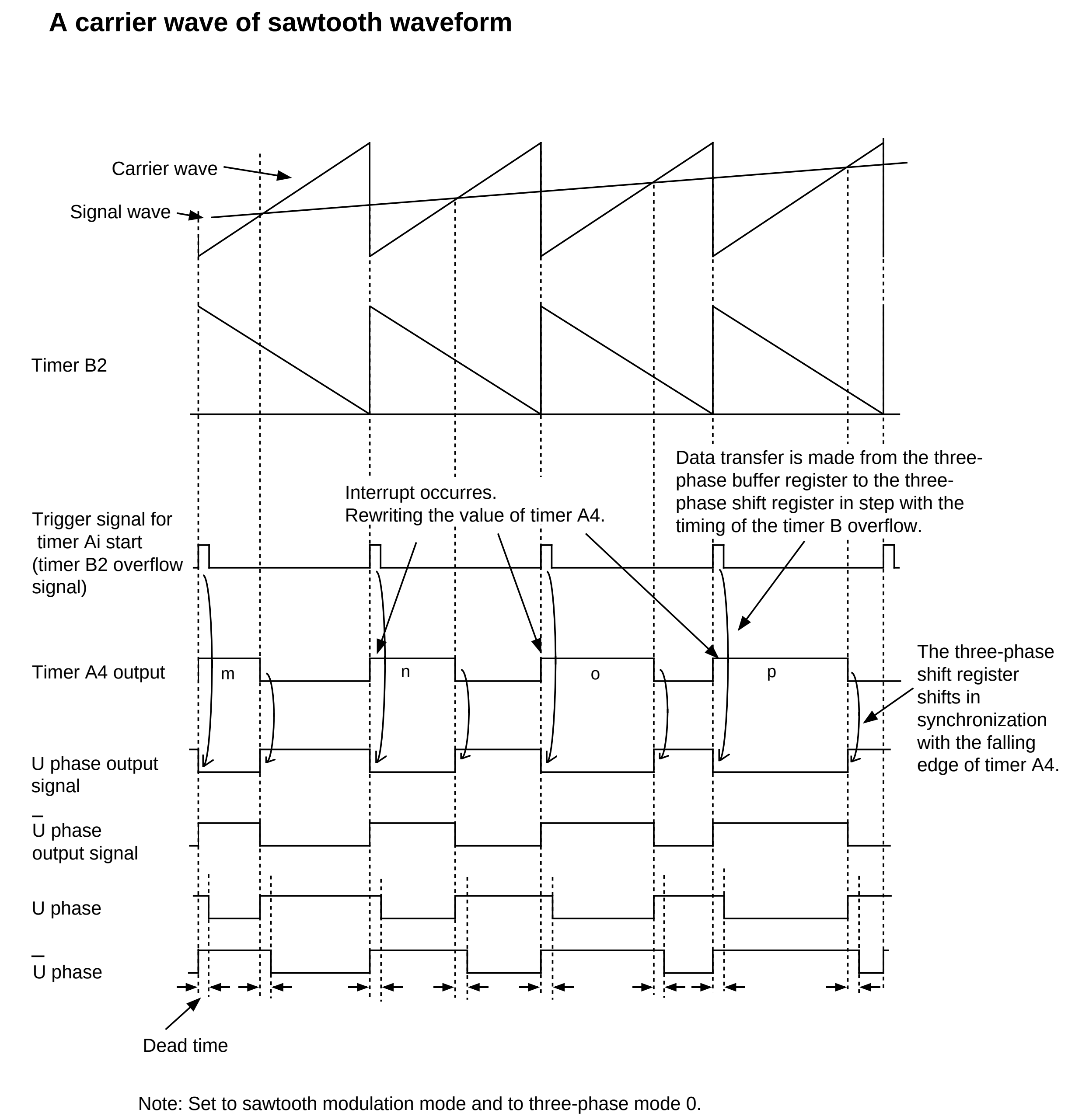
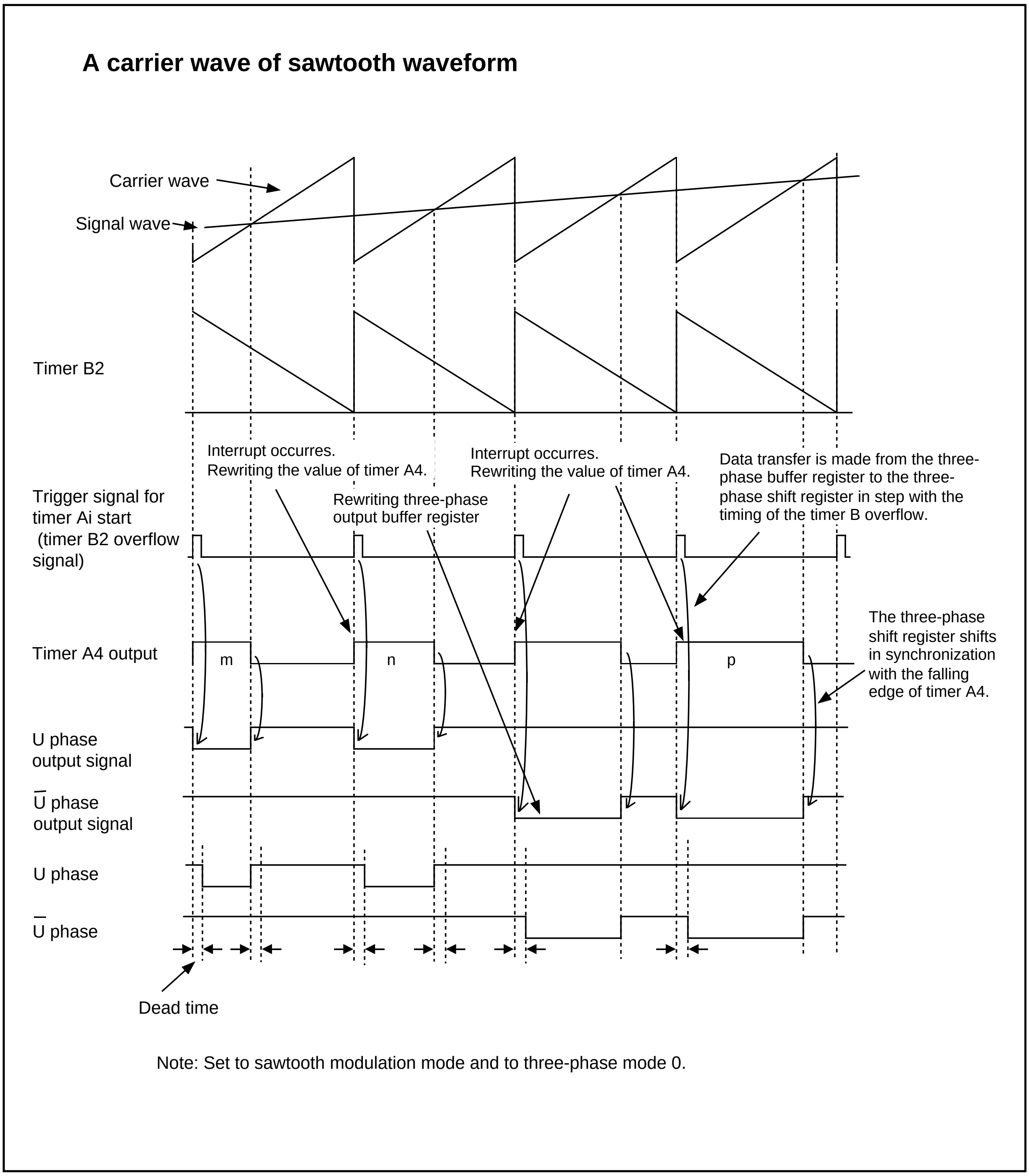

A carrier wave of sawtooth waveform
Carrier wave
Signal wave
Timer B2
Interrupt occurres.
Rewriting the value of timer A4.
Data transfer is made from the three-phase buffer register to the three-phase shift register in step with the timing of the timer B overflow.
Trigger signal for timer Ai start (timer B2 overflow signal)
Timer A4 output
m
n
o
p
The three-phase shift register shifts in synchronization with the falling edge of timer A4.
U phase output signal
$\overline{U}$ phase output signal
U phase
$\overline{U}$ phase
Dead time
Note: Set to sawtooth modulation mode and to three-phase mode 0.

Figure 1.15.8. Timing chart of operation (3)

Setting “1” both in DUB0 and in DUB1 provides a means to output the U phase alone and to fix the $\overline{U}$ phase output to “H” as shown in Figure 1.15.9.


A carrier wave of sawtooth waveform
Carrier wave
Signal wave
Timer B2
Interrupt occurres.
Rewriting the value of timer A4.
Interrupt occurres.
Rewriting the value of timer A4.
Data transfer is made from the three-phase buffer register to the three-phase shift register in step with the timing of the timer B overflow.
Rewriting three-phase output buffer register
Trigger signal for timer Ai start (timer B2 overflow signal)
Timer A4 output
m
n
p
The three-phase shift register shifts in synchronization with the falling edge of timer A4.
U phase output signal
$\overline{U}$ phase output signal
U phase
$\overline{U}$ phase
Dead time

Note: Set to sawtooth modulation mode and to three-phase mode 0.

Figure 1.15.9. Timing chart of operation (4)

Serial I/O

Serial I/O is configured as five channels: UART0 to UART4.

UART0 to 4

UART0 to UART4 each have an exclusive timer to generate a transfer clock, so they operate independently of each other.

Figure 1.16.1 and 1.16.2 show the block diagram of UARTi (i=0 to 4). Figures 1.16.3 and 1.16.4 show the block diagram of the transmit/receive unit.

UARTi has two operation modes: a clock synchronous serial I/O mode and a clock asynchronous serial I/O mode (UART mode). The contents of the serial I/O mode select bits (bits 0 to 2 at addresses 0360_{16}, 0368_{16}, 0338_{16}, 0328_{16} and $02F8_{16}$) determine whether UARTi is used as a clock synchronous serial I/O or as a UART.

Although a few functions are different, UART0 to UART4 have almost the same functions.

UART2 to UART4, in particular, are compliant with the SIM interface with some extra settings added in clock-asynchronous serial I/O mode (Note). It also has the bus collision detection function that generates an interrupt request if the TxD pin and the RxD pin are different in level.

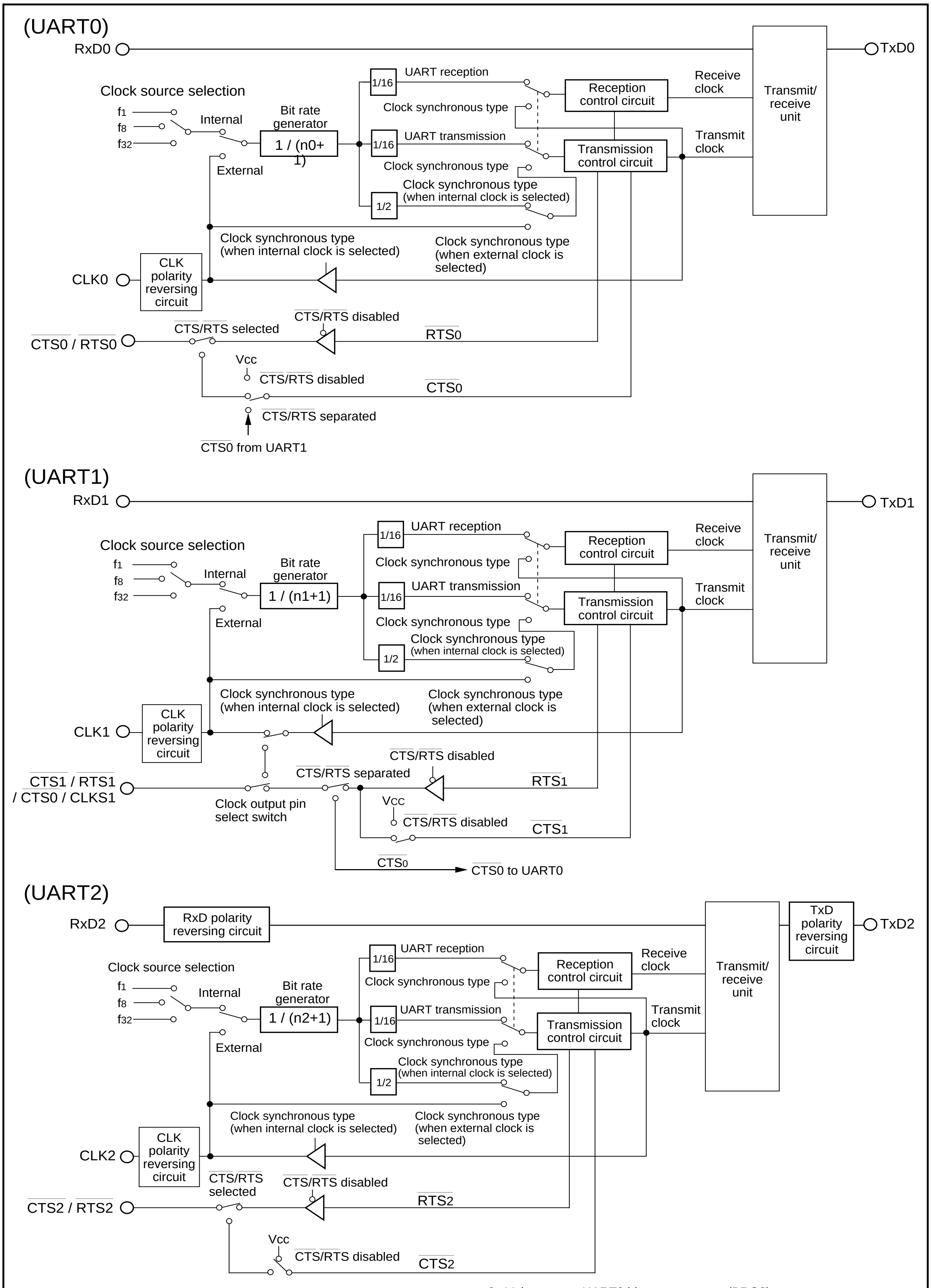
Table 1.16.1 shows the comparison of functions of UART0 to UART4, and Figures 1.16.5 through 1.16.11 show the registers related to UARTi.

Note: SIM : Subscriber Identity Module

Table 1.16.1. Comparison of functions of UART0 to UART4

Function	UART0	UART1	UART2	UART3	UART4
CLK polarity selection	Possible (Note 1)	Possible (Note 1)	Possible (Note 1)	Possible (Note 1)	Possible (Note 1)
LSB first / MSB first selection	Possible (Note 1)	Possible (Note 1)	Possible (Note 2)	Possible (Note 2)	Possible (Note 2)
Continuous receive mode selection	Possible (Note 1)	Possible (Note 1)	Possible (Note 1)	Possible (Note 1)	Possible (Note 1)
Transfer clock output from multiple pins selection	Impossible	Possible (Note 1)	Impossible	Impossible	Impossible
Separate $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ pins	Possible	Impossible	Impossible	Impossible	Impossible
Serial data logic switch	Impossible	Impossible	Possible (Note 4)	Possible (Note 4)	Possible (Note 4)
Sleep mode selection	Possible (Note 3)	Possible (Note 3)	Impossible	Impossible	Impossible
TxD, RxD I/O polarity switch	Impossible	Impossible	Possible	Possible	Possible
TxD, RxD port output format	CMOS output	CMOS output	N-channel open drain output	CMOS output	CMOS output
Parity error signal output	Impossible	Impossible	Possible (Note 4)	Possible (Note 4)	Possible (Note 4)
Bus collision detection	Impossible	Impossible	Possible	Possible	Possible

Note 1: Only when clock synchronous serial I/O mode.
Note 2: Only when clock synchronous serial I/O mode and 8-bit UART mode.
Note 3: Only when UART mode.
Note 4: Using for SIM interface.



n0 : Values set to UART0 bit rate generator (BRG0)
n1 : Values set to UART1 bit rate generator (BRG1)
n2 : Values set to UART2 bit rate generator (BRG2)

Figure 1.16.1. Block diagram of UARTi (i = 0 to 2)

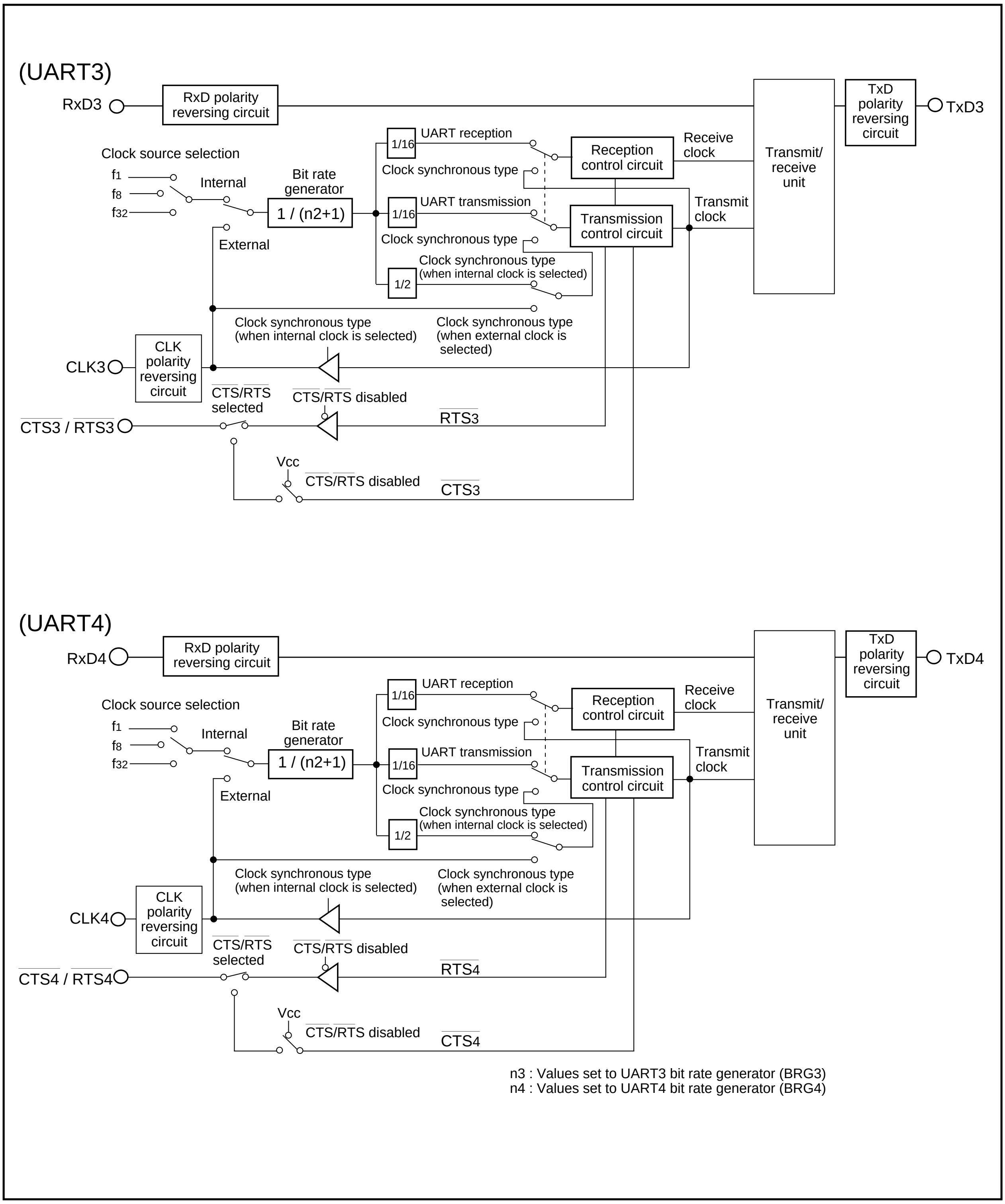

(UART3)
RxD3
RxD polarity reversing circuit
TxD polarity reversing circuit
TxD3
Transmit/ receive unit
1/16
UART reception
Clock synchronous type
Reception control circuit
Receive clock
Clock source selection
f1
f8
f32
Internal
Bit rate generator
1 / (n2+1)
1/16
UART transmission
Clock synchronous type
Transmission control circuit
Transmit clock
External
Clock synchronous type (when internal clock is selected)
1/2
Clock synchronous type (when internal clock is selected)
Clock synchronous type (when external clock is selected)
CLK3
CLK polarity reversing circuit
CTS/RTS selected
CTS/RTS disabled
CTS3 / RTS3
RTS3
Vcc
CTS/RTS disabled
CTS3
(UART4)
RxD4
RxD polarity reversing circuit
TxD polarity reversing circuit
TxD4
Transmit/ receive unit
1/16
UART reception
Clock synchronous type
Reception control circuit
Receive clock
Clock source selection
f1
f8
f32
Internal
Bit rate generator
1 / (n2+1)
1/16
UART transmission
Clock synchronous type
Transmission control circuit
Transmit clock
External
Clock synchronous type (when internal clock is selected)
1/2
Clock synchronous type (when internal clock is selected)
Clock synchronous type (when external clock is selected)
CLK4
CLK polarity reversing circuit
CTS/RTS selected
CTS/RTS disabled
CTS4 / RTS4
RTS4
Vcc
CTS/RTS disabled
CTS4
n3 : Values set to UART3 bit rate generator (BRG3)
n4 : Values set to UART4 bit rate generator (BRG4)

Figure 1.16.2. Block diagram of UARTi (i = 3, 4)

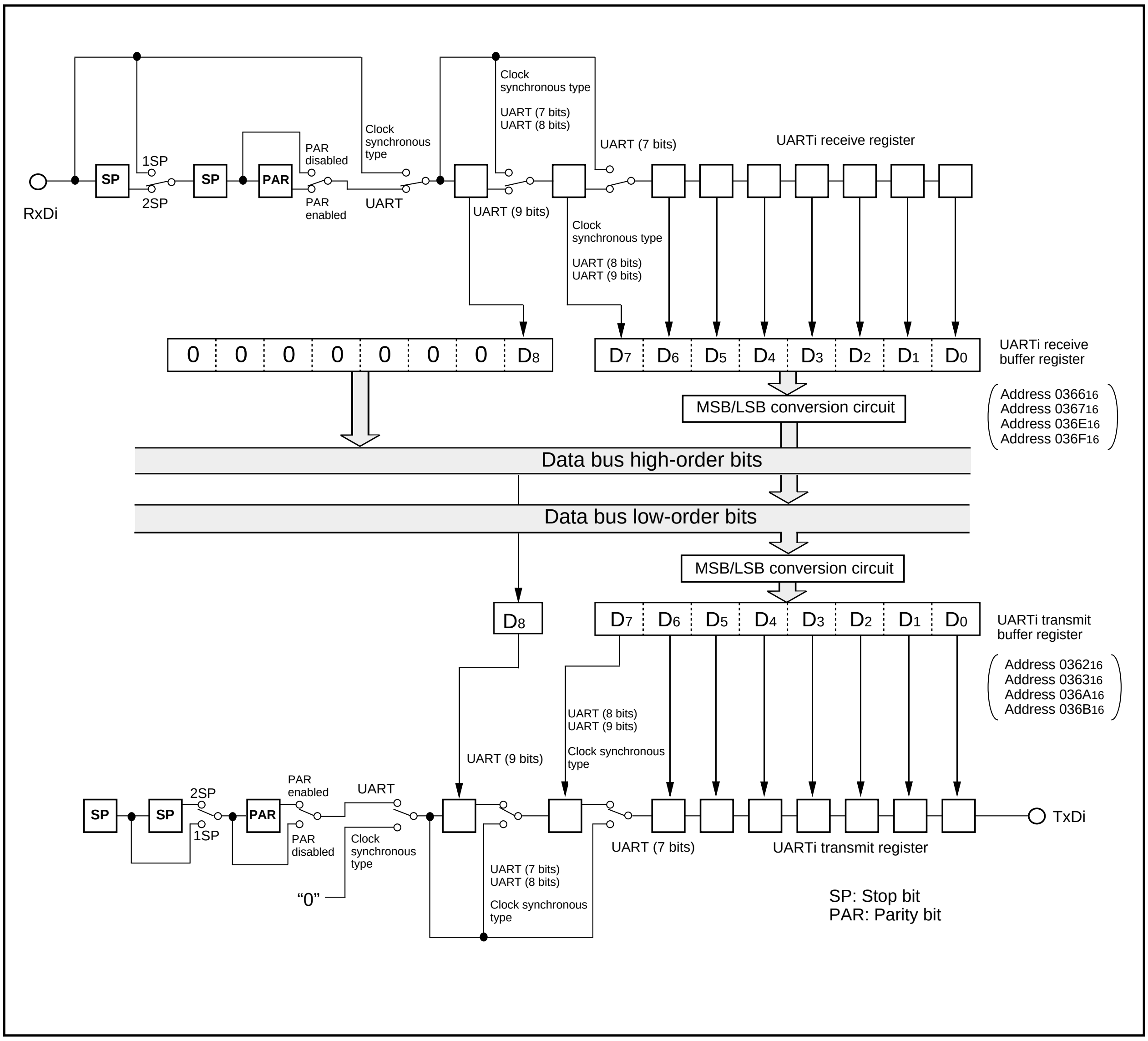


Figure 1.16.3. Block diagram of UARTi (i = 0, 1) transmit/receive unit

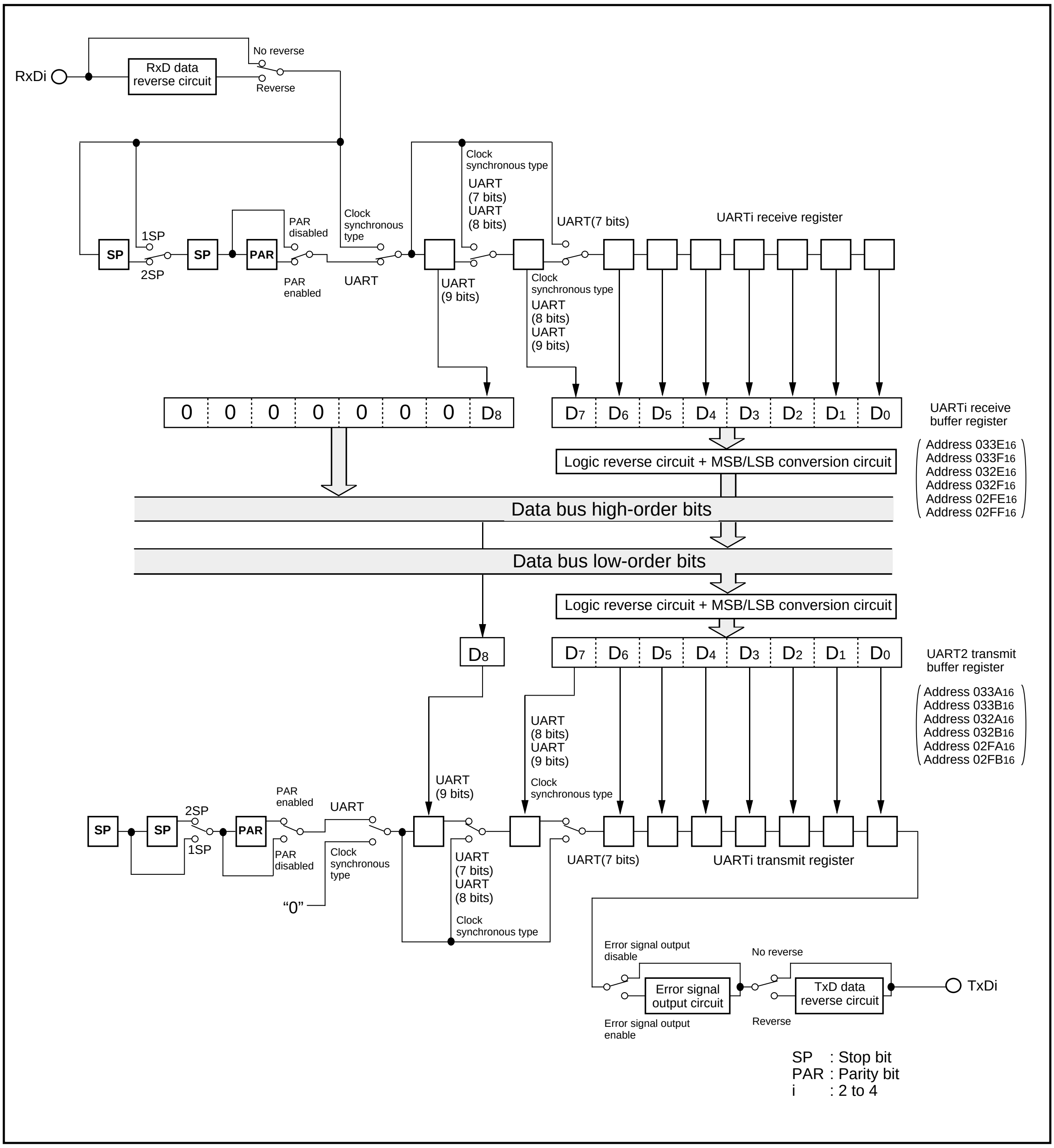
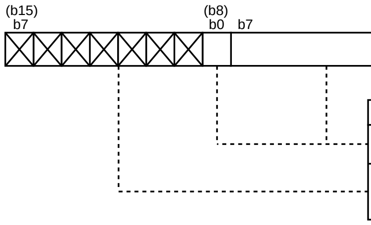


Figure 1.16.4. Block diagram of UARTi (i = 2 to 4) transmit/receive unit

UARTi transmit buffer register (Note)

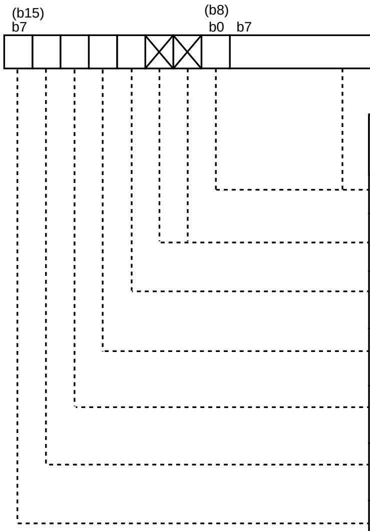


Symbol	Address	When reset
U0TB	0363$_{16}$, 0362$_{16}$	Indeterminate
U1TB	036B$_{16}$, 036A$_{16}$	Indeterminate
U2TB	033B$_{16}$, 033A$_{16}$	Indeterminate
U3TB	032B$_{16}$, 032A$_{16}$	Indeterminate
U4TB	02FB$_{16}$, 02FA$_{16}$	Indeterminate

Function	R	W
Transmit data	×	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.	—	—

Note: Use MOV instruction to write to this register.

UARTi receive buffer register



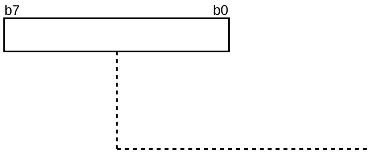
Symbol	Address	When reset
U0RB	0367$_{16}$, 0366$_{16}$	Indeterminate
U1RB	036F$_{16}$, 036E$_{16}$	Indeterminate
U2RB	033F$_{16}$, 033E$_{16}$	Indeterminate
U3RB	032F$_{16}$, 032E$_{16}$	Indeterminate
U4RB	02FF$_{16}$, 02FE$_{16}$	Indeterminate

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
—	——	Receive data	Receive data	○	×
Nothing is assigned. When write, set "0". When read, the value of these bits is “0”.				—	—
ABT	Arbitration lost detecting flag (Note 2)	0 : Not detected 1 : Detected	Invalid	○	○
OER	Overrun error flag (Note 1)	0 : No overrun error 1 : Overrun error found	0 : No overrun error 1 : Overrun error found	○	×
FER	Framing error flag (Note 1)	Invalid	0 : No framing error 1 : Framing error found	○	×
PER	Parity error flag (Note 1)	Invalid	0 : No parity error 1 : Parity error found	○	×
SUM	Error sum flag (Note 1)	Invalid	0 : No error 1 : Error found	○	×

Note 1: Bits 15 through 12 are set to “0” when the serial I/O mode select bit (bits 2 to 0 at addresses 0360$_{16}$, 0368$_{16}$, 0338$_{16}$, 0328$_{16}$ and 02F8$_{16}$) are set to “000$_2$” or the receive enable bit is set to “0”.
(Bit 15 is set to “0” when bits 14 to 12 all are set to “0”.) Bits 14 and 13 are also set to “0” when the lower byte of the UARTi receive buffer register (addresses 0366$_{16}$, 036E$_{16}$, 033E$_{16}$, 032E$_{16}$ and 02FE$_{16}$) is read out.

Note 2: Arbitration lost detecting flag is allocated to U2RB, U3RB and U4RB and nothing but “0” may be written. Nothing is assigned in bit 11 of U0RB and U1RB. When write, set "0". When read, the value of this bit is “0”.

UARTi bit rate generator (Note 1, 2)



Symbol	Address	When reset
U0BRG	0361$_{16}$	Indeterminate
U1BRG	0369$_{16}$	Indeterminate
U2BRG	0339$_{16}$	Indeterminate
U3BRG	0329$_{16}$	Indeterminate
U4BRG	02F9$_{16}$	Indeterminate

Function	Values that can be set	R	W
Assuming that set value = n, BRGi divides the count source by n + 1	00$_{16}$ to FF$_{16}$	×	○

Note 1: Use MOV instruction to write to this register.
Note 2: Write a value to this register while transmit/receive halts.

Figure 1.16.5. Serial I/O-related registers (1)

UARTi transmit/receive mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UiMR(i=0,1)	0360_{16}, 0368_{16}	00_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode select bit	Must be fixed to 001 b2 b1 b0 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited	○	○
SMD1				○	○
SMD2				○	○
CKDIR	Internal/external clock select bit	0 : Internal clock (Note 1) 1 : External clock (Note 2)	0 : Internal clock 1 : External clock (Note 2)	○	○
STPS	Stop bit length select bit	Invalid	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd/even parity select bit	Invalid	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	Invalid	0 : Parity disabled 1 : Parity enabled	○	○
SLEP	Sleep select bit	Set to “0”	0 : Sleep mode deselected 1 : Sleep mode selected	○	○

Note 1: Select CLK output by the corresponding function select registers A, B and C.
Note 2: Set the corresponding function select register A to the I/O port.

UARTi transmit/receive mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UiMR (i=2 to 4)	0338_{16}, 0328_{16}, $02F8_{16}$	00_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
SMD0	Serial I/O mode select bit	Must be fixed to 001 b2 b1 b0 0 0 0 : Serial I/O invalid 0 1 0 : (Note) 0 1 1 : Inhibited 1 1 1 : Inhibited	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long 0 0 0 : Serial I/O invalid 0 1 0 : Inhibited 0 1 1 : Inhibited 1 1 1 : Inhibited	○	○
SMD1				○	○
SMD2				○	○
CKDIR	Internal/external clock select bit	0 : Internal clock (Note 2) 1 : External clock (Note 3)	0 : Internal clock 1 : External clock (Note 3)	○	○
STPS	Stop bit length select bit	Invalid	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd/even parity select bit	Invalid	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	Invalid	0 : Parity disabled 1 : Parity enabled	○	○
IOPOL	TxD, RxD I/O polarity reverse bit	0 : No reverse 1 : Reverse Usually set to “0”	0 : No reverse 1 : Reverse Usually set to “0”	○	○

Note 1: Bit 2 to bit 0 are set to “010_2” when I^2C mode is used.
Note 2: Select CLK output by the corresponding function select registers A, B and C.
Note 3: Set the corresponding function select register A to the I/O port.

Figure 1.16.6. Serial I/O-related registers (2)

UARTi transmit/receive control register 0

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UiC0(i=0,1)	0364_{16}, $036C_{16}$	08_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source select bit	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : Inhibited	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : Inhibited	○	○
CLK1				○	○
CRS	$\overline{CTS}$/$\overline{RTS}$ function select bit	Valid when bit 4 = “0” 0 : $\overline{CTS}$ function is selected (Note 1) 1 : $\overline{RTS}$ function is selected (Note 2)	Valid when bit 4 = “0” 0 : $\overline{CTS}$ function is selected (Note 1) 1 : $\overline{RTS}$ function is selected (Note 2)	○	○
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	○	×
CRD	$\overline{CTS}$/$\overline{RTS}$ disable bit	0 : $\overline{CTS}$/$\overline{RTS}$ function enabled 1 : $\overline{CTS}$/$\overline{RTS}$ function disabled	0 : $\overline{CTS}$/$\overline{RTS}$ function enabled 1 : $\overline{CTS}$/$\overline{RTS}$ function disabled	○	○
NCH	Data output select bit	0 : TXDi pin is CMOS output 1 : TXDi pin is N-channel open drain output	0: TXDi pin is CMOS output 1: TXDi pin is N-channel open drain output	○	○
CKPOL	CLK polarity select bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	Set to “0”	○	○
UFORM	Transfer format select bit	0 : LSB first 1 : MSB first	Set to “0”	○	○

Note 1: Set the corresponding function select register A to I/O port, and port direction register to “0”.
Note 2: Select $\overline{RTS}$ output using the corresponding function select registers A and B.

UART2 transmit/receive control register 0

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
U2C0	$033C_{16}$	08_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source select bit	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : Inhibited	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : Inhibited	○	○
CLK1				○	○
CRS	$\overline{CTS}$/$\overline{RTS}$ function select bit	Valid when bit 4 = “0” 0 : $\overline{CTS}$ function is selected (Note 1) 1 : $\overline{RTS}$ function is selected (Note 2)	Valid when bit 4 = “0” 0 : $\overline{CTS}$ function is selected (Note 1) 1 : $\overline{RTS}$ function is selected (Note 2)	○	○
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	○	×
CRD	$\overline{CTS}$/$\overline{RTS}$ disable bit	0 : $\overline{CTS}$/$\overline{RTS}$ function enabled 1 : $\overline{CTS}$/$\overline{RTS}$ function disabled	0 : $\overline{CTS}$/$\overline{RTS}$ function enabled 1 : $\overline{CTS}$/$\overline{RTS}$ function disabled	○	○
Nothing is assigned. When write, set “0”. When read, the value of this bit is “0”.				—	—
CKPOL	CLK polarity select bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	Set to “0”	○	○
UFORM	Transfer format select bit (Note 3)	0 : LSB first 1 : MSB first	0 : LSB first 1 : MSB first	○	○

Note 1: Set the corresponding function select register A to I/O port, and port direction register to “0”.
Note 2: Select $\overline{RTS}$ output using the corresponding function select registers A and B.
Note 3: Only clock synchronous serial I/O mode and 8-bit UART mode are valid.

Figure 1.16.7. Serial I/O-related registers (3)

UARTi transmit/receive control register 0

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UiC0(i=3,4)	$032C_{16}$, $02FC_{16}$	08_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
CLK0	BRG count source select bit	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : Inhibited	b1 b0 0 0 : f_1 is selected 0 1 : f_8 is selected 1 0 : f_{32} is selected 1 1 : Inhibited	○	○
CLK1				○	○
CRS	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function select bit	Valid when bit 4 = “0” 0 : $\overline{\text{CTS}}$ function is selected (Note 1) 1 : $\overline{\text{RTS}}$ function is selected (Note 2)	Valid when bit 4 = “0” 0 : $\overline{\text{CTS}}$ function is selected (Note 1) 1 : $\overline{\text{RTS}}$ function is selected (Note 2)	○	○
TXEPT	Transmit register empty flag	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	0 : Data present in transmit register (during transmission) 1 : No data present in transmit register (transmission completed)	○	×
CRD	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit	0 : $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function enabled 1 : $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function disabled	0 : $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function enabled 1 : $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function disabled	○	○
NCH	Data output select bit	0 : TXDi pin is CMOS output 1 : TXDi pin is N-channel open drain output	0: TXDi pin is CMOS output 1: TXDi pin is N-channel open drain output	○	○
CKPOL	CLK polarity select bit	0 : Transmit data is output at falling edge of transfer clock and receive data is input at rising edge 1 : Transmit data is output at rising edge of transfer clock and receive data is input at falling edge	Set to “0”	○	○
UFORM	Transfer format select bit (Note 3)	0 : LSB first 1 : MSB first	0 : LSB first 1 : MSB first	○	○

Note 1: Set the corresponding function select register A to I/O port, and port direction register to “0”.
Note 2: Select $\overline{\text{RTS}}$ output using the corresponding function select registers A and B.
Note 3: Valid only in clock syncronous serial I/O mode and 8 bits UART mode.

Figure 1.16.8. Serial I/O-related registers (4)

UARTi transmit/receive control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UiC1(i=0,1)	0365_{16},$036D_{16}$	02_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0 : Transmission disabled 1 : Transmission enabled	○	○
TI	Transmit buffer empty flag	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	○	×
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	0 : Reception disabled 1 : Reception enabled	○	○
RI	Receive complete flag	0 : No data present in receive buffer register 1 : Data present in receive buffer register	0 : No data present in receive buffer register 1 : Data present in receive buffer register	○	×
Nothing is assigned. When write, set "0". When read, the value of these bits is “0”.				–	–

UARTi transmit/receive control register 1

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UiC1 (i=2 to 4)	$033D_{16}$, $032D_{16}$, $02FD_{16}$	02_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
TE	Transmit enable bit	0 : Transmission disabled 1 : Transmission enabled	0 : Transmission disabled 1 : Transmission enabled	○	○
TI	Transmit buffer empty flag	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	0 : Data present in transmit buffer register 1 : No data present in transmit buffer register	○	×
RE	Receive enable bit	0 : Reception disabled 1 : Reception enabled	0 : Reception disabled 1 : Reception enabled	○	○
RI	Receive complete flag	0 : No data present in receive buffer register 1 : Data present in receive buffer register	0 : No data present in receive buffer register 1 : Data present in receive buffer register	○	×
UiIRS	UARTi transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmit is completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmit is completed (TXEPT = 1)	○	○
UiRRM	UARTi continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	Set to “0”	○	○
UiLCH	Data logic select bit	0 : No reverse 1 : Reverse	0 : No reverse 1 : Reverse	○	○
UiERE	Error signal output enable bit	Set to “0”	0 : Output disabled 1 : Output enabled	○	○

Figure 1.16.9. Serial I/O-related registers (5)

UART transmit/receive control register 2

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UCON	0370_{16}	$X0000000_2$

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
U0IRS	UART0 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	○	○
U1IRS	UART1 transmit interrupt cause select bit	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	0 : Transmit buffer empty (TI = 1) 1 : Transmission completed (TXEPT = 1)	○	○
U0RRM	UART0 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enable	Set to "0"	○	○
U1RRM	UART1 continuous receive mode enable bit	0 : Continuous receive mode disabled 1 : Continuous receive mode enabled	Set to "0"	○	○
Nothing is assigned. When write, set "0". When read, its content is indeterminate.				–	–
RCSP	Separate $\overline{CTS}/\overline{RTS}$ bit	0 : $\overline{CTS}/\overline{RTS}$ shared pin 1 : $\overline{CTS}/\overline{RTS}$ separated	0 : $\overline{CTS}/\overline{RTS}$ shared pin 1 : $\overline{CTS}/\overline{RTS}$ separated	○	○
Nothing is assigned. When write, set "0". When read, its content is indeterminate.				–	–

UARTi special mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UiSMR (i=2 to 4)	0337_{16}, 0327_{16}, $02F7_{16}$	00_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
IICM	IIC mode select bit	0 : Normal mode 1 : IIC mode	Set to "0"	○	○
ABC	Arbitration lost detecting flag control bit	0 : Update per bit 1 : Update per byte	Set to "0"	○	○
BBS	Bus busy flag	0 : STOP condition detected 1 : START condition detected	Set to "0"	○	○ (Note1)
LSYN	SCLL sync output enable bit	0 : Disabled 1 : Enabled	Set to "0"	○	○
ABSCS	Bus collision detect sampling clock select bit	Set to "0"	0 : Rising edge of transfer clock 1 : Underflow signal of timer Ai (Note 2)	○	○
ACSE	Auto clear function select bit of transmit enable bit	Set to "0"	0 : No auto clear function 1 : Auto clear at occurrence of bus collision	○	○
SSS	Transmit start condition select bit	Set to "0"	0 : Ordinary 1 : Falling edge of RxDi	○	○
Nothing is assigned. When write, set "0". When read, its content is indeterminate.				–	–

Note 1: Nothing but "0" may be written.
Note 2: UART2 : timer A0 underflow signal, UART3 : timer A3 underflow signal, UART4 : timer A4 underflow signal.

Figure 1.16.10. Serial I/O-related registers (6)

UARTi special mode register 2

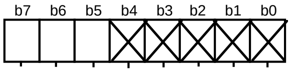
b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UiSMR2 (i=2 to 4)	0336_{16}, 0326_{16}, $02F6_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
IICM2	IIC mode select bit 2	0 : NACK/ACK interrupt DMA source - ACK Transfer to receive buffer at the rising edge of last bit of receive clock Receive interrupt is occurred at the rising edge of last bit of receive clock 1 : UART transfer/receive interrupt DMA source - UART receive Transfer to receive buffer at the falling edge of last bit of receive clock Receive interrupt is occurred at the falling edge of last bit of receive clock	○	○
CSC	Clock synchronous bit	0 : Disabled 1 : Enabled	○	○
SWC	SCL wait output bit	0 : Disabled 1 : Enabled	○	○
ALS	SDA output stop flag	0 : Disabled 1 : Enabled	○	○
STC	UARTi initialize bit	0 : Disabled 1 : Enabled	○	○
SWC2	SCL wait output bit 2	0 : UARTi clock 1 : 0 output	○	○
SDHI	SDA output inhibit bit	0 : Enabled 1 : Disabled (high impedance)	○	○
SHTC	Start/stop condition control bit	Must set to "1" in selecting IIC mode.	○	○

Figure 1.16.11. Serial I/O-related registers (7)

UART2 special mode register 3



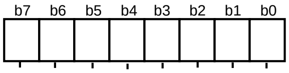
Symbol	Address	When reset
U2SMR3	0335_{16}	$000XXXXX_2$

Bit symbol	Bit name	Function	R	W
	Nothing is assigned. These bits can neither be set nor reset. When read, their contents are indeterminate.		—	—
DL0	SDA_2(TxD_2) digital delay time set bit (Note 1,2)	b7 b6 b5 000:Without delay 001:2-cycle of 1/f(X_{IN}) 010:3-cycle of 1/f(X_{IN}) 011:4-cycle of 1/f(X_{IN}) 100:5-cycle of 1/f(X_{IN}) 101:6-cycle of 1/f(X_{IN}) 110:7-cycle of 1/f(X_{IN}) 111:8-cycle of 1/f(X_{IN})	○	○
DL1			○	○
DL2			○	○

Note 1: These bits are used for SDA_2(TxD_2) output digital delay when using UART2 for IIC interface. Otherwise, must set to "000".

Note 2: When external clock is selected, delay is increased approx. 100ns.

UARTi special mode register 3 (i=3,4)



Symbol	Address	When reset
U3SMR3	0325_{16}	00000000_2
U4SMR3	$02F5_{16}$	00000000_2

Bit symbol	Bit name	Function	R	W
SSE	$\overline{SS}$ port function enable bit (Note 3)	0: $\overline{SS}$ function disable 1: $\overline{SS}$ function enable	○	○
CKPH	Clock phase set bit	0: Without clock delay 1: With clock delay	○	○
DINC	Serial input port set bit	0: Select TxDi and RxDi (master mode) (Note 5) 1: Select STxDi and SRxDi (slave mode) (Note 6)	○	○
NODC	Clock output select bit	0: CLKi is CMOS output 1: CLKi is N-channel open drain output	○	○
ERR	Fault error flag	0: Without fault error 1: With fault error	○	○ (Note 4)
DL0	SDAi(TxD_2) digital delay time set bit (Note 1,2)	b7 b6 b5 000 :Without delay 001 :2-cycle of 1/f(X_{IN}) 010 :3-cycle of 1/f(X_{IN}) 011 :4-cycle of 1/f(X_{IN}) 100 :5-cycle of 1/f(X_{IN}) 101 :6-cycle of 1/f(X_{IN}) 110 :7-cycle of 1/f(X_{IN}) 111 :8-cycle of 1/f(X_{IN})	○	○
DL1			○	○
DL2			○	○

Note 1: These bits are used for SDAi(TxDi) output digital delay when using UARTi for IIC interface. Otherwise, must set to "000".

Note 2: When external clock is selected, delay is increased approx. 100ns.

Note 3: Set $\overline{SS}$ function after setting $\overline{CTS}$/$\overline{RTS}$ disable bit (bit 4 of UARTi transfer/receive control register 0) to "1".

Note 4: Nothing but "0" may be written.

Note 5: Set CLKi and TxDi both for output using the CLKi and TxDi function select register A. Set the RxDi function select register A for input/output port and the port direction register to "0".

Note 6: Set STxDi for output using the STxDi function select registers A and B. Set the CLKi and SRxDi function select register A for input/output port and the port direction register to "0".

Figure 1.16.12. Serial I/O-related registers (8)

(1) Clock synchronous serial I/O mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Tables 1.17.1 and 1.17.2 list the specifications of the clock synchronous serial I/O mode. Figure 1.17.1 shows the UARTi transmit/receive mode register.

Table 1.17.1. Specifications of clock synchronous serial I/O mode (1)

Item	Specification
Transfer data format	• Transfer data length: 8 bits
Transfer clock	• When internal clock is selected (bit 3 at addresses 0360_{16}, 0368_{16}, 0338_{16}, 0328_{16}, $02F8_{16}$ = “0”) : fi/ 2(n+1) (Note) fi = f1, f8, f32 – CLK is selected by the corresponding port function select register, peripheral function select register and peripheral subfunction select register. • When external clock is selected (bit 3 at addresses 0360_{16}, 0368_{16}, 0338_{16} , 0328_{16}, $02F8_{16}$= “1”) : Input from CLKi pin – Set the corresponding function select register A to I/O port
Transmission/reception control	• $\overline{\text{CTS}}$ function/$\overline{\text{RTS}}$ function/$\overline{\text{CTS}}$, $\overline{\text{RTS}}$ function chosen to be invalid
Transmission start condition	• To start transmission, the following requirements must be met: – Transmit enable bit (bit 0 at addresses 0365_{16}, $036D_{16}$, $033D_{16}$, $032D_{16}$, $02FD_{16}$) = “1” – Transmit buffer empty flag (bit 1 at addresses 0365_{16}, $036D_{16}$, $033D_{16}$, $032D_{16}$, $02FD_{16}$) = “0” – When $\overline{\text{CTS}}$ function selected, $\overline{\text{CTS}}$ input level = “L” – TxD output selected by the corresponding function select register A, B and C. • Furthermore, if external clock is selected, the following requirements must also be met: – CLKi polarity select bit (bit 6 at addresses 0364_{16}, $036C_{16}$, $033C_{16}$, $032C_{16}$, $02FC_{16}$) = “0”: CLKi input level = “H” – CLKi polarity select bit (bit 6 at addresses 0364_{16}, $036C_{16}$, $033C_{16}$, $032C_{16}$, $02FC_{16}$) = “1”: CLKi input level = “L”
Reception start condition	• To start reception, the following requirements must be met: – Receive enable bit (bit 2 at addresses 0365_{16}, $036D_{16}$, $033D_{16}$, $032D_{16}$, $02FD_{16}$) = “1” – Transmit enable bit (bit 0 at addresses 0365_{16}, $036D_{16}$, $033D_{16}$, $032D_{16}$, $02FD_{16}$) = “1” – Transmit buffer empty flag (bit 1 at addresses 0365_{16}, $036D_{16}$, $033D_{16}$, $032D_{16}$, $02FD_{16}$) = “0” • Furthermore, if external clock is selected, the following requirements must also be met: – CLKi polarity select bit (bit 6 at addresses 0364_{16}, $036C_{16}$, $033C_{16}$, $032C_{16}$, $02FC_{16}$) = “0”: CLKi input level = “H” – CLKi polarity select bit (bit 6 at addresses 0364_{16}, $036C_{16}$, $033C_{16}$, $032C_{16}$, $02FC_{16}$) = “1”: CLKi input level = “L”
Interrupt request generation timing	• When transmitting – Transmit interrupt cause select bit (bits 0, 1 at address 0370_{16}, bit 4 at address $033D_{16}$, $032D_{16}$, $02FD_{16}$) = “0”: Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed – Transmit interrupt cause select bit (bits 0, 1 at address 0370_{16}, bit 4 at address $033D_{16}$, $032D_{16}$, $02FD_{16}$) = “1”: Interrupts requested when data transmission from UARTi transfer register is completed • When receiving – Interrupts requested when data transfer from UARTi receive register to UARTi receive buffer register is completed

Note : “n” denotes the value 00_{16} to FF_{16} that is set to the UART bit rate generator.

Table 1.17.2. Specifications of clock synchronous serial I/O mode (2)

Item	Specification
Error detection	• Overrun error (Note 1) This error occurs when the next data is ready before contents of UARTi receive buffer register are read out
Select function	• CLK polarity selection Whether transmit data is output/input at the rising edge or falling edge of the transfer clock can be selected • LSB first/MSB first selection Whether transmission/reception begins with bit 0 or bit 7 can be selected • Continuous receive mode selection Reception is enabled simultaneously by a read from the receive buffer register • Transfer clock output from multiple pins selection (UART1) (Note 2) UART1 transfer clock can be chosen by software to be output from one of the two pins set • Separate $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ pins (UART0) (Note 2) UART0 $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ pins each can be assigned to separate pins • Switching serial data logic (UART2 to UART4) Whether to reverse data in writing to the transmission buffer register or reading the reception buffer register can be selected. • TxD, RxD I/O polarity reverse (UART2 to UART4) This function is reversing TxD port output and RxD port input. All I/O data level is reversed.

Note 1: If an overrun error occurs, the UARTi receive buffer will have the next data written in. Note also that the UARTi receive interrupt request bit will not change.

Note 2: The transfer clock output from multiple pins and the separate $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ pins functions cannot be selected simultaneously.

UARTi transmit/receive mode registers

b7	b6	b5	b4	b3	b2	b1	b0
0					0	0	1

Symbol	Address	When reset
UiMR(i=0,1)	0360_{16}, 0368_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 0 0 1 : Clock synchronous serial I/O mode	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal/external clock select bit	0 : Internal clock (Note 1) 1 : External clock (Note 2)	○	○
STPS	Invalid in clock synchronous serial I/O mode		○	○
PRY			○	○
PRYE			○	○
SLEP	0 (Set to “0” in clock synchronous serial I/O mode)		○	○

Note 1: Select CLK output by the corresponding function select registers A, B and C.
Note 2: Set the corresponding function select register A to the I/O port.

UART2 transmit/receive mode register

b7	b6	b5	b4	b3	b2	b1	b0
0					0	0	1

Symbol	Address	When reset
UiMR (i=2 to 4)	0338_{16}, 0328_{16}, $02F8_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 0 0 1 : Clock synchronous serial I/O mode	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal/external clock select bit	0 : Internal clock (Note 2) 1 : External clock (Note 3)	○	○
STPS	Invalid in clock synchronous serial I/O mode		○	○
PRY			○	○
PRYE			○	○
IOPOL	TxD, RxD I/O polarity reverse bit (Note 1)	0 : No reverse 1 : Reverse	○	○

Note 1: Usually set to “0”.
Note 2: Select CLK output by the corresponding function select registers A, B and C.
Note 3: Set the corresponding function select register A to the I/O port.

Figure 1.17.1. UARTi transmit/receive mode register in clock synchronous serial I/O mode

Table 1.17.3 lists the functions of the input/output pins during clock synchronous serial I/O mode. This table shows the pin functions when the transfer clock output from multiple pins and the separate $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ pins functions are not selected. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs a “H”. (If the N-channel open drain is selected, this pin is in floating state.)

Table 1.17.3. Input/output pin functions in clock synchronous serial I/O mode

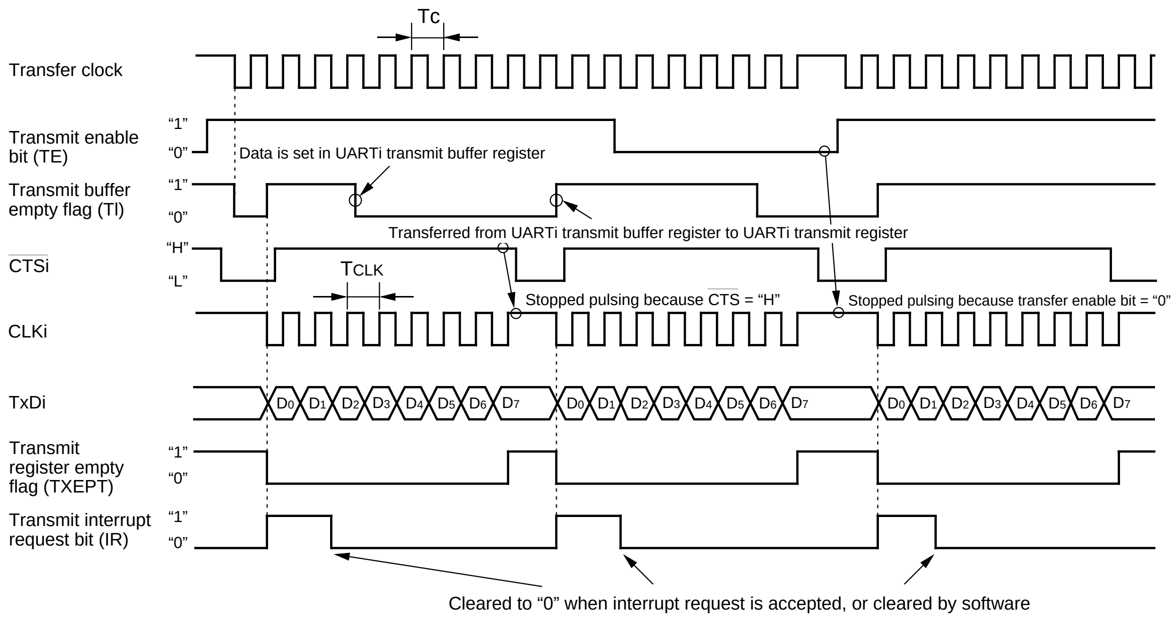
Pin name	Function	Method of selection
TxDi ($P6_3$, $P6_7$, $P7_0$, $P9_2$, $P9_6$)	Serial data output (Note 1)	(Outputs dummy data when performing reception only)
RxDi ($P6_2$, $P6_6$, $P7_1$, $P9_1$, $P9_7$)	Serial data input (Note 2)	Port $P6_2$, $P6_6$, $P7_1$, $P9_1$ and $P9_7$ direction register (bits 2 and 6 at address $03C2_{16}$, bit 1 at address $03C3_{16}$, bit 1 and 7 at address $03C7_{16}$)= “0” (Can be used as an input port when performing transmission only)
CLKi ($P6_1$, $P6_5$, $P7_2$, $P9_0$, $P9_5$)	Transfer clock output (Note 1)	Internal/external clock select bit (bit 3 at addresses 0360_{16}, 0368_{16}, 0338_{16}, 0328_{16}, $02F8_{16}$) = “0”
	Transfer clock input (Note 2)	Internal/external clock select bit (bit 3 at addresses 0360_{16}, 0368_{16}, 0338_{16}, 0328_{16}, $02F8_{16}$) = “1” Port $P6_1$, $P6_5$, $P7_2$, $P9_0$ and $P9_5$ direction register (bits 1 and 5 at address $03C2_{16}$, bit 2 at address $03C3_{16}$, bit 0 and 5 at address $03C7_{16}$) = “0”
$\overline{\text{CTSi}}$/$\overline{\text{RTSi}}$ ($P6_0$, $P6_4$, $P7_3$, $P9_3$, $P9_4$)	$\overline{\text{CTS}}$ input (Note 2)	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at addresses 0364_{16}, $036C_{16}$, $033C_{16}$, $032C_{16}$, $02FC_{16}$) =“0” CTS/RTS function select bit (bit 2 at addresses 0364_{16}, $036C_{16}$, $033C_{16}$, $032C_{16}$, $02FC_{16}$) = “0” Port $P6_0$, $P6_4$, $P7_3$, $P9_3$ and $P9_4$ direction register (bits 0 and 4 at address $03C2_{16}$, bit 3 at address $03C3_{16}$, bits 3 and 4 at address $03C7_{16}$) = “0”
	$\overline{\text{RTS}}$ output (Note 1)	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at addresses 0364_{16}, $036C_{16}$, $033C_{16}$, $032C_{16}$, $02FC_{16}$) = “0” CTS/RTS function select bit (bit 2 at addresses 0364_{16}, $036C_{16}$, $033C_{16}$, $032C_{16}$, $02FC_{16}$) = “1”
	Programmable I/O port (Note 2)	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at addresses 0364_{16}, $036C_{16}$, $033C_{16}$, $032C_{16}$, $02FC_{16}$) = “1”

(when transfer clock output from multiple pins and separate $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ pins functions are not selected)

Note 1: Select TxD output, CLK output and $\overline{\text{RTS}}$ output by the corresponding function select registers A, B and C.

Note 2: Select I/O port by the corresponding function select register A.

- Example of transmit timing (when internal clock is selected)

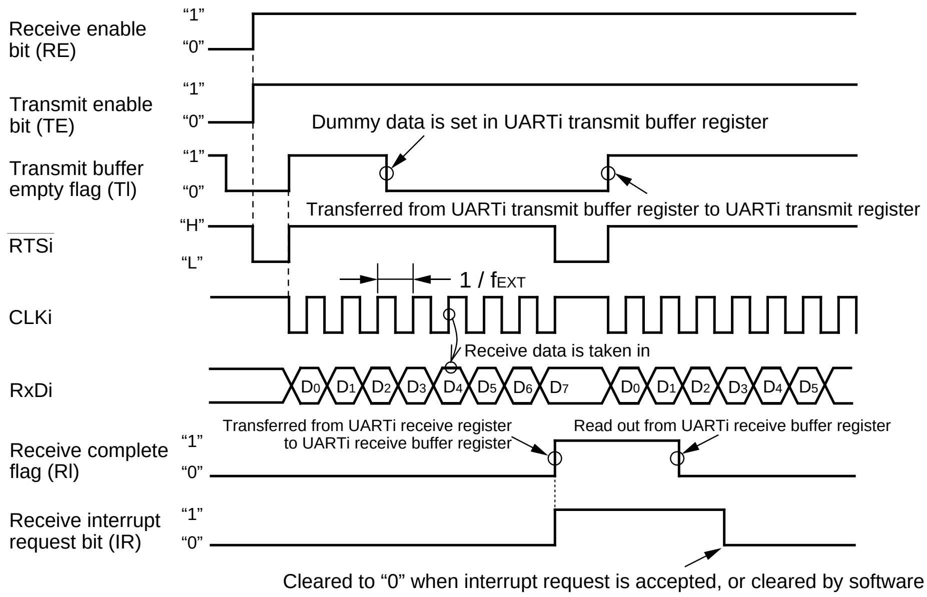

Tc
Transfer clock
Transmit enable bit (TE)
"1"
"0"
Data is set in UARTi transmit buffer register
Transmit buffer empty flag (TI)
"1"
"0"
Transferred from UARTi transmit buffer register to UARTi transmit register
CTSi
"H"
"L"
TCLK
Stopped pulsing because CTS = "H"
Stopped pulsing because transfer enable bit = "0"
CLKi
TxDi
D0 D1 D2 D3 D4 D5 D6 D7
Transmit register empty flag (TXEPT)
"1"
"0"
Transmit interrupt request bit (IR)
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.

The above timing applies to the following settings:
- Internal clock is selected.
- CTS function is selected.
- CLK polarity select bit = "0".
- Transmit interrupt cause select bit = "0".

Tc = TCLK = 2(n + 1) / fi
fi: frequency of BRGi count source (f1, f8, f32)
n: value set to BRGi

- Example of receive timing (when external clock is selected)


Receive enable bit (RE)
"1"
"0"
Transmit enable bit (TE)
"1"
"0"
Dummy data is set in UARTi transmit buffer register
Transmit buffer empty flag (TI)
"1"
"0"
Transferred from UARTi transmit buffer register to UARTi transmit register
RTSi
"H"
"L"
1 / fEXT
CLKi
Receive data is taken in
RxDi
D0 D1 D2 D3 D4 D5 D6 D7
Transferred from UARTi receive register to UARTi receive buffer register
Read out from UARTi receive buffer register
Receive complete flag (RI)
"1"
"0"
Receive interrupt request bit (IR)
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.

The above timing applies to the following settings:
- External clock is selected.
- RTS function is selected.
- CLK polarity select bit = "0".

fEXT: frequency of external clock

Meet the following conditions are met when the CLKi input before data reception = "H"
- Transmit enable bit → "1"
- Receive enable bit → "1"
- Dummy data write to UARTi transmit buffer register

Figure 1.17.2. Typical transmit/receive timings in clock synchronous serial I/O mode

(a) Polarity select function

As shown in Figure 1.17.3, the CLK polarity select bit (bit 6 at addresses 0364_{16}, $036C_{16}$, $033C_{16}$, $032C_{16}$, $02FC_{16}$) allows selection of the polarity of the transfer clock.

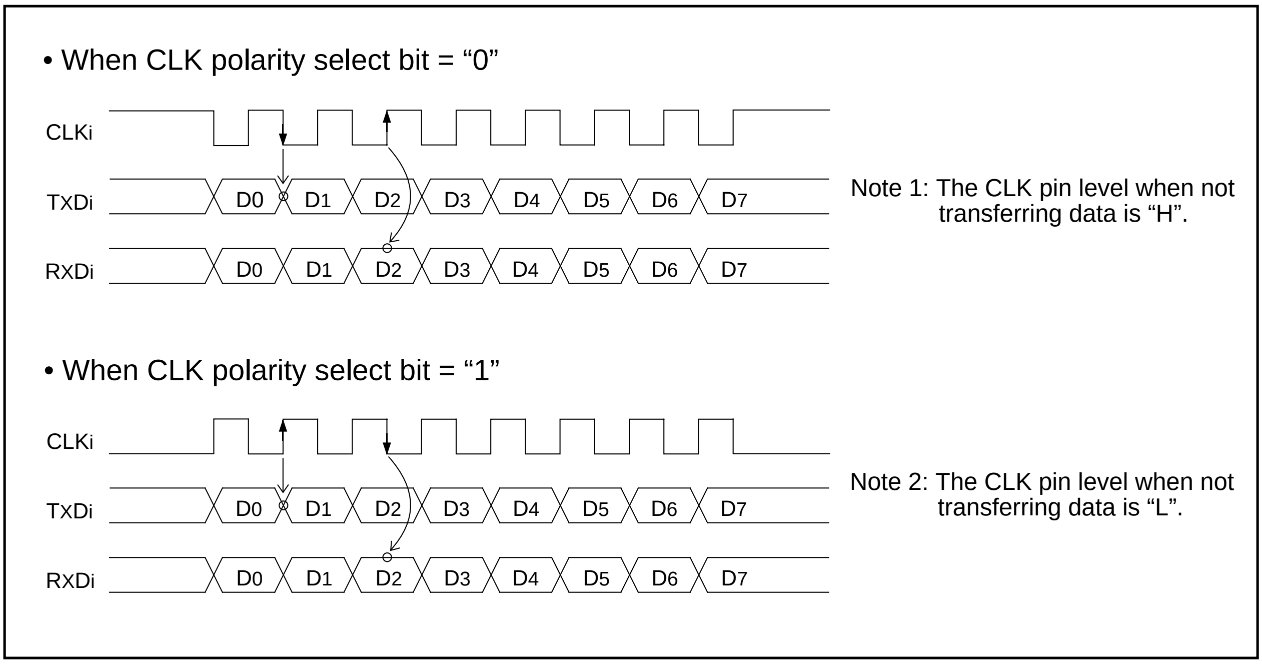

• When CLK polarity select bit = “0”
CLKi
TxDi
RxDi
D0 D1 D2 D3 D4 D5 D6 D7
Note 1: The CLK pin level when not transferring data is “H”.
• When CLK polarity select bit = “1”
CLKi
TxDi
RxDi
D0 D1 D2 D3 D4 D5 D6 D7
Note 2: The CLK pin level when not transferring data is “L”.

Figure 1.17.3. Polarity of transfer clock

(b) LSB first/MSB first select function

As shown in Figure 1.17.4, when the transfer format select bit (bit 7 at addresses 0364_{16}, $036C_{16}$, $033C_{16}$, $032C_{16}$, $02FC_{16}$) = “0”, the transfer format is “LSB first”; when the bit = “1”, the transfer format is “MSB first”.

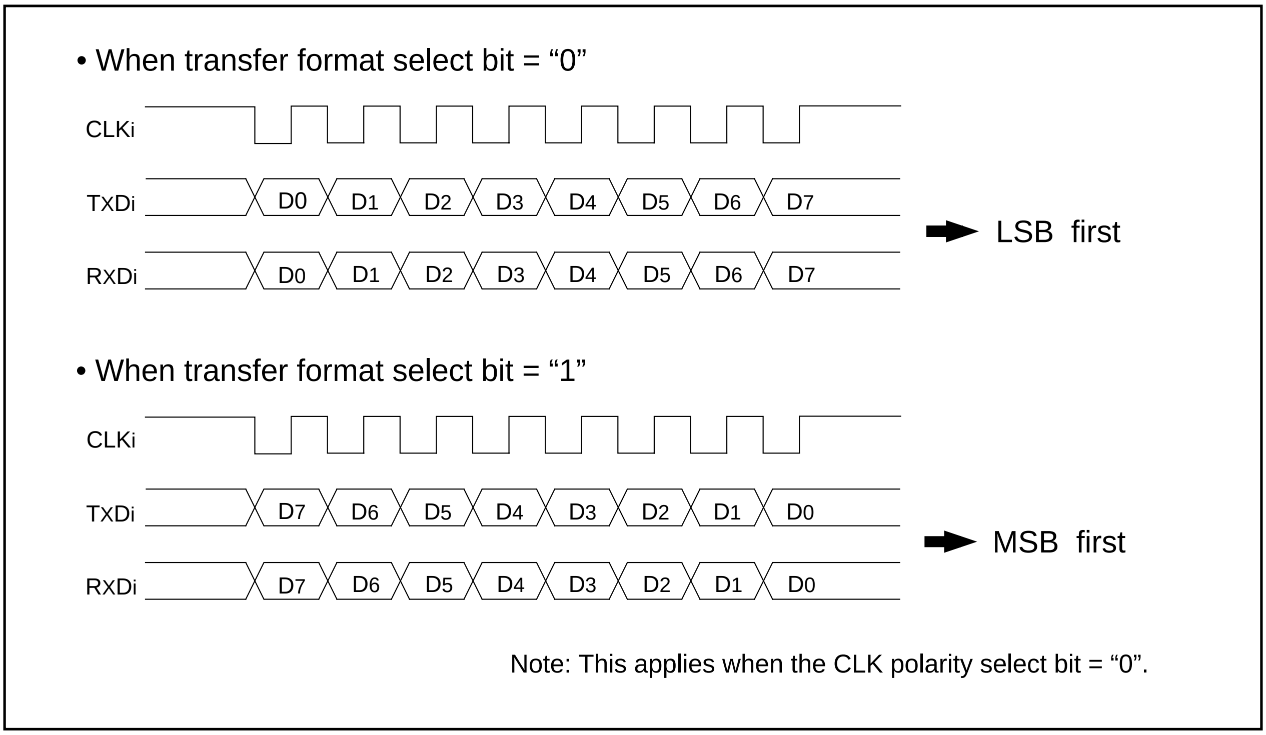
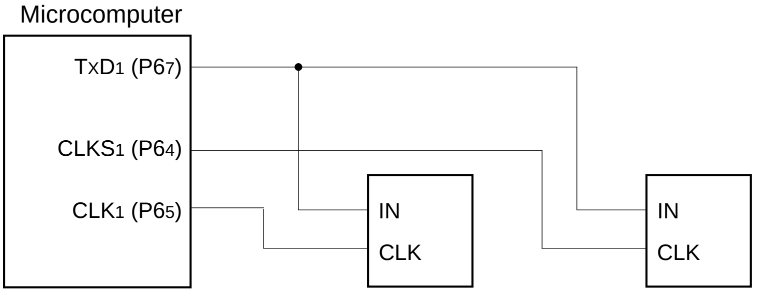

• When transfer format select bit = “0”
CLKi
TxDi
RxDi
D0 D1 D2 D3 D4 D5 D6 D7
LSB first
• When transfer format select bit = “1”
CLKi
TxDi
RxDi
D7 D6 D5 D4 D3 D2 D1 D0
MSB first
Note: This applies when the CLK polarity select bit = “0”.

Figure 1.17.4. Transfer format

(c) Transfer clock output from multiple pins function (UART1)

This function allows the setting two transfer clock output pins and choosing one of the two to output a clock by using the port function select register (bits of related to-P6$_4$ and P6$_5$). (See Figure 1.17.5.) The multiple pins function is valid only when the internal clock is selected for UART1. Note that when this function is selected, UART1 $\overline{\text{CTS}}/\overline{\text{RTS}}$ function cannot be used.



Note: This applies when the internal clock is selected and transmission is performed only in clock synchronous serial I/O mode.

Figure 1.17.5. The transfer clock output from the multiple pins function usage

(d) Continuous receive mode

If the continuous receive mode enable bit (bits 2 and 3 at address 0370$_{16}$, bit 5 at address 033D$_{16}$, 032D$_{16}$, 02FD$_{16}$) is set to “1”, the unit is placed in continuous receive mode. In this mode, when the receive buffer register is read out, the unit simultaneously goes to a receive enable state without having to set dummy data to the transmit buffer register back again.

(e) Separate $\overline{\text{CTS}}/\overline{\text{RTS}}$ pins function (UART0)

This function works the same way as in the clock asynchronous serial I/O (UART) mode. The method of setting and the input/output pin functions are both the same, so refer to select function in the next section, “(2) Clock asynchronous serial I/O (UART) mode.” Note that this function is invalid if the transfer clock output from the multiple pins function is selected.

(f) Serial data logic switch function (UART2 to UART4)

When the data logic select bit (bit6 at address 033D$_{16}$, 032D$_{16}$, 02FD$_{16}$) = “1”, and writing to transmit buffer register or reading from receive buffer register, data is reversed. Figure 1.17.6 shows the example of serial data logic switch timing.

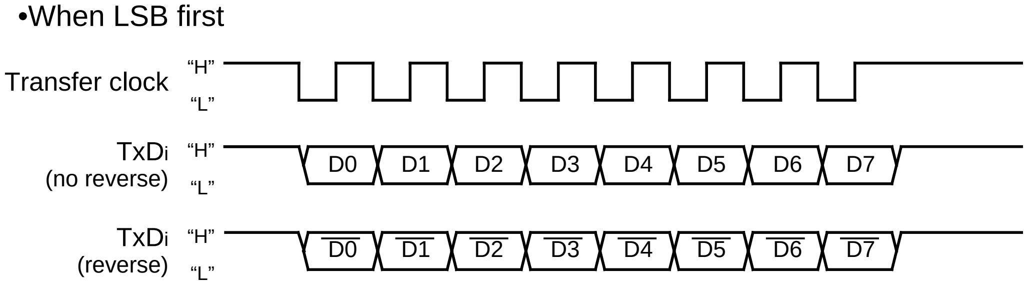

•When LSB first
Transfer clock
“H”
“L”
TxDi
(no reverse)
“H”
“L”
D0 D1 D2 D3 D4 D5 D6 D7
TxDi
(reverse)
“H”
“L”
$\overline{D0}$ $\overline{D1}$ $\overline{D2}$ $\overline{D3}$ $\overline{D4}$ $\overline{D5}$ $\overline{D6}$ $\overline{D7}$

Figure 1.17.6. Serial data logic switch timing

(2) Clock asynchronous serial I/O (UART) mode

The UART mode allows transmitting and receiving data after setting the desired transfer rate and transfer data format. Tables 1.18.1 and 1.18.2 list the specifications of the UART mode. Figure 1.18.1 shows the UARTi transmit/receive mode register.

Table 1.18.1. Specifications of UART Mode (1)

Item	Specification
Transfer data format	• Character bit (transfer data): 7 bits, 8 bits, or 9 bits as selected • Start bit: 1 bit • Parity bit: Odd, even, or nothing as selected • Stop bit: 1 bit or 2 bits as selected
Transfer clock	• When internal clock is selected (bit 3 at addresses 0360$_{16}$, 0368$_{16}$, 0338$_{16}$, 0328$_{16}$, 02F8$_{16}$ = "0") : fi/16(n+1) (Note 1) fi = f_1, f_8, f_{32} • When external clock is selected (bit 3 at addresses 0360$_{16}$, 0368$_{16}$, 0338$_{16}$, 0328$_{16}$, 02F8$_{16}$ ="1") : f_{EXT}/16(n+1)(Note 1) (Note 2)
Transmission/reception control	• $\overline{CTS}$ function/$\overline{RTS}$ function/$\overline{CTS}$, $\overline{RTS}$ function chosen to be invalid
Transmission start condition	• To start transmission, the following requirements must be met: - Transmit enable bit (bit 0 at addresses 0365$_{16}$, 036D$_{16}$, 033D$_{16}$, 032D$_{16}$, 02FD$_{16}$) = "1" - Transmit buffer empty flag (bit 1 at addresses 0365$_{16}$, 036D$_{16}$, 033D$_{16}$, 032D$_{16}$, 02FD$_{16}$) = "0" - When $\overline{CTS}$ function selected, $\overline{CTS}$ input level = "L" - TxD output is selected by the corresponding function select register A, B and C.
Reception start condition	• To start reception, the following requirements must be met: - Receive enable bit (bit 2 at addresses 0365$_{16}$, 036D$_{16}$, 033D$_{16}$, 032D$_{16}$, 02FD$_{16}$) = "1" - Start bit detection
Interrupt request generation timing	• When transmitting - Transmit interrupt cause select bits (bits 0,1 at address 0370$_{16}$, bit 4 at address 033D$_{16}$, 032D$_{16}$, 02FD$_{16}$) = "0": Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed - Transmit interrupt cause select bits (bits 0, 1 at address 0370$_{16}$, bit 4 at address 033D$_{16}$, 032D$_{16}$, 02FD$_{16}$) = "1": Interrupts requested when data transmission from UARTi transfer register is completed • When receiving - Interrupts requested when data transfer from UARTi receive register to UARTi receive buffer register is completed

Note 1: 'n' denotes the value 00$_{16}$ to FF$_{16}$ that is set to the UARTi bit rate generator.

Note 2: f_{EXT} is input from the CLKi pin.

Table 1.18.2. Specifications of UART Mode (2)

Item	Specification
Error detection	• Overrun error (Note) This error occurs when the next data is ready before contents of UARTi receive buffer register are read out • Framing error This error occurs when the number of stop bits set is not detected • Parity error This error occurs when if parity is enabled, the number of 1’s in parity and character bits does not match the number of 1’s set • Error sum flag This flag is set (= 1) when any of the overrun, framing, and parity errors is encountered
Select function	• Separate $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ pins (UART0) UART0 $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ pins each can be assigned to separate pins • Sleep mode selection (UART0, UART1) This mode is used to transfer data to and from one of multiple slave microcomputers • Serial data logic switch (UART2 to UART4) This function is reversing logic value of transferring data. Start bit, parity bit and stop bit are not reversed. • TxD, RxD I/O polarity switch (UART2 to UART4) This function is reversing TxD port output and RxD port input. All I/O data level is reversed.

Note: If an overrun error occurs, the UARTi receive buffer will have the next data written in. Note also that the UARTi receive interrupt request bit will not change.

UARTi transmit / receive mode registers

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UiMR(i=0,1)	0360_{16}, 0368_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal / external clock select bit	0 : Internal clock 1 : External clock (Note)	○	○
STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd / even parity select bit	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	○	○
SLEP	Sleep select bit	0 : Sleep mode deselected 1 : Sleep mode selected	○	○

Note: Set the corresponding port function select register A to I/O port.

UARTi transmit / receive mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UiMR (i=2 to 4)	0338_{16}, 0328_{16}, $02F8_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
SMD0	Serial I/O mode select bit	b2 b1 b0 1 0 0 : Transfer data 7 bits long 1 0 1 : Transfer data 8 bits long 1 1 0 : Transfer data 9 bits long	○	○
SMD1			○	○
SMD2			○	○
CKDIR	Internal / external clock select bit	0 : Internal clock 1 : External clock (Note 2)	○	○
STPS	Stop bit length select bit	0 : One stop bit 1 : Two stop bits	○	○
PRY	Odd / even parity select bit	Valid when bit 6 = “1” 0 : Odd parity 1 : Even parity	○	○
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	○	○
IOPOL	TxD, RxD I/O polarity reverse bit (Note 1)	0 : No reverse 1 : Reverse	○	○

Note 1: Usually set to “0”.
Note 2: Set the corresponding port function select register A to I/O port.

Figure 1.18.1. UARTi transmit/receive mode register in UART mode

Table 1.18.3 lists the functions of the input/output pins during UART mode. This table shows the pin functions when the separate $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ pins function is not selected. Note that for a period from when the UARTi operation mode is selected to when transfer starts, the TxDi pin outputs a “H”. (If the N-channel open drain is selected, this pin is in floating state.)

Table 1.18.3. Input/output pin functions in UART mode

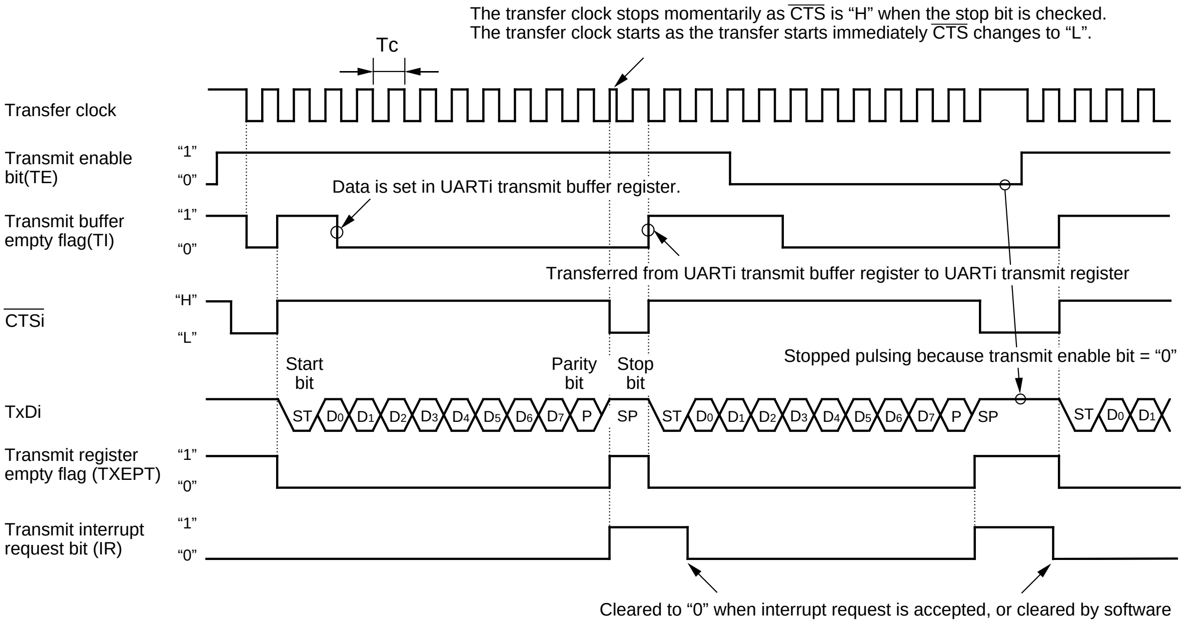
Pin name	Function	Method of selection
TxDi ($P6_3$, $P6_7$, $P7_0$, $P9_2$, $P9_6$)	Serial data output (Note 1)	
RxDi ($P6_2$, $P6_6$, $P7_1$, $P9_1$, $P9_7$)	Serial data input (Note 2)	Port $P6_2$, $P6_6$, $P7_1$, $P9_1$ and $P9_7$ direction register (bits 2 and 6 at address $03C2_{16}$, bit 1 at address $03C3_{16}$, bit 1 and 7 at address $03C7_{16}$)= “0” (Can be used as an input port when performing transmission only)
CLKi ($P6_1$, $P6_5$, $P7_2$, $P9_0$, $P9_5$)	Programmable I/O port (Note 2)	Internal/external clock select bit (bit 3 at addresses 0360_{16}, 0368_{16}, 0338_{16}, 0328_{16}, $02F8_{16}$) = “0”
	Transfer clock input (Note 2)	Internal/external clock select bit (bit 3 at addresses 0360_{16}, 0368_{16}, 0338_{16}, 0328_{16}, $02F8_{16}$) = “1” Port $P6_1$, $P6_5$, $P7_2$, $P9_0$ and $P9_5$ direction register (bits 1 and 5 at address $03C2_{16}$, bit 2 at address $03C3_{16}$, bits 0 and 5 at address $03C7_{16}$) = “0”
$\overline{\text{CTSi}}$/$\overline{\text{RTSi}}$ ($P6_0$, $P6_4$, $P7_3$, $P9_3$, $P9_4$)	$\overline{\text{CTS}}$ input (Note 2)	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at addresses 0364_{16}, $036C_{16}$, $033C_{16}$, $032C_{16}$, $02FC_{16}$) =“0” CTS/RTS function select bit (bit 2 at addresses 0364_{16}, $036C_{16}$, $033C_{16}$, $032C_{16}$, $02FC_{16}$) = “0” Port $P6_0$, $P6_4$, $P7_3$, $P9_3$ and $P9_4$ direction register (bits 0 and 4 at address $03C2_{16}$, bit 3 at address $03C3_{16}$, bits 3 and 4 at address $03C7_{16}$) = “0”
	$\overline{\text{RTS}}$ output (Note 1)	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at addresses 0364_{16}, $036C_{16}$, $033C_{16}$, $032C_{16}$, $02FC_{16}$) = “0” $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function select bit (bit 2 at addresses 0364_{16}, $036C_{16}$, $033C_{16}$, $032C_{16}$, $02FC_{16}$) = “1”
	Programmable I/O port (Note 2)	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit (bit 4 at addresses 0364_{16}, $036C_{16}$, $033C_{16}$, $032C_{16}$, $02FC_{16}$) = “1”

(When separate $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ pins function is not selected)

Note 1: Select TxD output, CLK output and $\overline{\text{RTS}}$ output by the corresponding function select registers A, B and C.

Note 2: Select I/O port by the corresponding function select register A.

- Example of transmit timing when transfer data is 8 bits long (parity enabled, one stop bit)

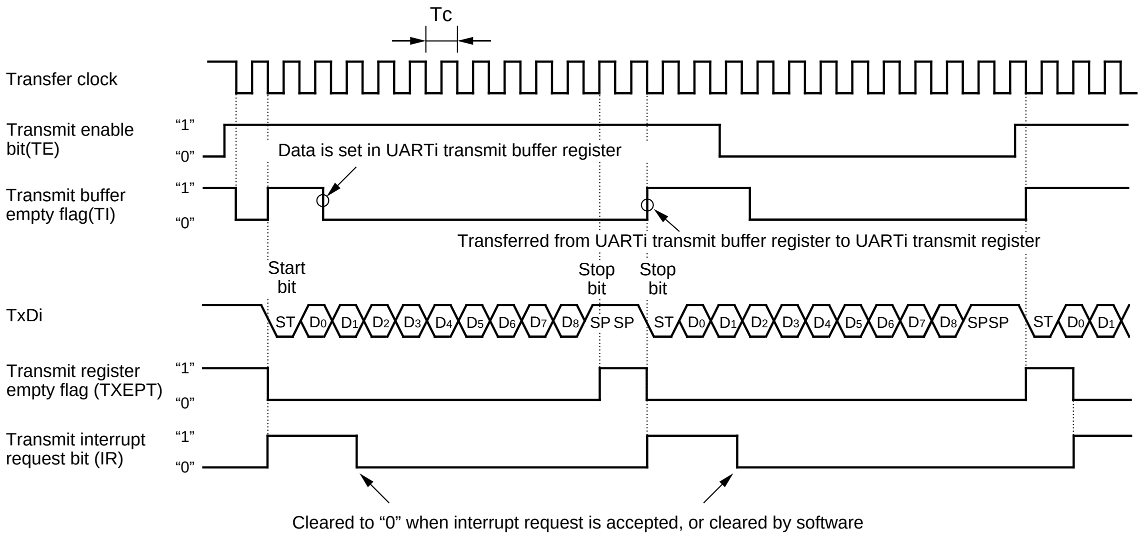

The transfer clock stops momentarily as CTS is "H" when the stop bit is checked.
The transfer clock starts as the transfer starts immediately CTS changes to "L".
Tc
Transfer clock
Transmit enable bit(TE)
"1"
"0"
Data is set in UARTi transmit buffer register.
Transmit buffer empty flag(TI)
Transferred from UARTi transmit buffer register to UARTi transmit register
CTSi
"H"
"L"
Stopped pulsing because transmit enable bit = "0"
Start bit
Parity bit
Stop bit
TxDi
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0 D1
Transmit register empty flag (TXEPT)
Transmit interrupt request bit (IR)
Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is enabled.
- One stop bit.
- CTS function is selected.
- Transmit interrupt cause select bit = "1".

Tc = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of BRGi count source (f1, f8, f32)
fEXT : frequency of BRGi count source (external clock)
n : value set to BRGi

- Example of transmit timing when transfer data is 9 bits long (parity disabled, two stop bits)


Tc
Transfer clock
Transmit enable bit(TE)
"1"
"0"
Data is set in UARTi transmit buffer register
Transmit buffer empty flag(TI)
Transferred from UARTi transmit buffer register to UARTi transmit register
Start bit
Stop bit
Stop bit
TxDi
ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0 D1
Transmit register empty flag (TXEPT)
Transmit interrupt request bit (IR)
Cleared to "0" when interrupt request is accepted, or cleared by software

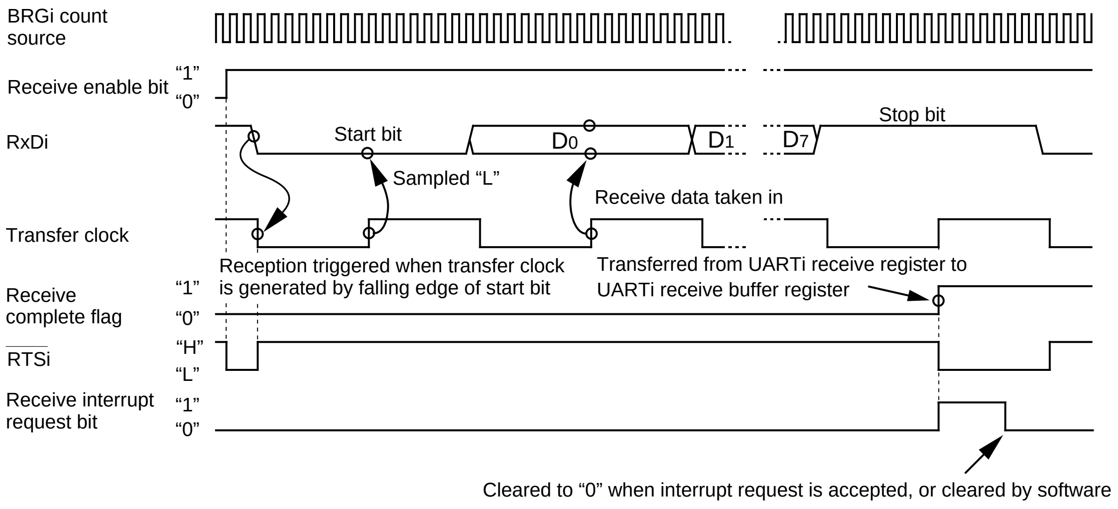
Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is disabled.
- Two stop bits.
- CTS function is disabled.
- Transmit interrupt cause select bit = "0".

Tc = 16 (n + 1) / fi or 16 (n + 1) / fEXT
fi : frequency of BRGi count source (f1, f8, f32)
fEXT : frequency of BRGi count source (external clock)
n : value set to BRGi

Figure 1.18.2. Typical transmit timings in UART mode

• Example of receive timing when transfer data is 8 bits long (parity disabled, one stop bit)


BRGi count source
Receive enable bit
"1"
"0"
RxDi
Start bit
D_0
D_1
D_7
Stop bit
Sampled "L"
Receive data taken in
Transfer clock
Reception triggered when transfer clock is generated by falling edge of start bit
Transferred from UARTi receive register to UARTi receive buffer register
Receive complete flag
"1"
"0"
$\overline{\text{RTSi}}$
"H"
"L"
Receive interrupt request bit
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

The above timing applies to the following settings :
- •Parity is disabled.
- •One stop bit.
- •RTS function is selected.

Figure 1.18.3. Typical receive timing in UART mode

(a) Separate $\overline{\text{CTS}}/\overline{\text{RTS}}$ pins function (UART0)

With the separate $\overline{\text{CTS}}/\overline{\text{RTS}}$ bit (bit 6 at address 0370_{16}) is set to "1", the unit outputs/inputs the $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ signals on different pins. (See Figure 1.18.4.) This function is valid only for UART0. Note that if this function is selected, the $\overline{\text{CTS}}/\overline{\text{RTS}}$ function for UART1 cannot be used.

Set both $\overline{\text{CTS}}/\overline{\text{RTS}}$ function select bit (bit 2 at address $036C_{16}$) of UART1and $\overline{\text{CTS}}/\overline{\text{RTS}}$ disable bit (bit 4 at address $036C_{16}$)of UART1 to "0" and set $P6_4$ to input port by the function select register.

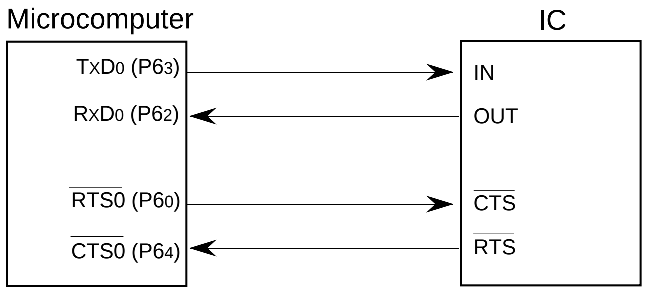


Figure 1.18.4. The separate $\overline{\text{CTS}}/\overline{\text{RTS}}$ pins function usage

(b) Sleep mode (UART0, UART1)

This mode is used to transfer data between specific microcomputers among multiple microcomputers connected using UARTi. The sleep mode is selected when the sleep select bit (bit 7 at addresses 0360_{16}, 0368_{16}) is set to "1" during reception. In this mode, the unit performs receive operation when the MSB of the received data = "1" and does not perform receive operation when the MSB = "0".

(c) Function for switching serial data logic (UART2 to UART4)

When the data logic select bit (bit 6 of address $033D_{16}$, $032D_{16}$, $02FD_{16}$) is assigned 1, data is inverted in writing to the transmission buffer register or reading the reception buffer register. Figure 1.18.5 shows the example of timing for switching serial data logic.

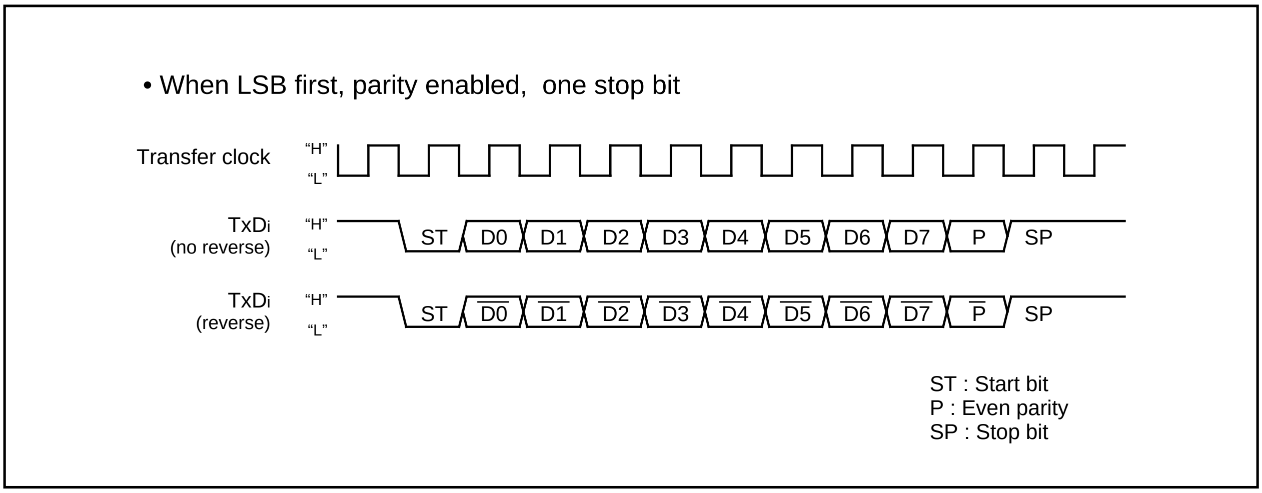

• When LSB first, parity enabled, one stop bit
Transfer clock
"H"
"L"
TxDi (no reverse)
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
TxDi (reverse)
ST $\overline{D0}$ $\overline{D1}$ $\overline{D2}$ $\overline{D3}$ $\overline{D4}$ $\overline{D5}$ $\overline{D6}$ $\overline{D7}$ $\overline{P}$ SP
ST : Start bit
P : Even parity
SP : Stop bit

Figure 1.18.5. Timing for switching serial data logic

(d) TxD, RxD I/O polarity reverse function (UART2 to UART4)

This function is to reverse TxD pin output and RxD pin input. The level of any data to be input or output (including the start bit, stop bit(s), and parity bit) is reversed. Set this function to "0" (not to reverse) for usual use.

(e) Bus collision detection function (UART2 to UART4)

This function is to sample the output level of the TxD pin and the input level of the RxD pin at the rising edge of the transfer clock; if their values are different, then an interrupt request occurs. Figure 1.18.6 shows the example of detection timing of a buss collision (in UART mode).

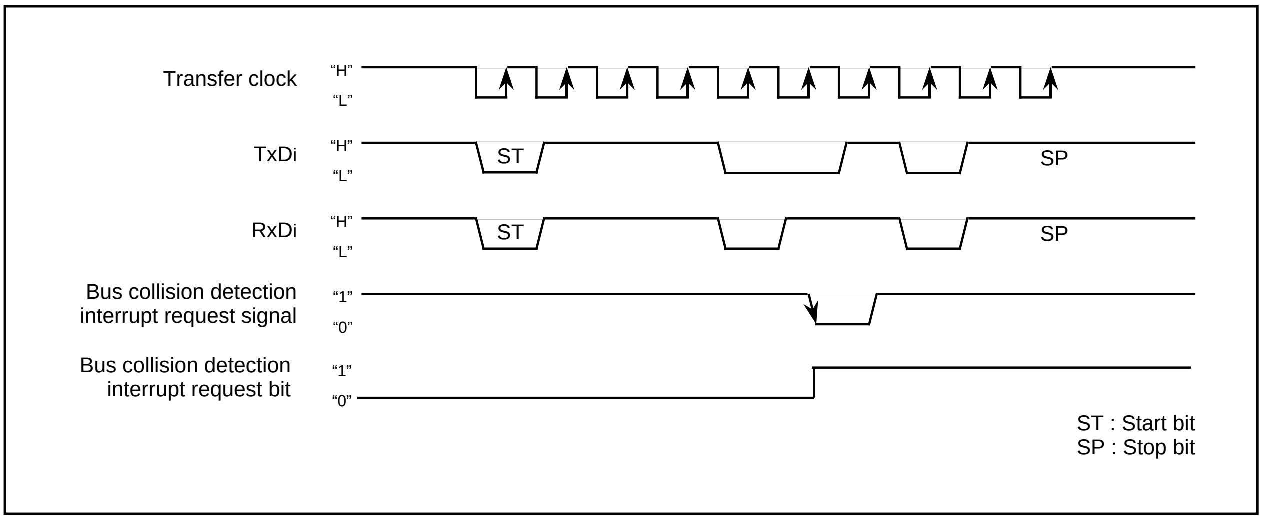

Transfer clock
"H"
"L"
TxDi
ST
SP
RxDi
ST
SP
Bus collision detection interrupt request signal
"1"
"0"
Bus collision detection interrupt request bit
"1"
"0"
ST : Start bit
SP : Stop bit

Figure 1.18.6. Detection timing of a bus collision (in UART mode)

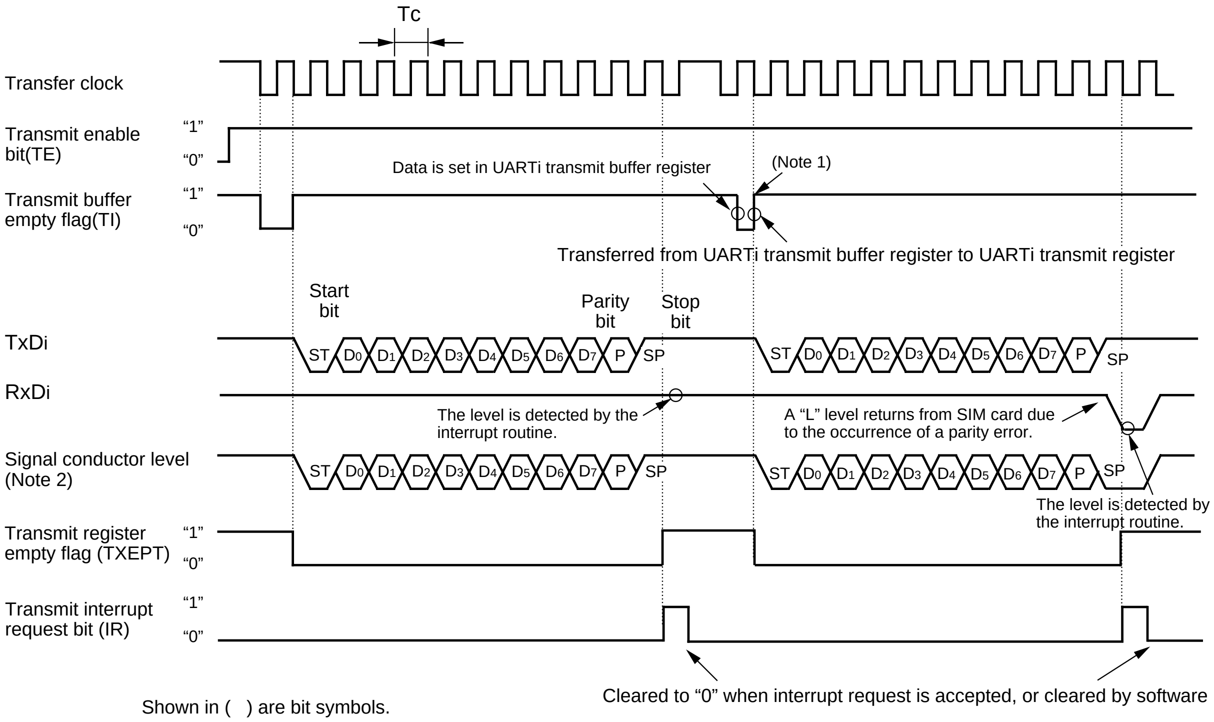
(3) Clock-asynchronous serial I/O mode (compliant with the SIM interface)

The SIM interface is used for connecting the microcomputer with a memory card I/C or the like; adding some extra settings in UART2 to UART4 clock-asynchronous serial I/O mode allows the user to effect this function. Table 1.19.1 shows the specifications of clock-asynchronous serial I/O mode (compliant with the SIM interface).

Table 1.19.1. Specifications of clock-asynchronous serial I/O mode (compliant with the SIM interface)

Item	Specification
Transfer data format	• Transfer data 8-bit UART mode (bit 2 to 0 of addresses 0338$_{16}$, 0328$_{16}$, 02F8$_{16}$ = “101$_2$”) • One stop bit (bit 4 of addresses 0338$_{16}$, 0328$_{16}$, 02F8$_{16}$ = “0”) • With the direct format chosen Set parity to “even” (bit 5 and 6 of addresses 0338$_{16}$, 0328$_{16}$, 02F8$_{16}$ = “1” and “1” respectively) Set data logic to “direct” (bit 6 of address 033D$_{16}$, 032D$_{16}$, 02FD$_{16}$ = “0”). Set transfer format to LSB (bit 7 of address 033C$_{16}$, 032C$_{16}$, 02FC$_{16}$ = “0”). • With the inverse format chosen Set parity to “odd” (bit 5 and 6 of addresses 0338$_{16}$, 0328$_{16}$, 02F8$_{16}$ = “0” and “1” respectively) Set data logic to “inverse” (bit 6 of address 033D$_{16}$, 032D$_{16}$, 02FD$_{16}$ = “1”) Set transfer format to MSB (bit 7 of address 033C$_{16}$, 032C$_{16}$, 02FC$_{16}$ = “1”)
Transfer clock	• With the internal clock chosen (bit 3 of addresses 0338$_{16}$, 0328$_{16}$, 02F8$_{16}$ = “0”) : fi / 16 (n + 1) (Note 1) : fi=f$_1$, f$_8$, f$_{32}$ • With an external clock chosen (bit 3 of addresses 0338$_{16}$, 0328$_{16}$, 02F8$_{16}$ = “1”) : f$_{EXT}$ / 16 (n+1) (Note 1) (Note 2)
Transmission / reception control	• Disable the $\overline{\text{CTS}}$ and $\overline{\text{RTS}}$ function (bit 4 of address 033C$_{16}$, 032C$_{16}$, 02FC$_{16}$ = “1”)
Other settings	• The sleep mode select function is not available for UART2 and UART3 • Set transmission interrupt factor to “transmission completed” (bit 4 of address 033D$_{16}$, 032D$_{16}$, 02FD$_{16}$ = “1”) • Set N-channel open drain output to TxD and RxD pins in UART3 and 4 (bit 5 of address 032C$_{16}$, 02FC$_{16}$ = “1”)
Transmission start condition	• To start transmission, the following requirements must be met: - Transmit enable bit (bit 0 of address 033D$_{16}$, 032D$_{16}$, 02FD$_{16}$) = “1” - Transmit buffer empty flag (bit 1 of address 033D$_{16}$, 032D$_{16}$, 02FD$_{16}$) = “0”
Reception start condition	• To start reception, the following requirements must be met: - Reception enable bit (bit 2 of address 033D$_{16}$, 032D$_{16}$, 02FD$_{16}$) = “1” - Detection of a start bit
Interrupt request generation timing	• When transmitting When data transmission from the UART2 to UART4 transfer register is completed (bit 4 of address 033D$_{16}$, 032D$_{16}$, 02FD$_{16}$ = “1”) • When receiving When data transfer from the UART2 to UART4 receive register to the UART2 to UART4 receive buffer register is completed
Error detection	• Overrun error (see the specifications of clock-asynchronous serial I/O) (Note 3) • Framing error (see the specifications of clock-asynchronous serial I/O) • Parity error (see the specifications of clock-asynchronous serial I/O) - On the reception side, an “L” level is output from the TxD$_i$ pin by use of the parity error signal output function (bit 7 of address 033D$_{16}$, 032D$_{16}$, 02FD$_{16}$ = “1”) when a parity error is detected - On the transmission side, a parity error is detected by the level of input to the RxD$_i$ pin when a transmission interrupt occurs • The error sum flag (see the specifications of clock-asynchronous serial I/O)

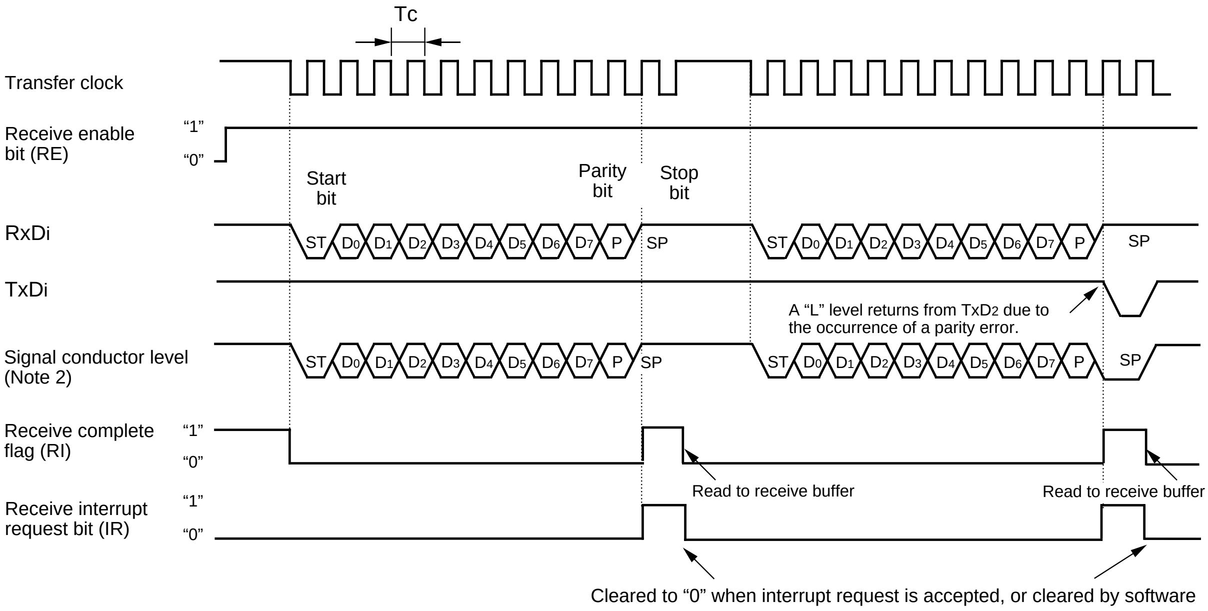
Note 1: ‘n’ denotes the value 00$_{16}$ to FF$_{16}$ that is set to the UARTi bit rate generator.
Note 2: f$_{EXT}$ is input from the CLKi pin.
Note 3: If an overrun error occurs, the UARTi receive buffer will have the next data written in. Note also that the UARTi receive interrupt request bit will not change.


Tc
Transfer clock
Transmit enable bit(TE)
"1"
"0"
Transmit buffer empty flag(TI)
"1"
"0"
Data is set in UARTi transmit buffer register
(Note 1)
Transferred from UARTi transmit buffer register to UARTi transmit register
Start bit
Parity bit
Stop bit
TxDi
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
RxDi
The level is detected by the interrupt routine.
A "L" level returns from SIM card due to the occurrence of a parity error.
Signal conductor level (Note 2)
The level is detected by the interrupt routine.
Transmit register empty flag (TXEPT)
"1"
"0"
Transmit interrupt request bit (IR)
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is enabled.
- One stop bit.
- Transmit interrupt cause select bit = "1".

Tc = 16 (n + 1) / fi or 16 (n + 1) / f_{EXT}
fi : frequency of BRGi count source (f_1, f_8, f_{32})
f_{EXT} : frequency of BRGi count source (external clock)
n : value set to BRGi


Tc
Transfer clock
Receive enable bit (RE)
"1"
"0"
Start bit
Parity bit
Stop bit
RxDi
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
TxDi
A "L" level returns from TxD2 due to the occurrence of a parity error.
Signal conductor level (Note 2)
Receive complete flag (RI)
"1"
"0"
Read to receive buffer
Read to receive buffer
Receive interrupt request bit (IR)
"1"
"0"
Cleared to "0" when interrupt request is accepted, or cleared by software

Shown in ( ) are bit symbols.

The above timing applies to the following settings :
- Parity is enabled.
- One stop bit.
- Transmit interrupt cause select bit = "0".

Tc = 16 (n + 1) / fi or 16 (n + 1) / f_{EXT}
fi : frequency of BRGi count source (f_1, f_8, f_{32})
f_{EXT} : frequency of BRGi count source (external clock)
n : value set to BRGi

Note 1: After writing to the transfer buffer at above timing, transmission starts at the timing of BRG overflow.
Note 2: Equal in waveform because TxD_i and RxD_i are connected.

Figure 1.19.1. Typical transmit/receive timing in UART mode (compliant with the SIM interface)

(a) Function for outputting a parity error signal

With the error signal output enable bit (bit 7 of address $033D_{16}$, $032D_{16}$, $02FD_{16}$) assigned “1”, you can output an “L” level from the TxD_i pin when a parity error is detected. In step with this function, the generation timing of a transmission completion interrupt changes to the detection timing of a parity error signal. Figure 1.19.2 shows the output timing of the parity error signal.

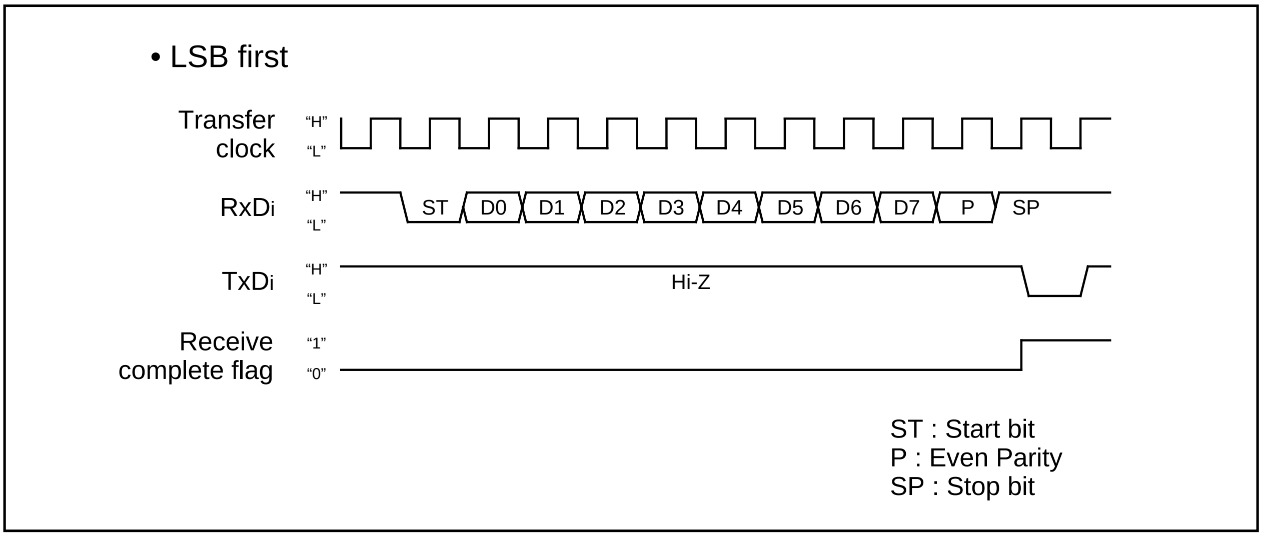


Figure 1.19.2. Output timing of the parity error signal

(b) Direct format/inverse format

Connecting the SIM card allows you to switch between direct format and inverse format. If you choose the direct format, D_0 data is output from TxDi. If you choose the inverse format, D_7 data is inverted and output from TxDi.

Figure 1.19.3 shows the SIM interface format.

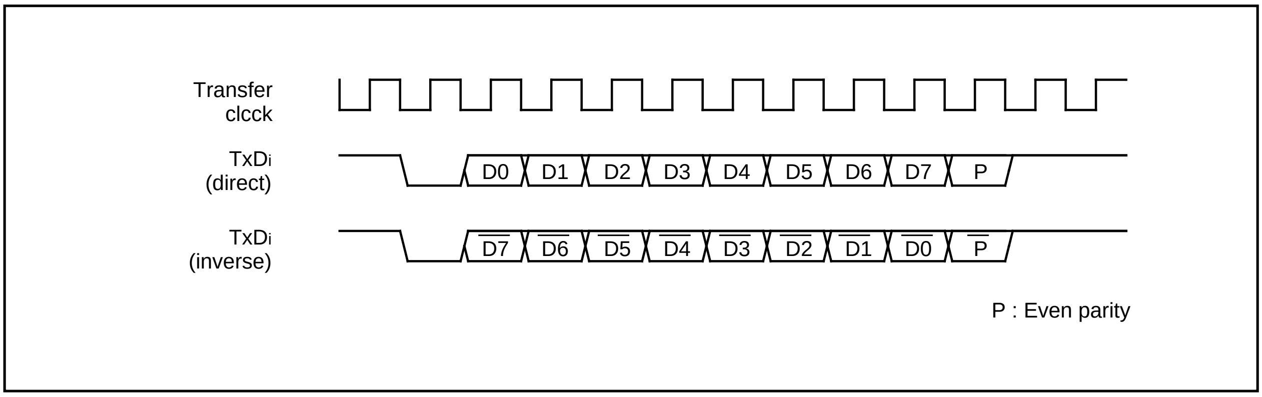
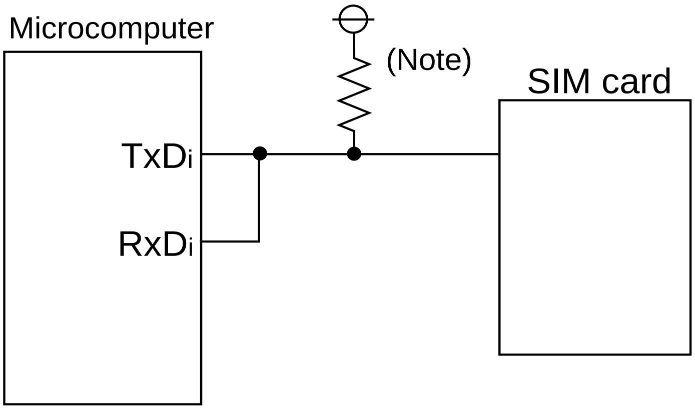


Figure 1.19.3. SIM interface format

Figure 1.19.4 shows the example of connecting the SIM interface. Connect TxDi and RxDi and apply pull-up.



Note :TxDi pin is N-channel open drain and needs a pull-up resistance.

Figure 1.19.4. Connecting the SIM interface

UARTi Special Mode Register (i = 2 to 4)

UART2 to UART4 operate the IIC bus interface (simple IIC bus) using the UARTi special mode register (addresses 0336_{16}, 0326_{16} and $02F6_{16}$ [i = 2 to 4]) and UARTi special mode register 2 (addresses 0336_{16}, 0326_{16} and $02F6_{16}$ [i = 2 to 4]). UART3 and UART4 add special functions using UARTi special mode resister 3 (addresses 0325_{16} and $02F5_{16}$ [i = 3 or 4]).

(1) IIC Bus Interface Mode

The I^2C bus interface mode is provided with UART2 to UART4.

Table 1.20.1 shows the construction of the UARTi special mode register and UARTi special mode register 2.

When the I^2C mode select bit (bit 0 in addresses 0337_{16}, 0327_{16} and $02F7_{16}$) is set to “1”, the I^2C bus (simple I^2C bus) interface circuit is enabled.

To use the I^2C bus, set the SCLi and the SDAi of both master and slave to output with the function select register. In UART3 and 4, set the data output select bit (bit 5 in address $032C_{16}$ and $02FC_{16}$) to N-channel open drain output.

Table 1.20.1 shows the relationship of the IIC mode select bit to control. To use the chip in the clock synchronized serial I/O mode or clock asynchronized serial I/O mode, always set this bit to “0”.

Table 1.20.1. Features in I^2C mode

	Function	Normal mode	I^2C mode (Note 1)
1	Factor of interrupt number 39 to 41 (Note 2)	Bus collision detection	Start condition detection or stop condition detection
2	Factor of interrupt number 33, 35, 37 (Note 2)	UARTi transmission	No acknowledgment detection (NACK)
3	Factor of interrupt number 34, 36, 38 (Note 2)	UARTi reception	Acknowledgment detection (ACK)
4	UARTi transmission output delay	Not delayed	Delayed
5	$P7_0$, $P9_2$, $P9_6$ at the time when UARTi is in use	TxD_i (output)	SDAi (input/output) (Note 3)
6	$P7_1$, $P9_1$, $P9_7$ at the time when UARTi is in use	RxD_i (input)	SCLi (input/output)
7	$P7_2$, $P9_0$, $P9_5$ at the time when UARTi is in use	CLKi	$P7_2$, $P9_0$, $P9_5$
8	DMA1 factor at the time when 1 1 0 1 is assigned to the DMA request factor selection bits	UARTi reception	Acknowledgment detection (ACK)
9	Noise filter width	15ns	50ns
10	Reading $P7_1$, $P9_1$, $P9_7$	Reading the terminal when 0 is assigned to the direction register	Reading the terminal regardless of the value of the direction register
11	Initial value of UARTi output	H level (when 0 is assigned to the CLK polarity select bit)	The value set in latch $P7_0$, $P9_2$, $P9_6$ when the port is selected (Note 3)

Note 1: Make the settings given below when I^2C mode is in use.
Set 0 1 0 in bits 2, 1, 0 of the UARTi transmission/reception mode register.
Disable the $\overline{RTS}/\overline{CTS}$ function. Choose the MSB First function.

Note 2: Follow the steps given below to switch from a factor to another.
1. Disable the interrupt of the corresponding number.
2. Switch from a factor to another.
3. Reset the interrupt request flag of the corresponding number.
4. Set an interrupt level of the corresponding number.

Note 3: Set an initial value of SDA transmission output when IIC mode (IIC mode select bit = "1") is valid and serial I/O is invalid.

UARTi special mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UiSMR (i=2 to 4)	0337_{16}, 0327_{16}, $02F7_{16}$	00_{16}

Bit symbol	Bit name	Function (During clock synchronous serial I/O mode)	Function (During UART mode)	R	W
IICM	I^{2}C mode select bit	0 : Normal mode 1 : I^{2}C mode	Set to “0”	○	○
ABC	Arbitration lost detecting flag control bit	0 : Update per bit 1 : Update per byte	Set to “0”	○	○
BBS	Bus busy flag	0 : STOP condition detected 1 : START condition detected	Set to “0”	○	○ (Note 1)
LSYN	SCLL sync output enable bit	0 : Disabled 1 : Enabled	Set to “0”	○	○
ABSCS	Bus collision detect sampling clock select bit	Set to “0”	0 : Rising edge of transfer clock (Note 2) 1 : Underflow signal of timer Ai	○	○
ACSE	Auto clear function select bit of transmit enable bit	Set to “0”	0 : No auto clear function 1 : Auto clear at occurrence of bus collision	○	○
SSS	Transmit start condition select bit	Set to “0”	0 : Ordinary 1 : Falling edge of RxDi	○	○
Nothing is assigned. When write, set "0". When read, the content is "0".				–	–

Note 1: Nothing but "0" may be written.
Note 2: UART2 : timer A0 underflow signal, UART3 : timer A3 underflow signal, UART4 : timer A4 underflow signal.

UARTi special mode register 2

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UiSMR2 (i=2 to 4)	0336_{16}, 0326_{16}, $02F6_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
IICM2	IIC mode select bit 2	Refer to Table 1.20.2.	○	○
CSC	Clock synchronous bit	0 : Disabled 1 : Enabled	○	○
SWC	SCL wait output bit	0 : Disabled 1 : Enabled	○	○
ALS	SDA output stop flag	0 : Disabled 1 : Enabled	○	○
STC	UARTi initialize bit	0 : Disabled 1 : Enabled	○	○
SWC2	SCL wait output bit 2	0 : UARTi clock 1 : 0 output	○	○
SDHI	SDA output inhibit bit	0 : Enabled 1 : Disabled (high impedance)	○	○
SHTC	Start/stop condition control bit	Set to "1" in selecting IIC mode.	○	○

Figure 1.20.1. UART2 special mode register

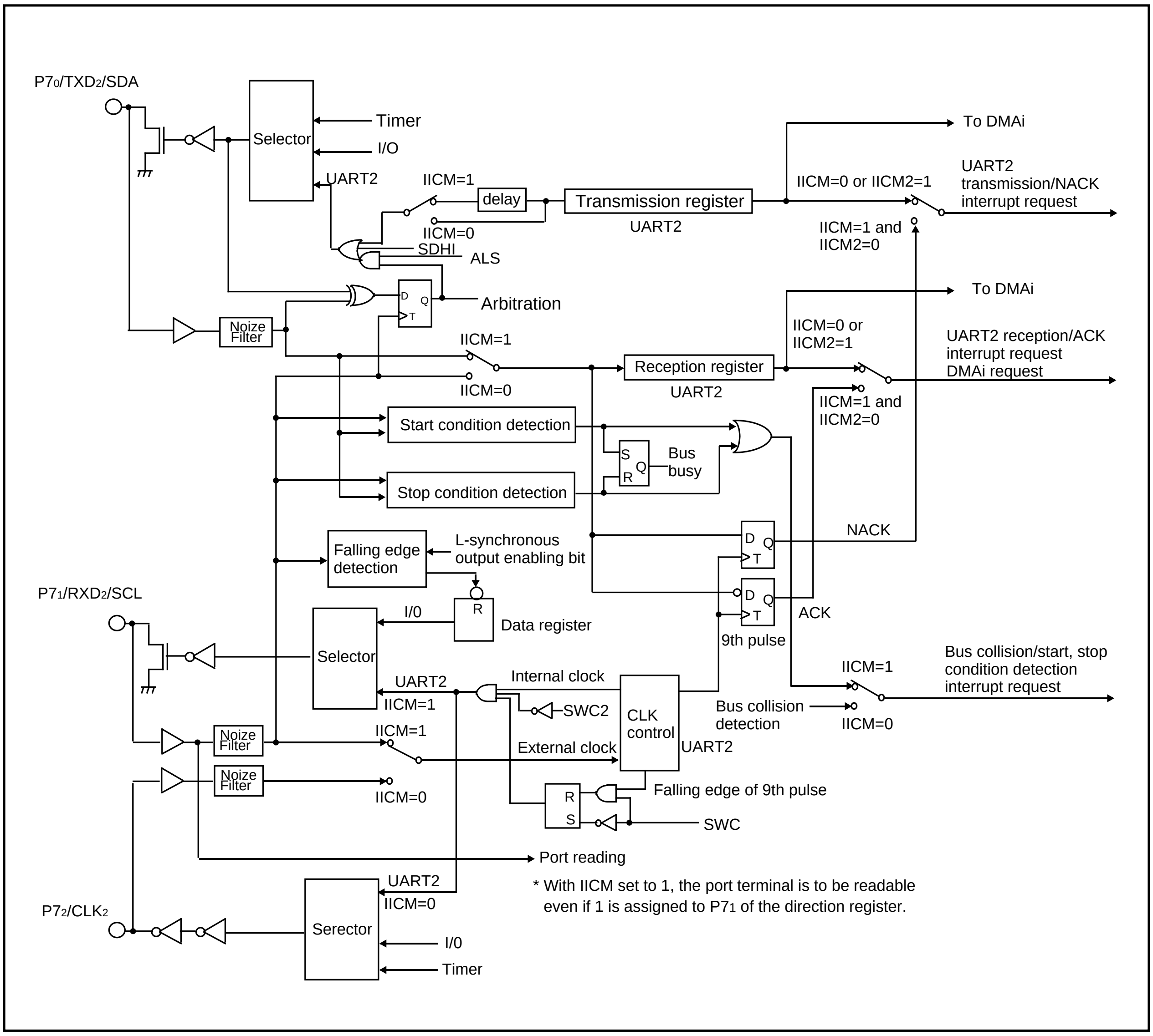


Figure 1.20.2. Functional block diagram for I^{2}C mode

Figure 1.20.2 is a block diagram of the IIC bus interface.
To explain the control bit of the IIC bus interface, UART2 is used as an example.

UART2 Special Mode Register (Address 0337$_{16}$)

Bit 0 is the IIC mode select bit. When set to “1”, ports P7$_0$, P7$_1$ and P7$_2$ operate respectively as the SDA2 data transmission-reception pin, SCL2 clock I/O pin and port P7$_2$. A delay circuit is added to SDA2 transmission output, therefore after SCL2 is sufficiently L level, SDA2 output changes. Port P7$_1$ (SCL2) is designed to read pin level regardless of the content of the port direction register. SDA2 transmission output is initially set to port P7$_0$ in this mode. Furthermore, interrupt factors for the bus collision detection interrupt, UART2 transmission interrupt and UART2 reception interrupt change respectively to the start/stop condition detection interrupts, acknowledge non-detection interrupt and acknowledge detection interrupt.

The start condition detection interrupt is generated when the fall at the SDA2 pin ($P7_0$) is detected while the SCL2 pin ($P7_1$) is in the H state. The stop condition detection interrupt is generated when the rise at the SDA2 pin ($P7_0$) is detected while the SCL2 pin ($P7_1$) is in the H state.
The acknowledge non-detection interrupt is generated when the H level at the SDA2 pin is detected at the 9th rise of the transmission clock.
The acknowledge detection interrupt is generated when the L level at the SDA2 pin is detected at the 9th rise of the transmission clock. Also, DMA transfer can be started when the acknowledge is detected if UART2 transmission is selected as the DMAi request factor.

Bit 2 is the bus busy flag. It is set to “1” when the start condition is detected, and reset to “0” when the stop condition is detected.

Bit 1 is the arbitration lost detection flag control bit. Arbitration detects a conflict between data transmitted at SCL2 rise and data at the SDA2 pin. This detection flag is allocated to bit 11 in UART2 transmission buffer register (address $033E_{16}$). It is set to “1” when a conflict is detected. With the arbitration lost detection flag control bit, it can be selected to update the flag in units of bits or bytes. When this bit is set to “1”, update is set to units of byte. If a conflict is then detected, the arbitration lost detection flag control bit will be set to “1” at the 9th rise of the clock. When updating in units of byte, always clear (“0” interrupt) the arbitration lost detection flag control bit after the 1st byte has been acknowledged but before the next byte starts transmitting.

Bit 3 is the SCL2 L synchronization output enable bit. When this bit is set to “1”, the $P7_1$ data register is set to “0” in sync with the L level at the SCL2 pin.

Bit 4 is the bus collision detection sampling clock select bit. The bus collision detection interrupt is generated when RxDi and TxDi level do not conflict with one another. When this bit is “0”, a conflict is detected in sync with the rise of the transfer clock. When this bit is “1”, detection is made when timer Ai (timer A0 with UART2, timer A3 with UART3 and timer A4 with UART4) underflows. Operation is shown in Figure 1.20.3.

Bit 5 is the transmission enable bit automatic clear select bit. By setting this bit to “1”, the transmission bit is automatically reset to “0” when the bus collision detection interrupt factor bit is “1” (when a conflict is detected).

Bit 6 is the transmission start condition select bit. By setting this bit to “1”, TxDi transmission starts in sync with the falling at the RxDi pin.

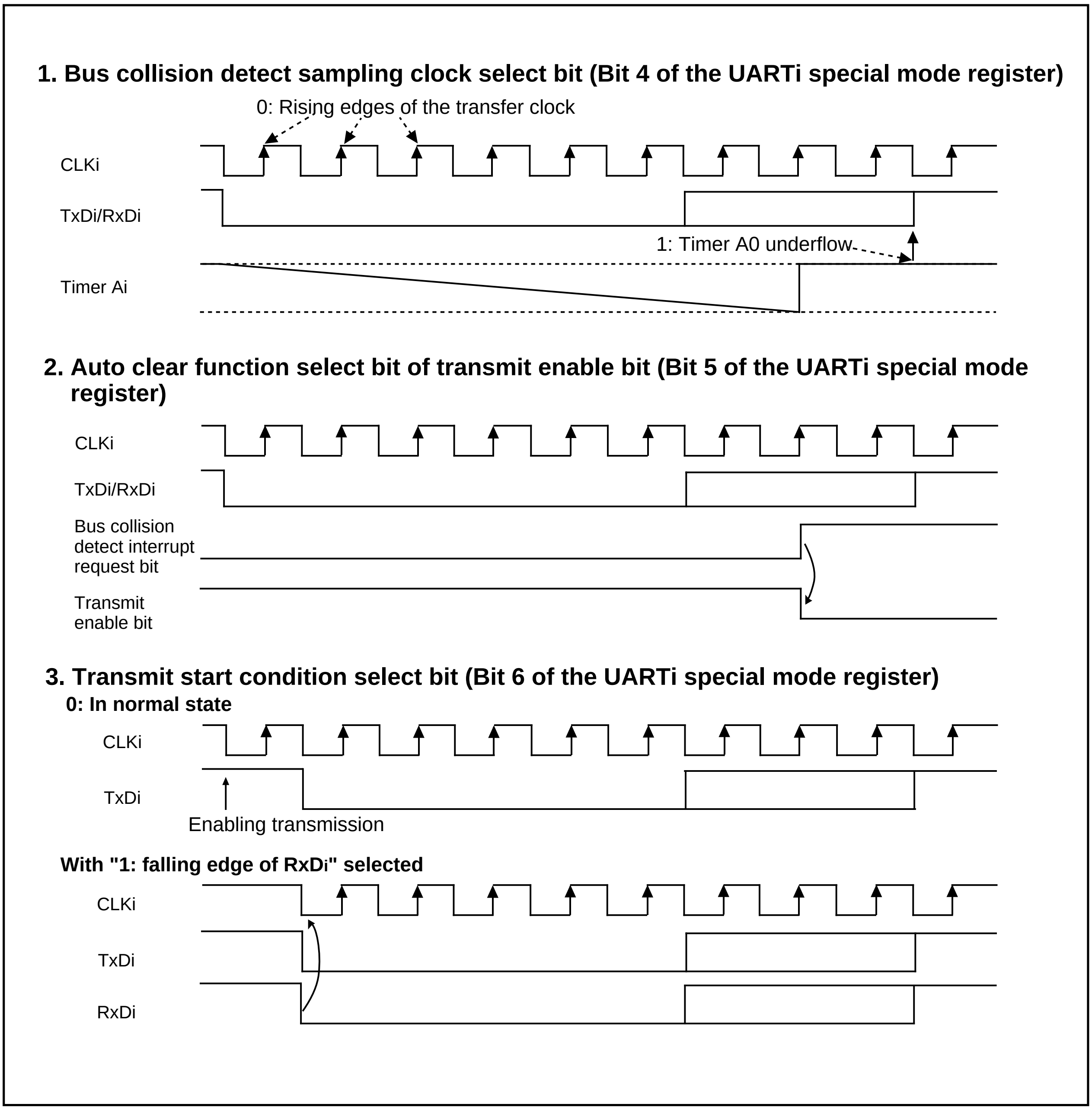

1. Bus collision detect sampling clock select bit (Bit 4 of the UARTi special mode register)
0: Rising edges of the transfer clock
CLKi
TxDi/RxDi
1: Timer A0 underflow
Timer Ai
2. Auto clear function select bit of transmit enable bit (Bit 5 of the UARTi special mode register)
CLKi
TxDi/RxDi
Bus collision detect interrupt request bit
Transmit enable bit
3. Transmit start condition select bit (Bit 6 of the UARTi special mode register)
0: In normal state
CLKi
TxDi
Enabling transmission
With "1: falling edge of RxDi" selected
CLKi
TxDi
RxDi

Figure 1.20.3. Some other functions added

UART2 Special Mode Register 2 (Address 0336_{16})

Bit 0 is the IIC mode select bit 2. Table 1.20.2 gives control changes by bit when the IIC mode select bit is “1”. Start and stop condition detection timing characteristics are shown in Figure 1.20.4. Always set bit 7 (start/stop condition control bit) to “1”.

Bit 1 is the clock synchronization bit. When this bit is set to “1”, if the rise edge is detected at pin SCL2 while the internal SCL is H level, the internal SCL is changed to L level, the baud rate generator value is reloaded and the L sector count starts. Also, while the SCL2 pin is L level, if the internal SCL changes from L level to H, baud rate generator stops counting. If the SCL2 pin is H level, counting restarts. Because of this function, the UART2 transmission-reception clock takes the AND condition for the internal SCL and SCL2 pin signals. This function operates from the clock half period before the 1st rise of the UART2 clock to the 9th rise. To use this function, select the internal clock as the transfer clock.

Bit 2 is the SCL wait output bit. When this bit is set to “1”, output from the SCL2 pin is fixed to L level at the clock’s 9th rise. When set to “0”, the L output lock is released.

Bit 3 is the SDA output stop bit. When this bit is set to “1”, an arbitration lost is generated. If the arbitration lost detection flag is “1”, the SDA2 pin simultaneously becomes high impedance.

Bit 4 is the UART2 initialize bit. While this bit is set to “1”, the following operations are performed when the start condition is detected.

1. The transmission shift register is initialized and the content of the transmission register is transmitted to the transmission shift register. As such, transmission starts with the 1st bit of the next input clock. However, the UART2 output value remains the same as when the start condition was detected, without changing from when the clock is input to when the 1st bit of data is output.
2. The reception shift register is initialized and reception starts with the 1st bit of the next input clock.
3. The SCL wait output bit is set to “1”. As such, the SCL2 pin becomes L level at the rise of the 9th bit of the clock.

When UART transmission-reception has been started using this function, the content of the transmission buffer available flag does not change. Also, to use this function, select an external clock as the transfer clock.

Bit 5 is SCL wait output bit 2. When this bit is set to “1” and serial I/O has been selected, an L level can be forcefully output from the SCL2 pin even during UART operation. When this bit is set to “0', the L output from the SCL2 pin is canceled and the UART2 clock is input and output.

Bit 6 is the SDA output disable bit. When this bit is set to “1”, the SDA2 pin is forcefully made high impedance. To overwrite this bit, do so at the rise of the UART2 transfer clock. The arbitration lost detection flag may be set.

Table 1.20.2. Functions changed by I²C mode select bit 2

Function	IICM2 = 0	IICM2 = 1
Interrupt no. 33, 35, 37 factor	Acknowrege not detect (NACK)	UART2 transfer (rising edge of)
Interrupt no. 34, 36, 38 factor	Acknowrege detect (ACK)	Acknowrege detect (ACK)
DMA factor	Acknowrege detect (ACK)	Acknowrege detect (ACK)
Data transfer timing from UARTi (i = 2 to 4) receive shift register to receive buffer	Rising edge of the last bit of receive clock	Rising edge of the last bit of receive clock
UARTi(i = 2 to 4) receive / ACK interrupt request generation timing	Rising edge of the last bit of receive clock	Rising edge of the last bit of receive clock

3 to 6 cycles < set up time (Note)
3 to 6 cycles < hold time (Note)

Note : Cycle number shows main clock input oscillation frequency f(XIN) cycle number.

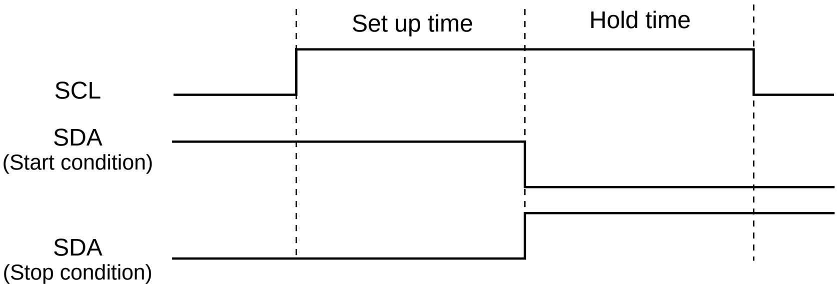


Figure 1.20.4. Start/stop condition detect timing characteristics

UART2 Special Mode Register 3 (Address 0335$_{16}$)

Bits 5 to 7 are the SDA2 digital delay setting bits. By setting these bits, it is possible to turn the SDA2 delay OFF or set the f(XIN) delay to 2 to 8 cycles.

(2) Serial Interface Special Function

UART 3 and UART4 can control communications on the serial bus using the $\overline{SSi}$ input pins (Figure 1.20.5). The master outputting the transfer clock transfers data to the slave inputting the transfer clock. In this case, in order to prevent a data collision on the bus, the master floats the output pin of other slaves/ masters using the $\overline{SSi}$ input pins. Figure 1.20.6 shows the structure of UARTi special mode register 3 (addresses 0325$_{16}$ and 02F5$_{16}$ [i = 3 or 4]) which controls this mode.
$\overline{SSi}$ input pins function between the master and slave are as follows.

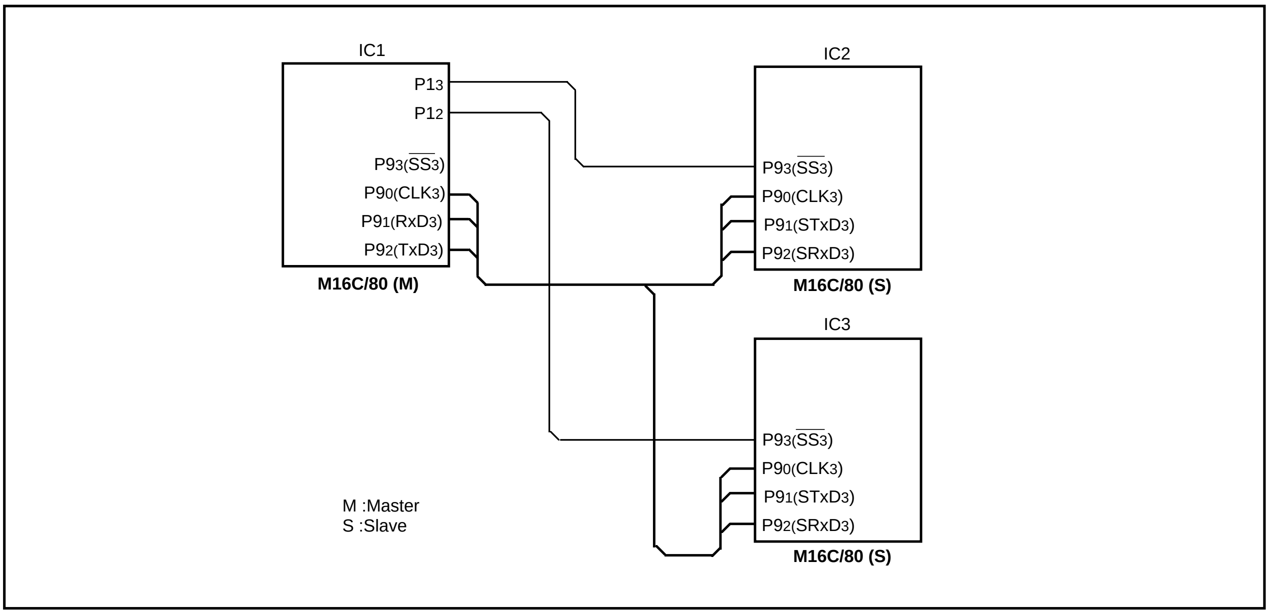


Figure 1.20.5 Serial bus communication control example using the SSi input pins

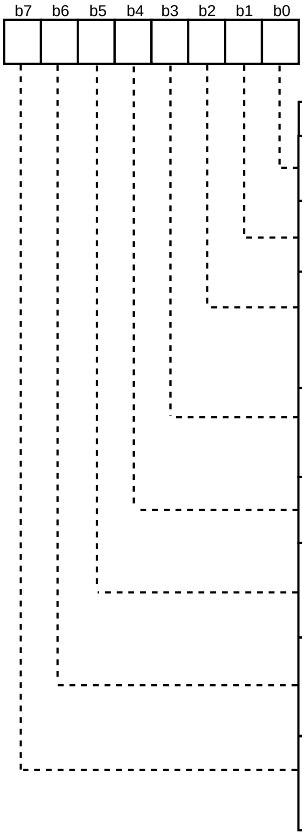
< Slave Mode (STxDi and SRxDi are selected, DINC = 1) >

When an H level signal is input to an $\overline{SSi}$ input pin, the STxDi and SRxDi pins both become high impedance, hence clock input is ignored. When an "L" level signal is input to an SSi input pin, clock input becomes effective and serial communications are enabled. (i = 3 or 4)

< Master Mode (TxDi and RxDi are selected, DINC = 0) >

The $\overline{SSi}$ input pins are used with a multiple master system. When an $\overline{SSi}$ input pin is H level, transmission has priority and serial communications are enabled. When an L signal is input to an $\overline{SSi}$ input pin, another master exists, and the TxDi, RxDi and CLKi pins all become high impedance. Moreover, the trouble error interrupt request bit becomes “1”. Communications do not stop even when a trouble error is generated during communications. To stop communications, set bits 0, 1 and 2 of the UARTi transmission-reception mode register (address 0328$_{16}$ and 02F8$_{16}$ [i = 3 or 4]) to “0”.
The trouble error interrupt is used by both the bus collision interrupt and start/stop condition detection interrupts, but the trouble error interrupt itself can be selected by setting bit 0 of UARTi special mode register 3 (address 0325$_{16}$ and 02F5$_{16}$ [i = 3 or 4]) to “1”.
When the trouble error flag is set to “0”, output is restored to the clock output and data output pins. In the master mode, if an $\overline{SSi}$ input pin is H level, “0” can be written for the trouble error flag. When an $\overline{SSi}$ input pin is L level, “0” cannot be written for the trouble error flag. In the slave mode, the “0” can be written for the trouble error flag regardless of the input to the $\overline{SSi}$ input pins.

UARTi special mode register 3 (i=3,4)



Symbol	Address	When reset
U3SMR3	0325_{16}	00000000_2
U4SMR3	$02F5_{16}$	00000000_2

Bit symbol	Bit name	Function	R	W
SSE	$\overline{SS}$ port function enable bit (Note 3)	0: $\overline{SS}$ function disable 1: $\overline{SS}$ function enable	○	○
CKPH	Clock phase set bit	0: Without clock delay 1: With clock delay	○	○
DINC	Serial input port set bit	0: Select TxDi and RxDi (master mode) (Note 5) 1: Select STxDi and SRxDi (slave mode) (Note 6)	○	○
NODC	Clock output select bit	0: CLKi is CMOS output 1: CLKi is N-channel open drain output	○	○
ERR	Fault error flag	0: Without fault error 1: With fault error	○	○ (Note 4)
DL0	SDAi(TxD$_2$) digital delay time set bit (Note 1,2)	b7 b6 b5 000 :Without delay 001 :2-cycle of 1/f(X_{IN}) 010 :3-cycle of 1/f(X_{IN}) 011 :4-cycle of 1/f(X_{IN}) 100 :5-cycle of 1/f(X_{IN}) 101 :6-cycle of 1/f(X_{IN}) 110 :7-cycle of 1/f(X_{IN}) 111 :8-cycle of 1/f(X_{IN})	○	○
DL1			○	○
DL2			○	○

Note 1: These bits are used for SDAi(TxDi) output digital delay when using UARTi for IIC interface. Otherwise, must set to "000".

Note 2: When external clock is selected, delay is increased approx. 100ns.

Note 3: Set $\overline{SS}$ function after setting $\overline{CTS}/\overline{RTS}$ disable bit (bit 4 of UARTi transfer/receive control register 0) to "1".

Note 4: Nothing but "0" may be written.

Note 5: Set CLKi and TxDi both for output using the CLKi and TxDi function select register A. Set the RxDi function select register A for input/output port and the port direction register to "0".

Note 6: Set STxDi for output using the STxDi function select registers A and B. Set the CLKi and SRxDi function select register A for input/output port and the port direction register to "0".

Figure 1.20.6. UARTi special mode register 3 (i=3,4)

Clock Phase Setting

With bit 1 of UARTi special mode register 3 (addresses 0325_{16} and $02F5_{16}$ [i = 3 or 4]) and bit 6 of UARTi transmission-reception control register 0 (addresses $032C_{16}$ and $02FC_{16}$ [i = 3 or 4]), four combinations of transfer clock phase and polarity can be selected.

Bit 6 of UARTi transmission-reception control register 0 (addresses $032C_{16}$ and $02FC_{16}$ [i = 3 or 4]) sets transfer clock polarity, whereas bit 1 of UARTi special mode register 3 (addresses 0325_{16} and $02F5_{16}$ [i = 3 or 4]) sets transfer clock phase.

Transfer clock phase and polarity must be the same between the master and slave involved in the transfer.

< Master (Internal Clock) (DINC = 0) >

Figure 1.20.7 shows the transmission and reception timing.

< Slave (External Clock) (DINC = 1) >

- With “0” for bit 1 (CKPH) of UARTi special mode register 3 (addresses 0325_{16} and $02F5_{16}$ [i = 3 or 4]), when an SSi input pin is H level, output data is high impedance. When an SSi input pin is L level, the serial transmission start condition is satisfied, though output is indeterminate. After that, serial transmission is synchronized with the clock. Figure 1.20.8 shows the timing.
- With “1” for bit 1 (CKPH) of UARTi special mode register 3 (addresses 0325_{16} and $02F5_{16}$ [i = 3 or 4]), when an SSi input pin is H level, output data is high impedance. When an SSi input pin is L level, the first data is output. After that, serial transmission is synchronized with the clock. Figure 1.20.9 shows the timing.

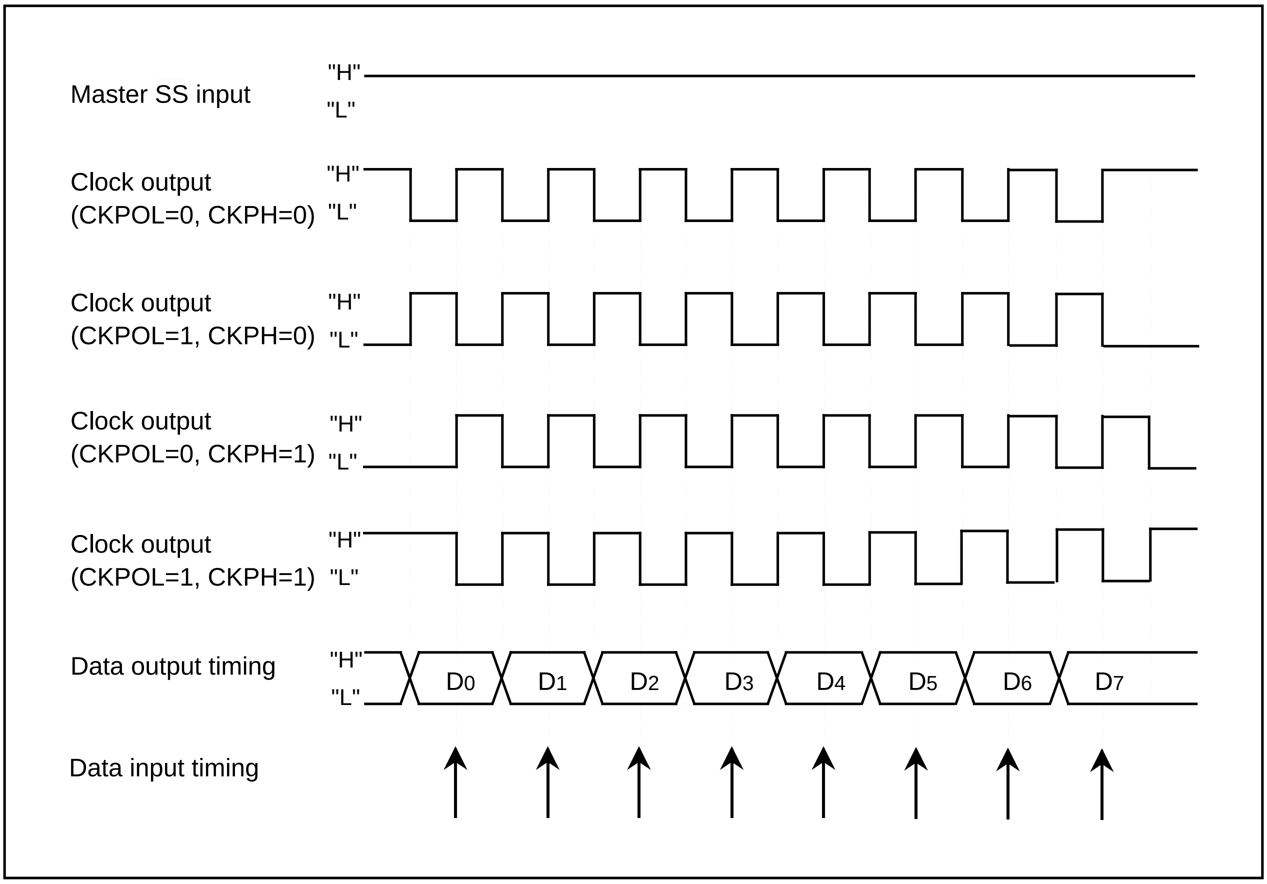


Figure 1.20.7. The transmission and reception timing in master mode (internal clock)

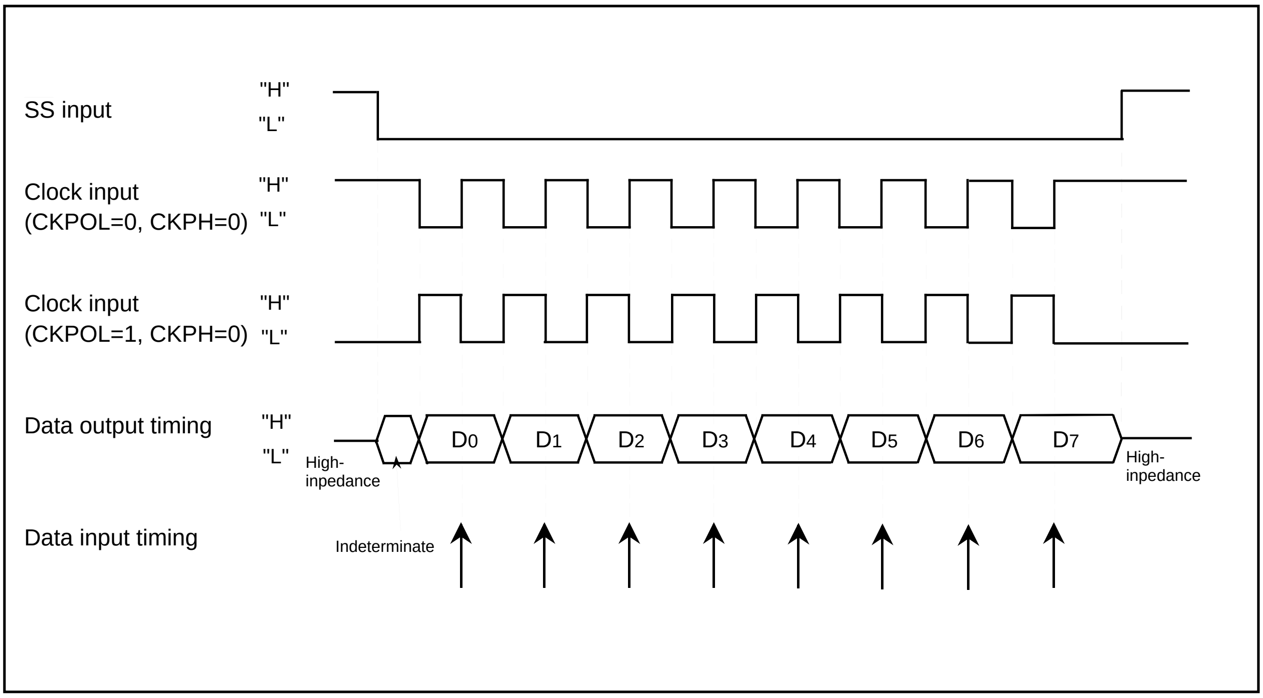


Figure 1.20.8. The transmission and reception timing (CKPH=0) in slave mode (external clock)

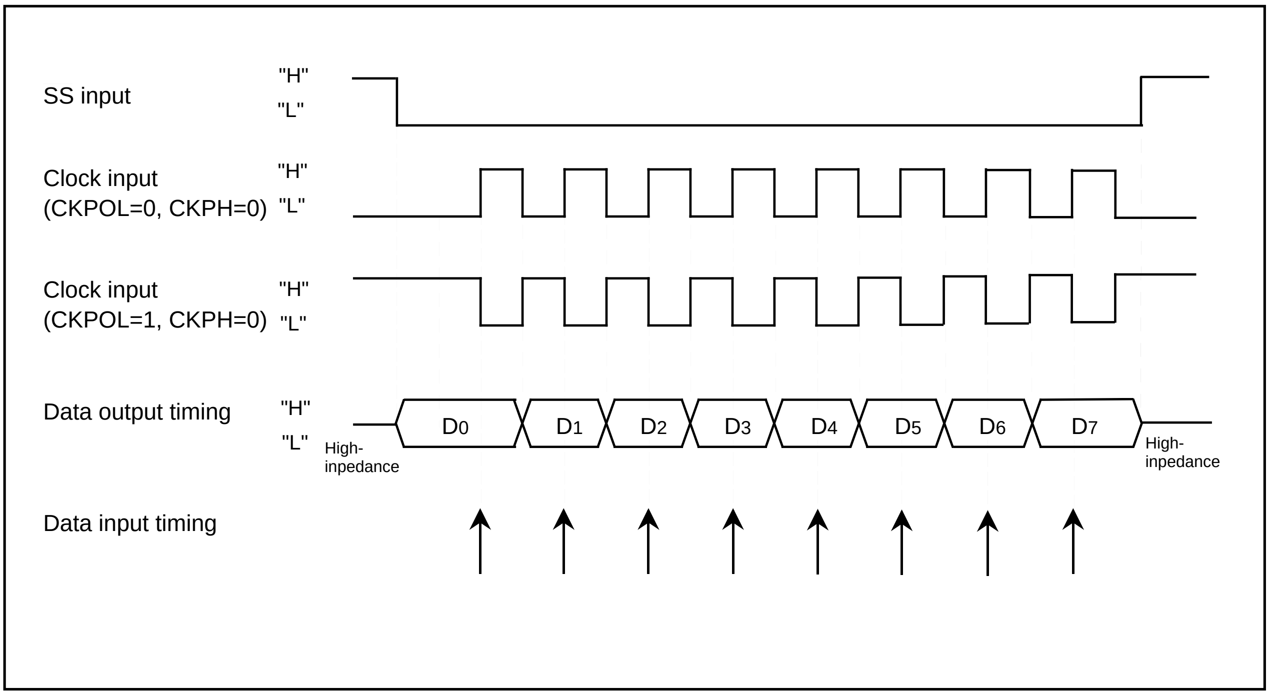


Figure 1.20.9. The transmission and reception timing (CKPH=1) in slave mode (external clock)

A-D Converter

The A-D converter consists of one 10-bit successive approximation A-D converter circuit with a capacitive coupling amplifier. Pins P10$_0$ to P10$_7$, P9$_5$, and P9$_6$ also function as the analog signal input pins. The direction registers of these pins for A-D conversion must therefore be set to input. The Vref connect bit (bit 5 at address 0397$_{16}$) can be used to isolate the resistance ladder of the A-D converter from the reference voltage input pin (VREF) when the A-D converter is not used. Doing so stops any current flowing into the resistance ladder from VREF, reducing the power dissipation. When using the A-D converter, start A-D conversion only after setting bit 5 of 0397$_{16}$ to connect VREF.

The result of A-D conversion is stored in the A-D registers of the selected pins. When set to 10-bit precision, the low 8 bits are stored in the even addresses and the high 2 bits in the odd addresses. When set to 8-bit precision, the low 8 bits are stored in the even addresses.

Table 1.21.1 shows the performance of the A-D converter. Figure 1.21.1 shows the block diagram of the A-D converter, and Figures 1.21.2 and 1.21.3 show the A-D converter-related registers.

Table 1.21.1. Performance of A-D converter

Item	Performance
Method of A-D conversion	Successive approximation (capacitive coupling amplifier)
Analog input voltage (Note 1)	0V to AV$_{CC}$ (V$_{CC}$)
Operating clock f$_{AD}$ (Note 2)	V$_{CC}$ = 5V f$_{AD}$, f$_{AD}$/2, f$_{AD}$/4 f$_{AD}$=f(X$_{IN}$)
	V$_{CC}$ = 3V f$_{AD}$/2, f$_{AD}$/4 f$_{AD}$=f(X$_{IN}$)
Resolution	8-bit or 10-bit (selectable)
Absolute precision	V$_{CC}$ = 5V • 8-bit resolution ±2LSB • 10-bit resolution ±3LSB However, when using AN$_0$ to AN$_7$ in the mode which external operation amp is connected : ±7LSB V$_{CC}$ = 3V • Without sample and hold function (8-bit resolution) ±2LSB
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1
Analog input pins	8 pins (AN$_0$ to AN$_7$) + 2 pins (ANEX$_0$ and ANEX$_1$)
A-D conversion start condition	• Software trigger A-D conversion starts when the A-D conversion start flag changes to “1” • External trigger (can be retriggered) A-D conversion starts when the A-D conversion start flag is “1” and the $\overline{ADTRG}$/P9$_7$ input changes from “H” to “L”
Conversion speed per pin	• Without sample and hold function 8-bit resolution: 49 f$_{AD}$ cycles, 10-bit resolution: 59 f$_{AD}$ cycles • With sample and hold function 8-bit resolution: 28 f$_{AD}$ cycles, 10-bit resolution: 33 f$_{AD}$ cycles

Note 1: Does not depend on use of sample and hold function.

Note 2: When f(X$_{IN}$) is over 10 MHz, the f$_{AD}$ frequency must be under 10 MHz by dividing.
Without sample and hold function, set the f$_{AD}$ frequency to 250kHz min.
With the sample and hold function, set the f$_{AD}$ frequency to 1MHz min.

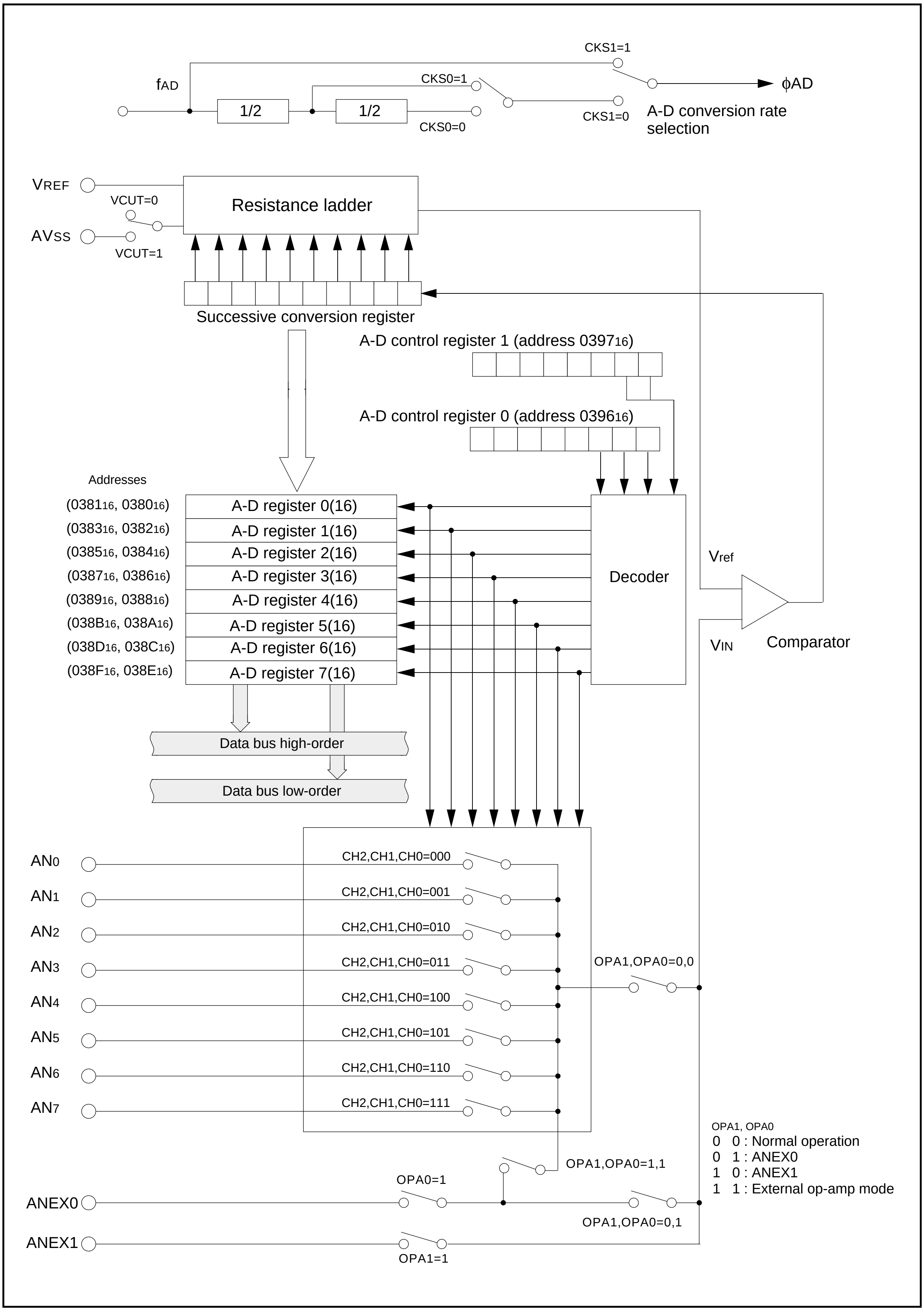

CKS1=1
fAD
CKS0=1
φAD
1/2
1/2
CKS0=0
CKS1=0
A-D conversion rate selection
VREF
VCUT=0
Resistance ladder
AVSS
VCUT=1
Successive conversion register
A-D control register 1 (address 0397₁₆)
A-D control register 0 (address 0396₁₆)
Addresses
(0381₁₆, 0380₁₆) A-D register 0(16)
(0383₁₆, 0382₁₆) A-D register 1(16)
(0385₁₆, 0384₁₆) A-D register 2(16)
(0387₁₆, 0386₁₆) A-D register 3(16)
(0389₁₆, 0388₁₆) A-D register 4(16)
(038B₁₆, 038A₁₆) A-D register 5(16)
(038D₁₆, 038C₁₆) A-D register 6(16)
(038F₁₆, 038E₁₆) A-D register 7(16)
Decoder
Vref
VIN
Comparator
Data bus high-order
Data bus low-order
AN0 CH2,CH1,CH0=000
AN1 CH2,CH1,CH0=001
AN2 CH2,CH1,CH0=010
AN3 CH2,CH1,CH0=011
AN4 CH2,CH1,CH0=100
AN5 CH2,CH1,CH0=101
AN6 CH2,CH1,CH0=110
AN7 CH2,CH1,CH0=111
OPA1,OPA0=0,0
OPA1,OPA0=1,1
OPA0=1
ANEX0
OPA1,OPA0=0,1
ANEX1
OPA1=1
OPA1, OPA0
0 0 : Normal operation
0 1 : ANEX0
1 0 : ANEX1
1 1 : External op-amp mode

Figure 1.21.1. Block diagram of A-D converter

A-D control register 0 (Note 1)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
ADCON0	0396_{16}	$00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN$_0$ is selected 0 0 1 : AN$_1$ is selected 0 1 0 : AN$_2$ is selected 0 1 1 : AN$_3$ is selected 1 0 0 : AN$_4$ is selected 1 0 1 : AN$_5$ is selected 1 1 0 : AN$_6$ is selected 1 1 1 : AN$_7$ is selected (Note 2)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 (Note 2) Repeat sweep mode 1	○	○
MD1			○	○
TRG	Trigger select bit	0 : Software trigger 1 : AD$_{TRG}$ trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : f_{AD}/4 is selected 1 : f_{AD}/2 is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: When changing A-D operation mode, set analog input pin again.

A-D control register 1 (Note 1)

b7 b6 b5 b4 b3 b2 b1 b0

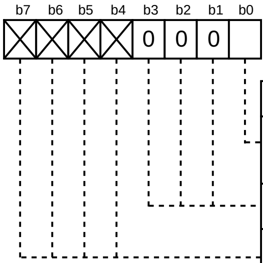
Symbol	Address	When reset
ADCON1	0397_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep and repeat sweep mode 0 are selected b1 b0 0 0 : AN$_0$, AN$_1$ (2 pins) 0 1 : AN$_0$ to AN$_3$ (4 pins) 1 0 : AN$_0$ to AN$_5$ (6 pins) 1 1 : AN$_0$ to AN$_7$ (8 pins) When repeat sweep mode 1 is selected b1 b0 0 0 : AN$_0$ (1 pin) 0 1 : AN$_0$, AN$_1$ (2 pins) 1 0 : AN$_0$ to AN$_2$ (3 pins) 1 1 : AN$_0$ to AN$_3$ (4 pins)	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1 1 : Repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1 (Note 2)	0 : f_{AD}/2 or f_{AD}/4 is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	0 : Vref not connected 1 : Vref connected	○	○
OPA0	External op-amp connection mode bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used(Note 3) 0 1 : ANEX0 input is A-D converted(Note 4) 1 0 : ANEX1 input is A-D converted(Note 5) 1 1 : External op-amp connection mode(Note 6)	○	○
OPA1			○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: When f(X_{IN}) is over 10 MHz, the f_{AD} frequency must be under 10 MHz by dividing.
Note 3: Set "0" to PSL3_5 and PSL3_6 of the function select register B3.
Note 4: Set "1" to PSL3_5 of the function select register B3.
Note 5: Set "1" to PSL3_6 of the function select register B3.
Note 6: Set "1" to PSL3_5 and PSL3_6 of the function select register B3.

Figure 1.21.2. A-D converter-related registers (1)

A-D control register 2 (Note)



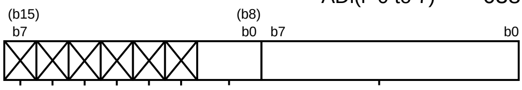
Symbol	Address	When reset
ADCON2	0394_{16}	$XXXXXXX0_2$

Bit symbol	Bit name	Function	R	W
SMP	A-D conversion method select bit	0 : Without sample and hold 1 : With sample and hold	○	○
Reserved bit		Must always set to “0”	○	○
Nothing is assigned. When write, set "0". When read, their content is "0".			–	–

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D register i

Symbol	Address	When reset
ADi(i=0 to 7)	0380_{16} to $038F_{16}$	Indeterminate



Function	R	W
Eight low-order bits of A-D conversion result	○	×
• During 10-bit mode Two high-order bits of A-D conversion result	○	×
• During 8-bit mode When read, the content is indeterminate	×	×
Nothing is assigned. When write, set "0". When read, their content is "0".	–	–

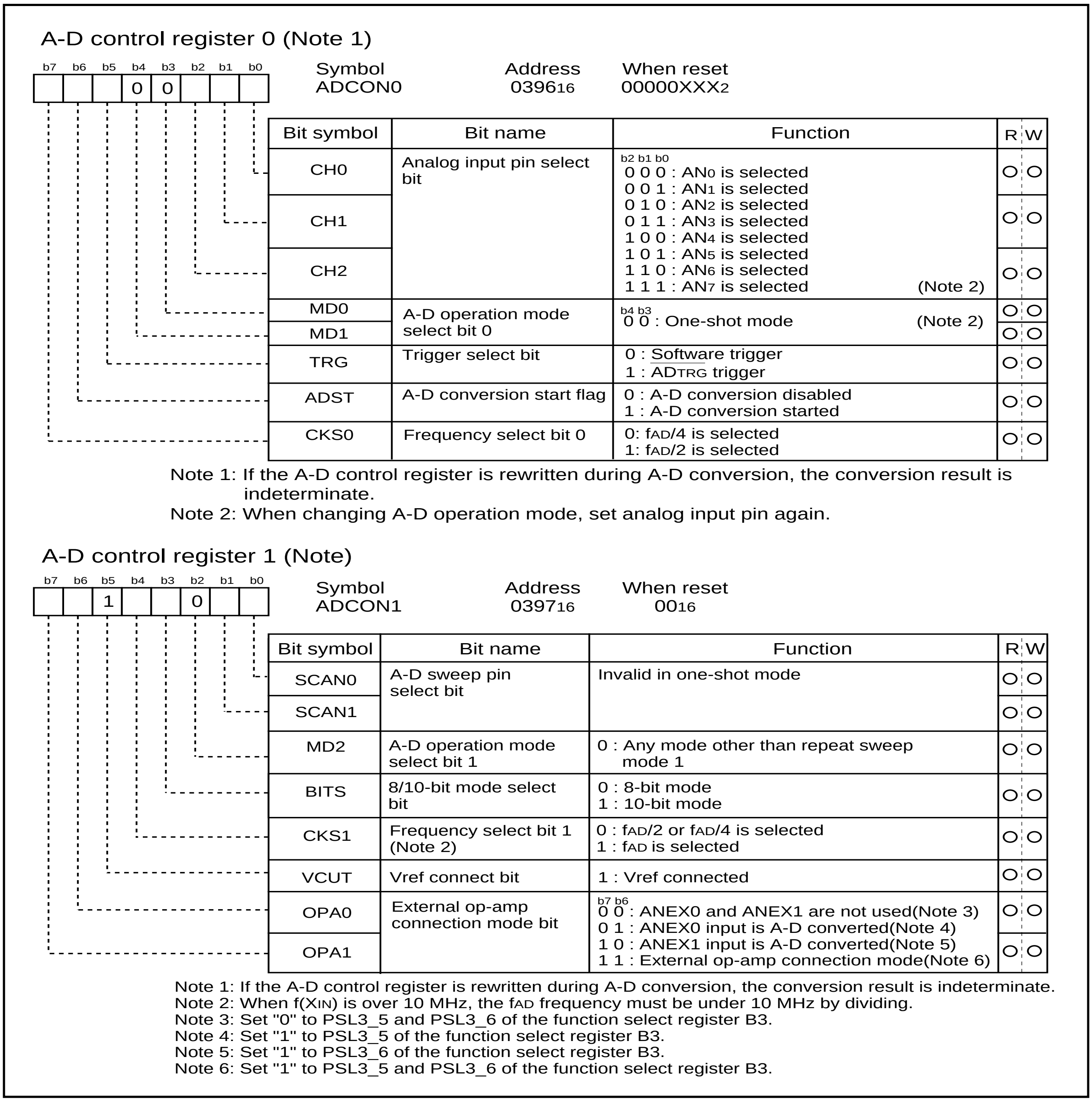
Figure 1.21.3. A-D converter-related registers (2)

(1) One-shot mode

In one-shot mode, the pin selected using the analog input pin select bit is used for one-shot A-D conversion. Table 1.21.2 shows the specifications of one-shot mode. Figure 1.21.4 shows the A-D control register in one-shot mode.

Table 1.21.2. One-shot mode specifications

Item	Specification
Function	The pin selected by the analog input pin select bit is used for one A-D conversion
Start condition	Writing “1” to A-D conversion start flag
Stop condition	• End of A-D conversion (A-D conversion start flag changes to “0”, except when external trigger is selected) • Writing “0” to A-D conversion start flag
Interrupt request generation timing	End of A-D conversion
Input pin	One of AN_0 to AN_7, as selected
Reading of result of A-D converter	Read A-D register corresponding to selected pin



A-D control register 0 (Note 1)

Symbol	Address	When reset
ADCON0	0396_{16}	$00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN_0 is selected 0 0 1 : AN_1 is selected 0 1 0 : AN_2 is selected 0 1 1 : AN_3 is selected 1 0 0 : AN_4 is selected 1 0 1 : AN_5 is selected 1 1 0 : AN_6 is selected 1 1 1 : AN_7 is selected (Note 2)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 0 : One-shot mode (Note 2)	○	○
MD1			○	○
TRG	Trigger select bit	0 : Software trigger 1 : ADTRG trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0: f_{AD}/4 is selected 1: f_{AD}/2 is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: When changing A-D operation mode, set analog input pin again.

A-D control register 1 (Note)

Symbol	Address	When reset
ADCON1	0397_{16}	00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	Invalid in one-shot mode	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1 (Note 2)	0 : f_{AD}/2 or f_{AD}/4 is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
OPA0	External op-amp connection mode bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used(Note 3) 0 1 : ANEX0 input is A-D converted(Note 4) 1 0 : ANEX1 input is A-D converted(Note 5) 1 1 : External op-amp connection mode(Note 6)	○	○
OPA1			○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: When f(X_{IN}) is over 10 MHz, the f_{AD} frequency must be under 10 MHz by dividing.
Note 3: Set "0" to PSL3_5 and PSL3_6 of the function select register B3.
Note 4: Set "1" to PSL3_5 of the function select register B3.
Note 5: Set "1" to PSL3_6 of the function select register B3.
Note 6: Set "1" to PSL3_5 and PSL3_6 of the function select register B3.

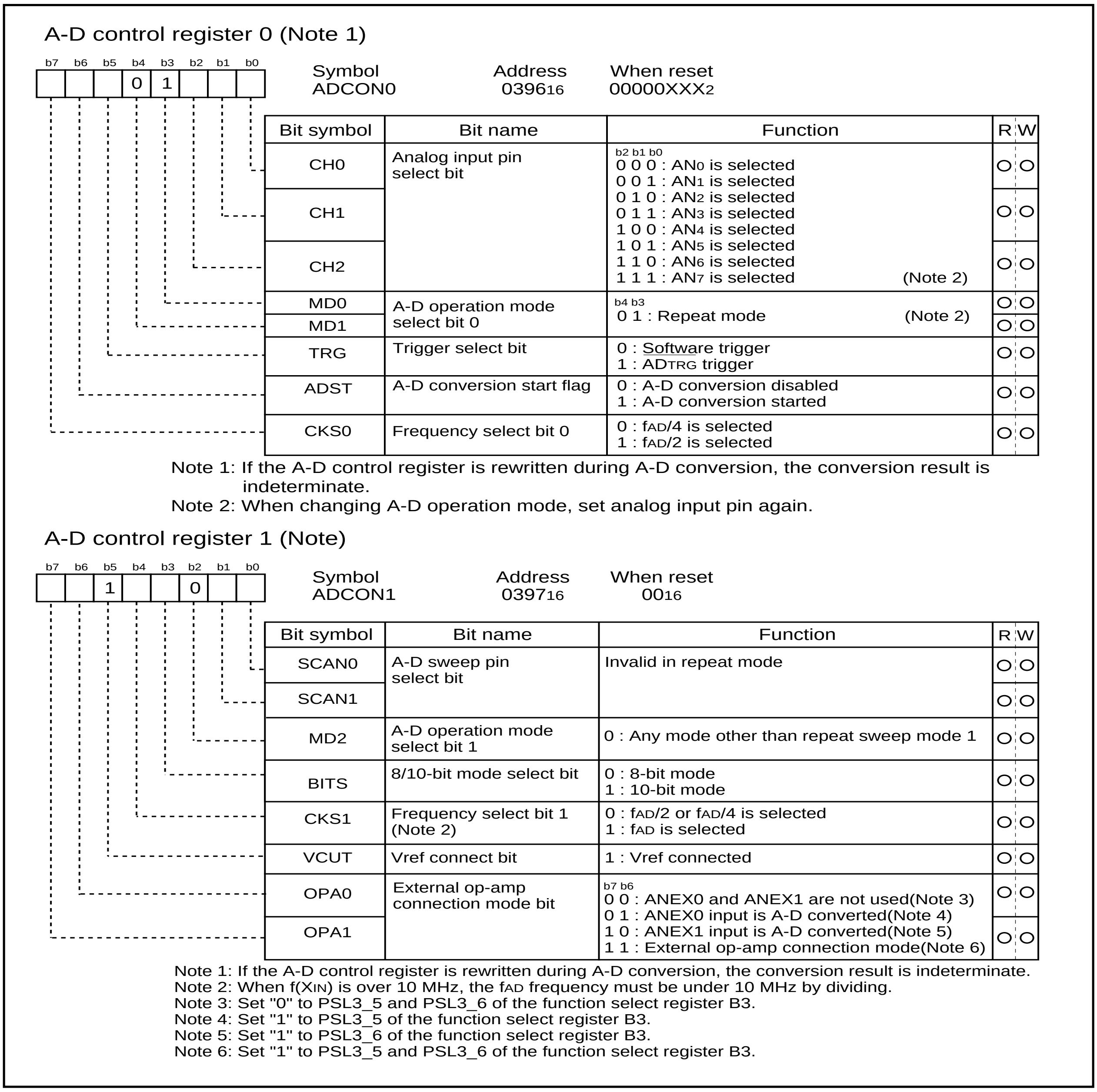
Figure 1.21.4. A-D conversion register in one-shot mode

(2) Repeat mode

In repeat mode, the pin selected using the analog input pin select bit is used for repeated A-D conversion. Table 1.21.3 shows the specifications of repeat mode. Figure 1.21.5 shows the A-D control register in repeat mode.

Table 1.21.3. Repeat mode specifications

Item	Specification
Function	The pin selected by the analog input pin select bit is used for repeated A-D conversion
Star condition	Writing “1” to A-D conversion start flag
Stop condition	Writing “0” to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	One of AN_0 to AN_7, as selected
Reading of result of A-D converter	Read A-D register corresponding to selected pin (at any time)



A-D control register 0 (Note 1)

Symbol ADCON0 Address 0396_{16} When reset $00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	b2 b1 b0 0 0 0 : AN_0 is selected 0 0 1 : AN_1 is selected 0 1 0 : AN_2 is selected 0 1 1 : AN_3 is selected 1 0 0 : AN_4 is selected 1 0 1 : AN_5 is selected 1 1 0 : AN_6 is selected 1 1 1 : AN_7 is selected (Note 2)	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 0 1 : Repeat mode (Note 2)	○	○
MD1			○	○
TRG	Trigger select bit	0 : Software trigger 1 : AD_{TRG} trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : f_{AD}/4 is selected 1 : f_{AD}/2 is selected	○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: When changing A-D operation mode, set analog input pin again.

A-D control register 1 (Note)

Symbol ADCON1 Address 0397_{16} When reset 00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	Invalid in repeat mode	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1 (Note 2)	0 : f_{AD}/2 or f_{AD}/4 is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
OPA0	External op-amp connection mode bit	b7 b6 0 0 : ANEX0 and ANEX1 are not used(Note 3) 0 1 : ANEX0 input is A-D converted(Note 4) 1 0 : ANEX1 input is A-D converted(Note 5) 1 1 : External op-amp connection mode(Note 6)	○	○
OPA1			○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: When $f(X_{IN})$ is over 10 MHz, the f_{AD} frequency must be under 10 MHz by dividing.
Note 3: Set "0" to PSL3_5 and PSL3_6 of the function select register B3.
Note 4: Set "1" to PSL3_5 of the function select register B3.
Note 5: Set "1" to PSL3_6 of the function select register B3.
Note 6: Set "1" to PSL3_5 and PSL3_6 of the function select register B3.

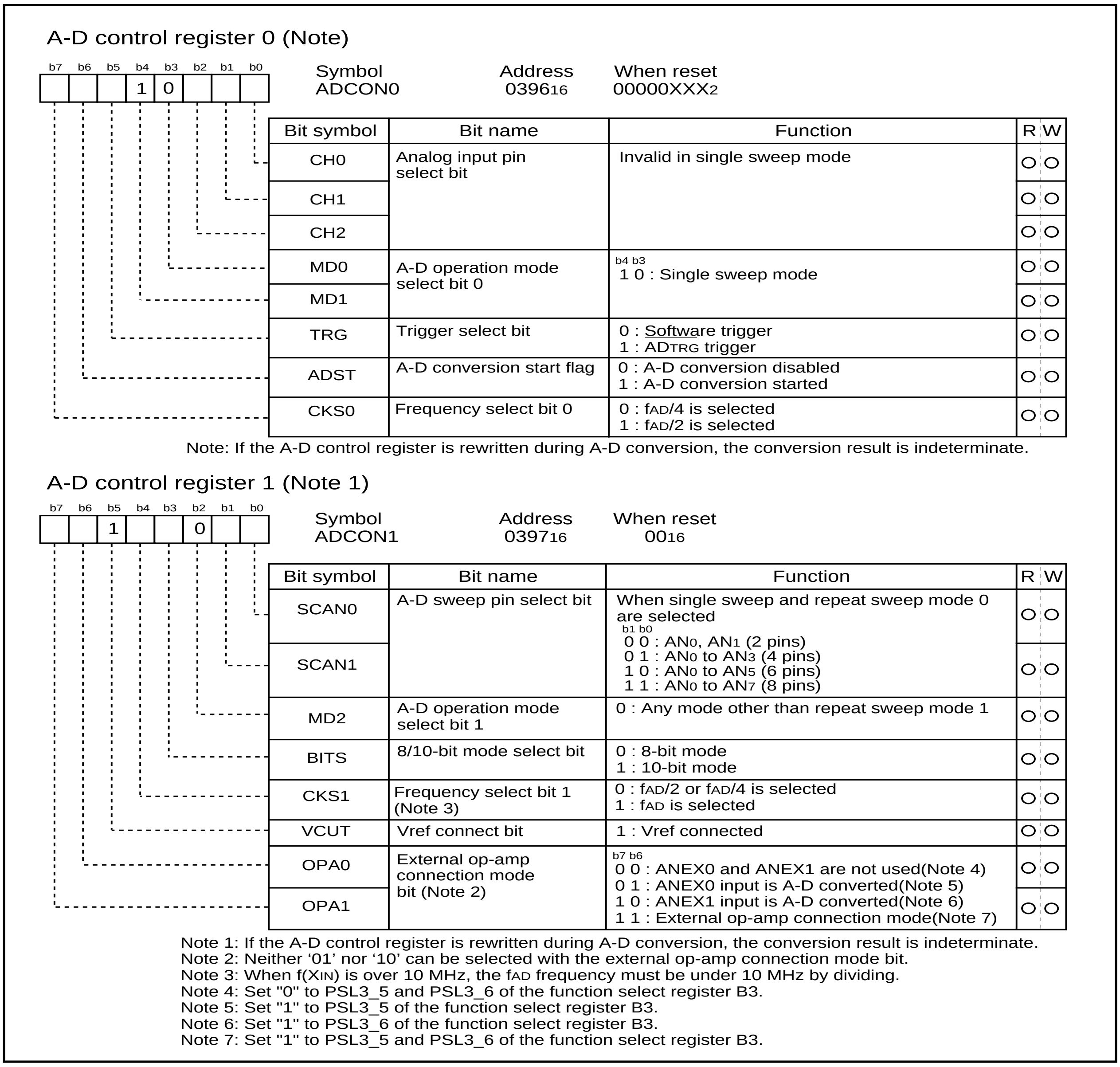
Figure 1.21.5. A-D conversion register in repeat mode

(3) Single sweep mode

In single sweep mode, the pins selected using the A-D sweep pin select bit are used for one-by-one A-D conversion. Table 1.21.4 shows the specifications of single sweep mode. Figure 1.21.6 shows the A-D control register in single sweep mode.

Table 1.21.4. Single sweep mode specifications

Item	Specification
Function	The pins selected by the A-D sweep pin select bit are used for one-by-one A-D conversion
Start condition	Writing “1” to A-D converter start flag
Stop condition	• End of A-D conversion (A-D conversion start flag changes to “0”, except when external trigger is selected) • Writing “0” to A-D conversion start flag
Interrupt request generation timing	End of A-D conversion
Input pin	AN$_0$ and AN$_1$ (2 pins), AN$_0$ to AN$_3$ (4 pins), AN$_0$ to AN$_5$ (6 pins), or AN$_0$ to AN$_7$ (8 pins)
Reading of result of A-D converter	Read A-D register corresponding to selected pin



Symbol ADCON0 Address 0396$_{16}$ When reset 00000XXX$_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	Invalid in single sweep mode	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 1 0 : Single sweep mode	○	○
MD1			○	○
TRG	Trigger select bit	0 : Software trigger 1 : AD$_{TRG}$ trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : f$_{AD}$/4 is selected 1 : f$_{AD}$/2 is selected	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

Symbol ADCON1 Address 0397$_{16}$ When reset 00$_{16}$

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep and repeat sweep mode 0 are selected b1 b0 0 0 : AN$_0$, AN$_1$ (2 pins) 0 1 : AN$_0$ to AN$_3$ (4 pins) 1 0 : AN$_0$ to AN$_5$ (6 pins) 1 1 : AN$_0$ to AN$_7$ (8 pins)	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1 (Note 3)	0 : f$_{AD}$/2 or f$_{AD}$/4 is selected 1 : f$_{AD}$ is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
OPA0	External op-amp connection mode bit (Note 2)	b7 b6 0 0 : ANEX0 and ANEX1 are not used(Note 4) 0 1 : ANEX0 input is A-D converted(Note 5) 1 0 : ANEX1 input is A-D converted(Note 6) 1 1 : External op-amp connection mode(Note 7)	○	○
OPA1			○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: Neither ‘01’ nor ‘10’ can be selected with the external op-amp connection mode bit.
Note 3: When f(X$_{IN}$) is over 10 MHz, the f$_{AD}$ frequency must be under 10 MHz by dividing.
Note 4: Set "0" to PSL3_5 and PSL3_6 of the function select register B3.
Note 5: Set "1" to PSL3_5 of the function select register B3.
Note 6: Set "1" to PSL3_6 of the function select register B3.
Note 7: Set "1" to PSL3_5 and PSL3_6 of the function select register B3.

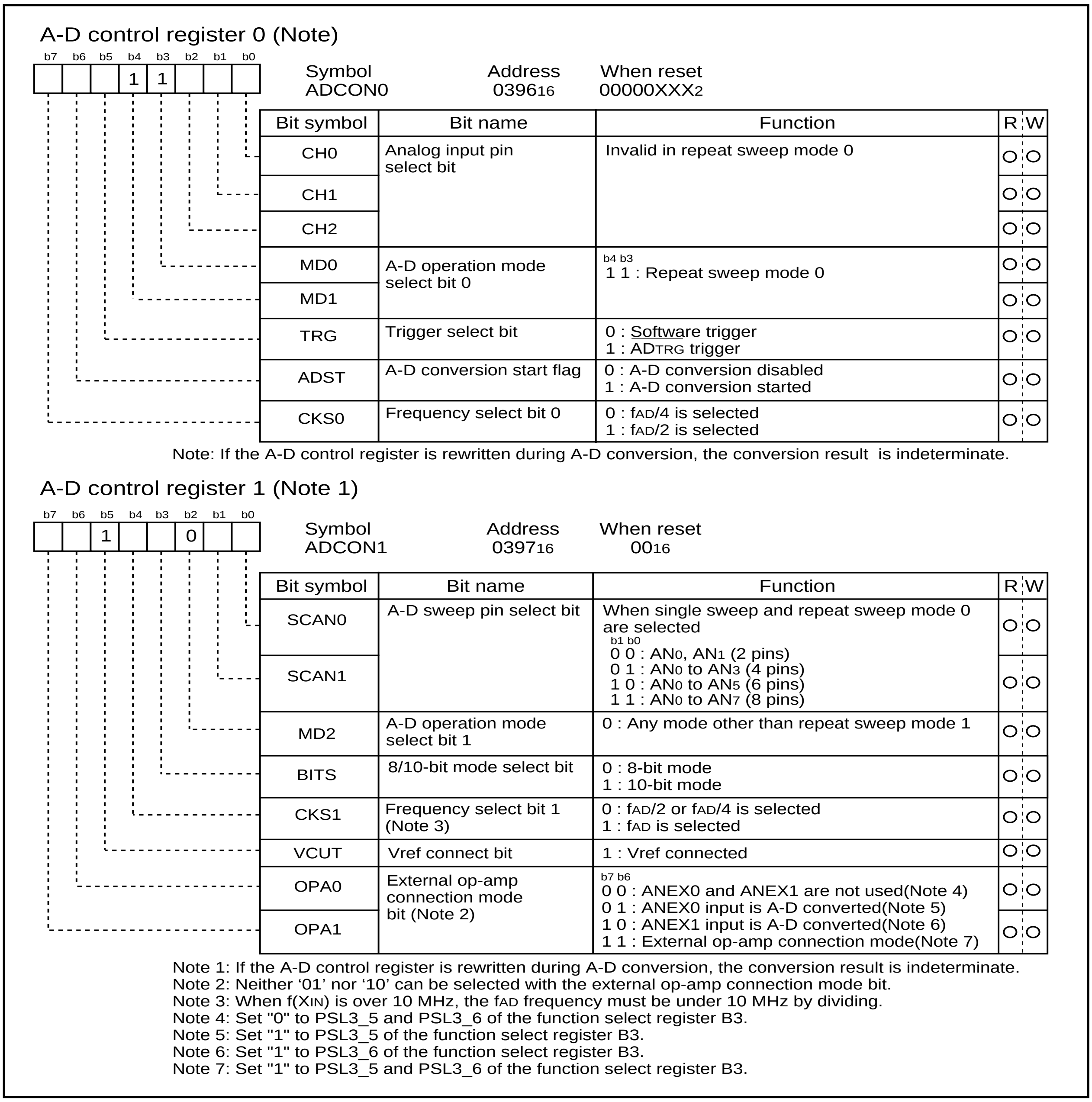
Figure 1.21.6. A-D conversion register in single sweep mode

(4) Repeat sweep mode 0

In repeat sweep mode 0, the pins selected using the A-D sweep pin select bit are used for repeat sweep A-D conversion. Table 1.21.5 shows the specifications of repeat sweep mode 0. Figure 1.21.7 shows the A-D control register in repeat sweep mode 0.

Table 1.21.5. Repeat sweep mode 0 specifications

Item	Specification
Function	The pins selected by the A-D sweep pin select bit are used for repeat sweep A-D conversion
Start condition	Writing “1” to A-D conversion start flag
Stop condition	Writing “0” to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	AN_0 and AN_1 (2 pins), AN_0 to AN3 (4 pins), AN_0 to AN5 (6 pins), or AN_0 to AN_7 (8 pins)
Reading of result of A-D converter	Read A-D register corresponding to selected pin (at any time)



A-D control register 0 (Note)

Symbol ADCON0 Address 0396_{16} When reset $00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	Invalid in repeat sweep mode 0	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 1 1 : Repeat sweep mode 0	○	○
MD1			○	○
TRG	Trigger select bit	0 : Software trigger 1 : AD_{TRG} trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : f_{AD}/4 is selected 1 : f_{AD}/2 is selected	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D control register 1 (Note 1)

Symbol ADCON1 Address 0397_{16} When reset 00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When single sweep and repeat sweep mode 0 are selected b1 b0 0 0 : AN_0, AN_1 (2 pins) 0 1 : AN_0 to AN_3 (4 pins) 1 0 : AN_0 to AN_5 (6 pins) 1 1 : AN_0 to AN_7 (8 pins)	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	0 : Any mode other than repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1 (Note 3)	0 : f_{AD}/2 or f_{AD}/4 is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
OPA0	External op-amp connection mode bit (Note 2)	b7 b6 0 0 : ANEX0 and ANEX1 are not used(Note 4) 0 1 : ANEX0 input is A-D converted(Note 5) 1 0 : ANEX1 input is A-D converted(Note 6) 1 1 : External op-amp connection mode(Note 7)	○	○
OPA1			○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: Neither ‘01’ nor ‘10’ can be selected with the external op-amp connection mode bit.
Note 3: When $f(X_{IN})$ is over 10 MHz, the f_{AD} frequency must be under 10 MHz by dividing.
Note 4: Set "0" to PSL3_5 and PSL3_6 of the function select register B3.
Note 5: Set "1" to PSL3_5 of the function select register B3.
Note 6: Set "1" to PSL3_6 of the function select register B3.
Note 7: Set "1" to PSL3_5 and PSL3_6 of the function select register B3.

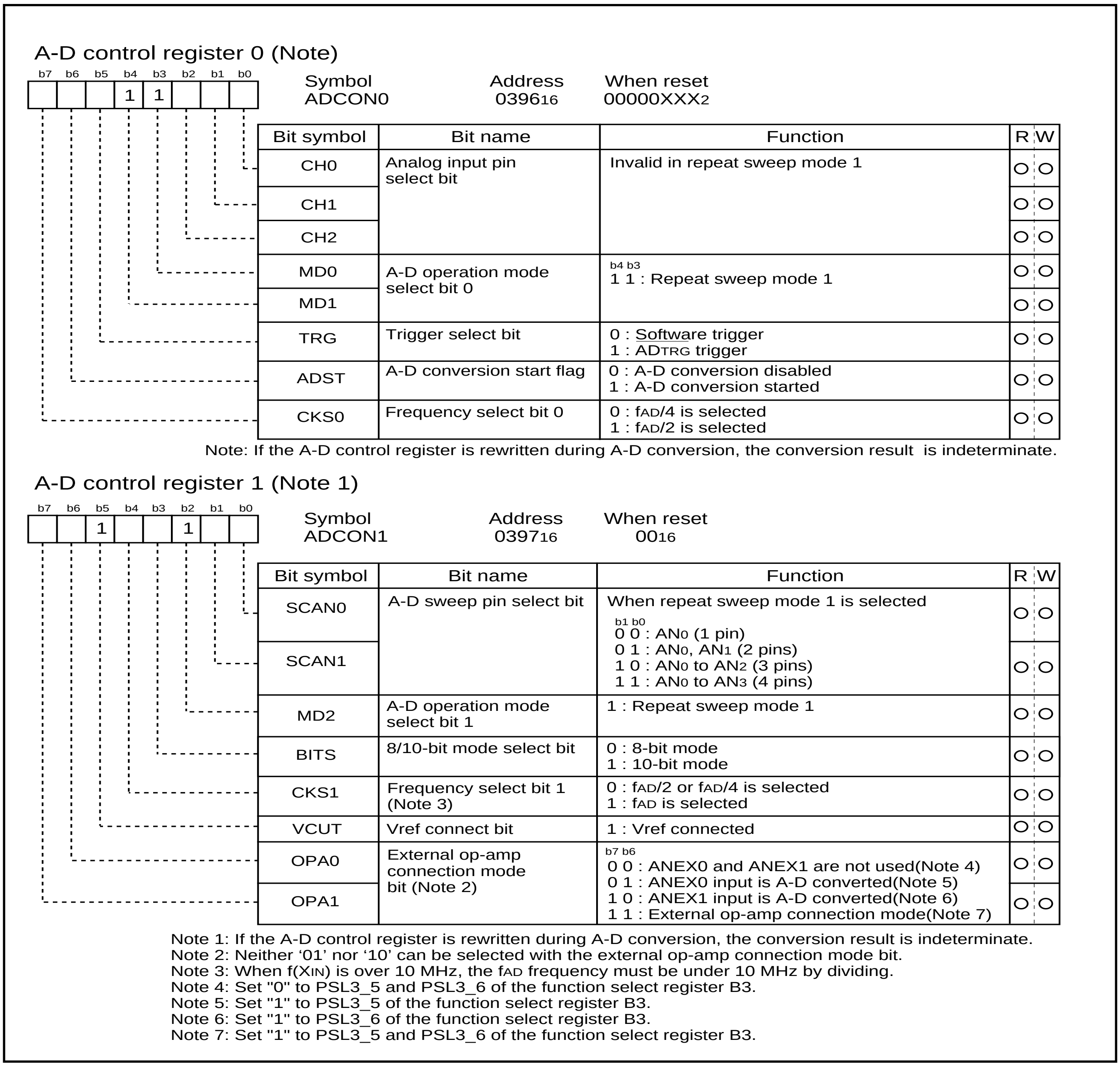
Figure 1.21.7. A-D conversion register in repeat sweep mode 0

(5) Repeat sweep mode 1

In repeat sweep mode 1, all pins are used for A-D conversion with emphasis on the pin or pins selected using the A-D sweep pin select bit. Table 1.21.6 shows the specifications of repeat sweep mode 1. Figure 1.21.8 shows the A-D control register in repeat sweep mode 1.

Table 1.21.6. Repeat sweep mode 1 specifications

Item	Specification
Function	All pins perform repeat sweep A-D conversion, with emphasis on the pin or pins selected by the A-D sweep pin select bit Example : AN_0 selected AN_0 → AN_1 → AN_0 → AN_2 → AN_0 → AN_3, etc
Start condition	Writing “1” to A-D conversion start flag
Stop condition	Writing “0” to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	AN_0 to AN_7
With emphasis on the pin	AN_0 (1 pin), AN_0 and AN_1 (2 pins), AN_0 to AN_2 (3 pins), AN_0 to AN_3 (4 pins)
Reading of result of A-D converter	Read A-D register corresponding to selected pin (at any time)



A-D control register 0 (Note)

Symbol ADCON0 Address 0396_{16} When reset $00000XXX_2$

Bit symbol	Bit name	Function	R	W
CH0	Analog input pin select bit	Invalid in repeat sweep mode 1	○	○
CH1			○	○
CH2			○	○
MD0	A-D operation mode select bit 0	b4 b3 1 1 : Repeat sweep mode 1	○	○
MD1			○	○
TRG	Trigger select bit	0 : Software trigger 1 : ADTRG trigger	○	○
ADST	A-D conversion start flag	0 : A-D conversion disabled 1 : A-D conversion started	○	○
CKS0	Frequency select bit 0	0 : $f_{AD}/4$ is selected 1 : $f_{AD}/2$ is selected	○	○

Note: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.

A-D control register 1 (Note 1)

Symbol ADCON1 Address 0397_{16} When reset 00_{16}

Bit symbol	Bit name	Function	R	W
SCAN0	A-D sweep pin select bit	When repeat sweep mode 1 is selected b1 b0 0 0 : AN_0 (1 pin) 0 1 : AN_0, AN_1 (2 pins) 1 0 : AN_0 to AN_2 (3 pins) 1 1 : AN_0 to AN_3 (4 pins)	○	○
SCAN1			○	○
MD2	A-D operation mode select bit 1	1 : Repeat sweep mode 1	○	○
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	○	○
CKS1	Frequency select bit 1 (Note 3)	0 : $f_{AD}/2$ or $f_{AD}/4$ is selected 1 : f_{AD} is selected	○	○
VCUT	Vref connect bit	1 : Vref connected	○	○
OPA0	External op-amp connection mode bit (Note 2)	b7 b6 0 0 : ANEX0 and ANEX1 are not used(Note 4) 0 1 : ANEX0 input is A-D converted(Note 5) 1 0 : ANEX1 input is A-D converted(Note 6) 1 1 : External op-amp connection mode(Note 7)	○	○
OPA1			○	○

Note 1: If the A-D control register is rewritten during A-D conversion, the conversion result is indeterminate.
Note 2: Neither ‘01’ nor ‘10’ can be selected with the external op-amp connection mode bit.
Note 3: When $f(X_{IN})$ is over 10 MHz, the f_{AD} frequency must be under 10 MHz by dividing.
Note 4: Set "0" to PSL3_5 and PSL3_6 of the function select register B3.
Note 5: Set "1" to PSL3_5 of the function select register B3.
Note 6: Set "1" to PSL3_6 of the function select register B3.
Note 7: Set "1" to PSL3_5 and PSL3_6 of the function select register B3.

Figure 1.21.8. A-D conversion register in repeat sweep mode 1

(a) Sample and hold

Sample and hold is selected by setting bit 0 of the A-D control register 2 (address 0394_{16}) to “1”. When sample and hold is selected, the rate of conversion of each pin increases. As a result, a 28 f_{AD} cycle is achieved with 8-bit resolution and 33 f_{AD} with 10-bit resolution. Sample and hold can be selected in all modes. However, in all modes, be sure to specify before starting A-D conversion whether sample and hold is to be used.

(b) Extended analog input pins

In one-shot mode and repeat mode, the input via the extended analog input pins $ANEX_0$ and $ANEX_1$ can also be converted from analog to digital.

When bit 6 of the A-D control register 1 (address 0397_{16}) is “1” and bit 7 is “0”, input via $ANEX_0$ is converted from analog to digital. The result of conversion is stored in A-D register 0.

When bit 6 of the A-D control register 1 (address 0397_{16}) is “0” and bit 7 is “1”, input via $ANEX_1$ is converted from analog to digital. The result of conversion is stored in A-D register 1.

Set the related input peripheral function of the function select register B3 to disabled.

(c) External operation amp connection mode

In this mode, multiple external analog inputs via the extended analog input pins, $ANEX_0$ and $ANEX_1$, can be amplified together by just one operation amp and used as the input for A-D conversion.

When bit 6 of the A-D control register 1 (address 0397_{16}) is “1” and bit 7 is “1”, input via AN_0 to AN_7 is output from $ANEX_0$. The input from $ANEX_1$ is converted from analog to digital and the result stored in the corresponding A-D register. The speed of A-D conversion depends on the response of the external operation amp. Do not connect the $ANEX_0$ and $ANEX_1$ pins directly. Figure 1.21.9 is an example of how to connect the pins in external operation amp mode.

Set the related input peripheral function of the function select register B3 to disabled.

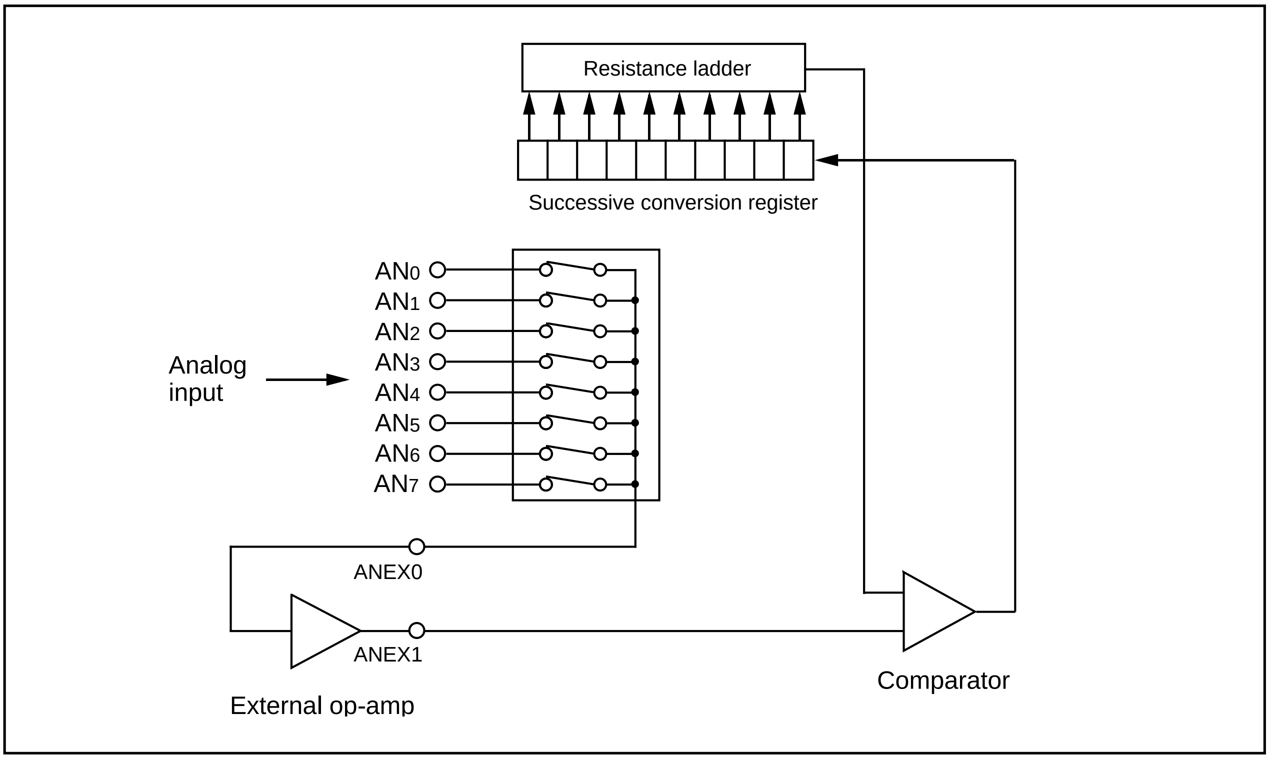


Figure 1.21.9. Example of external op-amp connection mode

D-A Converter

This is an 8-bit, R-2R type D-A converter. The microcomputer contains two independent D-A converters of this type. D-A conversion is performed when a value is written to the corresponding D-A register. Bits 0 and 1 (D-A output enable bits) of the D-A control register decide if the result of conversion is to be output. Set the function select register A to I/O port, the related input peripheral function of the function select register B3 to disabled and the direction register to input mode. When the D-A output is enabled, the pull-up function of the corresponding port is automatically disabled.

Output analog voltage (V) is determined by a set value (n : decimal) in the D-A register.

$V = V_{REF} \times n / 256$ (n = 0 to 255)

V_{REF} : reference voltage

Table 1.22.1 lists the performance of the D-A converter. Figure 1.22.1 shows the block diagram of the D-A converter. Figure 1.22.2 shows the D-A control register.

Table 1.22.1. Performance of D-A converter

Item	Performance
Conversion method	R-2R method
Resolution	8 bits
Analog output pin	2 channels

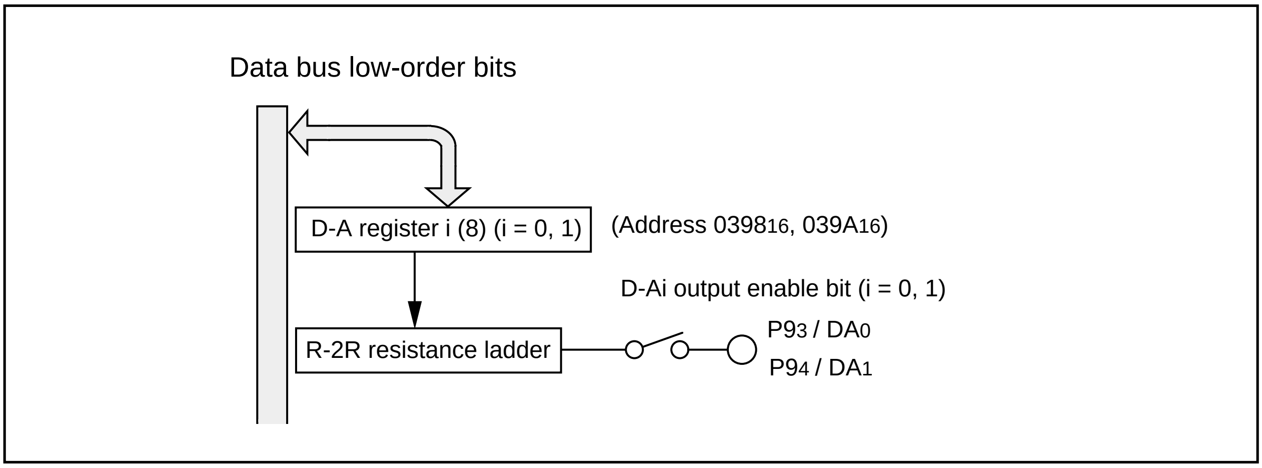
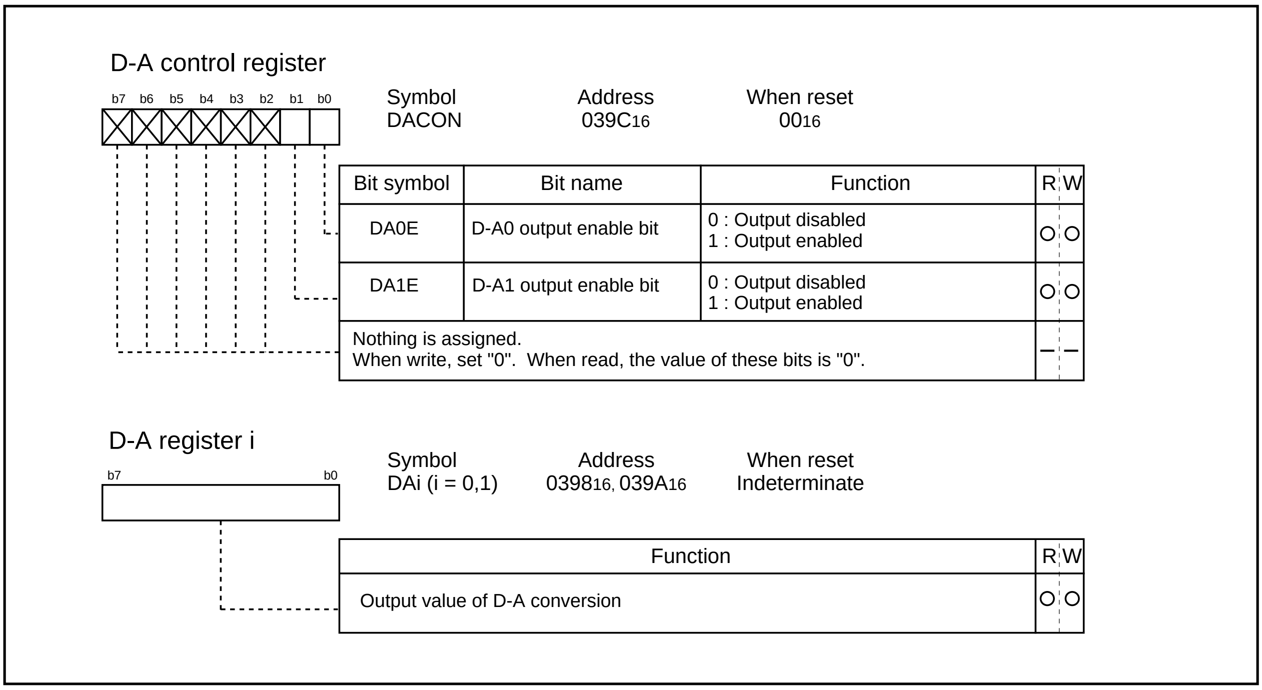


Figure 1.22.1. Block diagram of D-A converter



D-A control register

Symbol	Address	When reset
DACON	039C$_{16}$	00$_{16}$

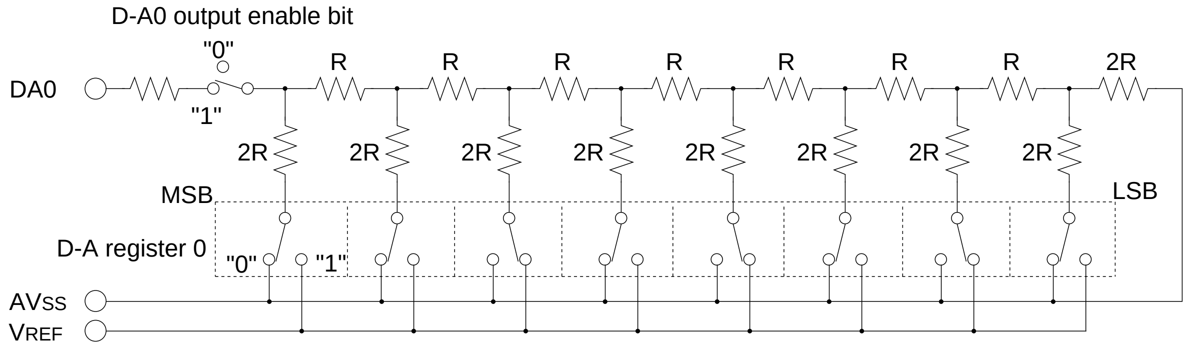
Bit symbol	Bit name	Function	R	W
DA0E	D-A0 output enable bit	0 : Output disabled 1 : Output enabled	○	○
DA1E	D-A1 output enable bit	0 : Output disabled 1 : Output enabled	○	○
Nothing is assigned. When write, set "0". When read, the value of these bits is "0".			–	–

D-A register i

Symbol	Address	When reset
DAi (i = 0,1)	0398$_{16}$, 039A$_{16}$	Indeterminate

Function	R	W
Output value of D-A conversion	○	○

Figure 1.22.2. D-A control register



Note 1: In the above figure, the D-A register value is "2A$_{16}$".
Note 2: This circuit is the same in D-A1.
Note 3: To save power when not using the D-A converter, set the D-A output enable bit to "0" and the D-A register to "00$_{16}$", and prevent current flowing to the R-2R resistance.

Figure 1.22.3. D-A converter equivalent circuit

CRC Calculation Circuit

The Cyclic Redundancy Check (CRC) calculation circuit detects an error in data blocks. The microcomputer uses a generator polynomial of CRC_CCITT ($X^{16} + X^{12} + X^{5} + 1$) to generate CRC code.

The CRC code is a 16-bit code generated for a block of a given data length in multiples of 8 bits. The CRC code is set in a CRC data register each time one byte of data is transferred to a CRC input register after writing an initial value into the CRC data register. Generation of CRC code for one byte of data is completed in two machine cycles.

Figure 1.23.1 shows the block diagram of the CRC circuit. Figure 1.23.2 shows the CRC-related registers.

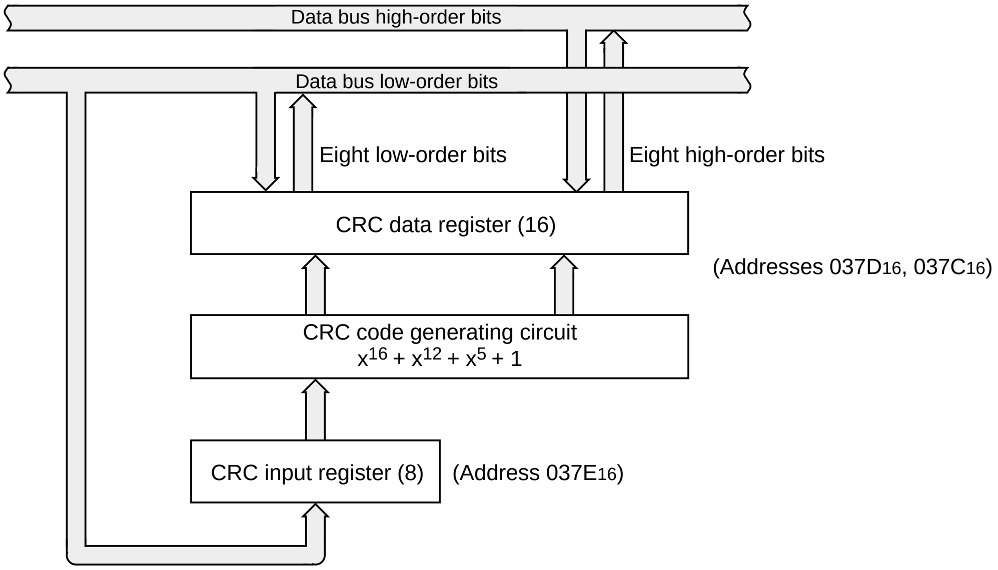
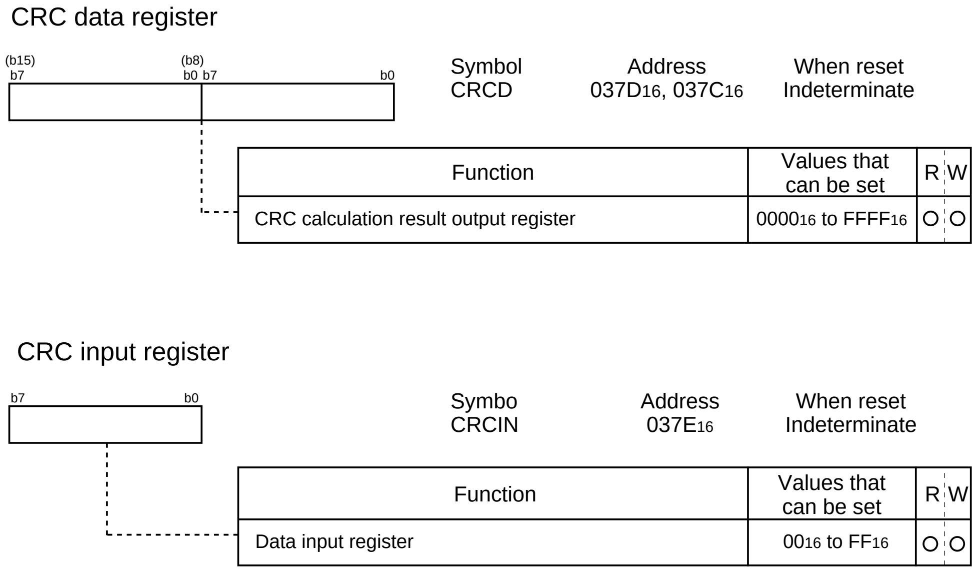


Figure 1.23.1. Block diagram of CRC circuit



CRC data register

Symbol CRCD

Address $037D_{16}$, $037C_{16}$

When reset Indeterminate

Function	Values that can be set	R	W
CRC calculation result output register	0000_{16} to $FFFF_{16}$	○	○

CRC input register

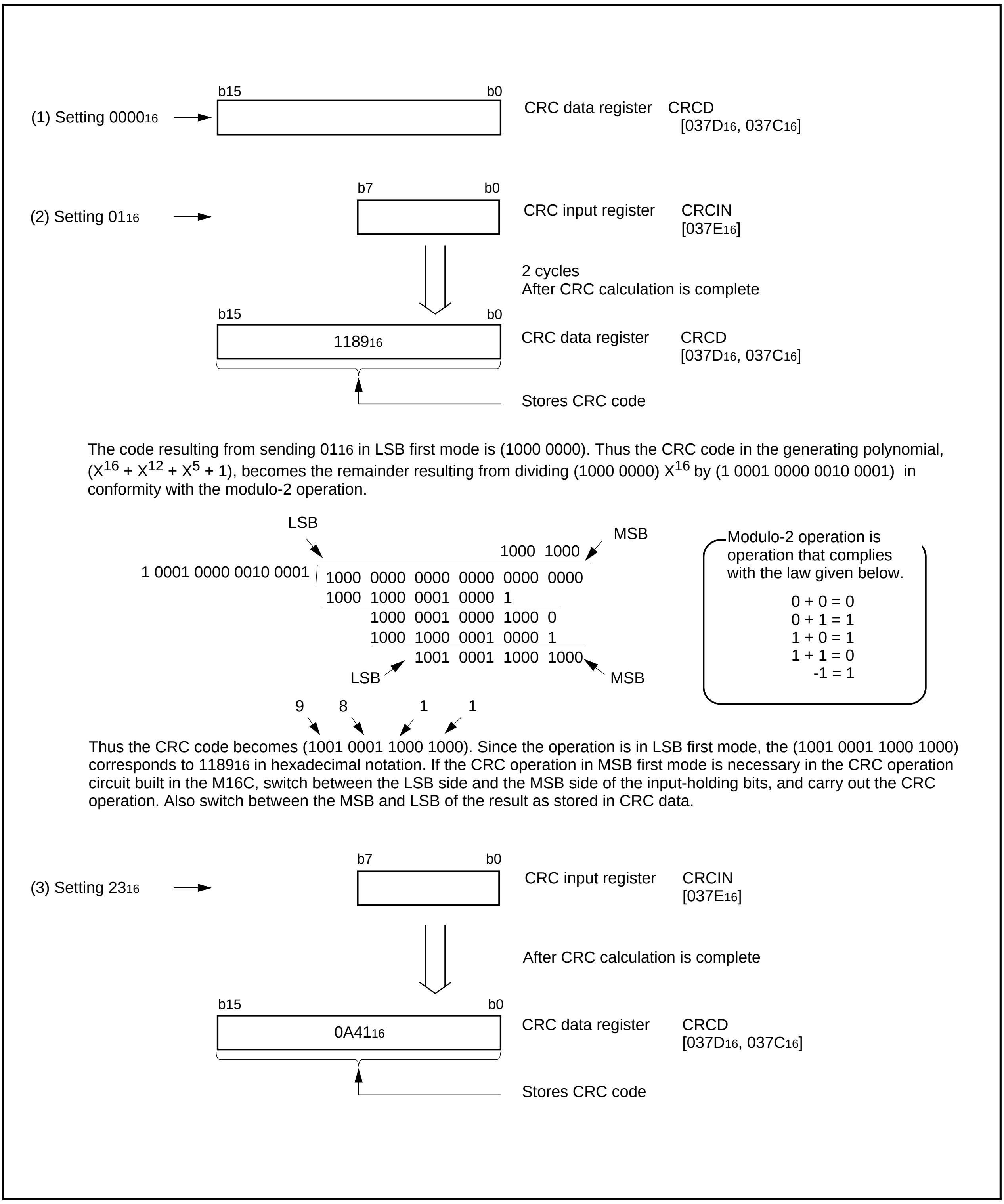
Symbo CRCIN

Address $037E_{16}$

When reset Indeterminate

Function	Values that can be set	R	W
Data input register	00_{16} to FF_{16}	○	○

Figure 1.23.2. CRC-related registers



The code resulting from sending 01_{16} in LSB first mode is (1000 0000). Thus the CRC code in the generating polynomial, ($X^{16} + X^{12} + X^5 + 1$), becomes the remainder resulting from dividing (1000 0000) X^{16} by (1 0001 0000 0010 0001) in conformity with the modulo-2 operation.

```
                    LSB                                MSB
                                            1000  1000
1 0001 0000 0010 0001 ) 1000  0000  0000  0000  0000  0000
                        1000  1000  0001  0000  1
                              1000  0001  0000  1000  0
                              1000  1000  0001  0000  1
                                    1001  0001  1000  1000
                              LSB                           MSB
                    9     8           1     1
```

Modulo-2 operation is operation that complies with the law given below.
0 + 0 = 0
0 + 1 = 1
1 + 0 = 1
1 + 1 = 0
-1 = 1

Thus the CRC code becomes (1001 0001 1000 1000). Since the operation is in LSB first mode, the (1001 0001 1000 1000) corresponds to 1189_{16} in hexadecimal notation. If the CRC operation in MSB first mode is necessary in the CRC operation circuit built in the M16C, switch between the LSB side and the MSB side of the input-holding bits, and carry out the CRC operation. Also switch between the MSB and LSB of the result as stored in CRC data.

Figure 1.23.3. CRC example

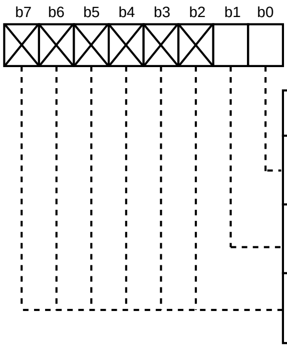
X-Y Converter

X-Y conversion rotates the 16 x 16 matrix data by 90 degrees. It can also be used to invert the top and bottom of the 16-bit data. Figure 1.24.1 shows the XY control register.

The Xi and the Yi registers are 16-bit registers. There are 16 of each (where i= 0 to 15).

The Xi and Yi registers are mapped to the same address. The Xi register is a write-only register, while the Yi register is a read-only register. Be sure to access the Xi and Yi registers in 16-bit units from an even address. Operation cannot be guaranteed if you attempt to access these registers in 8-bit units.

XY control register



Symbol	Address	When reset
XYC	02E0$_{16}$	XXXXXX00$_2$

Bit symbol	Bit name	Function	R	W
XYC0	Read-mode set bit	0 : Data conversion 1 : No data conversion	○	○
XYC1	Write-mode set bit	0 : No bit mapping conversion 1 : Bit mapping conversion	○	○
Nothing is assigned. When write, set "0". When read, the value of these bits is indeterminate.			—	—

Figure 1.24.1. XY control register

The reading of the Yi register is controlled by the read-mode set bit (bit 0 at address $02E0_{16}$).
When the read-mode set bit (bit 0 at address $02E0_{16}$) is “0”, specific bits in the Xi register can be read at the same time as the Yi register is read.
For example, when you read the Y0 register, bit 0 is read as bit 0 of the X0 register, bit 1 is read as bit 0 of the X1 register, ..., bit 14 is read as bit 0 of the X14 register, bit 15 as bit 0 of the X15 register. Similarly, when you read the Y15 register, bit 0 is bit 15 of the X0 register, bit 1 is bit 15 of the X1 register, ..., bit 14 is bit 15 of the X14 register, bit 15 is bit 15 of the X15 register.
Figure 1.24.2 shows the conversion table when the read mode set bit = “0”. Figure 1.24.3 shows the X-Y conversion example.

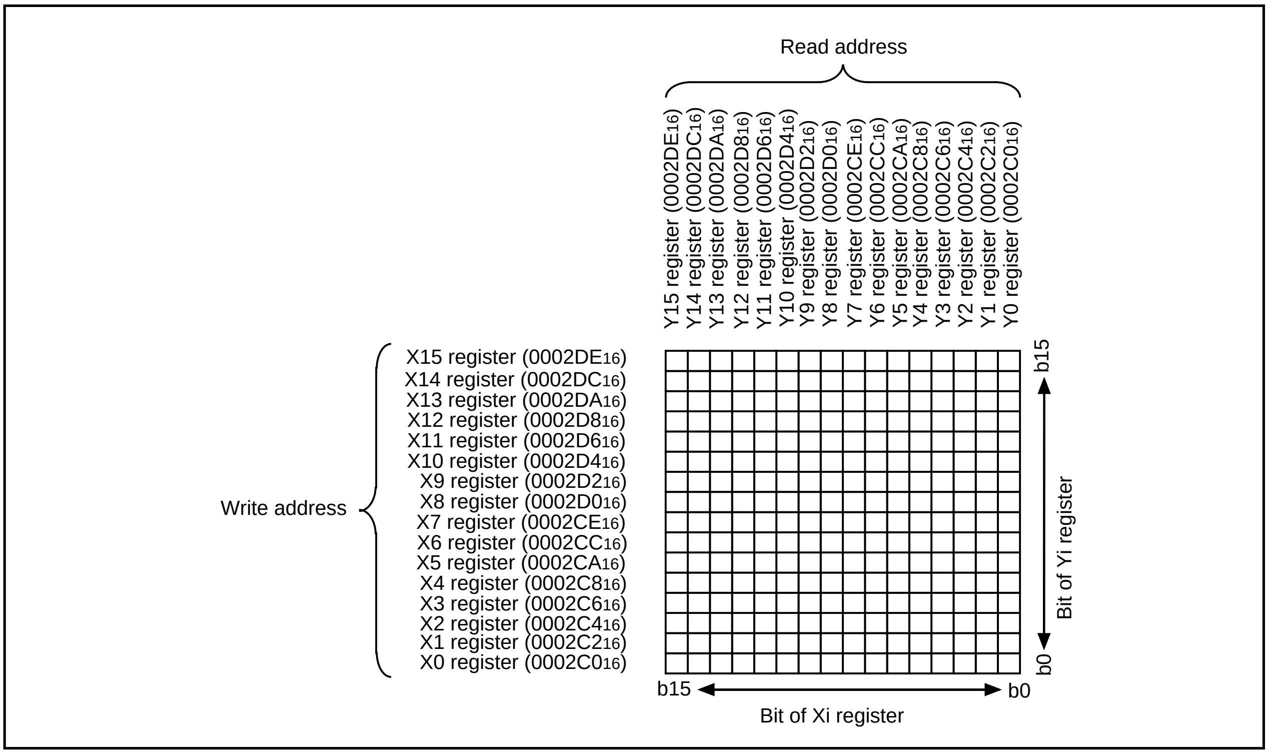


Figure 1.24.2. Conversion table when the read mode set bit = “0”

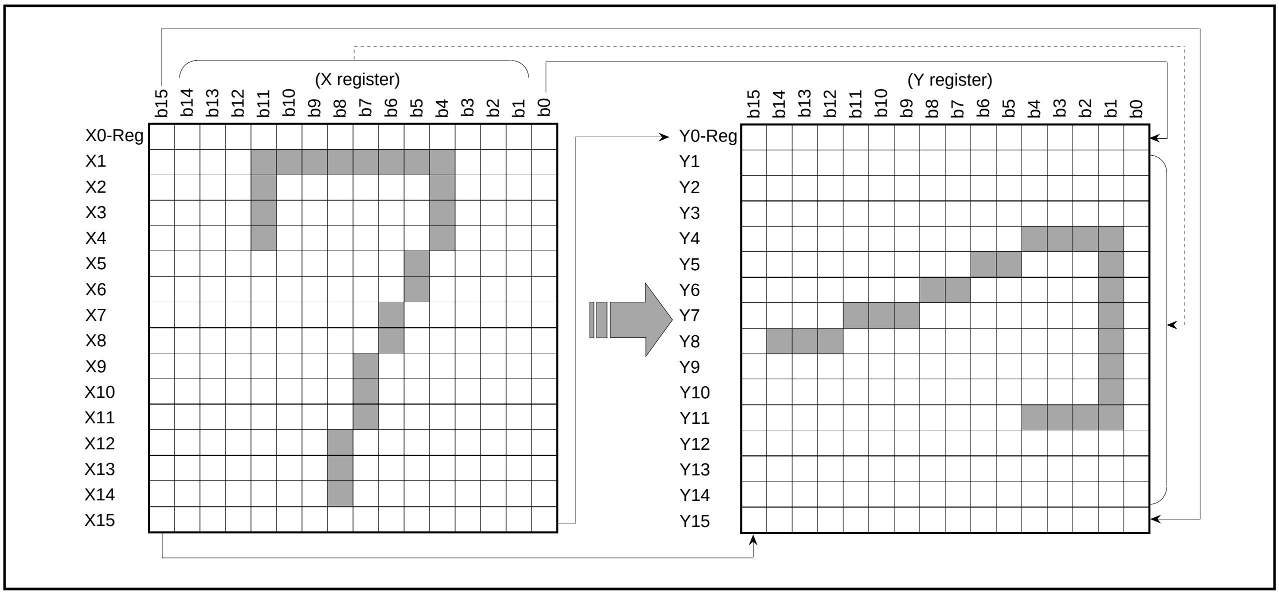


Figure 1.24.3. X-Y conversion example

When the read-mode set bit (bit 0 at address $02E0_{16}$) is “1”, you can read the value written to the Xi register by reading the Yi register. Figure 1.24.4 shows the conversion table when the read mode set bit = “1”.

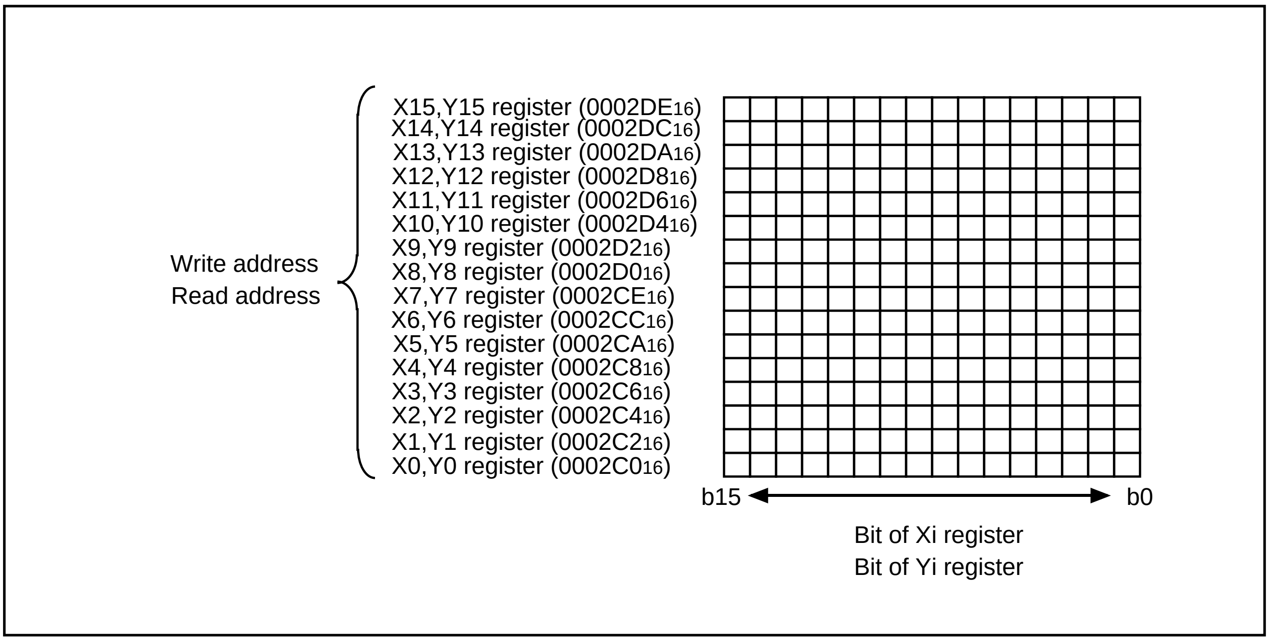

Write address
Read address
X15,Y15 register (0002DE16)
X14,Y14 register (0002DC16)
X13,Y13 register (0002DA16)
X12,Y12 register (0002D816)
X11,Y11 register (0002D616)
X10,Y10 register (0002D416)
X9,Y9 register (0002D216)
X8,Y8 register (0002D016)
X7,Y7 register (0002CE16)
X6,Y6 register (0002CC16)
X5,Y5 register (0002CA16)
X4,Y4 register (0002C816)
X3,Y3 register (0002C616)
X2,Y2 register (0002C416)
X1,Y1 register (0002C216)
X0,Y0 register (0002C016)
b15
b0
Bit of Xi register
Bit of Yi register

Figure 1.24.4. Conversion table when the read mode set bit = “1”

The value written to the Xi register is controlled by the write mode set bit (bit 1 at address $02E0_{16}$).
When the write mode set bit (bit 1 at address $02E0_{16}$) is “0” and data is written to the Xi register, the bit stream is written directly.
When the write mode set bit (bit 1 at address $02E0_{16}$) is “1” and data is written to the Xi register, the bit sequence is reversed so that the high becomes low and vice versa. Figure 1.24.5 shows the conversion table when the write mode set bit = “1”.

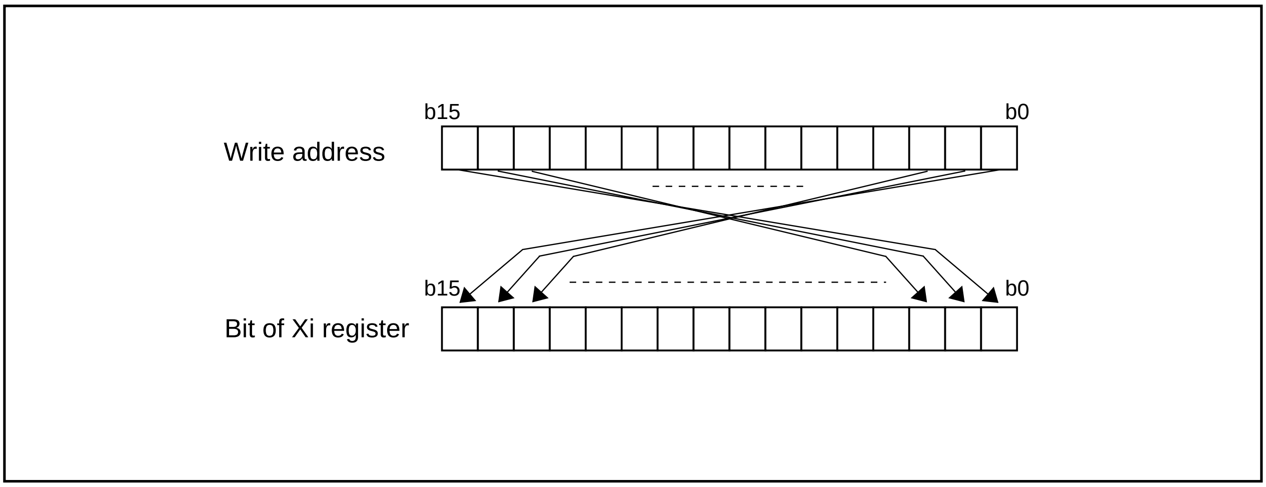


Figure 1.24.5. Conversion table when the write mode set bit = “1”

DRAM Controller

There is a built in DRAM controller to which it is possible to connect between 512 Kbytes and 8 Mbytes of DRAM. Table 1.25.1 shows the functions of the DRAM controller.

Table 1.25.1. DRAM Controller Functions

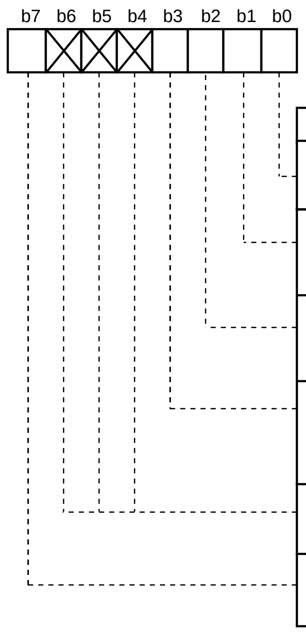
DRAM space	512KB, 1MB, 2MB, 4MB, 8MB
Bus control	2CAS/1W
Refresh	$\overline{\text{CAS}}$ before $\overline{\text{RAS}}$ refresh Self refresh-compatible
Function modes	EDO-compatible, fast page mode-compatible
Waits	1 wait or 2 waits, programmable

To use the DRAM controller, use the DRAM space select bit of the DRAM control register (address 0040$_{16}$) to specify the DRAM size. Figure 1.25.1 shows the DRAM control register.

The DRAM controller cannot be used in external memory mode 3 (bits 1 and 2 at address 0005$_{16}$ are “11$_2$”). Always use the DRAM controller in external memory modes 0, 1, or 2.

When the data bus width is 16-bit in DRAM area, set "1" to R/W mode select bit (bit 2 at address 0004$_{16}$).

Set wait time between after DRAM power ON and before memory processing, and processing necessary for dummy cycle to refresh DRAM by software.



DRAM control register

Symbol	Address	When reset
DRAMCONT	0040$_{16}$	Indeterminate (Note 4)

Bit symbol	Bit name	Function	R	W
WT	Wait select bit (Note 1)	0 : Two wait 1 : One wait	○	○
AR0	DRAM space select bit	b3 b2 b1 0 0 0 : DRAM ignored 0 0 1 : Inhibit 0 1 0 : 0.5MB 0 1 1 : 1MB 1 0 0 : 2MB 1 0 1 : 4MB 1 1 0 : 8MB 1 1 1 : Inhibit	○	○
AR1			○	○
AR2			○	○
Nothing is assigned. When write, set "0". When read, the value of these bits is indeterminate.			–	–
SREF	Self-refresh mode bit (Note 2)	0: Self-refresh OFF 1: Self-refresh ON	○	○

Note 1: The number of cycles with 2 waits is 3-2-2. With 1 wait, it is 2-1-1.

Note 2: When you set "1", both $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ change to "L". When you set "0", $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ change to "H" and then normal operation (read/write, refresh) is resumed. In Stop mode, there is no control.

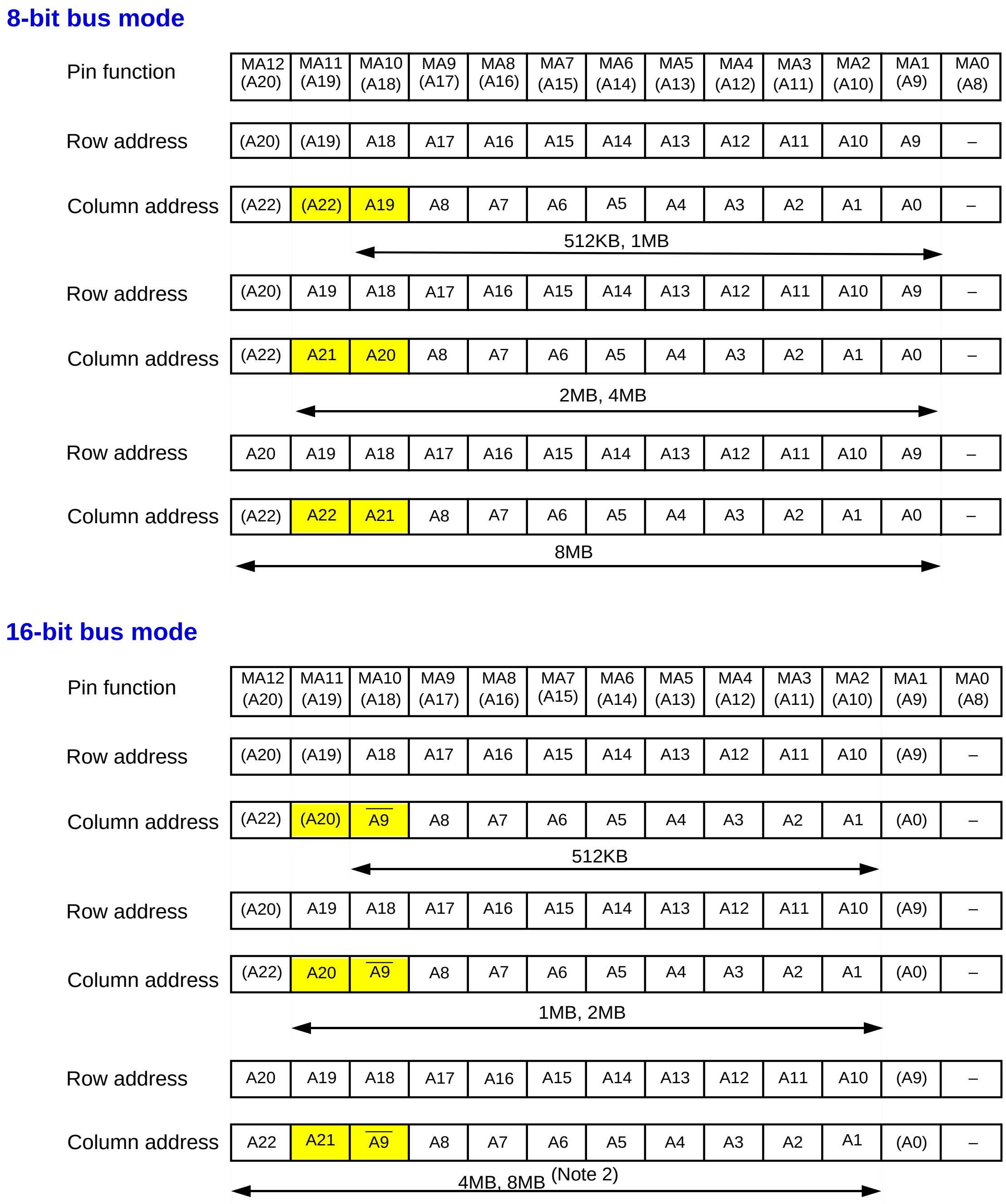
Note 3: Set the bus width using the external data bus width control register (address 000B$_{16}$). When selecting 8-bit bus width, $\overline{\text{CASH}}$ is indeterminate.

Note 4: After reset, the content of this register is indeterminate. DRAM controller starts the operation after writing to this register.

Figure 1.25.1. DRAM control register

• DRAM Controller Multiplex Address Output

The DRAM controller outputs the row addresses and column addresses as a multiplexed signal to the address bus A8 to A20. Figure 1.25.2 shows the output format for multiplexed addresses.



8-bit bus mode

Pin function	MA12 (A20)	MA11 (A19)	MA10 (A18)	MA9 (A17)	MA8 (A16)	MA7 (A15)	MA6 (A14)	MA5 (A13)	MA4 (A12)	MA3 (A11)	MA2 (A10)	MA1 (A9)	MA0 (A8)
Row address	(A20)	(A19)	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	–
Column address	(A22)	(A22)	A19	A8	A7	A6	A5	A4	A3	A2	A1	A0	–
			512KB, 1MB										
Row address	(A20)	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	–
Column address	(A22)	A21	A20	A8	A7	A6	A5	A4	A3	A2	A1	A0	–
		2MB, 4MB											
Row address	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	–
Column address	(A22)	A22	A21	A8	A7	A6	A5	A4	A3	A2	A1	A0	–
	8MB												

16-bit bus mode

Pin function	MA12 (A20)	MA11 (A19)	MA10 (A18)	MA9 (A17)	MA8 (A16)	MA7 (A15)	MA6 (A14)	MA5 (A13)	MA4 (A12)	MA3 (A11)	MA2 (A10)	MA1 (A9)	MA0 (A8)
Row address	(A20)	(A19)	A18	A17	A16	A15	A14	A13	A12	A11	A10	(A9)	–
Column address	(A22)	(A20)	$\overline{A9}$	A8	A7	A6	A5	A4	A3	A2	A1	(A0)	–
			512KB										
Row address	(A20)	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	(A9)	–
Column address	(A22)	A20	$\overline{A9}$	A8	A7	A6	A5	A4	A3	A2	A1	(A0)	–
		1MB, 2MB											
Row address	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	(A9)	–
Column address	A22	A21	$\overline{A9}$	A8	A7	A6	A5	A4	A3	A2	A1	(A0)	–
	4MB, 8MB (Note 2)												

Note 1: ( ) invalid bit: 🟨 bits that change according to selected mode (8-bit/16-bit bus mode, DRAM space).

Note 2: The figure is for 4Mx1 or 4Mx4 memory configuration. If you are using a 4Mx16 configuration, use combinations of the following: For row addresses, MA0 to MA12; for column addresses MA2 to MA8, MA11, and MA12. Or for row addresses MA1 to MA12; for column addresses MA2 to MA9, MA11, MA12.

Note 3: "–" is indeterminate.

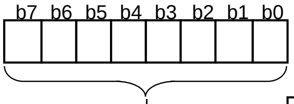
Figure 1.25.2. Output format for multiplexed addresses

• Refresh

The refresh method is $\overline{\text{CAS}}$ before $\overline{\text{RAS}}$. The refresh interval is set by the DRAM refresh interval set register (address 0041_{16}). The refresh signal is not output in HOLD state. Figure 1.25.3 shows the DRAM refresh interval set register.
Use the following formula to determine the value to set in the refresh interval set register.

Refresh interval set register value (0 to 255) = refresh interval time / (BCLK frequency X 32) - 1

DRAM refresh interval set register



Symbol REFCNT — Address 0041_{16} — When reset Indeterminate

Bit symbol	Bit name	Function	R	W
REFCNT0	Refresh interval set bit	b7 b6 b5 b4 b3 b2 b1 b0 0 0 0 0 0 0 0 0 : 1.6 µs 0 0 0 0 0 0 0 1 : 3.2 µs 0 0 0 0 0 0 1 0 : 4.8 µs • • • 1 1 1 1 1 1 1 1 : 409.6 µs (Note)	○	○
REFCNT1			○	○
REFCNT2			○	○
REFCNT3			○	○
REFCNT4			○	○
REFCNT5			○	○
REFCNT6			○	○
REFCNT7			○	○

Note: Refresh interval at 20 MHz operating (no division)
Refresh interval = BCLK frequency X (refresh interval set bit + 1) X 32

Figure 1.25.3. DRAM refresh interval set register

The DRAM self-refresh operates in STOP mode, etc.
When shifting to self-refresh, select DRAM ignored by the DRAM space select bit. In the next instruction, simultaneously set the DRAM space select bit and self-refresh ON by self-refresh mode bit. Also, insert two NOPs after the instruction that sets the self-refresh mode bit to "1".
Do not access external memory while operating in self-refresh. (All external memory space access is inhibited.)
When disabling self-refresh, simultaneously select DRAM ignored by the DRAM space select bit and self-refresh OFF by self-refresh mode bit. In the next instruction, set the DRAM space select bit.
Do not access the DRAM space immediately after setting the DRAM space select bit.

Example) One wait is selected by the wait select bit and 4MB is selected by the DRAM space select bit

Shifting to self-refresh

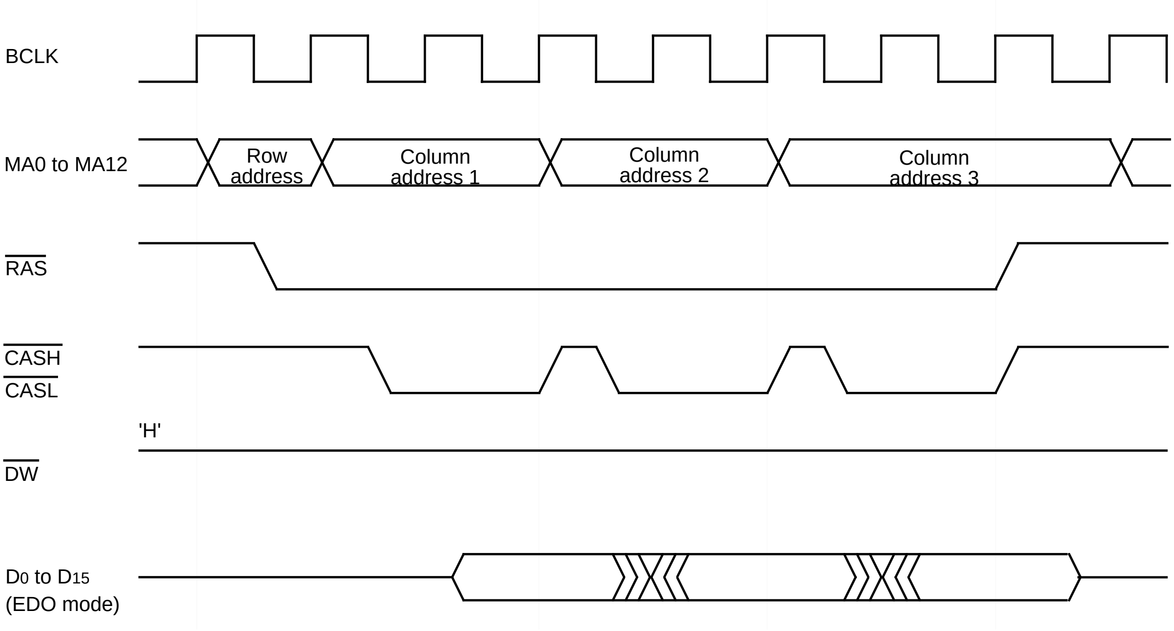
```
•••
mov.b   #00000001b,DRAMCONT    ;DRAM ignored, one wait is selected
mov.b   #10001011b,DRAMCONT    ;Set self-refresh, select 4MB and one wait
nop                            ;Two nops are needed
nop                            ;
•••
```

Disable self-refresh

```
•••
mov.b   #00000001b,DRAMCONT    ;Disable self-refresh, DRAM ignored, one wait is
                               ;selected
mov.b   #00001011b,DRAMCONT    ;Select 4MB and one wait
nop                            ;Inhibit instruction to access DRAM area
nop
•••
```

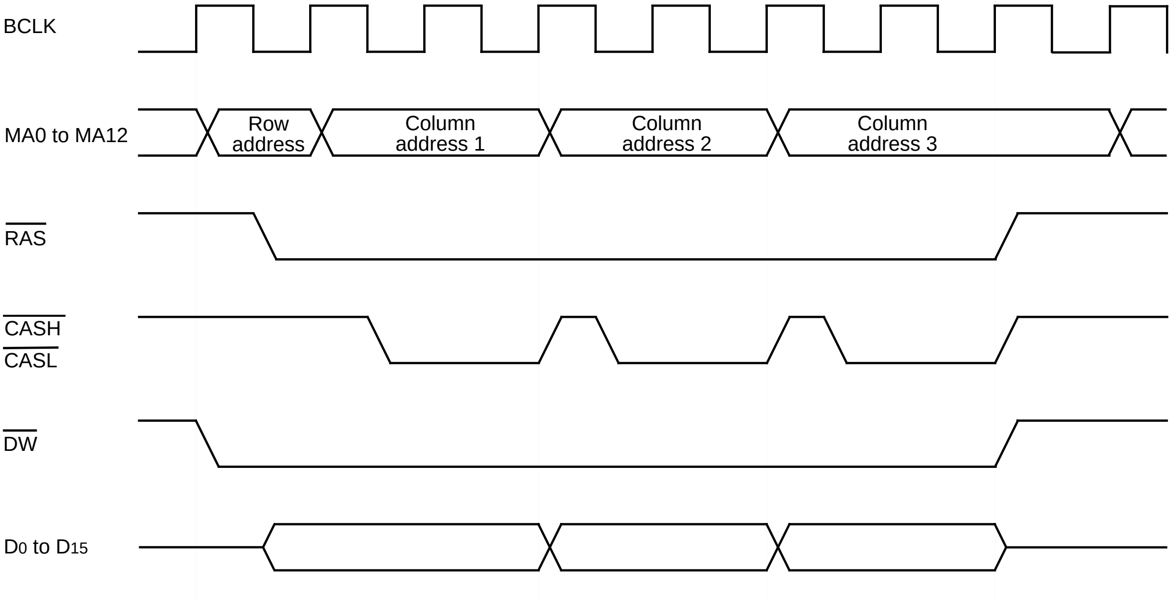
Figures 1.25.4 to 1.25.6 show the bus timing during DRAM access.

< Read cycle (wait control bit = 0) >



Note : Only CASL is operating in 8-bit data bus width.

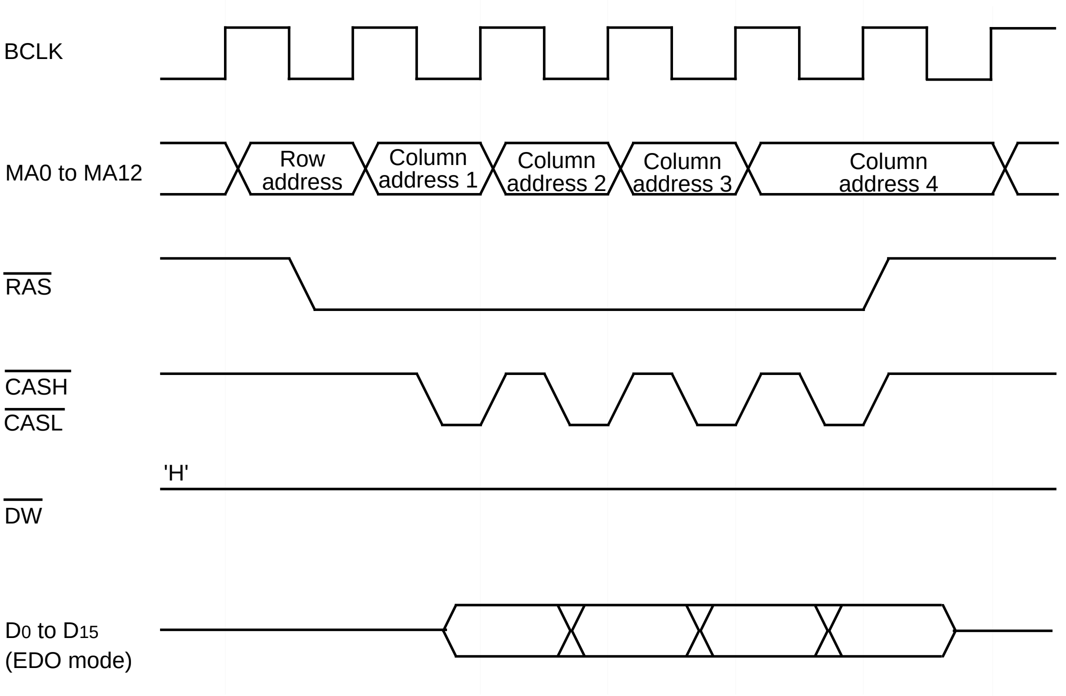
< Write cycle (wait control bit = 0) >



Note : Only CASL is operating in 8-bit data bus width.

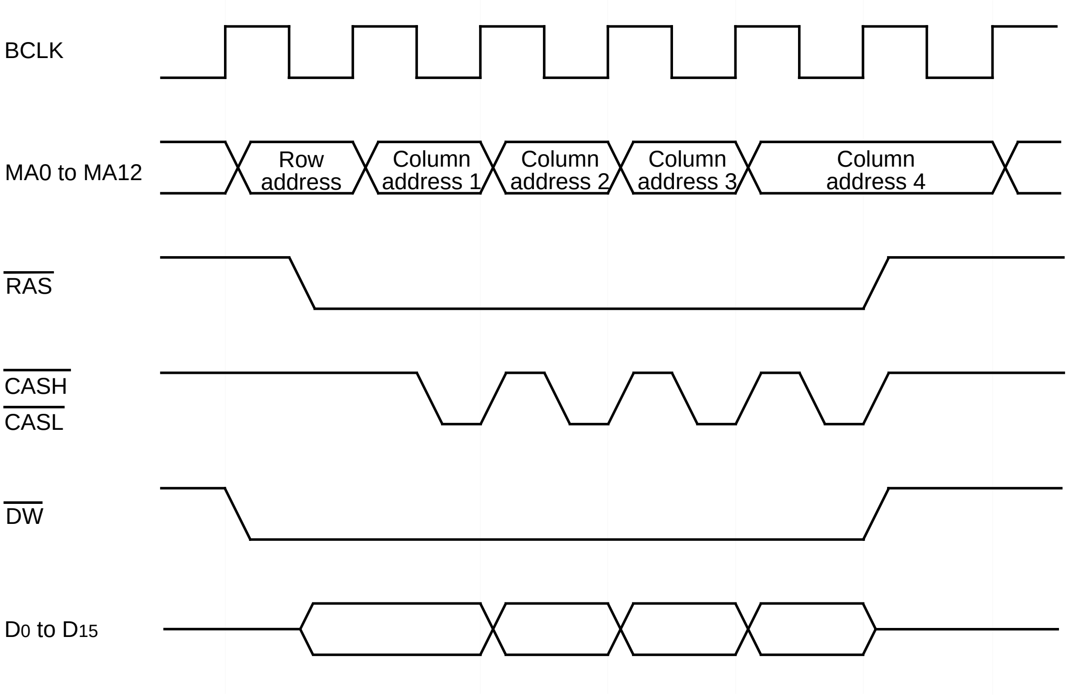
Figure 1.25.4. The bus timing during DRAM access (1)

< Read cycle (wait control bit = 1) >



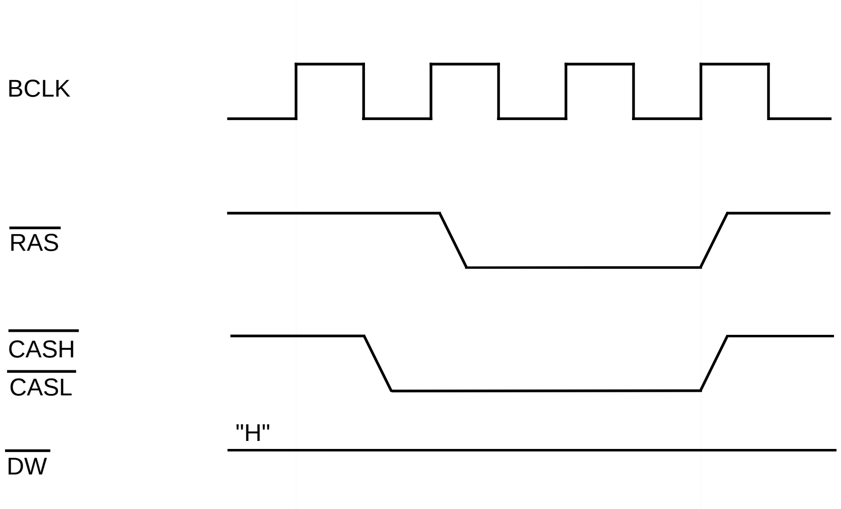
Note : Only $\overline{\text{CASL}}$ is operating in 8-bit data bus width.

< Write cycle (wait control bit = 1) >



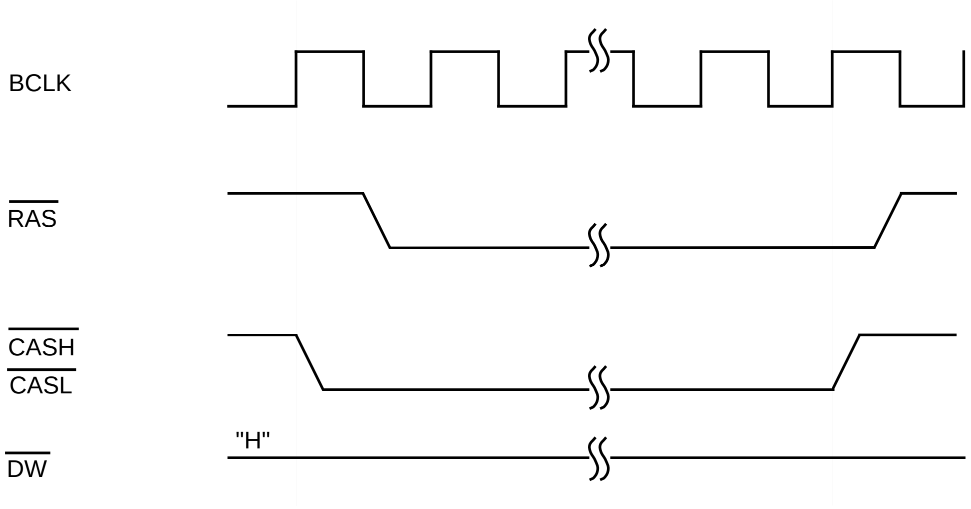
Note : Only $\overline{\text{CASL}}$ is operating in 8-bit data bus width.

Figure 1.25.5. The bus timing during DRAM access (2)



< $\overline{\text{CAS}}$ before $\overline{\text{RAS}}$ refresh cycle >

Note : Only $\overline{\text{CASL}}$ is operating in 8-bit data bus width.



< Self refresh cycle >

Note : Only $\overline{\text{CASL}}$ is operating in 8-bit data bus width.

Figure 1.25.6. The bus timing during DRAM access (3)

Programmable I/O Ports

There are 87 programmable I/O ports for 100-pin version: P0 to P10 (excluding P8$_5$). There are 123 programmable I/O ports for 144-pin version: P0 to P15 (excluding P8$_5$). Each port can be set independently for input or output using the direction register. A pull-up resistance for each block of 4 ports can be set. P8$_5$ is an input-only port and has no built-in pull-up resistance.
Figures 1.26.1 to 1.26.3 show the programmable I/O ports.
Each pin functions as a programmable I/O port and as the I/O for the built-in peripheral devices.
To use the pins as the inputs for the built-in peripheral devices, set the direction register of each pin to input mode. When the pins are used as the outputs for the built-in peripheral devices (other than the D-A converter), set the corresponding function select registers A, B and C. When pins are to be used as the outputs for the D-A converter, set the function select register of each pin to I/O port, and set the direction registers to input mode.
Table 1.26.1 lists each port and peripheral function.
See the descriptions of the respective functions for how to set up the built-in peripheral devices.

(1) Direction registers

Figures 1.26.4 and 1.26.5 show the direction registers.
These registers are used to choose the direction of the programmable I/O ports. Each bit in these registers corresponds one for one to each I/O pin.
In memory expansion and microprocessor mode, the contents of corresponding direction register of pins A$_0$ to A$_{22}$, $\overline{\text{A}_{23}}$, D$_0$ to D$_{15}$, MA$_0$ to MA$_{12}$, $\overline{\text{CS0}}$ to $\overline{\text{CS 3}}$, $\overline{\text{WRL}}$/$\overline{\text{WR}}$/$\overline{\text{CASL}}$, $\overline{\text{WRH}}$/$\overline{\text{BHE}}$/$\overline{\text{CASH}}$, $\overline{\text{RD}}$/$\overline{\text{DW}}$, BCLK/ALE/CLK$_{\text{OUT}}$, $\overline{\text{HLDA}}$/ALE, $\overline{\text{HOLD}}$, ALE/$\overline{\text{RAS}}$, and $\overline{\text{RDY}}$ are not changed.
Note: There is no direction register bit for P8$_5$.

(2) Port registers

Figures 1.26.6 and 1.26.7 show the port registers.
These registers are used to write and read data for input and output to and from an external device. A port register consists of a port latch to hold output data and a circuit to read the status of a pin. Each bit in port registers corresponds one for one to each I/O pin.
In memory expansion and microprocessor mode, the contents of corresponding port register of pins A$_0$ to A$_{22}$, $\overline{\text{A}_{23}}$, D$_0$ to D$_{15}$, MA$_0$ to MA$_{12}$, $\overline{\text{CS0}}$ to $\overline{\text{CS 3}}$, $\overline{\text{WRL}}$/$\overline{\text{WR}}$/$\overline{\text{CASL}}$, $\overline{\text{WRH}}$/$\overline{\text{BHE}}$/$\overline{\text{CASH}}$, $\overline{\text{RD}}$/$\overline{\text{DW}}$, BCLK/ALE/CLK$_{\text{OUT}}$, $\overline{\text{HLDA}}$/ALE, $\overline{\text{HOLD}}$, ALE/$\overline{\text{RAS}}$, and $\overline{\text{RDY}}$ are not changed.

(3) Function select register A

Figures 1.26.8 and 1.26.9 show the function select registers A.
The register is used to select port output and peripheral function output when the port functions for both port output and peripheral function output.
Each bit of this register corresponds to each pin that functions for both port output and peripheral function output.

(4) Function select register B

Figures 1.26.10 and 1.26.11 show the function select registers B.

This register selects the 1st peripheral function output and second peripheral function output when multiple peripheral function outputs are assigned to a pin. For pins with a third peripheral function, this register selects whether to enable the function select register C, or output the second peripheral function.

Each bit of this register corresponds to each pin that has multiple peripheral function outputs assigned to it.

This register is enabled when the bits of the corresponding function select register A are set for peripheral functions.

The bit 3 to bit 6 of function select register B3 is ignored bit for input peripheral function. When using DA0/DA1 and ANEX0/ANEX1, set related bit to "1". When not using DA0/DA1 or ANEX0/ANEX1, set related bit to "0".

(5) Function select register C

Figure 1.26.12 shows the function select register C.

This register is used to select the first peripheral function output and the third peripheral function output when three peripheral function outputs are assigned to a pin.

This register is effective when the bits of the function select register A of the counterpart pin have selected a peripheral function and when the function select register B has made effective the function select register C.

The bit 7 (PSC_7) is assigned the key-in interrupt inhibit bit. Setting 1 in the key-in interrupt inhibit bit causes no key-in interrupts regardless of the settings in the interrupt control register even if L is entered in pins KI_0 to KI_3. With 1 set in the key-in interrupt inhibit bit, input from a port pin cannot be effected even if the port direction register is set to input mode.

(6) Pull-up control registers

Figures 1.26.13 and 1.26.14 show the pull-up control registers.

The pull-up control register can be set to apply a pull-up resistance to each block of 4 ports. When ports are set to have a pull-up resistance, the pull-up resistance is connected only when the direction register is set for input.

Since P0 to P5 operate as the bus in memory expansion mode and microprocessor mode, do not set the pull-up control register. However, it is possible to select pull-up resistance presence to the usable port as I/O port by setting.

(7) Port control register

Figure 1.26.15 shows the port control register.

This register is used to choose whether to make port P1 a CMOS port or an Nch open drain. In the Nch open drain, the port P1 has no function that a complete open drain but keeps the CMOS port’s Pch always turned off. Thus the absolute maximum rating of the input voltage falls within the range from - 0.3 V to Vcc + 0.3 V.

The port control register functions similarly to the above also in the case in which port P1 can be used as a port when the bus width in the full external areas comprises 8 bits in either microprocessor mode or in memory expansion mode.

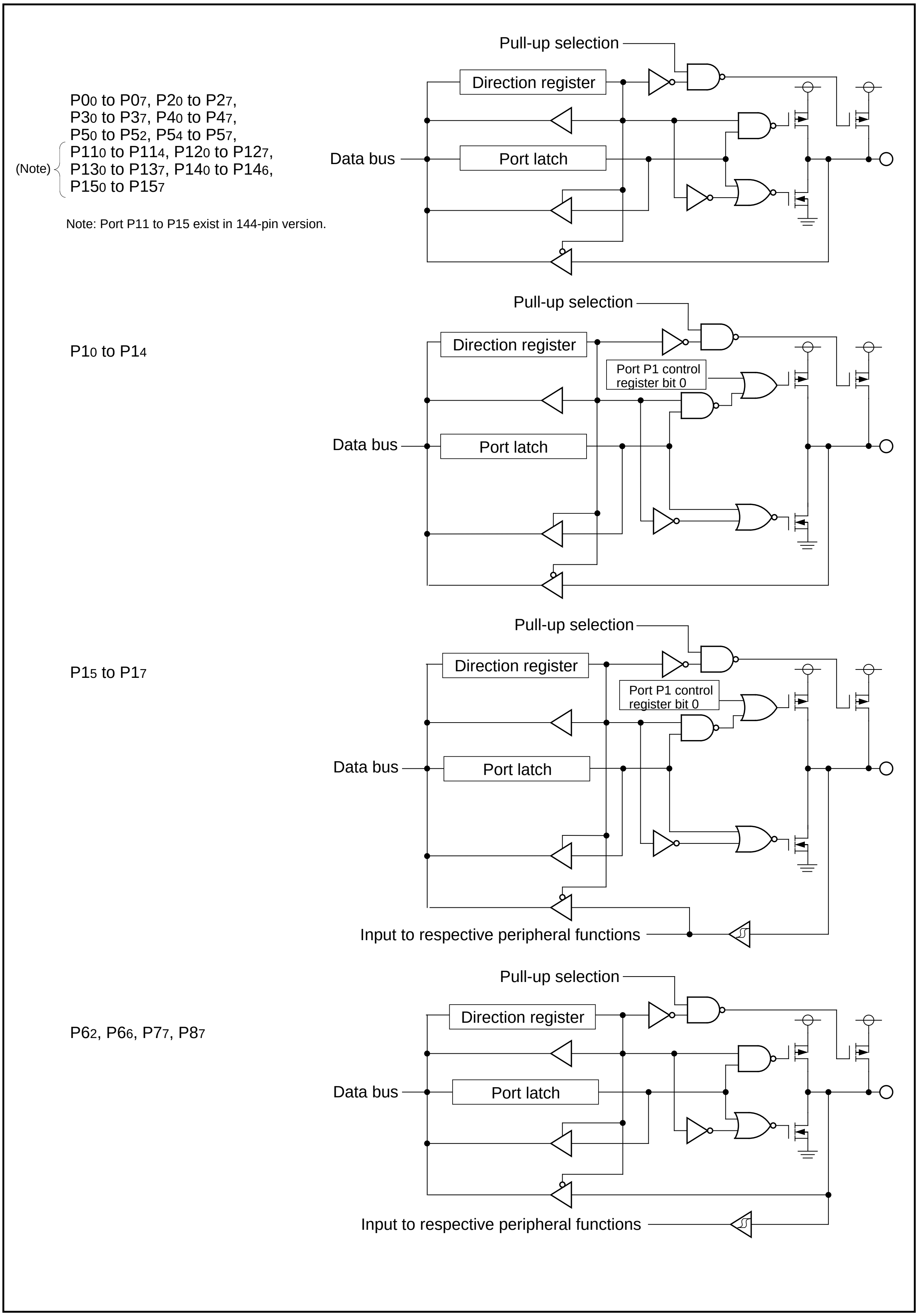

P0$_0$ to P0$_7$, P2$_0$ to P2$_7$,
P3$_0$ to P3$_7$, P4$_0$ to P4$_7$,
P5$_0$ to P5$_2$, P5$_4$ to P5$_7$,
(Note)
P11$_0$ to P11$_4$, P12$_0$ to P12$_7$,
P13$_0$ to P13$_7$, P14$_0$ to P14$_6$,
P15$_0$ to P15$_7$
Note: Port P11 to P15 exist in 144-pin version.
Pull-up selection
Direction register
Data bus
Port latch
P1$_0$ to P1$_4$
Pull-up selection
Direction register
Port P1 control register bit 0
Data bus
Port latch
P1$_5$ to P1$_7$
Pull-up selection
Direction register
Port P1 control register bit 0
Data bus
Port latch
Input to respective peripheral functions
P6$_2$, P6$_6$, P7$_7$, P8$_7$
Pull-up selection
Direction register
Data bus
Port latch
Input to respective peripheral functions

Figure 1.26.1. Programmable I/O ports (1)

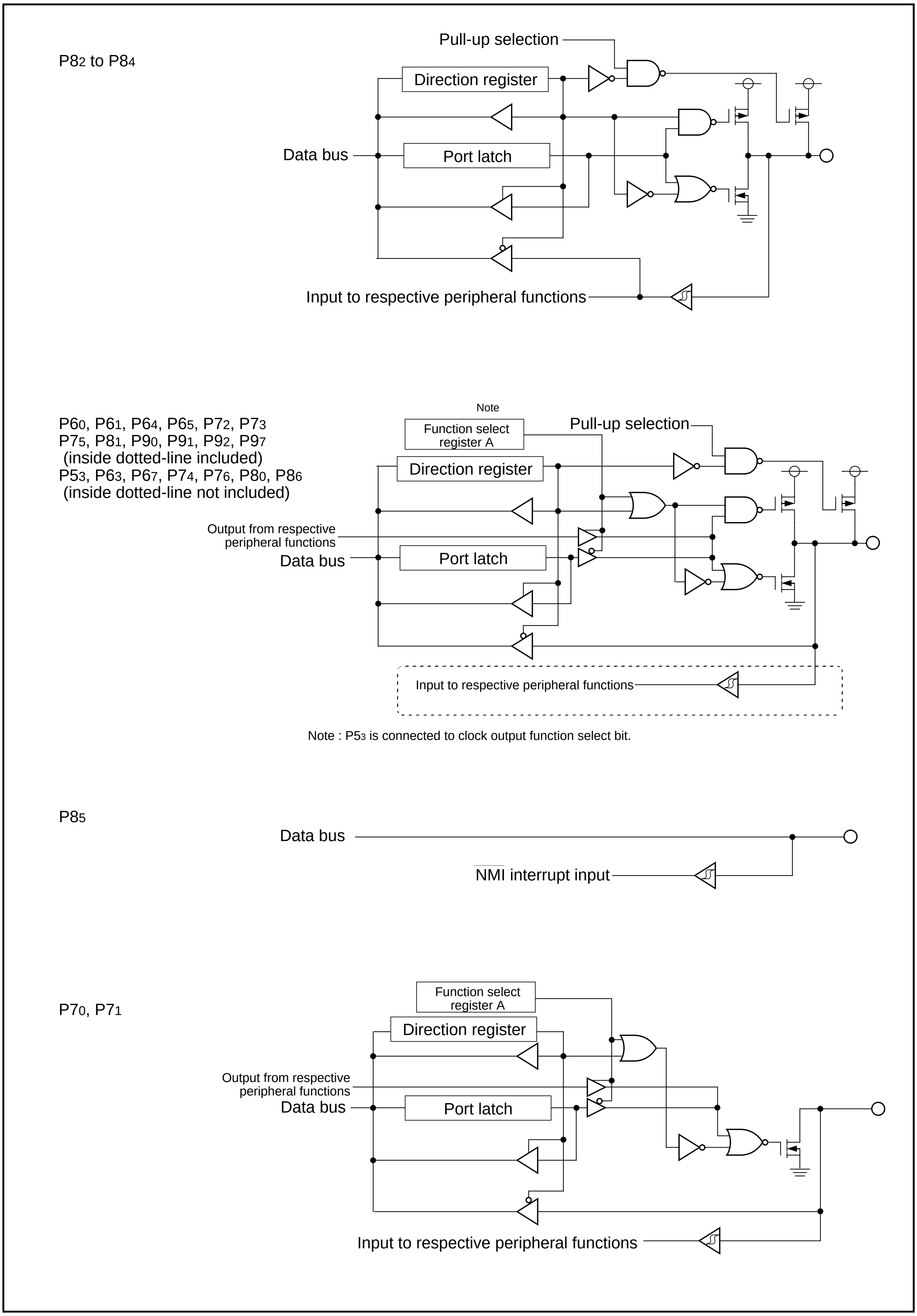

$P8_2$ to $P8_4$
Pull-up selection
Direction register
Data bus
Port latch
Input to respective peripheral functions
$P6_0$, $P6_1$, $P6_4$, $P6_5$, $P7_2$, $P7_3$
$P7_5$, $P8_1$, $P9_0$, $P9_1$, $P9_2$, $P9_7$
(inside dotted-line included)
$P5_3$, $P6_3$, $P6_7$, $P7_4$, $P7_6$, $P8_0$, $P8_6$
(inside dotted-line not included)
Note
Function select register A
Pull-up selection
Direction register
Output from respective peripheral functions
Data bus
Port latch
Input to respective peripheral functions
Note : $P5_3$ is connected to clock output function select bit.
$P8_5$
Data bus
$\overline{\text{NMI}}$ interrupt input
$P7_0$, $P7_1$
Function select register A
Direction register
Output from respective peripheral functions
Data bus
Port latch
Input to respective peripheral functions

Figure 1.26.2. Programmable I/O ports (2)

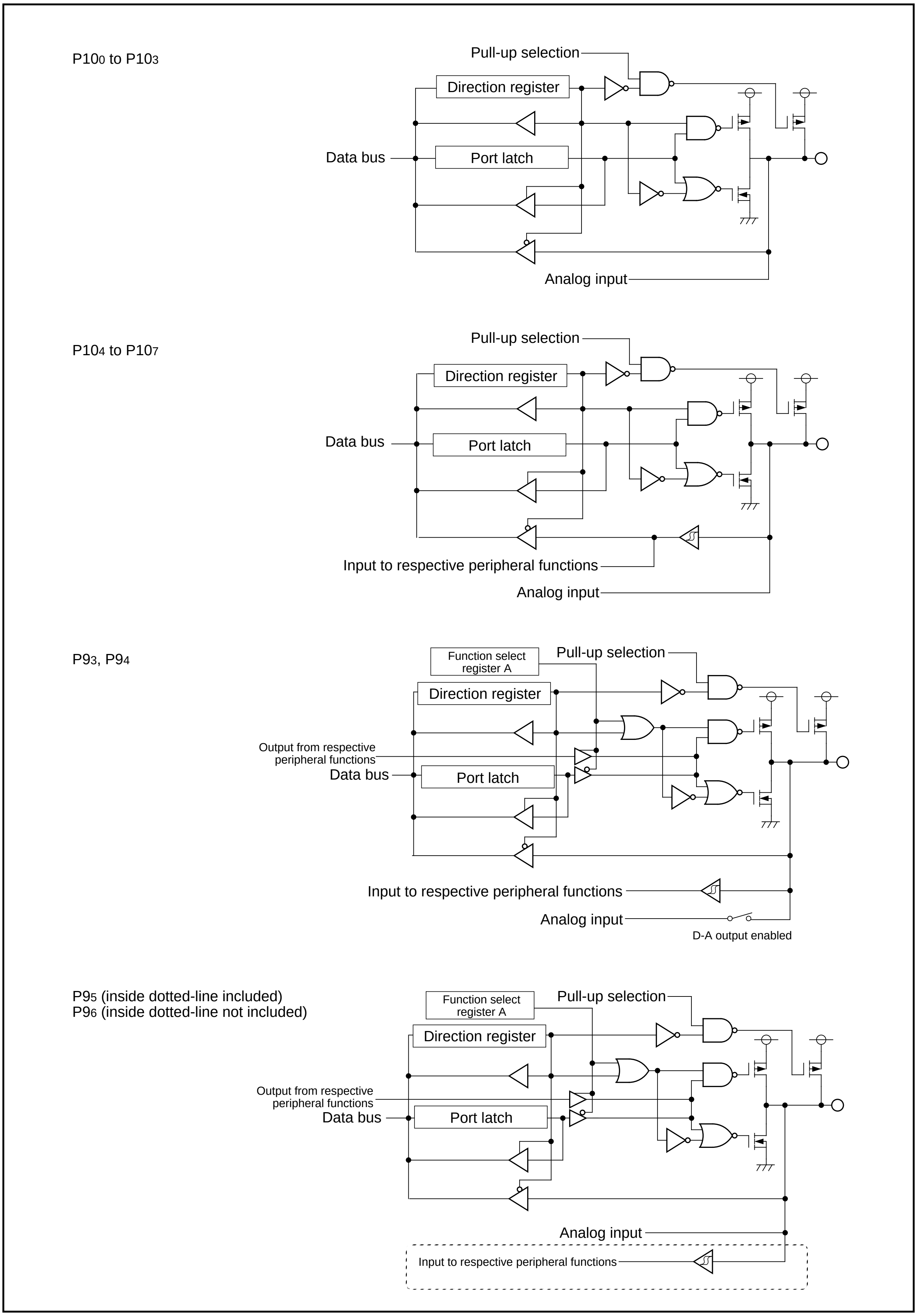
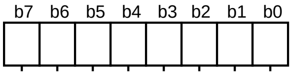

P10$_0$ to P10$_3$
Pull-up selection
Direction register
Data bus
Port latch
Analog input
P10$_4$ to P10$_7$
Pull-up selection
Direction register
Data bus
Port latch
Input to respective peripheral functions
Analog input
P9$_3$, P9$_4$
Function select register A
Pull-up selection
Direction register
Output from respective peripheral functions
Data bus
Port latch
Input to respective peripheral functions
Analog input
D-A output enabled
P9$_5$ (inside dotted-line included)
P9$_6$ (inside dotted-line not included)
Function select register A
Pull-up selection
Direction register
Output from respective peripheral functions
Data bus
Port latch
Analog input
Input to respective peripheral functions

Figure 1.26.3. Programmable I/O ports (3)

Port Pi direction register (Note 1,2, 3)



Symbol	Address	When reset
PDi (i = 0 to 15, except 8, 11 and 14)	$03E2_{16}$, $03E3_{16}$, $03E6_{16}$, $03E7_{16}$, $03EA_{16}$, $03EB_{16}$, $03C2_{16}$, $03C3_{16}$, $03C7_{16}$, $03CA_{16}$, $03CE_{16}$, $03CF_{16}$, $03D3_{16}$	00_{16}

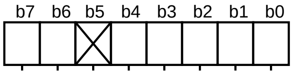
Bit symbol	Bit name	Function	R	W
PDi_0	Port Pi$_0$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port) (i = 0 to 15 except 8, 11 and 14)	○	○
PDi_1	Port Pi$_1$ direction register		○	○
PDi_2	Port Pi$_2$ direction register		○	○
PDi_3	Port Pi$_3$ direction register		○	○
PDi_4	Port Pi$_4$ direction register		○	○
PDi_5	Port Pi$_5$ direction register		○	○
PDi_6	Port Pi$_6$ direction register		○	○
PDi_7	Port Pi$_7$ direction register		○	○

Note 1: Set bit 2 of protect register (address $000A_{16}$) to “1” before rewriting to the port P9 direction register.

Note 2: In memory expansion and microprocessor mode, the contents of corresponding port direction register of pins A_0 to A_{22}, $\overline{A_{23}}$, D_0 to D_{15}, MA_0 to MA_{12}, $\overline{CS0}$ to $\overline{CS}$ 3, $\overline{WRL}/\overline{WR}/\overline{CASL}$, $\overline{WRH}/\overline{BHE}/\overline{CASH}$, $\overline{RD}/\overline{DW}$, BCLK/ALE/CLKOUT, $\overline{HLDA}$/ALE, $\overline{HOLD}$, ALE/$\overline{RAS}$, and $\overline{RDY}$ are not changed.

Note 3: Port P12, P13 and P15 direction registers exist in 144-pin version.

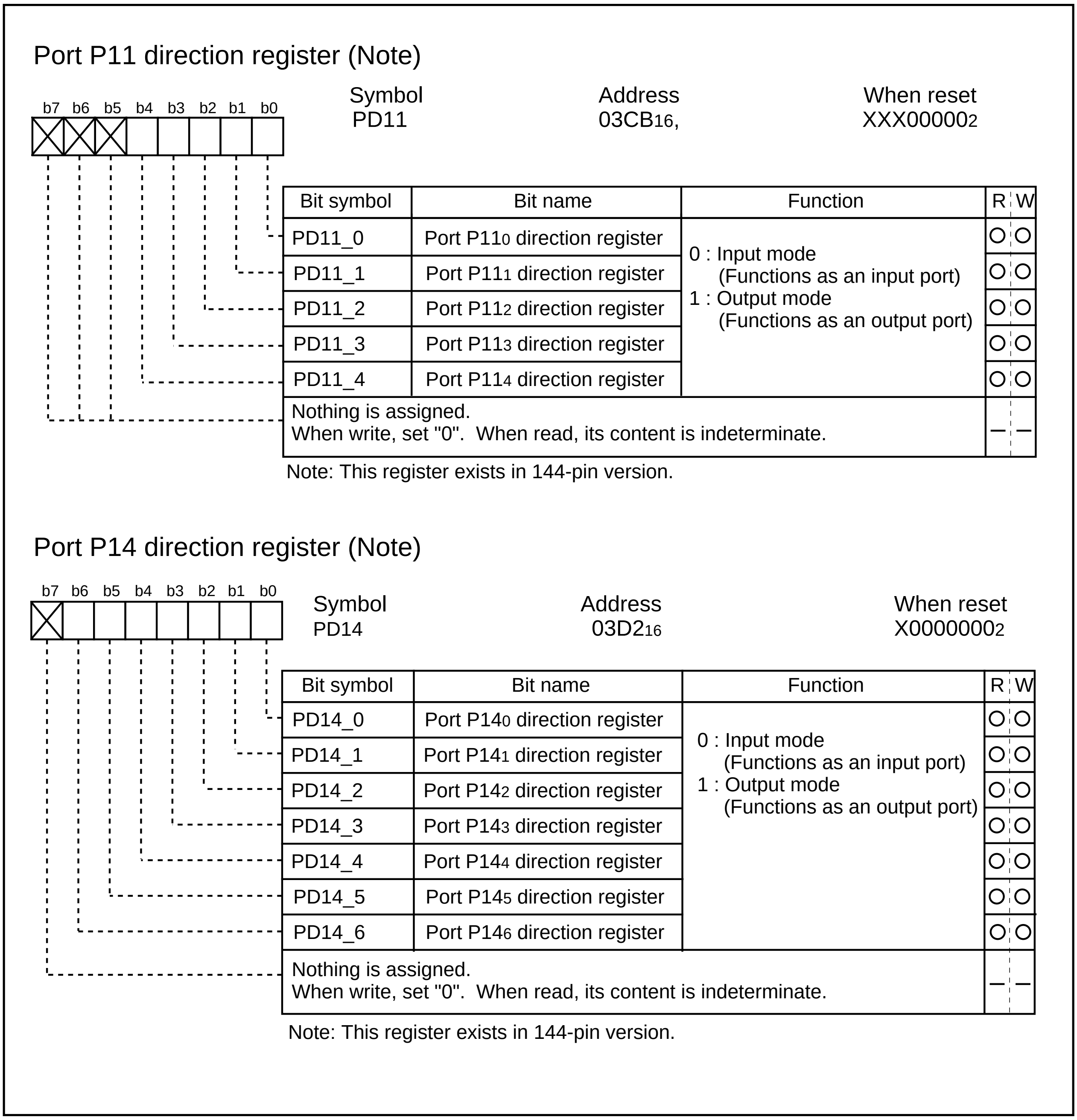
Port P8 direction register



Symbol	Address	When reset
PD8	$03C6_{16}$	$00X00000_2$

Bit symbol	Bit name	Function	R	W
PD8_0	Port P8$_0$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	○	○
PD8_1	Port P8$_1$ direction register		○	○
PD8_2	Port P8$_2$ direction register		○	○
PD8_3	Port P8$_3$ direction register		○	○
PD8_4	Port P8$_4$ direction register		○	○
Nothing is assigned. When write, set "0". When read, its content is indeterminate.			—	—
PD8_6	Port P8$_6$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	○	○
PD8_7	Port P8$_7$ direction register		○	○

Figure 1.26.4. Direction register (1)



Port P11 direction register (Note)

Symbol	Address	When reset
PD11	03CB$_{16}$,	XXX00000$_2$

Bit symbol	Bit name	Function	R	W
PD11_0	Port P11$_0$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	○	○
PD11_1	Port P11$_1$ direction register		○	○
PD11_2	Port P11$_2$ direction register		○	○
PD11_3	Port P11$_3$ direction register		○	○
PD11_4	Port P11$_4$ direction register		○	○
Nothing is assigned. When write, set "0". When read, its content is indeterminate.			—	—

Note: This register exists in 144-pin version.

Port P14 direction register (Note)

Symbol	Address	When reset
PD14	03D2$_{16}$	X0000000$_2$

Bit symbol	Bit name	Function	R	W
PD14_0	Port P14$_0$ direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	○	○
PD14_1	Port P14$_1$ direction register		○	○
PD14_2	Port P14$_2$ direction register		○	○
PD14_3	Port P14$_3$ direction register		○	○
PD14_4	Port P14$_4$ direction register		○	○
PD14_5	Port P14$_5$ direction register		○	○
PD14_6	Port P14$_6$ direction register		○	○
Nothing is assigned. When write, set "0". When read, its content is indeterminate.			—	—

Note: This register exists in 144-pin version.

Figure 1.26.5. Direction register (2)

Port Pi register (Note 1, 3)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
Pi (i = 0 to 15, except 8, 11 and 14)	$03E0_{16}$, $03E1_{16}$, $03E4_{16}$, $03E5_{16}$, $03E8_{16}$, $03E9_{16}$, $03C0_{16}$, $03C1_{16}$, $03C5_{16}$, $03C8_{16}$, $03CC_{16}$, $03CD_{16}$, $03D1_{16}$	Indeterminate

Bit symbol	Bit name	Function	R	W
PDi_0	Port Pi_0 register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit 0 : “L” level data 1 : “H” level data (Note 2) (i = 0 to 15 except 8, 11 and 14)	○	○
PDi_1	Port Pi_1 register		○	○
PDi_2	Port Pi_2 register		○	○
PDi_3	Port Pi_3 register		○	○
PDi_4	Port Pi_4 register		○	○
PDi_5	Port Pi_5 register		○ ○	○ ○
PDi_6	Port Pi_6 register		○	○
PDi_7	Port Pi_7 register		○	○

Note 1: In memory expansion and microprocessor mode, the contents of corresponding port Pi direction register of pins A_0 to A_{22}, $\overline{A_{23}}$, D_0 to D_{15}, MA_0 to MA_{12}, $\overline{CS0}$ to $\overline{CS}$ 3, $\overline{WRL}/\overline{WR}/\overline{CASL}$, $\overline{WRH}/\overline{BHE}/\overline{CASH}$, $\overline{RD}/\overline{DW}$, BCLK/ALE/$CLK_{OUT}$, $\overline{HLDA}$/ALE, $\overline{HOLD}$, ALE/$\overline{RAS}$, and $\overline{RDY}$ are not changed.
Note 2: $P7_0$ and $P7_1$ are N-channel open drain ports and high inpedance outputs.
Note 3: Port P12, P13 and P15 registers exist in 144-pin version.

Port P8 register

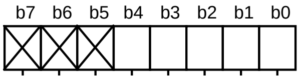
b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
P8	$03C4_{16}$	Indeterminate

Bit symbol	Bit name	Function	R	W
PD8_0	Port P80 register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit (except for $P8_5$) 0 : “L” level data 1 : “H” level data	○	○
PD8_1	Port P81 register		○	○
PD8_2	Port P82 register		○	○
PD8_3	Port P83 register		○ ○	○ ○
PD8_4	Port P84 register		○	○
PD8_5	Port P85 register		○	×
PD8_6	Port P86 register		○	○
PD8_7	Port P87 register		○	○

Figure 1.26.6. Port register (1)

Port P11 register (Note)

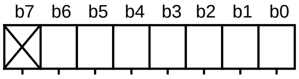


Symbol	Address	When reset
P11	03C9$_{16}$	Indeterminate

Bit symbol	Bit name	Function	R	W
P11_0	Port P11$_0$ register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit 0 : “L” level data 1 : “H” level data	○	○
P11_1	Port P11$_1$ register		○	○
P11_2	Port P11$_2$ register		○	○
P11_3	Port P11$_3$ register		○	○
P11_4	Port P11$_4$ register		○	○
Nothing is assigned. When write, set "0". When read, its content is indeterminate.			—	—

Note: This register exists in 144-pin version.

Port P14 register (Note)



Symbol	Address	When reset
P14	03D0$_{16}$	Indeterminate

Bit symbol	Bit name	Function	R	W
P14_0	Port P14$_0$ register	Data is input and output to and from each pin by reading and writing to and from each corresponding bit 0 : “L” level data 1 : “H” level data	○	○
P14_1	Port P14$_1$ register		○	○
P14_2	Port P14$_2$ register		○	○
P14_3	Port P14$_3$ register		○	○
P14_4	Port P14$_4$ register		○	○
P14_5	Port P14$_5$ register		○	○
P14_6	Port P14$_6$ register		○	○
Nothing is assigned. When set, write "0". When read, its content is indeterminate.			—	—

Note: This register exists in 144-pin version.

Figure 1.26.7. Port register (2)

Table 1.26.1. Each port and peripheral output function (Note 1)

Port	Periphral output function 1	Periphraloutput function 2	Periphral output function 3
P6$_0$	$\overline{\text{RTS}_0}$ output	——	——
P6$_1$	CLK$_0$ output	——	——
P6$_2$	——	——	——
P6$_3$	TxD$_0$ output	——	——
P6$_4$	$\overline{\text{RTS}_1}$ output	CLKS$_1$ output	——
P6$_5$	CLK$_1$ output	——	——
P6$_6$	——	——	——
P6$_7$	TxD$_1$ output	——	——
P7$_0$ (Note 2)	TxD$_2$(SDA$_2$) output	TA0$_{\text{OUT}}$ output	——
P7$_1$ (Note 2)	SCL$_2$ output	——	——
P7$_2$	CLK$_2$ output	TA1$_{\text{OUT}}$ output	V phase output
P7$_3$	$\overline{\text{RTS}_2}$ output	$\overline{\text{V}}$ phase output	——
P7$_4$	TA2$_{\text{OUT}}$ output	W phase output	——
P7$_5$	$\overline{\text{W}}$ phase output	——	——
P7$_6$	TA3$_{\text{OUT}}$ output	——	——
P7$_7$	——	——	——
P8$_0$	TA4$_{\text{OUT}}$ output	U phase output	——
P8$_1$	$\overline{\text{U}}$ phase output	——	——
P8$_2$	——	——	——
P8$_3$	——	——	——
P8$_4$	——	——	——
P8$_5$	——	——	——
P8$_6$	——	——	——
P8$_7$	——	——	——
P9$_0$	CLK$_3$ output	——	——
P9$_1$	SCL$_3$ output	STxD$_3$ output	——
P9$_2$	TxD$_3$(SDA$_3$) output	——	——
P9$_3$	$\overline{\text{RTS}_3}$ output	——	——
P9$_4$	$\overline{\text{RTS}_4}$ output	——	——
P9$_5$	CLK$_4$ output	——	——
P9$_6$	TxD$_4$(SDA$_4$) output	——	——
P9$_7$	SCL$_3$ output	STxD$_4$ output	——

Note 1: When using peripheral input function, set the corresponding function select register A to "0" (I/O port).

Note 2: N-channel open drain output.

Function select register A0

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
PS0	$03B0_{16}$	$0X000X00_2$

Bit symbol	Bit name	Function	R	W
PS0_0	Port $P6_0$ function select bit	0 : I/O port 1 : $\overline{RTS_0}$ output	○	○
PS0_1	Port $P6_1$ function select bit	0 : I/O port 1 : CLK_0 output	○	○
Nothing is assigned. When write, set "0". When read, the content is indeterminate.			—	—
PS0_3	Port $P6_3$ function select bit	0 : I/O port 1 : TXD_0 output	○	○
PS0_4	Port $P6_4$ function select bit	0 : I/O port 1 : Peripheral function output (PSL0_4 enabled)	○	○
PS0_5	Port $P6_5$ function select bit	0 : I/O port 1 : CLK_1 output	○	○
Nothing is assigned. When write, set "0". When read, the content is indeterminate.			—	—
PS0_7	Port $P6_7$ function select bit	0 : I/O port 1 : TXD_1 output	○	○

Function select register A1

b7 b6 b5 b4 b3 b2 b1 b0

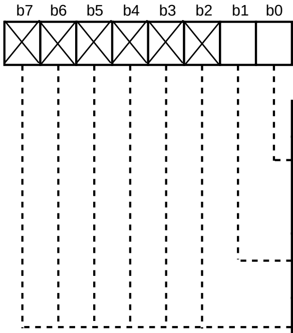
Symbol	Address	When reset
PS1	$03B1_{16}$	$X0000000_2$

Bit symbol	Bit name	Function	R	W
PS1_0	Port $P7_0$ function select bit (Note)	0 : I/O port 1 : Peripheral function output (PSL1_0 enabled)	○	○
PS1_1	Port $P7_1$ function select bit (Note)	0 : I/O port 1 : SCL_2 output	○	○
PS1_2	Port $P7_2$ function select bit	0 : I/O port 1 : Peripheral function output (PSL1_2, PSC_0 enabled)	○	○
PS1_3	Port $P7_3$ function select bit	0 : I/O port 1 : Peripheral function output (PSL1_3 enabled)	○	○
PS1_4	Port $P7_4$ function select bit	0 : I/O port 1 : Peripheral function output (PSL1_4 enabled)	○	○
PS1_5	Port $P7_5$ function select bit	0 : I/O port 1 : $\overline{W}$ phase output	○	○
PS1_6	Port $P7_6$ function select bit	0 : I/O port 1 : $TA3_{OUT}$ output	○	○
Nothing is assigned. When write, set "0". When read, the content is indeterminate.			—	—

Note: This port is N-channel open drain output.

Figure 1.26.8. Function select register A (1)

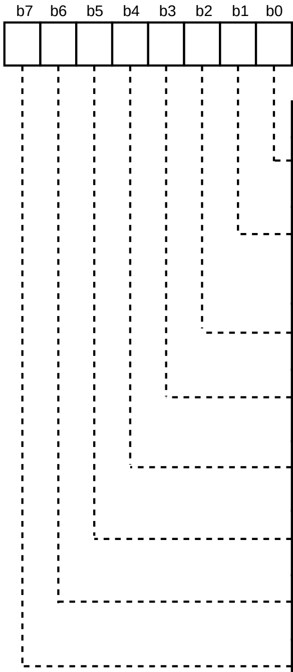
Function select register A2



Symbol	Address	When reset
PS2	03B4$_{16}$	XXXXXX00$_2$

Bit symbol	Bit name	Function	R	W
PS2_0	Port P8$_0$ function select bit	0 : I/O port 1 : Peripheral function output (PSL2_0 enabled)	○	○
PS2_1	Port P8$_1$ function select bit	0 : I/O port 1 : $\overline{U}$ phase output	○	○
Nothing is assigned. When write, set "0". When read, the content is indeterminate.			–	–

Function select register A3 (Note)



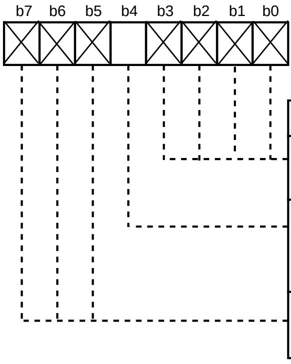
Symbol	Address	When reset
PS3	03B5$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
PS3_0	Port P9$_0$ function select bit	0 : I/O port 1 : CLK3 output	○	○
PS3_1	Port P9$_1$ function select bit	0 : I/O port 1 : Peripheral function output (PSL3_1 enabled)	○	○
PS3_2	Port P9$_2$ function select bit	0 : I/O port 1 : TxD$_3$(SDA$_3$) output	○	○
PS3_3	Port P9$_3$ function select bit	0 : I/O port 1 : RTS3 output	○	○
PS3_4	Port P9$_4$ function select bit	0 : I/O port 1 : RTS4 output	○	○
PS3_5	Port P9$_5$ function select bit	0 : I/O port 1 : CLK4 output	○	○
PS3_6	Port P9$_6$ function select bit	0 : I/O port 1 : TxD$_4$(SDA$_4$) output	○	○
PS3_7	Port P9$_7$ function select bit	0 : I/O port 1 : Peripheral function output (PSL3_7 enabled)	○	○

Note: Set bit 2 of protect register (address 000A$_{16}$) to “1” before rewriting to this register.

Figure 1.26.9. Function select register A (2)

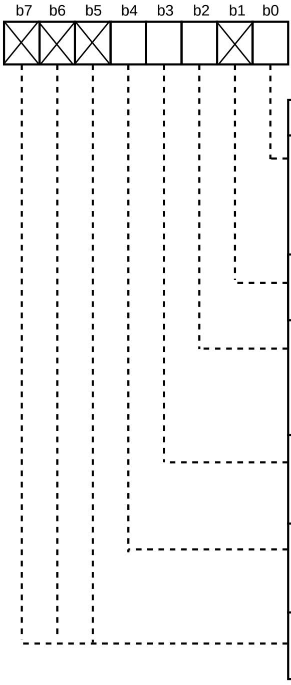
Function select register B0



Symbol	Address	When reset
PSL0	$03B2_{16}$	$XXX0XXXX_2$

Bit symbol	Bit name	Function	R	W
Nothing is assigned. When write, set "0". When read, the content is indeterminate.			—	—
PSL0_4	Port P6$_4$ peripheral function select bit (Enabled when PS0_4 = 1)	0 : $\overline{RTS1}$ output 1 : CLKS1 output	○	○
Nothing is assigned. When write, set "0". When read, the content is indeterminate.			—	—

Function select register B1

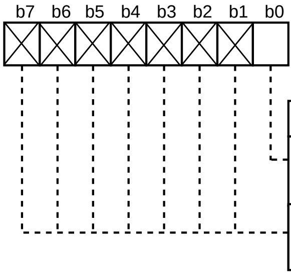


Symbol	Address	When reset
PSL1	$03B3_{16}$	$XXX000X0_2$

Bit symbol	Bit name	Function	R	W
PSL1_0	Port P7$_0$ peripheral function select bit (Enabled when PS1_0 = 1) (Note 2)	0 : TxD$_2$(SDA$_2$) port 1 : TA0$_{OUT}$ output	○	○
Nothing is assigned. When write, set "0". When read, the content is indeterminate.			—	—
PSL1_2	Port P7$_2$ peripheral function select bit (Enabled when PS1_2 = 1)	0 : Port P7$_2$ peripheral subfunction select bit (PSC_0) is enabled 1 : TA1$_{OUT}$ output (Note 1)	○	○
PSL1_3	Port P7$_3$ peripheral function select bit (Enabled when PS1_3 = 1)	0 : $\overline{RTS2}$ port 1 : $\overline{V}$ phase output	○	○
PSL1_4	Port P7$_4$ peripheral function select bit (Enabled when PS1_4 = 1)	0 : TA2$_{OUT}$ port 1 : W phase output	○	○
Nothing is assigned. When write, set "0". When read, the content is indeterminate.			—	—

Note 1: Set PSC_0 to “1”.
Note 2: This port is N-channel open drain output.

Function select register B2

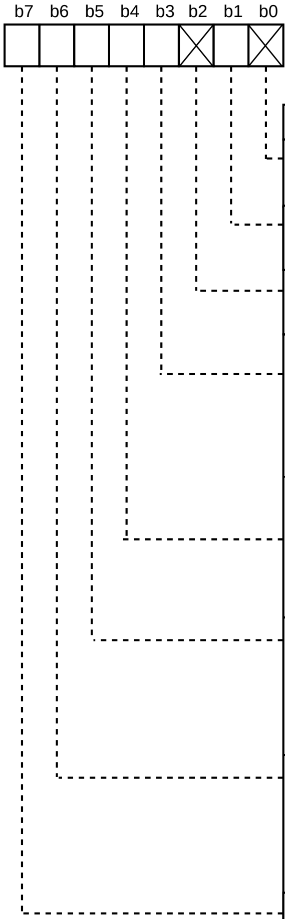


Symbol	Address	When reset
PSL2	$03B6_{16}$	$XXXXXXX0_2$

Bit symbol	Bit name	Function	R	W
PSL2_0	Port P8$_0$ peripheral function select bit (Enabled when PS2_0 = 1)	0 : TA4$_{OUT}$ output 1 : U phase output	○	○
Nothing is assigned. When write, set "0". When read, the content is indeterminate.			—	—

Figure 1.26.10. Function select register B (1)

Function select register B3



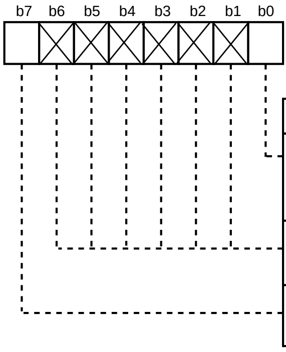
Symbol	Address	When reset
PSL3	03B7$_{16}$	00000X0X$_2$

Bit symbol	Bit name	Function	R	W
Nothing is assigned. When write, set "0". When read, the content is indeterminate.			–	–
PSL3_1	Port $P9_1$ peripheral function select bit	0 : SCL$_3$ output 1 : STxD$_3$ output	○	○
Nothing is assigned. When write, set "0". When read, the content is indeterminate.			–	–
PSL3_3	Port $P9_3$ peripheral function	0 : Input peripheral function enabled (Except DA0 output) (Note) 1 : Input peripheral function disabled (DA0 output)	○	○
PSL3_4	Port $P9_4$ peripheral function	0 : Input peripheral function enabled (Except DA1 output) (Note) 1 : Input peripheral function disabled (DA1 output)	○	○
PSL3_5	Port $P9_5$ peripheral function	0 : Input peripheral function enabled (Except ANEX0 use) (Note) 1 : Input peripheral function disabled (ANEX0 use)	○	○
PSL3_6	Port $P9_6$ peripheral function	0 : Input peripheral function enabled (Except ANEX1 use) (Note) 1 : Input peripheral function disabled (ANEX1 use)	○	○
PSL3_7	Port $P9_7$ peripheral function select bit	0 : SCL$_4$ output 1 : STxD$_4$ output	○	○

Note: Although DA0, DA1 output and ANEX0, ANEX1 can be used when "0" is set in these bits, the power supply may be increased.

Figure 1.26.11. Function select register B (2)

Function select register C



Symbol	Address	When reset
PSC	03AF$_{16}$	0XXXXXX0$_2$

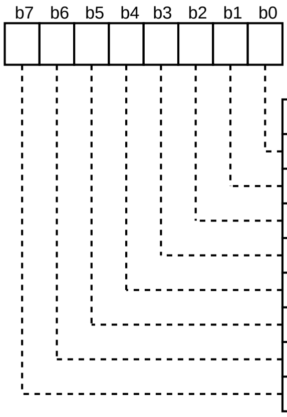
Bit symbol	Bit name	Function	R	W
PSC_0 (Note 1)	Port $P7_2$ peripheral subfunction select bit (Enabled when PS1_2 = 1 and PSL1_2 = 0)	0 : CLK2 output 1 : V phase output	○	○
Nothing is assigned. When write, set "0". When read, the content is indeterminate.			–	–
PSC_7 (Note 2)	Key input interrupt disable bit	0 : Enabled 1 : Disabled	○	○

Note 1: Set this bit to "0" when PSL1_2 = "1".

Note 2: When this bit is "1", key input interrupt for interrupt controller is disabled regardless of port input and setting of interrupt control register.
When changing this bit, set key input interrupt disabled by key input interrupt control register.

Figure 1.26.12. Function select register C

Pull-up control register 0 (Note)

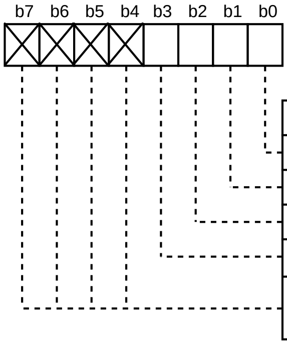


Symbol	Address	When reset
PUR0	$03F0_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
PU00	$P0_0$ to $P0_3$ pull-up	The corresponding port is pulled high with a pull-up resistance 0 : Not pulled high 1 : Pulled high	○	○
PU01	$P0_4$ to $P0_7$ pull-up		○	○
PU02	$P1_0$ to $P1_3$ pull-up		○	○
PU03	$P1_4$ to $P1_7$ pull-up		○	○
PU04	$P2_0$ to $P2_3$ pull-up		○	○
PU05	$P2_4$ to $P2_7$ pull-up		○	○
PU06	$P3_0$ to $P3_3$ pull-up		○	○
PU07	$P3_4$ to $P3_7$ pull-up		○	○

Note: Since P0 to P5 operate as the bus in memory expansion mode and microprocessor mode, do not set the pull-up control register. However, it is possible to select pull-up resistance presence to the usable port as I/O port by setting.

Pull-up control register 1 (Note)

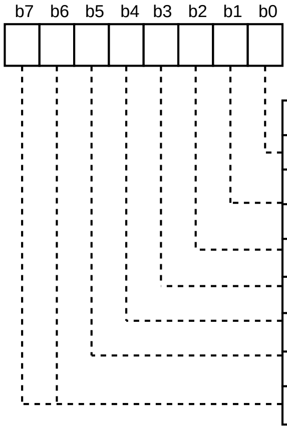


Symbol	Address	When reset
PUR1	$03F1_{16}$	$X0_{16}$

Bit symbol	Bit name	Function	R	W
PU10	$P4_0$ to $P4_3$ pull-up	The corresponding port is pulled high with a pull-up resistance 0 : Not pulled high 1 : Pulled high	○	○
PU11	$P4_4$ to $P4_7$ pull-up		○	○
PU12	$P5_0$ to $P5_3$ pull-up		○	○
PU13	$P5_4$ to $P5_7$ pull-up		○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—

Note: Since P0 to P5 operate as the bus in memory expansion mode and microprocessor mode, do not set the pull-up control register. However, it is possible to select pull-up resistance presence to the usable port as I/O port by setting.

Pull-up control register 2



Symbol	Address	When reset
PUR2	$03DA_{16}$	00_{16}

Bit symbol	Bit name	Function	R	W
PU20	$P6_0$ to $P6_3$ pull-up	The corresponding port is pulled high with a pull-up resistance 0 : Not pulled high 1 : Pulled high	○	○
PU21	$P6_4$ to $P6_7$ pull-up		○	○
PU22	$P7_0$ to $P7_3$ pull-up (Note 1)		○	○
PU23	$P7_4$ to $P7_7$ pull-up		○	○
PU24	$P8_0$ to $P8_3$ pull-up		○	○
PU25	$P8_4$ to $P8_7$ pull-up (Note 2)		○	○
PU26	$P9_0$ to $P9_3$ pull-up		○	○
PU27	$P9_4$ to $P9_7$ pull-up		○	○

Note 1: Since $P7_0$ and $P7_1$ are N-channel open drain ports, pull-up is not available for them.
Note 2: Except port $P8_5$.

Figure 1.26.13. Pull-up control register (1)

100-pin version

Pull-up control register 3

b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	0		

Symbol	Address	When reset
PUR3	03DB$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	R	W
PU30	$P10_0$ to $P10_3$ pull-up	The corresponding port is pulled high with a pull-up resistance 0 : Not pulled high 1 : Pulled high	○	○
PU31	$P10_4$ to $P10_7$ pull-up		○	○
Reserved bit		Must always be set to "0"	○	○

144-pin version

Pull-up control register 3

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
PUR3	03DB$_{16}$	00$_{16}$

Bit	Bit	Function	R	W
PU30	$P10_0$ to $P10_3$ pull-up	The corresponding port is pulled high with a pull-up resistance 0 : Not pulled high 1 : Pulled high	○	○
PU31	$P10_4$ to $P10_7$ pull-up		○	○
PU32	$P11_0$ to $P11_3$ pull-up		○	○
PU33	$P11_4$ pull-up		○	○
PU34	$P12_0$ to $P12_3$ pull-up		○	○
PU35	$P12_4$ to $P12_7$ pull-up		○	○
PU36	$P13_0$ to $P13_3$ pull-up		○	○
PU37	$P13_4$ to $P13_7$ pull-up		○	○

Pull-up control register 4 (Note)

b7	b6	b5	b4	b3	b2	b1	b0
☒	☒	☒	☒				

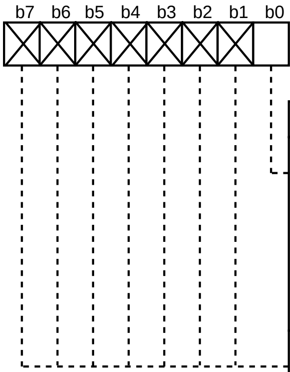
Symbol	Address	When reset
PUR4	03DC$_{16}$	XXXX0000$_2$

Bit symbol	Bit name	Function	R	W
PU40	$P14_0$ to $P14_3$ pull-up	The corresponding port is pulled high with a pull-up resistance 0 : Not pulled high 1 : Pulled high	○	○
PU41	$P14_4$ to $P14_6$ pull-up		○	○
PU42	$P15_0$ to $P15_3$ pull-up		○	○
PU43	$P15_4$ to $P15_7$ pull-up		○	○
Nothing is assigned. When write, set "0". When read, their contents are “0”.			–	–

Note: This register exists in 144-pin version.

Figure 1.26.14. Pull-up control register (2)

Port control register (Note 1)



Symbpl	Address	When reset
PCR	03FF$_{16}$	XXXXXXX0$_2$

Bit symbol	Bit name	Function	R	W
PCR0	Port P1 control register	0 : Function as common CMOS port 1 : Function as N-ch open drain port (Note 2)	○	○
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—

Note 1: Since P1 operates as the data bus in memory expansion mode and microprocessor mode, do not set the port control register. However, it is possible to select the CMOS port or N-channel open drain to the usable port as I/O port by setting.

Note 2: This function is designed to permanently turn OFF the Pch of the CMOS port. It does not make port 1 a full open drain. Therefore, the absolute maximum input voltage rating is [-3 to Vcc + 0.3V].

Figure 1.26.15. Port control register

Table 1.26.2. Example connection of unused pins in single-chip mode

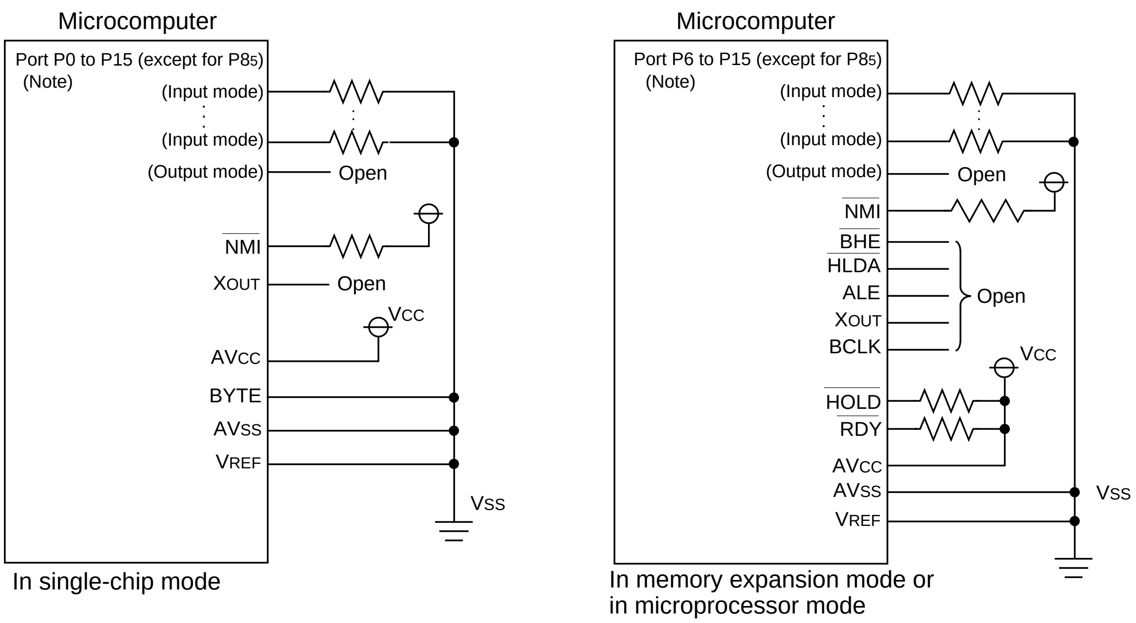
Pin name	Connection
Ports P0 to P15 (excluding P8$_5$) (Note 1)	After setting for input mode, connect every pin to VSS via a resistance (pull-down); or after setting for output mode, leave these pins open.
XOUT (Note 2)	Open
$\overline{\text{NMI}}$	Connect via resistance to VCC (pull-up)
AVCC	Connect to VCC
AVSS, VREF, BYTE	Connect to VSS

Note 1: Port P11 to P15 exist in 144-pin version.
Note 2: With external clock input to XIN pin.

Table 1.26.3. Example connection of unused pins in memory expansion mode and microprocessor mode

Pin name	Connection
Ports P6 to P15(excluding P8$_5$) (Note 1)	After setting for input mode, connect every pin to VSS via a resistance (pull-down); or after setting for output mode, leave these pins open.
$\overline{\text{BHE}}$, ALE, $\overline{\text{HLDA}}$, XOUT(Note 2), BCLK	Open
$\overline{\text{HOLD}}$, $\overline{\text{RDY}}$, $\overline{\text{NMI}}$	Connect via resistance to VCC (pull-up)
AVCC	Connect to VCC
AVSS, VREF	Connect to VSS

Note 1: Port P11 to P15 exist in 144-pin version.
Note 2: With external clock input to XIN pin.



Note: Port P11 to P15 exist in 144-pin version.

Figure 1.26.16. Example connection of unused pins

Usage Precaution

SFR (100-pin version)

(1) Addresses 03C9$_{16}$, 03CB$_{16}$ to 03D3$_{16}$, 03DC$_{16}$ area is for future plan. Must set "FF$_{16}$" to address 03CB$_{16}$, 03CE$_{16}$, 03CF$_{16}$, 03D2$_{16}$, 03D3$_{16}$ and "00$_{16}$" to address 03DC$_{16}$ at initial setting.

Timer A (timer mode)

(1) Reading the timer Ai register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer Ai register with the reload timing gets “FFFF$_{16}$”. Reading the timer Ai register after setting a value in the timer Ai register with a count halted but before the counter starts counting gets a proper value.

Timer A (event counter mode)

(1) Reading the timer Ai register while a count is in progress allows reading, with arbitrary timing, the value of the counter. Reading the timer Ai register with the reload timing gets “FFFF$_{16}$” by underflow or “0000$_{16}$” by overflow. Reading the timer Ai register after setting a value in the timer Ai register with a count halted but before the counter starts counting gets a proper value.

(2) When stop counting in free run type, set timer again.

(3) In the case of using as “Free-Run type”, the timer register contents may be unknown when counting begins. If the timer register is set before counting has started, then the starting value will be unknown.

- In the case where the up/down count will not be changed.

  Enable the “Reload” function and write to the timer register before counting begins. Rewrite the value to the timer register immediately after counting has started. If counting up, rewrite “0000$_{16}$” to the timer register. If counting down, rewrite “FFFF$_{16}$” to the timer register. This will cause the same operation as “Free-Run type” mode.

- In the case where the up/down count has changed.

  First set to “Reload type” operation. Once the first counting pulse has occurred, the timer may be changed to “Free-Run type”.

Timer A (one-shot timer mode)

(1) Setting the count start flag to “0” while a count is in progress causes as follows:

- The counter stops counting and a content of reload register is reloaded.
- The TAi$_{OUT}$ pin outputs “L” level.
- The interrupt request generated and the timer Ai interrupt request bit goes to “1”.

(2) The output from the one-shot timer synchronizes with the count source generated internally. Therefore, when an external trigger has been selected, a delay of one cycle of count source as maximum occurs between the trigger input to the TAi$_{IN}$ pin and the one-shot timer output.

(3) The timer Ai interrupt request bit goes to "1" if the timer's operation mode is set using any of the following procedures:

- Selecting one-shot timer mode after reset.
- Changing operation mode from timer mode to one-shot timer mode.
- Changing operation mode from event counter mode to one-shot timer mode.

Therefore, to use timer Ai interrupt (interrupt request bit), set timer Ai interrupt request bit to "0" after the above listed changes have been made.

(4) If a trigger occurs while a count is in progress, after the counter performs one down count following the reoccurrence of a trigger, the reload register contents are reloaded, and the count continues. To generate a trigger while a count is in progress, generate the second trigger after an elapse longer than one cycle of the timer's count source after the previous trigger occurred.

Timer A (pulse width modulation mode)

(1) The timer Ai interrupt request bit becomes "1" if setting operation mode of the timer in compliance with any of the following procedures:

- Selecting PWM mode after reset.
- Changing operation mode from timer mode to PWM mode.
- Changing operation mode from event counter mode to PWM mode.

Therefore, to use timer Ai interrupt (interrupt request bit), set timer Ai interrupt request bit to "0" after the above listed changes have been made.

(2) Setting the count start flag to "0" while PWM pulses are being output causes the counter to stop counting. If the TAiOUT pin is outputting an "H" level in this instance, the output level goes to "L", and the timer Ai interrupt request bit goes to "1". If the TAiOUT pin is outputting an "L" level in this instance, the level does not change, and the timer Ai interrupt request bit does not becomes "1".

Timer B (timer mode, event counter mode)

(1) Reading the timer Bi register while a count is in progress allows reading , with arbitrary timing, the value of the counter. Reading the timer Bi register with the reload timing gets "$FFFF_{16}$". Reading the timer Bi register after setting a value in the timer Bi register with a count halted but before the counter starts counting gets a proper value.

Timer B (pulse period/pulse width measurement mode)

(1) If changing the measurement mode select bit is set after a count is started, the timer Bi interrupt request bit goes to "1".

(2) When the first effective edge is input after a count is started, an indeterminate value is transferred to the reload register. At this time, timer Bi interrupt request is not generated.

(3) The value of the counter is indeterminate at the beginning of a count. Therefore, the timer Bi overflow flag may go to "1" and timer Bi interrupt request may be generated during the interval between a count start and an effective edge input.

Stop Mode and Wait Mode

(1) When returning from stop mode by hardware reset, $\overline{\text{RESET}}$ pin must be set to "L" level until main clock oscillation is stabilized.

(2) When shifting to WAIT mode or STOP mode, the program stops after reading from the WAIT instruction and the instruction that sets all clock stop control bits to "1" in the instruction queue. Therefore, insert a minimum of 4 NOPs after the WAIT instruction and the instruction that sets all clock stop control bits to "1" in order to flush the instruction queue.

A-D Converter

(1) Write to each bit (except bit 6) of A-D control register 0, to each bit of A-D control register 1, and to bit 0 of A-D control register 2 when A-D conversion is stopped (before a trigger occurs).

In particular, when the Vref connection bit is changed from "0" to "1", start A-D conversion after an elapse of 1 µs or longer.

(2) When changing A-D operation mode, select analog input pin again.

(3) Using one-shot mode or single sweep mode

Read the correspondence A-D register after confirming A-D conversion is finished. (It is known by A-D conversion interrupt request bit.)

(4) Using repeat mode, repeat sweep mode 0 or repeat sweep mode 1

Use the undivided main clock as the internal CPU clock.

(5) When $f(X_{IN})$ is faster than 10 MHz, make the frequency 10 MHz or less by dividing.

(6) Output impedance of sensor at A-D conversion (Reference value)

To carry out A-D conversion properly, charging the internal capacitor C shown in Figure 1.27.1 has to be completed within a specified period of time T. Let output impedance of sensor equivalent circuit be R0, microcomputer's internal resistance be R, precision (error) of the A-D converter be X, and the A-D converter's resolution be Y (Y is 1024 in the 10-bit mode, and 256 in the 8-bit mode).

Vc is generally $V_C = V_{IN} \{1 - e^{-\frac{t}{C(R0+R)}}\}$

And when t = T, $V_C = V_{IN} - \frac{X}{Y} V_{IN} = V_{IN}(1 - \frac{X}{Y})$

$$e^{-\frac{T}{C(R0+R)}} = \frac{X}{Y}$$

$$-\frac{T}{C(R0+R)} = \ln\frac{X}{Y}$$

Hence, $R0 = -\frac{T}{C \bullet \ln\frac{X}{Y}} - R$

With the model shown in Figure 1.27.1 as an example, when the difference between V_{IN} and V_C becomes 0.1LSB, we find impedance R0 when voltage between pins V_C changes from 0 to V_{IN}-(0.1/1024) V_{IN} in time T. (0.1/1024) means that A-D precision drop due to insufficient capacitor charge is held to 0.1LSB at time of A-D conversion in the 10-bit mode. Actual error however is the value of absolute precision added to 0.1LSB. When $f(X_{IN})$ = 10 MHz, T = 0.3 us in the A-D conversion mode with sample & hold. Output impedance R0 for sufficiently charging capacitor C within time T is determined as follows.

T = 0.3 µs, R = 7.8 kΩ, C = 3 pF, X = 0.1, and Y = 1024 . Hence,

$$R0 = -\frac{0.3 \times 10^{-6}}{3.0 \times 10^{-12} \bullet \ln\frac{0.1}{1024}} - 7.8 \times 10^3 \fallingdotseq 3.0 \times 10^3$$

Thus, the allowable output impedance of the sensor circuit capable of thoroughly driving the A-D converter turns out to be approximately 3.0 kΩ. Tables 1.27.1 and 1.27.2 show output impedance values based on the LSB values.

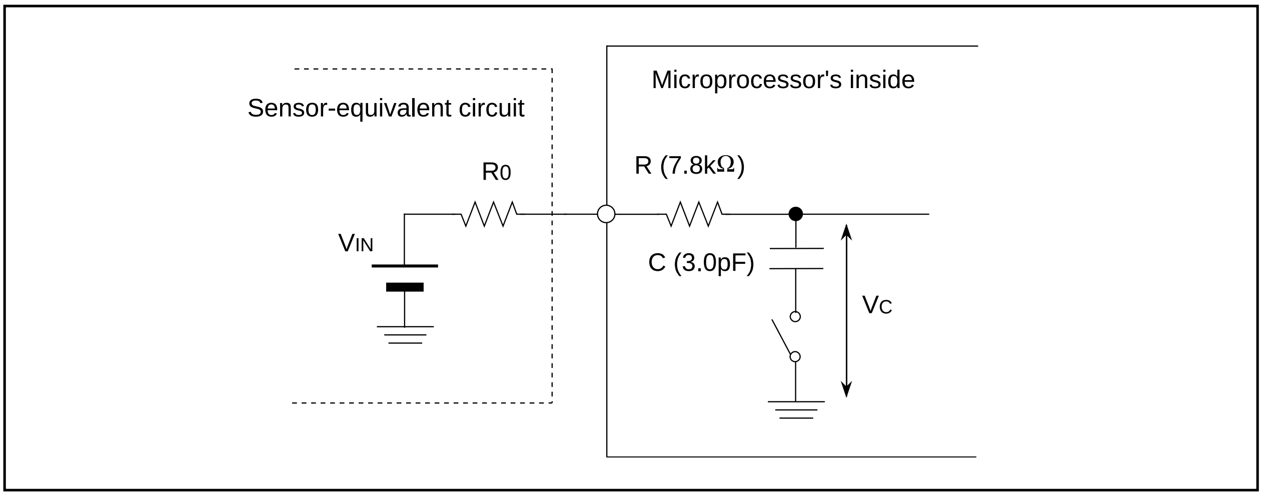


Figure 1.27.1 A circuit equivalent to the A-D conversion terminal

Tables 1.27.1. Output impedance values based on the LSB values (10-bit mode) Reference value

f(XIN) (MHz)	Cycle (µs)	Sampling time (µs)	R (Kohm)	C (pF)	Resolution (LSB)	R0max (Kohm)
10	0.1	0.3 (3 X cycle, Sample & hold bit is enabled)	7.8	3.0	0.1	3.0
					0.3	4.5
					0.5	5.3
					0.7	5.9
					0.9	6.4
					1.1	6.8
					1.3	7.2
					1.5	7.5
					1.7	7.8
					1.9	8.1
10	0.1	0.2 (2 X cycle, Sample & hold bit is enabled)	7.8	3.0	0.3	0.4
					0.5	0.9
					0.7	1.3
					0.9	1.7
					1.1	2.0
					1.3	2.2
					1.5	2.4
					1.7	2.6
					1.9	2.8

Tables 1.27.2. Output impedance values based on the LSB values (8-bit mode) Reference value

f(XIN) (MHz)	Cycle (µs)	Sampling time (µs)	R (Kohm)	C (pF)	Resolution (LSB)	R0max (Kohm)
10	0.1	0.3 (3 X cycle, Sample & hold bit is enabled)	7.8	3.0	0.1	4.9
					0.3	7.0
					0.5	8.2
					0.7	9.1
					0.9	9.9
					1.1	10.5
					1.3	11.1
					1.5	11.7
					1.7	12.1
					1.9	12.6
10	0.1	0.2 (2 X cycle, Sample & hold bit is enabled)	7.8	3.0	0.1	0.7
					0.3	2.1
					0.5	2.9
					0.7	3.5
					0.9	4.0
					1.1	4.4
					1.3	4.8
					1.5	5.2
					1.7	5.5
					1.9	5.8

Interrupts

(1) Setting the stack pointer

- The value of the stack pointer is initialized to 000000_{16} immediately after reset. Accepting an interrupt before setting a value in the stack pointer may cause runaway. Be sure to set a value in the stack pointer before accepting an interrupt.
  When using the $\overline{\text{NMI}}$ interrupt, initialize the stack pointer at the beginning of a program. Regarding the first instruction immediately after reset, generating any interrupts including the $\overline{\text{NMI}}$ interrupt is prohibited.
  Set an even address to the stack pointer so that operating efficiency is increased.

(2) The $\overline{\text{NMI}}$ interrupt

- As for the $\overline{\text{NMI}}$ interrupt pin, an interrupt cannot be prohibited. Connect it to the V_{CC} pin via a resistance (pulled-up) if unused.
- The $\overline{\text{NMI}}$ pin also serves as $P8_5$, which is exclusively input. Reading the contents of the P8 register allows reading the pin value. Use the reading of this pin only for establishing the pin level at the time when the $\overline{\text{NMI}}$ interrupt is input.
- Signal of "L" level width more than 1 clock of CPU operation clock (BCLK) is necessary for $\overline{\text{NMI}}$ pin.

(3) Address match interrupt

- Do not set the following addresses to the address match interrupt register.
  1. The address of the starting instruction in an interrupt routine.
  2. Any of the next 7 instructions addresses immediately after an instruction to clear an interrupt request bit of an interrupt control register or an instruction to rewrite an interrupt priority level to a smaller value.
  3. Any of the next 3 instructions addresses immediately after an instruction to set the interrupt enable flag (I flag).
  4. Any of the next 3 instructions addresses immediately after an instruction to rewrite a processor interrupt priority level (IPL) to a smaller value.

Example 1)

```
Interrupt_A:                          ; Interrupt A routine
          pushm   R0,R1,R2,R3,A0,A1   ; <---- Do not set address match interrupt to the
          ••••                        ;       start address of an interrupt instruction
```

Example 2)

```
mov.b    #0,TA0IC        ;Change TA0 interrupt priority level to a smaller value
nop                      ; 1st instruction  ┐
nop                      ; 2nd instruction  │
nop                      ; 3rd instruction  │
nop                      ; 4th instruction  ├ Do not set address match interrupt
nop                      ; 5th instruction  │ during this period
nop                      ; 6th instruction  │
nop                      ; 7th instruction  ┘
```

Example 3)

```
fset   I       ; Set I flag ( interrupt enabled)
nop            ; 1st instruction  ┐
nop            ; 2nd instruction  ├ Do not set address match interrupt
nop            ; 3rd instruction  ┘ during this period
```

Example 4)

```
        ldipl   #0     ; Rewrite IPL to a smaller value
        nop            ; 1st instruction  }
        nop            ; 2nd instruction  } Do not set address match interrupt
        nop            ; 3rd instruction  } during this period
```

- To return from an interrupt to the address set in an address match interrupt register using return instruction (reit or freit)

  To rewrite the interrupt control register within the interrupt routine, add the below processing to the end of the routine (immediately before the reit or freit instruction). Also, if multiple interrupts are enabled with other interrupts, add the below processing to the end of the interrupt that enables the multiple interrupts.

If the interrupt control register is being rewritten within the non-maskable interrupt routine, add the below processing to the end of all interrupts.

Additional process

```
                                   ; Execute after the register reset instruction (popm instruction)
    fclr    U                      ; Select ISP (Unnecessary if the ISP has been selected)
    pushm   R0                     ; Store R0 register
    mov.w   6[SP],R0               ; Read FLG on stack (use "stc SVF,R0" when high-speed
                                   ;                               interrupt)
    ldc     R0,FLG                 ; Set in FLG
    popm    R0                     ; Restore R0 register
    nop                            ; Dummy
    reit                           ; Interrupt completed (use freit when high-speed interrupt)
```

Example 5)

If rewriting the interrupt control register for interrupt B with the interrupt A routine and enabling multiple interrupts with interrupt C, the above processing is required at the end of the interrupt A and interrupt C routines.

```
            Interrupt A routine
Interrupt_A:
    pushm   R0,R1,R2,R3,A0,A1      ; Store registers
    ••••
    bclr    3,TA0IC                ; Rewrite interrupt control register of interrupt B
    ••••
    popm    R0,R1,R2,R3,A0,A1      ; Restore registers
    fclr    U                      ; Select ISP (Unnecessary if the ISP has been selected)
    pushm   R0                     ; Store R0 register
    mov.w   6[SP],R0               ; Read FLG on stack
    ldc     R0,FLG                 ; Set in FLG
    popm    R0                     ; Restore R0 register
    nop                            ; Dummy
    reit                           ; Interrupt completed
```

```
                Interrupt C routine
    Interrupt_C:
        pushm R0,R1,R2,R3,A0,A1      ; Store registers
        fset  I                      ; Multiple interrupt enabled
        ••••
        ••••
        popm  R0,R1,R2,R3,A0,A1      ;Restore registers
        fclr  U                      ; Select ISP (Unnecessary if the ISP has been selected)
        pushm R0                     ; Store R0 register
        mov.w 6[SP],R0               ; Read FLG on stack
        ldc   R0,FLG                 ; Set in FLG
        popm  R0                     ; Restore R0 register
        nop                          ; Dummy
        reit                         ; Interrupt completed
```

(4) External interrupt

- Edge sense

Either an “L” level or an “H” level of at least 250 ns width is necessary for the signal input to pins INT_0 to INT_5 regardless of the CPU operation clock.

- Level sense

Either an “L” level or an “H” level of 1 cycle of BCLK + at least 200 ns width is necessary for the signal input to pins INT_0 to INT_5 regardless of the CPU operation clock. (When X_{IN}=20MHz and no division mode, at least 250 ns width is necessary.)

- When the polarity of the INT_0 to INT_5 pins is changed, the interrupt request bit is sometimes set to "1". After changing the polarity, set the interrupt request bit to "0". Figure 1.27.2 shows the procedure for changing the $\overline{INT}$ interrupt generate factor.

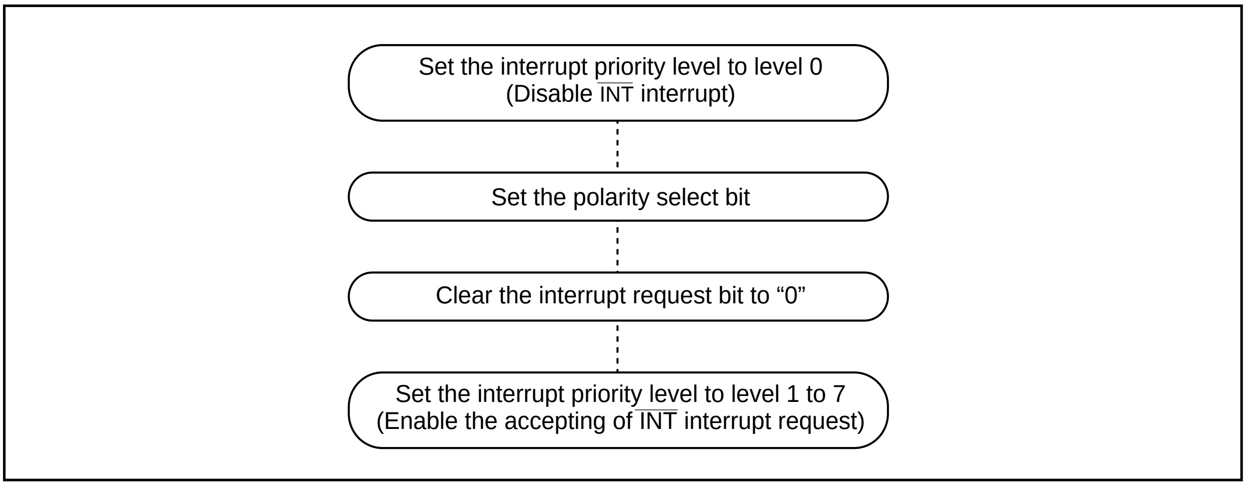

Set the interrupt priority level to level 0
(Disable $\overline{INT}$ interrupt)
Set the polarity select bit
Clear the interrupt request bit to “0”
Set the interrupt priority level to level 1 to 7
(Enable the accepting of $\overline{INT}$ interrupt request)

Figure 1.27.2. Switching condition of $\overline{INT}$ interrupt request

(5) Rewrite the interrupt control register

- When a instruction to rewrite the interrupt control register is executed but the interrupt is disabled, the interrupt request bit is not set sometimes even if the interrupt request for that register has been generated. This will depend on the instruction. If this creates problems, use the below instructions to change the register.

  Instructions : AND, OR, BCLR, BSET

DMAC

(1) Do not clear the DMA request bit of the DMAi request cause select register.

In M16C/80, when a DMA request is generated while the channel is disabled (Note), the DMA transfer is not executed and the DMA request bit is cleared automatically.

Note :The DMA is disabled or the transfer count register is "0".

(2) When DMA transfer is done by a software trigger, set DSR and DRQ of the DMAi request cause select register to "1" simultaneously using the OR instruction.

```
e.g.) OR.B   #0A0h, DMiSL        ; DMiSL is DMAi request cause select register
```

(3) When changing the DMAi request cause select bit of the DMAi request cause select register, set "1" to the DMA request bit, simultaneously. In this case, set the corresponding DMA channel to disabled before changing the DMAi request cause select bit. At least 2 instructions are needed from the instruction to write to the DMAi request cause select register to enable DMA.

Example) When DMA request cause is changed to timer A0 and using DMA0 in single transfer after DMA initial setting

```
push.w    R0                   ; Store R0 register
stc       DMD0, R0             ; Read DMA mode register 0
and.b     #11111100b, R0L      ; Clear DMA0 transfer mode select bit to "00"
ldc       R0, DMD0             ; DMA0 disabled
mov.b     #10000011b, DM0SL    ; Select timer A0
                               ; (Write "1" to DMA request bit simultaneously)
mov.b     R0L, R0L             ; Dummy cycle                }  At least 2 instructions
or.b      #00000001b, R0L      ; Set DMA0 single transfer   }  are needed until DMA
ldc       R0, DMD0             ; DMA0 enabled                  enabled.
pop.w     R0                   ; Restore R0 register
```

Noise

(1) A bypass capacitor should be inserted between Vcc-Vss line for reducing noise and latch-up

Connect a bypass capacitor (approx. 0.1µF) between the Vcc and Vss pins using short wiring and thicker circuit traces.

Precautions for using CLKOUT pin

When using the Clock Output function of P5$_3$/CLKOUT pin (f$_8$, f$_{32}$ or fc output) in single chip mode, use port P5$_7$ as an input only port (port P5$_7$ direction register is "0").

Although port P5$_7$ may be set as an output port, it will become high impedance and will not output "H" or "L" levels.

$\overline{\text{HOLD}}$ signal

When using the $\overline{\text{HOLD}}$ input while P4$_0$ to P4$_7$ and P5$_0$ to P5$_2$ are set as output ports in single-chip mode, you must first set all pins for P4$_0$ to P4$_7$ and P5$_0$ to P5$_2$ as input ports, then shift to microprocessor mode or memory expansion mode.

Reducing power consumption

(1) When A-D conversion is not performed, select the Vref not connected with the Vref connect bit of A-D control register 1. When A-D conversion is performed, start the A-D conversion at least 1 µs or longer after connecting Vref.

(2) When using AN4 ($P10_4$) to AN7 ($P10_7$), select the input disable of the key input interrupt signal with the key input interrupt disable bit of the function select register C .
When selecting the input disable of the key input interrupt signal, the key input interrupt cannot be used. Also, the port cannot be input even if the direction register of $P10_4$ to $P10_7$ is set to input (the input result becomes undefined). When the input disable of the key input interrupt signal is selected, use all AN4 to AN7 as A-D inputs.

(3) When ANEX0 and ANEX1 are used, select the input peripheral function disable with port $P9_5$ and $P9_6$ input peripheral function select bit of the function select register B3.
When the input peripheral function disable is selected, the port cannot be input even if the port direction register is set to input (the input result becomes undefined).
Also, it is not possible to input a peripheral function except ANEX0 and ANEX1.

(4) When D-A converter is not used, set output disabled with the D-A output enable bit of D-A control register and set the D-A register to "00_{16}".

(5) When D-A conversion is used, select the input peripheral function disabled with port $P9_3$ and $P9_4$ input peripheral function select bit of the function select register B3.
When the input peripheral function disabled is selected, the port cannot be input even if the port direction register is set to input (the input result becomes undefined).
Also, it is not possible to input a peripheral function.

DRAM controller

When shifting to self-refresh, select DRAM ignored by the DRAM space select bit. In the next instruction, simultaneously set the DRAM space select bit and self-refresh ON by self-refresh mode bit. Also, insert two NOPs after the instruction that sets the self-refresh mode bit to "1".
Do not access external memory while operating in self-refresh. (All external memory space access is inhibited.)
When disabling self-refresh, simultaneously select DRAM ignored by the DRAM space select bit and self-refresh OFF by self-refresh mode bit. In the next instruction, set the DRAM space select bit.
Do not access the DRAM space immediately after setting the DRAM space select bit.

Example) One wait is selected by the wait select bit and 4MB is selected by the DRAM space select bit

Shifting to self-refresh

```
        •••
        mov.b   #00000001b,DRAMCONT     ;DRAM ignored, one wait is selected
        mov.b   #10001011b,DRAMCONT     ;Set self-refresh, select 4MB and one wait
        nop                             ;Two nops are needed
        nop                             ;
        •••
```

Disable self-refresh

```
        •••
        mov.b   #00000001b,DRAMCONT     ;Disable self-refresh, DRAM ignored, one wait is
                                        ;selected
        mov.b   #00001011b,DRAMCONT     ;Select 4MB and one wait
        nop                             ;Inhibit instruction to access DRAM area
        nop
        •••
```

Setting the registers

The registers shown in Table 1.27.3 include indeterminate bit when read. Set immidiate to these registers.

Store the content of the frequently used register to RAM, change the content of RAM, then transfer to the register.

Table 1.27.3 The object registers

Register name	Symbol	Address
UART4 bit rate generator	U4BRG	$02F9_{16}$
UART4 transfer buffer register	U4TB	$02FB_{16}$, $02FA_{16}$
Dead time timer	DTT	$030C_{16}$
Timer B2 interrupt occurrence frequency set counter	ICTB2	$030D_{16}$
UART3 bit rate generator	U3BRG	0329_{16}
UART3 transfer buffer register	U3TB	$032B_{16}$, $032A_{16}$
UART2 bit rate generator	U2BRG	0339_{16}
UART2 transfer buffer register	U2TB	$033B_{16}$, $033A_{16}$
Up-down flag	UDF	0344_{16}
Timer A0 register (Note)	TA0	0347_{16}, 0346_{16}
Timer A1 register (Note)	TA1	0349_{16}, 0348_{16}
Timer A2 register (Note)	TA2	$034B_{16}$, $034A_{16}$
Timer A3 register (Note)	TA3	$034D_{16}$, $034C_{16}$
Timer A4 register (Note)	TA4	$034F_{16}$, $034E_{16}$
UART0 bit rate generator	U0BRG	0361_{16}
UART0 transfer buffer register	U0TB	0363_{16}, 0362_{16}
UART1 bit rate generator	U1BRG	0369_{16}
UART1 transfer buffer register	U1TB	$036B_{16}$, $036A_{16}$

Note: In one-shot timer mode and pulse widt modulation mode.

External ROM version (144-pin version)

The external ROM version is operated only in microprocessor mode, so be sure to perform the following:

- Connect CNVss pin to Vcc.

Notes on the microprocessor mode and transition after shifting from the microprocessor mode to the memory expansion mode / sigle-chip mode

In microprocessor mode, the SFR, internal RAM, and external memory space can be accessed.
For that reason, the internal ROM area cannot be accessed.
After the reset has been released and the operation of shifting from the microprocessor mode has started (“H” applied to the CNVSS pin), the internal ROM area cannot be accessed even if the CPU shifts to the memory expansion mode or single-chip mode.

Flash memory version

Bit 7 and bit 6 of the processor mode register 1 (address 0005_{16}) must be set to "11_2" and this setting should be done when the main clock is divided by 8.

Electrical characteristics

Table 1.28.1. Absolute maximum ratings

Symbol	Parameter		Condition	Rated value	Unit
Vcc	Supply voltage		V_{CC}=AV_{CC}	-0.3 to 6.5	V
AVcc	Analog supply voltage		V_{CC}=AV_{CC}	-0.3 to 6.5	V
V_I	Input voltage	$\overline{RESET}$, (maskROM : CNV_{SS}, BYTE), $P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_2$-$P7_7$, $P8_0$-$P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$, $P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$, V_{REF}, X_{IN} (Note 1)		-0.3 to Vcc+0.3	V
		$P7_0$, $P7_1$		-0.3 to 6.5	V
V_O	Output voltage	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$,$P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$,$P7_2$-$P7_7$, $P8_0$-$P8_4$, $P8_6$, $P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$, $P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$,X_{OUT} (Note 1)		-0.3 to Vcc+0.3	V
		$P7_0$, $P7_1$		-0.3 to 6.5	V
P_d	Power dissipation		Topr=25°C	500	mW
T_{opr}	Operating ambient temperature			-20 to 85 / -40 to 85 (Note 2)	°C
T_{stg}	Storage temperature			-65 to 150	°C

Note 1: Port P11 to P15 exist in 144-pin version.

Note 2: Specify a product of -40 to 85°C to use it.

Table 1.28.2. Recommended operating conditions (referenced to V_{CC} = 2.7V to 5.5V at Topr = – 20 to 85°C / – 40 to 85°C(Note3) unless otherwise specified)

Symbol	Parameter			Standard			Unit
				Min	Typ.	Max.	
Vcc	Supply voltage			2.7	5.0	5.5	V
AVcc	Analog supply voltage				Vcc		V
Vss	Supply voltage				0		V
AVss	Analog supply voltage				0		V
V_{IH}	HIGH input voltage	$P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_2$-$P7_7$, $P8_0$-$P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$, $P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$ (Note 5), X_{IN}, RESET, CNV_{SS}, BYTE		0.8Vcc		Vcc	V
		$P7_0$, $P7_1$		0.8Vcc		6.5	V
		$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$ (during single-chip mode)		0.8Vcc		Vcc	V
		$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$ (data input function during memory expansion and microprocessor modes)		0.5Vcc		Vcc	V
V_{IL}	LOW input voltage	$P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_0$-$P7_7$, $P8_0$-$P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$, $P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$ (Note 5), X_{IN}, RESET, CNV_{SS}, BYTE		0		0.2Vcc	V
		$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$ (during single-chip mode)		0		0.2Vcc	V
		$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$ (data input function during memory expansion and microprocessor modes)		0		0.16Vcc	V
$I_{OH\ (peak)}$	HIGH peak output current	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$ $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_2$-$P7_7$, $P8_0$-$P8_4$, $P8_6$, $P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$,$P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$ (Note 5)				-10.0	mA
$I_{OH\ (avg)}$	HIGH average output current	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$ $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_2$-$P7_7$, $P8_0$-$P8_4$, $P8_6$, $P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$,$P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$ (Note 5)				-5.0	mA
$I_{OL\ (peak)}$	LOW peak output current	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$ $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_0$-$P7_7$, $P8_0$-$P8_4$, $P8_6$, $P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$,$P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$ (Note 5)				10.0	mA
$I_{OL\ (avg)}$	LOW average output current	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$ $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_0$-$P7_7$, $P8_0$-$P8_4$, $P8_6$, $P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$,$P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$ (Note 5)				5.0	mA
f (X_{IN})	Main clock input oscillation frequency	No wait	Vcc=4.2V to 5.5V	0		20	MHz
			Vcc=2.7V to 4.2V	0		10	MHz
f (X_{CIN})	Subclock oscillation frequency				32.768	50	kHz

Note 1: The mean output current is the mean value within 100ms.

Note 2: The total I_{OL} (peak) for ports P0, P1, P2, $P8_6$, $P8_7$, P9, P10, P11, P14 and P15 must be 80mA max. The total I_{OH} (peak) for ports P0, P1, P2, $P8_6$, $P8_7$, P9, P10, P11, P14 and P15 must be -80mA max. The total I_{OL} (peak) for ports P3, P4, P5, P6, P7,$P8_0$ to $P8_4$, P12 and P13 must be 80mA max. The total I_{OH} (peak) for ports P3, P4, P5, P6, $P7_2$ to $P7_7$, $P8_0$ to $P8_4$, P12 and P13 must be -80mA max.

Note 3: Specify a product of -40 to 85°C to use it.

Note 4: The specification of V_{IH} and V_{IL} of $P8_7$ is not when using as X_{CIN} but when using programmable input port.

Note 5: Port P11 to P15 exist in 144-pin version.

V_{CC} = 5V

Table 1.28.3. Electrical characteristics (referenced to V_{CC}=5V, V_{SS}=0V at Topr=25^{O}C, f(X_{IN})=20MHz unless otherwise specified)

Symbol	Parameter	Measuring condition	Standard Min	Standard Typ.	Standard Max.	Unit
V_{OH}	HIGH output voltage: P0$_0$-P0$_7$, P1$_0$-P1$_7$, P2$_0$-P2$_7$, P3$_0$-P3$_7$, P4$_0$-P4$_7$, P5$_0$-P5$_7$, P6$_0$-P6$_7$, P7$_2$-P7$_7$, P8$_0$-P8$_4$, P8$_6$, P8$_7$, P9$_0$-P9$_7$, P10$_0$-P10$_7$, P11$_0$-P11$_4$, P12$_0$-P12$_7$, P13$_0$-P13$_7$, P14$_0$-P14$_6$, P15$_0$-P15$_7$ (Note 1)	I_{OH}= - 5mA	3.0			V
V_{OH}	HIGH output voltage: P0$_0$-P0$_7$, P1$_0$-P1$_7$, P2$_0$-P2$_7$, P3$_0$-P3$_7$, P4$_0$-P4$_7$, P5$_0$-P5$_7$, P6$_0$-P6$_7$, P7$_2$-P7$_7$, P8$_0$-P8$_4$, P8$_6$, P8$_7$, P9$_0$-P9$_7$, P10$_0$-P10$_7$, P11$_0$-P11$_4$, P12$_0$-P12$_7$, P13$_0$-P13$_7$, P14$_0$-P14$_6$, P15$_0$-P15$_7$ (Note 1)	I_{OH}= - 200μA	4.7			V
V_{OH}	HIGH output voltage X_{OUT} HIGHPOWER	I_{OH}= - 1mA	3.0			V
	HIGH output voltage X_{OUT} LOWPOWER	I_{OH}= - 0.5mA	3.0			
	HIGH output voltage X_{COUT} HIGHPOWER	With no load applied		3.0		V
	HIGH output voltage X_{COUT} LOWPOWER	With no load applied		1.6		
V_{OL}	LOW output voltage: P0$_0$-P0$_7$, P1$_0$-P1$_7$, P2$_0$-P2$_7$, P3$_0$-P3$_7$, P4$_0$-P4$_7$, P5$_0$-P5$_7$, P6$_0$-P6$_7$, P7$_0$-P7$_7$, P8$_0$-P8$_4$, P8$_6$, P8$_7$, P9$_0$-P9$_7$, P10$_0$-P10$_7$, P11$_0$-P11$_4$, P12$_0$-P12$_7$, P13$_0$-P13$_7$, P14$_0$-P14$_6$, P15$_0$-P15$_7$ (Note 1)	I_{OL}=5mA			2.0	V
V_{OL}	LOW output voltage: P0$_0$-P0$_7$, P1$_0$-P1$_7$, P2$_0$-P2$_7$, P3$_0$-P3$_7$, P4$_0$-P4$_7$, P5$_0$-P5$_7$, P6$_0$-P6$_7$, P7$_0$-P7$_7$, P8$_0$-P8$_4$, P8$_6$, P8$_7$, P9$_0$-P9$_7$, P10$_0$-P10$_7$, P11$_0$-P11$_4$, P12$_0$-P12$_7$, P13$_0$-P13$_7$, P14$_0$-P14$_6$, P15$_0$-P15$_7$ (Note 1)	I_{OL}=200μA			0.45	V
V_{OL}	LOW output voltage X_{OUT} HIGHPOWER	I_{OL}=1mA			2.0	V
	LOW output voltage X_{OUT} LOWPOWER	I_{OL}=0.5mA			2.0	
	LOW output voltage X_{COUT} HIGHPOWER	With no load applied		0		V
	LOW output voltage X_{COUT} LOWPOWER	With no load applied		0		
V_{T+}-V_{T-}	Hysteresis: $\overline{HOLD}$, $\overline{RDY}$, TA0$_{IN}$-TA4$_{IN}$, TB0$_{IN}$-TB5$_{IN}$, $\overline{INT_0}$-$\overline{INT_5}$, $\overline{AD_{TRG}}$, $\overline{CTS_0}$-$\overline{CTS_4}$, CLK$_0$-CLK$_4$, TA0$_{OUT}$-TA4$_{OUT}$, $\overline{NMI}$, $\overline{KI_0}$-$\overline{KI_3}$, RxD0-RxD4, SCL$_2$-SCL$_4$, SDA$_2$-SDA$_4$		0.2		1.0	V
V_{T+}-V_{T-}	Hysteresis: $\overline{RESET}$		0.2		1.8	V
I_{IH}	HIGH input current: P0$_0$-P0$_7$, P1$_0$-P1$_7$, P2$_0$-P2$_7$, P3$_0$-P3$_7$, P4$_0$-P4$_7$, P5$_0$-P5$_7$, P6$_0$-P6$_7$, P7$_0$-P7$_7$, P8$_0$-P8$_7$, P9$_0$-P9$_7$,P10$_0$-P10$_7$, P11$_0$-P11$_4$, P12$_0$-P12$_7$,P13$_0$-P13$_7$, P14$_0$-P14$_6$, P15$_0$-P15$_7$, X_{IN}, $\overline{RESET}$, CNVss, BYTE (Note 1)	V_I=5V			5.0	μA
I_{IL}	LOW input current: P0$_0$-P0$_7$, P1$_0$-P1$_7$, P2$_0$-P2$_7$, P3$_0$-P3$_7$, P4$_0$-P4$_7$, P5$_0$-P5$_7$, P6$_0$-P6$_7$, P7$_0$-P7$_7$, P8$_0$-P8$_7$, P9$_0$-P9$_7$,P10$_0$-P10$_7$, P11$_0$-P11$_4$, P12$_0$-P12$_7$,P13$_0$-P13$_7$, P14$_0$-P14$_6$, P15$_0$-P15$_7$, X_{IN}, $\overline{RESET}$, CNVss, BYTE (Note 1)	V_I=0V			- 5.0	μA
R_{PULLUP}	Pull-up resistance: P0$_0$-P0$_7$, P1$_0$-P1$_7$, P2$_0$-P2$_7$, P3$_0$-P3$_7$, P4$_0$-P4$_7$, P5$_0$-P5$_7$, P6$_0$-P6$_7$, P7$_2$-P7$_7$, P8$_0$-P8$_4$, P8$_6$, P8$_7$, P9$_0$-P9$_7$,P10$_0$-P10$_7$, P11$_0$-P11$_4$, P12$_0$-P12$_7$, P13$_0$-P13$_7$, P14$_0$-P14$_6$, P15$_0$-P15$_7$ (Note 1)	V_I=0V	30.0	50.0	167.0	kΩ
R_{fXIN}	Feedback resistance X_{IN}			1.0		MΩ
R_{fXCIN}	Feedback resistance X_{CIN}			6.0		MΩ
V_{RAM}	RAM retention voltage	When clock is stopped	2.0			V
Icc	Power supply current Measuring condition: In single-chip mode, the output pins are open and other pins are V_{SS}	f(X_{IN})=20MHz Square wave, no division Mask ROM 128 KB version ROMless RAM 10 KB version(Note 2)		45.0	72.0	mA
		Mask ROM 256 KB version ROMless RAM 24 KB version (Note 2)		50.0	80.0	
		Flash memory version		50.0	80.0	
		f(X_{CIN})=32kHz Square wave Mask ROM 128 KB version ROMless RAM 10 KB version(Note 2)		90.0		μA
		Mask ROM 256 KB version ROMless RAM 24 KB version (Note 2)		100.0		
		Flash memory version		7.0		mA
		f(XCIN)=32kHz When a WAIT instruction is executed		4.0		μA
		Topr=25°C when clock is stopped Mask ROM 128 KB version ROMless RAM 10KB version (Note 2)			1.0	μA
		Mask ROM 256 KB version ROMless RAM 24KB version (Note 2)			2.0	
		Flash memory version			1.0	
		Topr=85°C when clock is stopped			20.0	

Note 1: Port P11 to P15 exist in 144-pin version.

Note 2: ROMless version exists in 144-pin version.

V_{CC} = 5V

Table 1.28.4. A-D conversion characteristics (referenced to V_{CC} = AV_{CC} = V_{REF} = 5V, Vss = AV_{SS} = 0V at Topr = 25°C, f(X_{IN}) = 20MHz unless otherwise specified)

Symbol	Parameter		Measuring condition		Standard			Unit
					Min.	Typ.	Max.	
-	Resolution		V_{REF} = V_{CC}				10	Bits
-	Absolute accuracy	Sample & hold function not available	V_{REF} = V_{CC} = 5V				±3	LSB
		Sample & hold function available (10bit)	V_{REF} = V_{CC} = 5V	AN_0 to AN_7 input			±3	LSB
				ANEX0, ANEX1 input, External op-amp connection mode			±7	LSB
		Sample & hold function available (8bit)	V_{REF} = V_{CC} = 5V				±2	LSB
R_{LADDER}	Ladder resistance		V_{REF} = V_{CC}		10		40	kΩ
t_{CONV}	Conversion time (10bit)				3.3			µs
t_{CONV}	Conversion time (8bit)				2.8			µs
t_{SAMP}	Sampling time				0.3			µs
V_{REF}	Reference voltage				2		V_{CC}	V
V_{IA}	Analog input voltage				0		V_{REF}	V

Note: Divide the frequency if f(X_{IN}) exceeds 10 MHz, and make ØAD equal to or lower than 10 MHz.

Table 1.28.5. D-A conversion characteristics (referenced to V_{CC} = 5V, V_{SS} = AV_{SS} = 0V, V_{REF} = 5V at Topr = 25°C, f(X_{IN}) = 20MHz unless otherwise specified)

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
−	Resolution				8	Bits
−	Absolute accuracy				1.0	%
t_{su}	Setup time				3	µs
R_O	Output resistance		4	10	20	kΩ
I_{VREF}	Reference power supply input current	(Note)			1.5	mA

Note: This applies when using one D-A converter, with the D-A register for the unused D-A converter set to "00_{16}".

The A-D converter's ladder resistance is not included.

Also, when the contents of D-A register is except "00_{16}" and the Vref is unconnected at the A-D control register 1, I_{VREF} is sent.

VCC = 5V

Timing requirements (referenced to VCC = 5V, VSS = 0V at Topr = 25°C unless otherwise specified)

Table 1.28.6. External clock input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External clock input cycle time	50		ns
$t_{w(H)}$	External clock input HIGH pulse width	22		ns
$t_{w(L)}$	External clock input LOW pulse width	22		ns
t_r	External clock rise time		5	ns
t_f	External clock fall time		5	ns

Table 1.28.7. Memory expansion and microprocessor modes

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{ac1(RD-DB)}$	Data input access time (RD standard, no wait)		(Note)	ns
$t_{ac1(AD-DB)}$	Data input access time (AD standard, CS standard, no wait)		(Note)	ns
$t_{ac2(RD-DB)}$	Data input access time (RD standard, with wait)		(Note)	ns
$t_{ac2(AD-DB)}$	Data input access time (AD standard, CS standard, with wait)		(Note)	ns
$t_{ac3(RD-DB)}$	Data input access time (RD standard, when accessing multiplex bus area)		(Note)	ns
$t_{ac3(AD-DB)}$	Data input access time (AD standard, CS standard, when accessing multiplex bus area)		(Note)	ns
$t_{ac4(RAS-DB)}$	Data input access time (RAS standard, DRAM access)		(Note)	ns
$t_{ac4(CAS-DB)}$	Data input access time (CAS standard, DRAM access)		(Note)	ns
$t_{ac4(CAD-DB)}$	Data input access time (CAD standard, DRAM access)		(Note)	ns
$t_{su(DB-BCLK)}$	Data input setup time	26		ns
$t_{su(RDY-BCLK)}$	$\overline{\text{RDY}}$ input setup time	26		ns
$t_{su(HOLD-BCLK)}$	$\overline{\text{HOLD}}$ input setup time	30		ns
$t_{h(RD-DB)}$	Data input hold time	0		ns
$t_{h(CAS-DB)}$	Data input hold time	0		ns
$t_{h(BCLK-RDY)}$	$\overline{\text{RDY}}$ input hold time	0		ns
$t_{h(BCLK-HOLD)}$	$\overline{\text{HOLD}}$ input hold time	0		ns
$t_{d(BCLK-HLDA)}$	HLDA output delay time		25	ns

Note: Calculated according to the BCLK frequency as follows:
Note that inserting wait or using lower operation frequency f(BCLK) is needed when calculated value is negative.

$$t_{ac1(RD-DB)} = \frac{10^9}{f_{(BCLK)} \text{ X } 2} - 35 \quad \text{[ns]}$$

$$t_{ac1(AD-DB)} = \frac{10^9}{f_{(BCLK)}} - 35 \quad \text{[ns]}$$

$$t_{ac2(RD-DB)} = \frac{10^9 \text{ X } m}{f_{(BCLK)} \text{ X } 2} - 35 \quad \text{[ns] (m=3, 5 and 7 when 1 wait, 2 wait and 3 wait, respectively)}$$

$$t_{ac2(AD-DB)} = \frac{10^9 \text{ X } n}{f_{(BCLK)}} - 35 \quad \text{[ns] (n=2, 3 and 4 when 1 wait, 2 wait and 3 wait, respectively)}$$

$$t_{ac3(RD-DB)} = \frac{10^9 \text{ X } m}{f_{(BCLK)} \text{ X } 2} - 35 \quad \text{[ns] (m=3 and 5 when 2 wait and 3 wait, respectively)}$$

$$t_{ac3(AD-DB)} = \frac{10^9 \text{ X } n}{f_{(BCLK)} \text{ X } 2} - 35 \quad \text{[ns] (n=5 and 7 when 2 wait and 3 wait, respectively)}$$

$$t_{ac4(RAS-DB)} = \frac{10^9 \text{ X } m}{f_{(BCLK)} \text{ X } 2} - 35 \quad \text{[ns] (m=3 and 5 when 1 wait and 2 wait, respectively)}$$

$$t_{ac4(CAS-DB)} = \frac{10^9 \text{ X } n}{f_{(BCLK)} \text{ X } 2} - 35 \quad \text{[ns] (n=1 and 3 when 1 wait and 2 wait, respectively)}$$

$$t_{ac4(CAD-DB)} = \frac{10^9 \text{ X } l}{f_{(BCLK)}} - 35 \quad \text{[ns] (l=1 and 2 when 1 wait and 2 wait, respectively)}$$

VCC = 5V

Timing requirements (referenced to VCC = 5V, VSS = 0V at Topr = 25°C unless otherwise specified)

Table 1.28.8. Timer A input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN input cycle time	100		ns
tw(TAH)	TAiIN input HIGH pulse width	40		ns
tw(TAL)	TAiIN input LOW pulse width	40		ns

Table 1.28.9. Timer A input (gating input in timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN input cycle time	400		ns
tw(TAH)	TAiIN input HIGH pulse width	200		ns
tw(TAL)	TAiIN input LOW pulse width	200		ns

Table 1.28.10. Timer A input (external trigger input in one-shot timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(TA)	TAiIN input cycle time	200		ns
tw(TAH)	TAiIN input HIGH pulse width	100		ns
tw(TAL)	TAiIN input LOW pulse width	100		ns

Table 1.28.11. Timer A input (external trigger input in pulse width modulation mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tw(TAH)	TAiIN input HIGH pulse width	100		ns
tw(TAL)	TAiIN input LOW pulse width	100		ns

Table 1.28.12. Timer A input (up/down input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
tc(UP)	TAiOUT input cycle time	2000		ns
tw(UPH)	TAiOUT input HIGH pulse width	1000		ns
tw(UPL)	TAiOUT input LOW pulse width	1000		ns
tsu(UP-TIN)	TAiOUT input setup time	400		ns
th(TIN-UP)	TAiOUT input hold time	400		ns

VCC = 5V

Timing requirements (referenced to VCC = 5V, VSS = 0V at Topr = 25°C unless otherwise specified)

Table 1.28.13. Timer B input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time (counted on one edge)	100		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on one edge)	40		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on one edge)	40		ns
$t_{c(TB)}$	TBiIN input cycle time (counted on both edges)	200		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width (counted on both edges)	80		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width (counted on both edges)	80		ns

Table 1.28.14. Timer B input (pulse period measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	400		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	200		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	200		ns

Table 1.28.15. Timer B input (pulse width measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBiIN input cycle time	400		ns
$t_{w(TBH)}$	TBiIN input HIGH pulse width	200		ns
$t_{w(TBL)}$	TBiIN input LOW pulse width	200		ns

Table 1.28.16. A-D trigger input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{\text{ADTRG}}$ input cycle time (trigger able minimum)	1000		ns
$t_{w(ADL)}$	$\overline{\text{ADTRG}}$ input LOW pulse width	125		ns

Table 1.28.17. Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	200		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	100		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	100		ns
$t_{d(C-Q)}$	TxDi output delay time		80	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	30		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

Table 1.28.18. External interrupt $\overline{\text{INTi}}$ inputs

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{\text{INTi}}$ input HIGH pulse width	250		ns
$t_{w(INL)}$	$\overline{\text{INTi}}$ input LOW pulse width	250		ns

VCC = 5V

Switching characteristics (referenced to VCC = 5V, VSS = 0V at Topr = 25°C, CM15 = “1” unless otherwise specified)

Table 1.28.19. Memory expansion mode and microprocessor mode (no wait)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.28.1		18	ns
$t_{h(BCLK-AD)}$	Address output hold time (BCLK standard)		-3		ns
$t_{h(RD-AD)}$	Address output hold time (RD standard)		0		ns
$t_{h(WR-AD)}$	Address output hold time (WR standard)		(Note)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			18	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (BCLK standard)		-3		ns
$t_{h(RD-CS)}$	Chip select output hold time (RD standard)		0		ns
$t_{h(WR-CS)}$	Chip select output hold time (WR standard)		(Note)		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time			18	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time		– 2		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			10	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		-5		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			18	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		-3		ns
$t_{d(DB-WR)}$	Data output delay time (WR standard)		(Note)		ns
$t_{h(WR-DB)}$	Data output hold time (WR standard)		(Note)		ns
$t_{w(WR)}$	WR signal width		(Note)		ns

Note: Calculated according to the BCLK frequency as follows:

$$t_{d(DB-WR)} = \frac{10^9}{f_{(BCLK)}} - 20 \quad [ns]$$

$$t_{h(WR-DB)} = \frac{10^9}{f_{(BCLK)} \text{ X } 2} - 10 \quad [ns]$$

$$t_{h(WR-AD)} = \frac{10^9}{f_{(BCLK)} \text{ X } 2} - 10 \quad [ns]$$

$$t_{h(WR-CS)} = \frac{10^9}{f_{(BCLK)} \text{ X } 2} - 10 \quad [ns]$$

$$t_{w(WR)} = \frac{10^9}{f_{(BCLK)} \text{ X } 2} - 15 \quad [ns]$$

VCC = 5V

Switching characteristics (referenced to VCC = 5V, VSS = 0V at Topr = 25°C unless otherwise specified)

Table 1.28.20. Memory expansion mode and microprocessor mode (with wait, accessing external memory)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.28.1		18	ns
$t_{h(BCLK-AD)}$	Address output hold time (BCLK standard)		– 3		ns
$t_{h(RD-AD)}$	Address output hold time (RD standard)		0		ns
$t_{h(WR-AD)}$	Address output hold time (WR standard)		(Note)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			18	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (BCLK standard)		– 3		ns
$t_{h(RD-CS)}$	Chip select output hold time (RD standard)		0		ns
$t_{h(WR-CS)}$	Chip select output hold time (WR standard)		(Note)		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time			18	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time		– 2		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			10	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		– 5		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			18	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		– 3		ns
$t_{d(DB-WR)}$	Data output delay time (WR standard)		(Note)		ns
$t_{h(WR-DB)}$	Data output hold time (WR standard)		(Note)		ns
$t_{w(WR)}$	WR signal width		(Note)		ns

Note: Calculated according to the BCLK frequency as follows:

$$t_{d(DB-WR)} = \frac{10^9 \times n}{f_{(BCLK)}} - 20$$ [ns] (n=1, 2 and 3 when 1 wait, 2 wait and 3 wait, respectively)

$$t_{h(WR-DB)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10$$ [ns]

$$t_{h(WR-AD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10$$ [ns]

$$t_{h(WR-CS)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10$$ [ns]

$$t_{w(WR)} = \frac{10^9 \times n}{f_{(BCLK)} \times 2} - 15$$ [ns] (n=1, 3 and 5 when 1 wait, 2 wait and 3 wait, respectively)

VCC = 5V

Switching characteristics (referenced to VCC = 5V, VSS = 0V at Topr = 25°C unless otherwise specified)

Table 1.28.21. Memory expansion mode and microprocessor mode (with wait, accessing external memory, multiplex bus area selected)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.28.1		18	ns
$t_{h(BCLK-AD)}$	Address output hold time (BCLK standard)		-3		ns
$t_{h(RD-AD)}$	Address output hold time (RD standard)		(Note)		ns
$t_{h(WR-AD)}$	Address output hold time (WR standard)		(Note)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			18	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (BCLK standard)		-3		ns
$t_{h(RD-CS)}$	Chip select output hold time (RD standard)		(Note)		ns
$t_{h(WR-CS)}$	Chip select output hold time (WR standard)		(Note)		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			18	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		-5		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			18	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		-3		ns
$t_{d(DB-WR)}$	Data output delay time (WR standard)		(Note)		ns
$t_{h(WR-DB)}$	Data output hold time (WR standard)		(Note)		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time (BCLK standard)			18	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time (BCLK standard)		– 2		ns
$t_{d(AD-ALE)}$	ALE signal output delay time (address standard)		(Note)		ns
$t_{h(ALE-AD)}$	ALE signal output hold time (address standard)		(Note)		ns
$t_{dz(RD-AD)}$	Address output flowting start time			8	ns
$t_{h(BCLK-DB)}$	Data output hold time (BCLK standard)		-5		ns

Note: Calculated according to the BCLK frequency as follows:

$$t_{h(RD-AD)} = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$t_{h(WR-AD)} = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$t_{h(RD-CS)} = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$t_{h(WR-CS)} = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$t_{d(DB-WR)} = \frac{10^9 \times m}{f(BCLK) \times 2} - 25 \quad [ns]$$ (m=3 and 5 when 2 wait and 3 wait, respectively)

$$t_{h(WR-DB)} = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$t_{d(AD-ALE)} = \frac{10^9}{f(BCLK) \times 2} - 23 \quad [ns]$$

$$t_{h(ALE-AD)} = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

VCC = 5V

Switching characteristics (referenced to Vcc = 5V, Vss = 0V at Topr = 25°C unless otherwise specified)

Table 1.28.22. Memory expansion mode and microprocessor mode (with wait, accessing external memory, DRAM area selected)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-RAD)}$	Row address output delay time	Figure 1.28.1		18	ns
$t_{h(BCLK-RAD)}$	Row address output hold time (BCLK standard)		-3		ns
$t_{d(BCLK-CAD)}$	String address output delay time			18	ns
$t_{h(BCLK-CAD)}$	String address output hold time (BCLK standard)		-3		ns
$t_{h(RAS-RAD)}$	Row address output hold time after RAS output		(Note)		ns
$t_{d(BCLK-RAS)}$	RAS output delay time (BCLK standard)			18	ns
$t_{h(BCLK-RAS)}$	RAS output hold time (BCLK standard)		-3		ns
t_{RP}	RAS "H" hold time		(Note)		ns
$t_{d(BCLK-CAS)}$	CAS output delay time (BCLK standard)			18	ns
$t_{h(BCLK-CAS)}$	CAS output hold time (BCLK standard)		-3		ns
$t_{d(BCLK-DW)}$	Data output delay time (BCLK standard)			18	ns
$t_{h(BCLK-DW)}$	Data output hold time (BCLK standard)		-5		ns
$t_{su(DB-CAS)}$	CAS after DB output setup time		(Note)		ns
$t_{h(BCLK-DB)}$	DB signal output hold time (BCLK standard)		-7		ns
$t_{su(CAS-RAS)}$	CAS before RAS setup time (refresh)		(Note)		ns

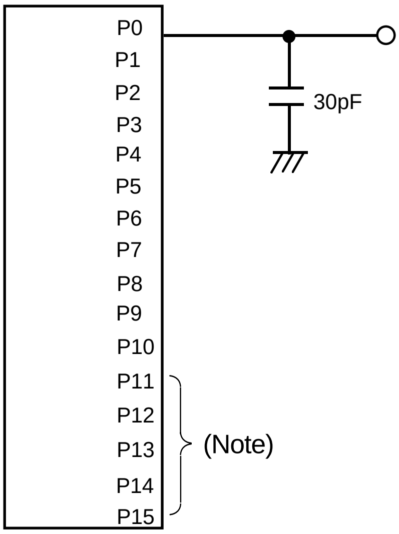
Note: Calculated according to the BCLK frequency as follows:

$$t_{h(RAS-RAD)} = \frac{10^{9}}{f_{(BCLK)} \text{ X } 2} - 13 \quad [\text{ns}]$$

$$t_{RP} = \frac{10^{9} \text{ X } 3}{f_{(BCLK)} \text{ X } 2} - 20 \quad [\text{ns}]$$

$$t_{su(DB-CAS)} = \frac{10^{9}}{f_{(BCLK)}} - 20 \quad [\text{ns}]$$

$$t_{su(CAS-RAS)} = \frac{10^{9}}{f_{(BCLK)} \text{ X } 2} - 13 \quad [\text{ns}]$$

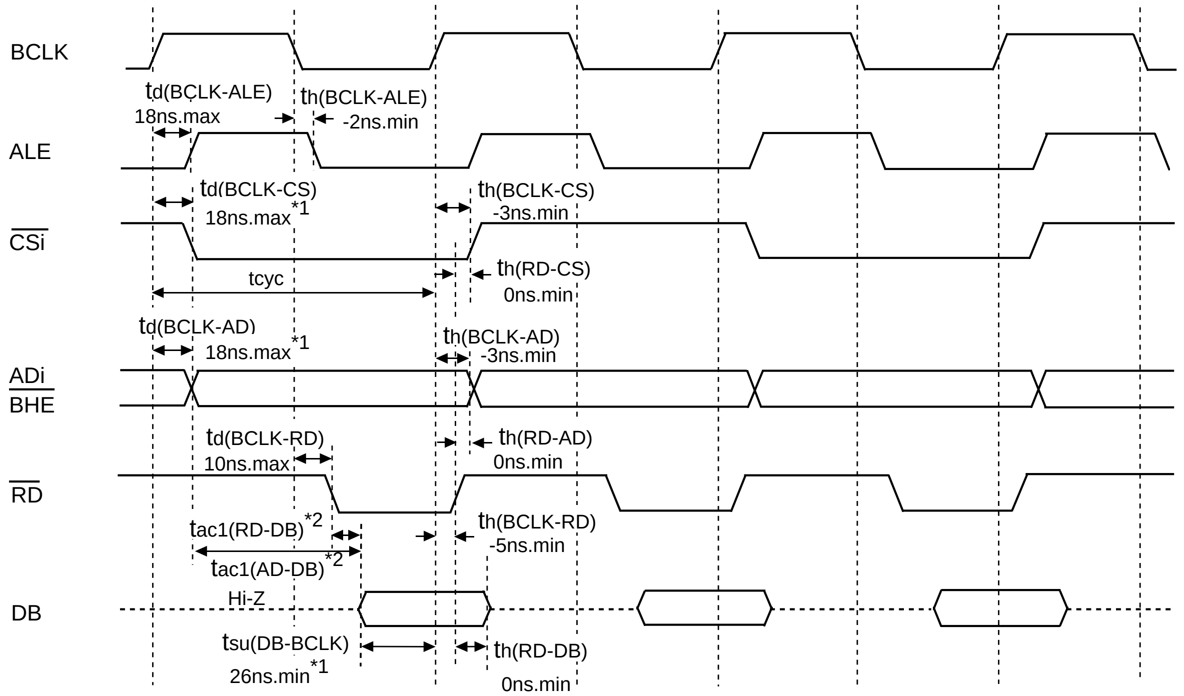


Note: Port P11 to P15 exist in 144-pin version.

Figure 1.28.1. Port P0 to P15 measurement circuit

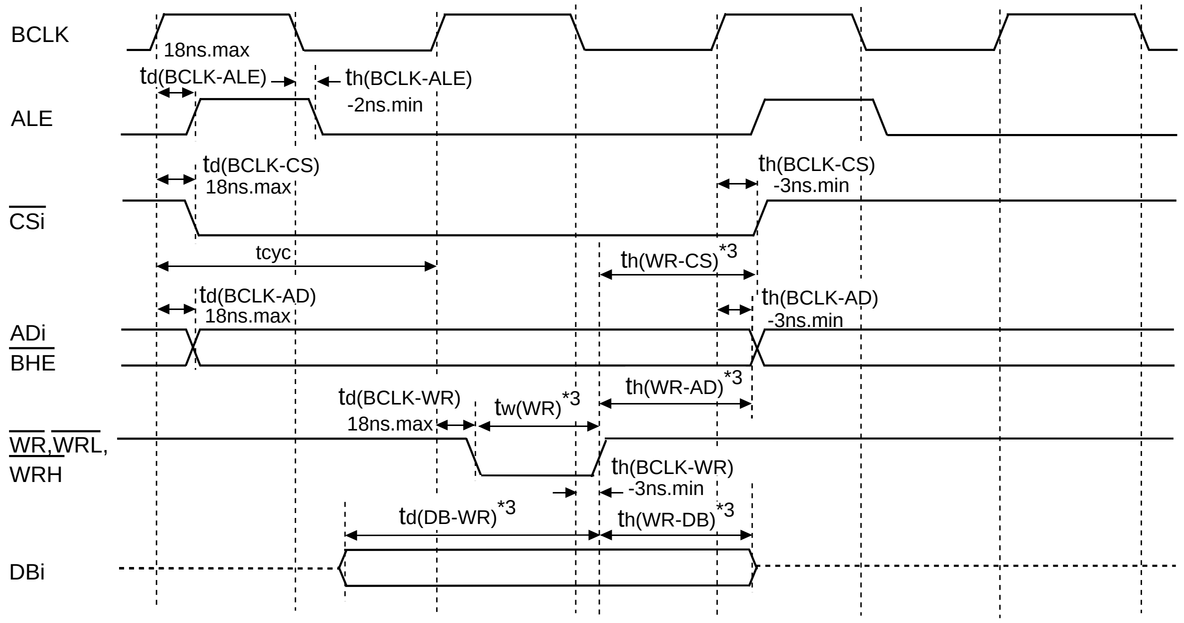
Vcc=5V

Memory expansion Mode and Microprocessor Mode (without wait)
Read Timing



*1:It is a guarantee value with being alone. 35ns.max garantees as $t_{d(BCLK-AD)}$+$t_{su(DB-BCLK)}$.
*2:It depends on operation frequency.
$t_{ac1(RD-DB)}$=(tcyc/2-35)ns.max
$t_{ac1(AD-DB)}$=(tcyc-35)ns.max

Write Timing (Written by 2 cycles in selecting no wait)



*3:It depends on operation frequency.
$t_{d(DB-WR)}$=(tcyc-20)ns.min
$t_{h(WR-DB)}$=(tcyc/2-10)ns.min
$t_{h(WR-AD)}$=(tcyc/2-10)ns.min
$t_{h(WR-CS)}$=(tcyc/2-10)ns.min
$t_{w(WR)}$=(tcyc/2-15)ns.min

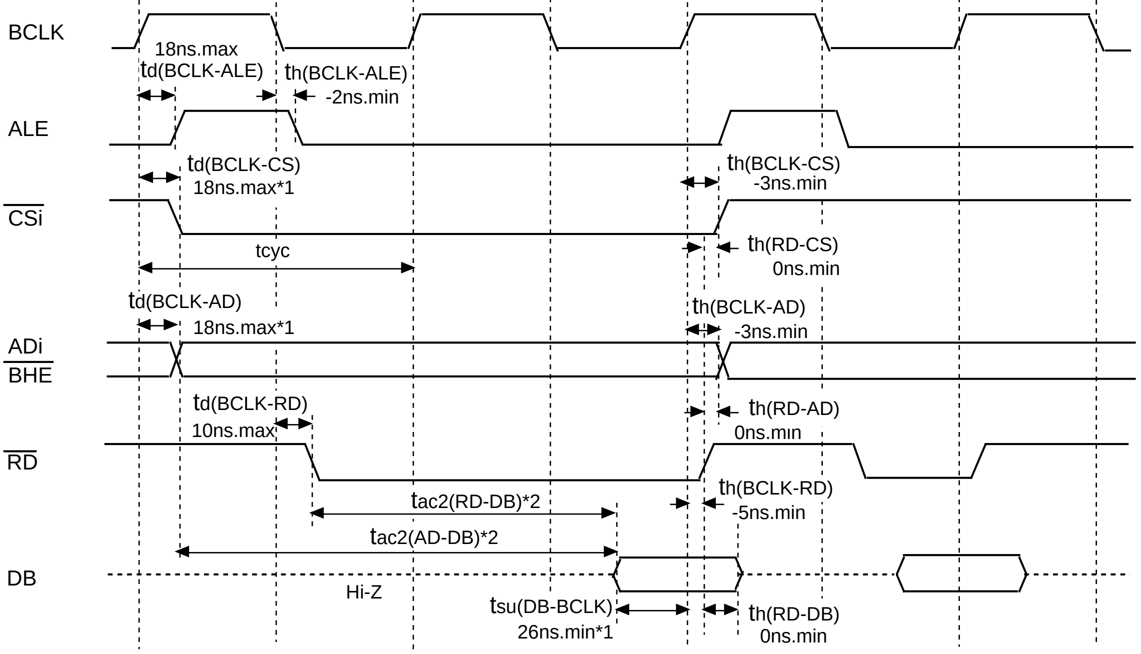
Measuring conditions
- V_{CC}=5V±10%
- Input timing voltage :Determined with V_{IH}=2.5V, V_{IL}=0.8V
- Output timing voltage :Determined with V_{OH}=2.0V, V_{OL}=0.8V

Figure 1.28.2. VCC=5V timing diagram (1)

Vcc=5V

Memory expansion Mode and Microprocessor Mode (with 1 wait)

Read Timing



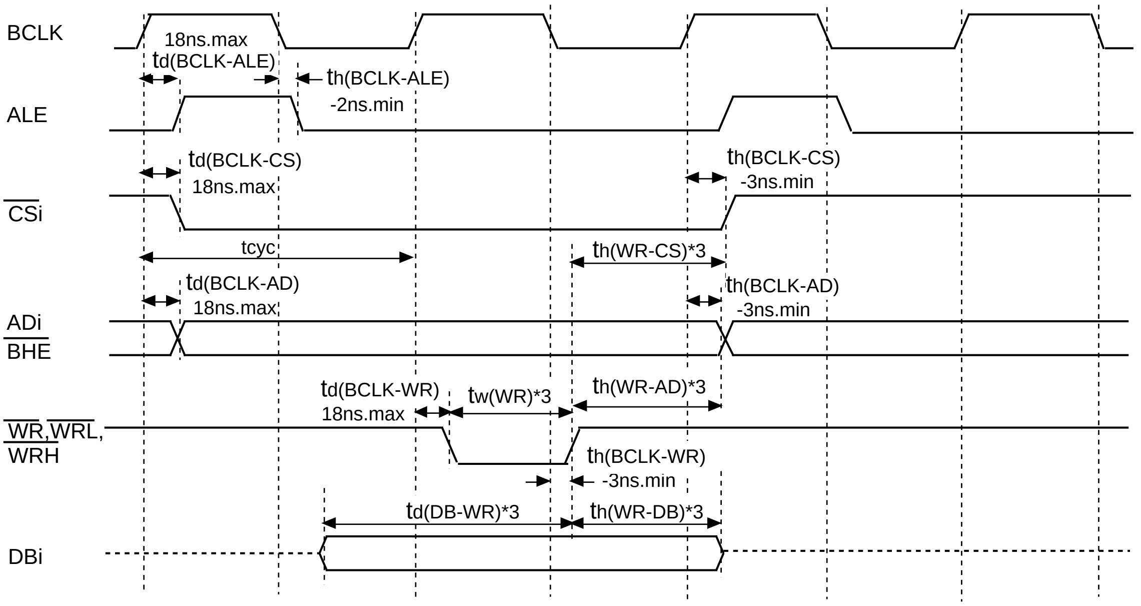
*1:It is a guarantee value with being alone. 35ns.max garantees as $t_{d(BCLK-AD)}$+$t_{su(DB-BCLK)}$.

*2:It depends on operation frequency.

$t_{ac2(RD-DB)}$=(tcyc/2 x m-35)ns.max (m=3, 5 and 7 when 1 wait, 2 wait and 3 wait, respectively.)

$t_{ac2(AD-DB)}$=(tcyc x n-35)ns.max (n=2, 3 and 4 when 1 wait, 2 wait and 3 wait, respectively.)

Write Timing



*3:It depends on operation frequency.

$t_{d(DB-WR)}$=(tcyc x n-20)ns.min
(n=1, 2 and 3 when 1 wait, 2 wait and 3 wait, respectively.)

$t_{h(WR-DB)}$=(tcyc/2-10)ns.min

$t_{h(WR-AD)}$=(tcyc/2-10)ns.min

$t_{h(WR-CS)}$=(tcyc/2-10)ns.min

$t_{w(WR)}$=(tcyc/2 x n-15)ns.min
(n=1, 3 and 5 when 1 wait, 2 wait and 3 wait, respectively.)

Measuring conditions

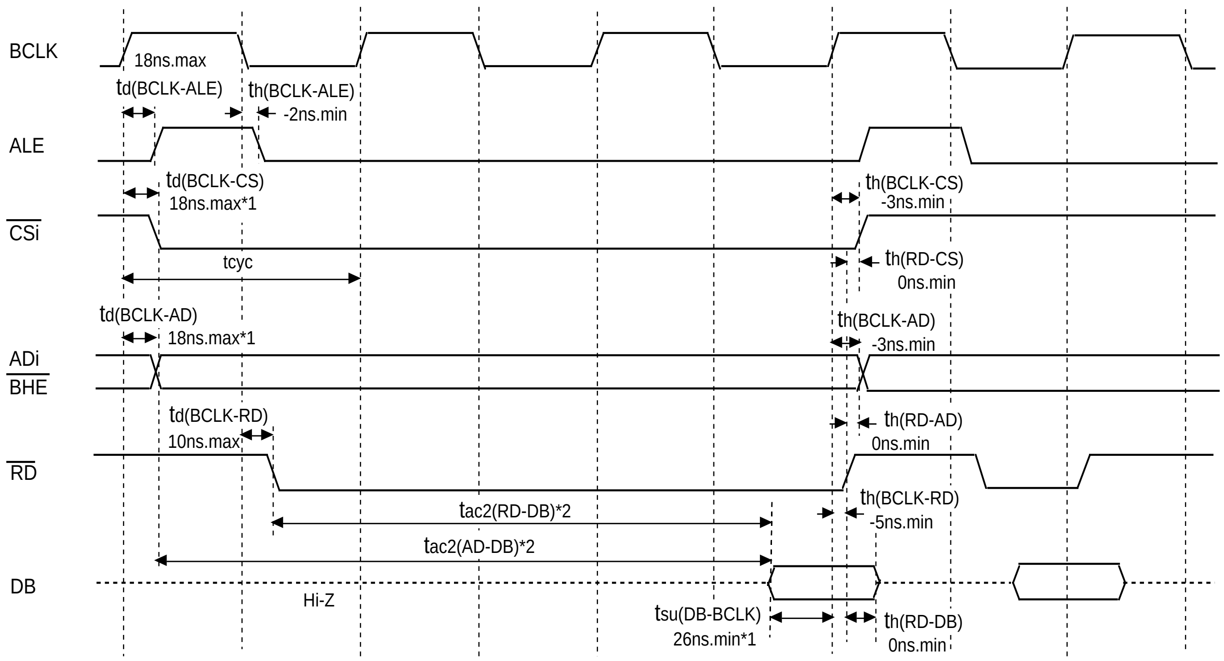
- V_{CC}=5V±10%
- Input timing voltage
  :Determined with V_{IH}=2.5V, V_{IL}=0.8V
- Output timing voltage
  :Determined with V_{OH}=2.0V, V_{OL}=0.8V

Figure 1.28.3. VCC=5V timing diagram (2)

Memory expansion Mode and Microprocessor Mode (with 2 wait)

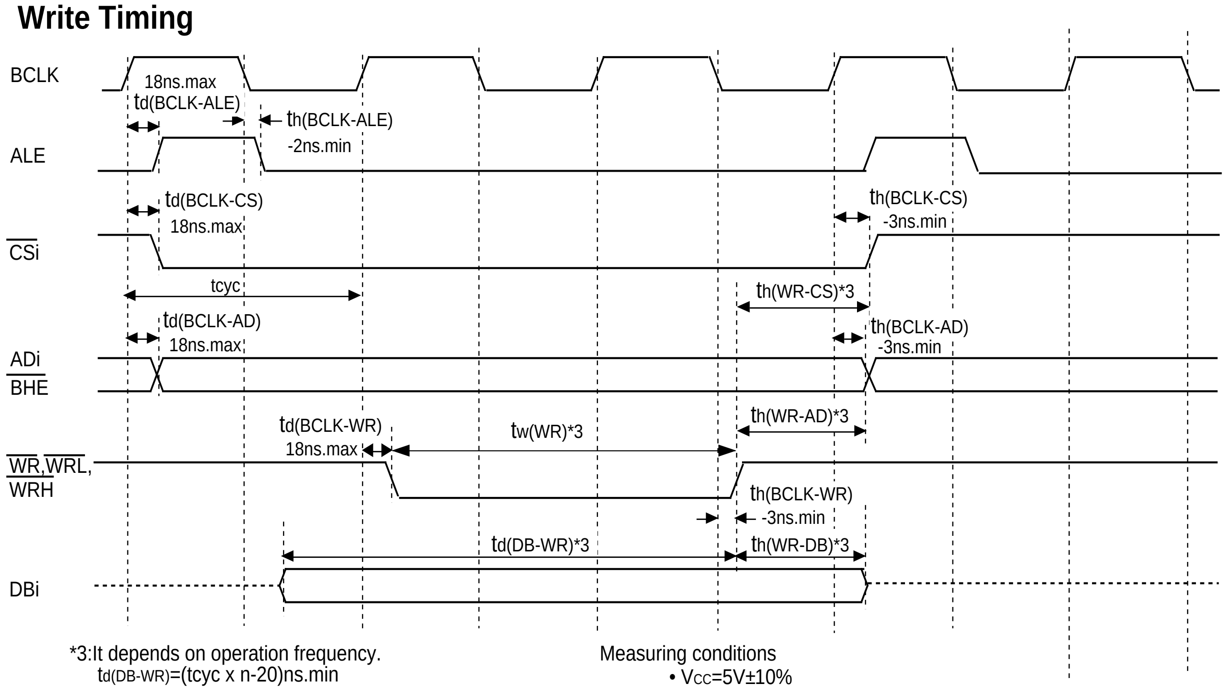
Vcc=5V

Read Timing



*1:It is a guarantee value with being alone. 35ns.max garantees as td(BCLK-AD)+tsu(DB-BCLK).
*2:It depends on operation frequency.
tac2(RD-DB)=(tcyc/2 x m-35)ns.max (m=3, 5 and 7 when 1 wait, 2 wait and 3 wait, respectively.)
tac2(AD-DB)=(tcyc x n-35)ns.max (n=2, 3 and 4 when 1 wait, 2 wait and 3 wait, respectively.)

Write Timing



*3:It depends on operation frequency.
td(DB-WR)=(tcyc x n-20)ns.min
(n=1, 2 and 3 when 1 wait, 2 wait and 3 wait, respectively.)
th(WR-DB)=(tcyc/2-10)ns.min
th(WR-AD)=(tcyc/2-10)ns.min
th(WR-CS)=(tcyc/2-10)ns.min
tw(WR)=(tcyc/2 x n-15)ns.min
(n=1, 3 and 5 when 1 wait, 2 wait and 3 wait, respectively.)

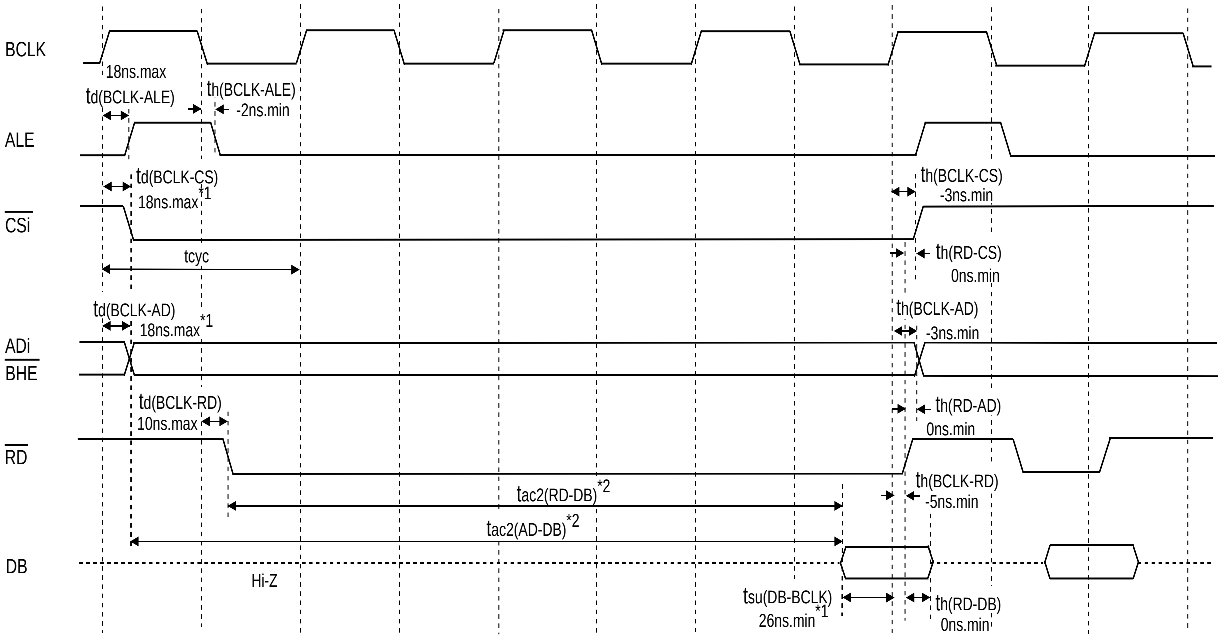
Measuring conditions
- Vcc=5V±10%
- Input timing voltage
  :Determined with VIH=2.5V, VIL=0.8V
- Output timing voltage
  :Determined with VOH=2.0V, VOL=0.8V

Figure 1.28.4. Vcc=5V timing diagram (3)

Memory expansion Mode and Microprocessor Mode (with 3 wait)

Vcc=5V

Read Timing



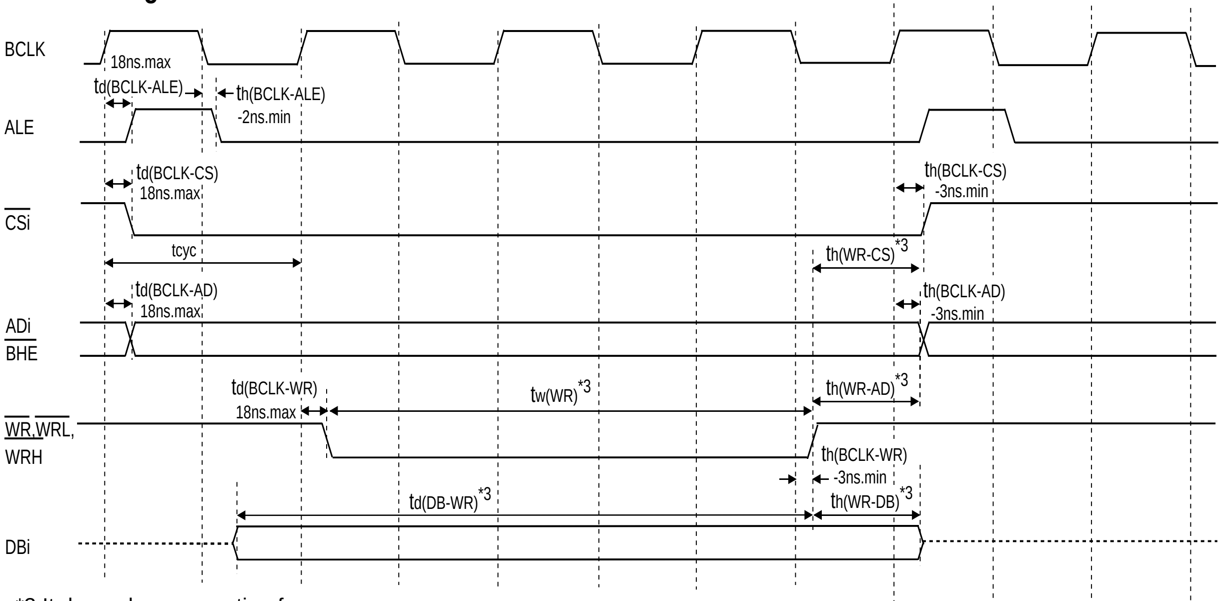
*1:It is a guarantee value with being alone. 35ns.max garantees as td(BCLK-AD)+tsu(DB-BCLK).

*2:It depends on operation frequency.

tac2(RD-DB)=(tcyc/2 x m-35)ns.max (m=3, 5 and 7 when 1 wait, 2 wait and 3 wait, respectively.)

tac2(AD-DB)=(tcyc x n-35)ns.max (n=2, 3 and 4 when 1 wait, 2 wait and 3 wait, respectively.)

Write Timing



*3:It depends on operation frequency.

td(DB-WR)=(tcyc x n-20)ns.min

(n=1, 2 and 3 when 1 wait, 2 wait and 3 wait, respectively.)

th(WR-DB)=(tcyc/2-10)ns.min

th(WR-AD)=(tcyc/2-10)ns.min

th(WR-CS)=(tcyc/2-10)ns.min

tw(WR)=(tcyc/2 x n-15)ns.min

(n=1, 3 and 5 when 1 wait, 2 wait and 3 wait, respectively.)

Measuring conditions

- VCC=5V±10%
- Input timing voltage
  :Determined with VIH=2.5V, VIL=0.8V
- Output timing voltage
  :Determined with VOH=2.0V, VOL=0.8V

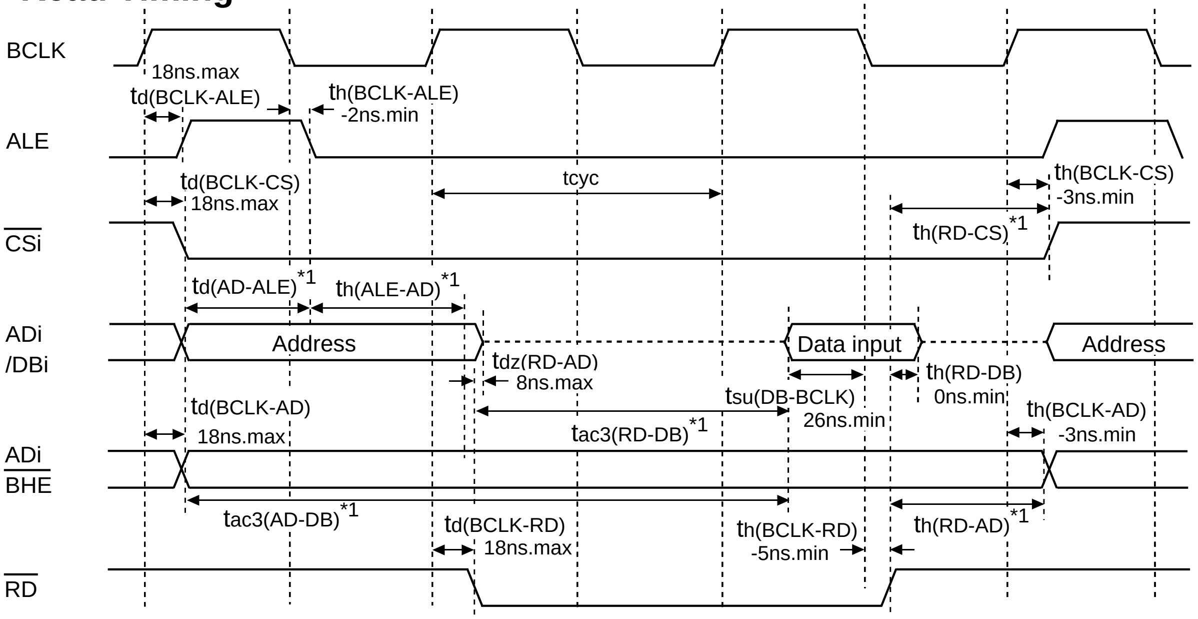
Figure 1.28.5. VCC=5V timing diagram (4)

Memory expansion Mode and Microprocessor Mode

Vcc=5V

(When accessing external memory area with 2 wait, and select multiplexed bus))

Read Timing



*1:It depends on operation frequency.

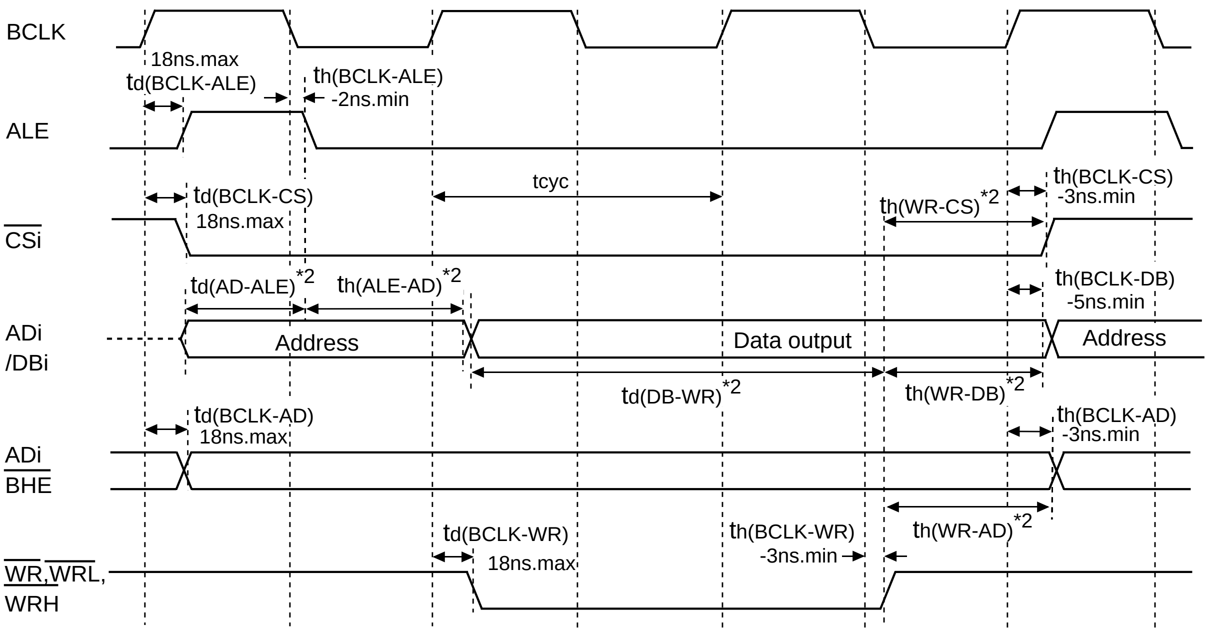
td(AD-ALE)=(tcyc/2-23)ns.min

th(ALE-AD)=(tcyc/2-10)ns.min, th(RD-AD)=(tcyc/2-10)ns.min, th(RD-CS)=(tcyc/2-10)ns.min

tac3(RD-DB)=(tcyc/2 x m-35)ns.max (m=3 and 5 when 2 wait and 3 wait, respectively.)

tac3(AD-DB)=(tcyc/2 x n-35)ns.max (n=5 and 7 when 2 wait and 3 wait, respectively.)

Write Timing



*2:It depends on operation frequency.

td(AD-ALE)=(tcyc/2-23)ns.min

th(ALE-AD)=(tcyc/2-10)ns.min, th(WR-AD)=(tcyc/2-10)ns.min

th(WR-CS)=(tcyc/2-10)ns.min, th(WR-DB)=(tcyc/2-10)ns.min

td(DB-WR)=(tcyc/2 x m-25)ns.min

(m=3 and 5 when 2 wait and 3 wait, respectively.)

Measuring conditions

- Vcc=5V±10%
- Input timing voltage
  :Determined with VIH=2.5V, VIL=0.8V
- Output timing voltage
  :Determined with VOH=2.0V, VOL=0.8V

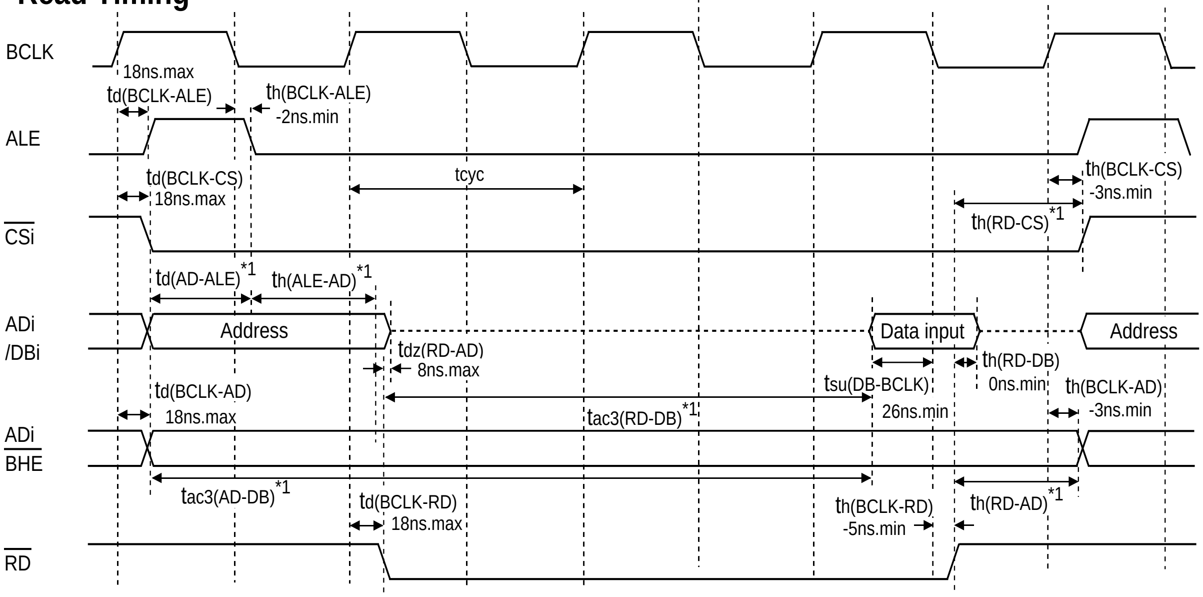
Figure 1.28.6. Vcc=5V timing diagram (5)

Memory expansion Mode and Microprocessor Mode

Vcc=5V

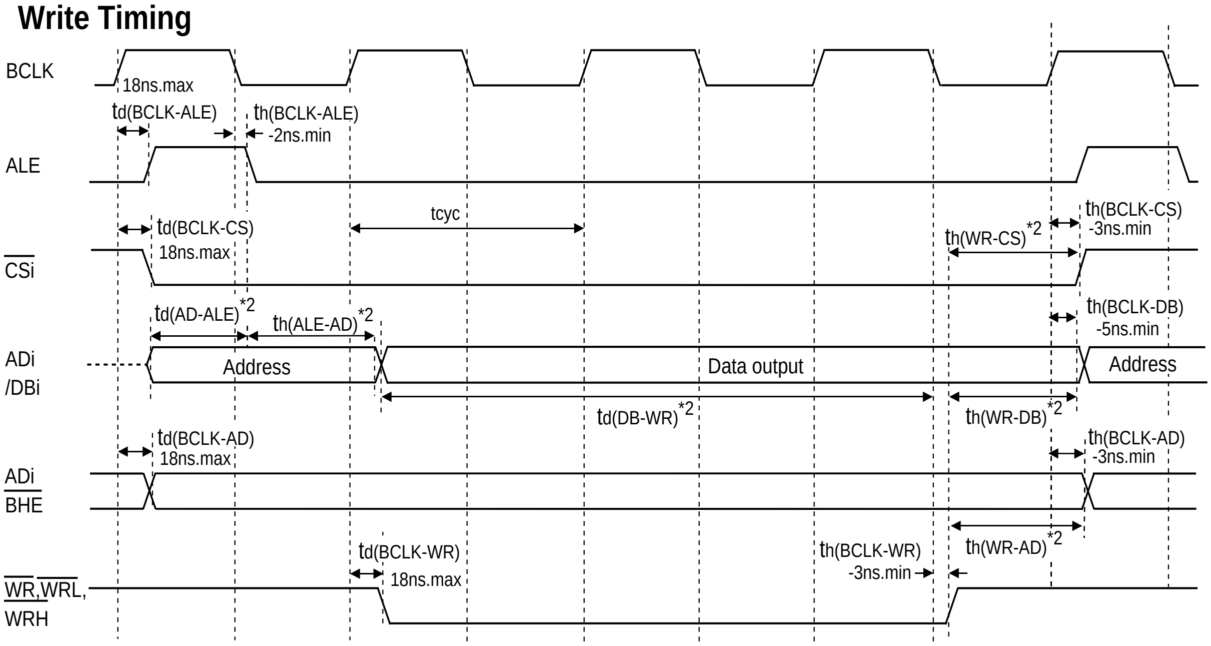
(When accessing external memory area with 3 wait, and select multiplexed bus))

Read Timing



*1:It depends on operation frequency.
$t_{d(AD-ALE)}$=(tcyc/2-23)ns.min
$t_{h(ALE-AD)}$=(tcyc/2-10)ns.min, $t_{h(RD-AD)}$=(tcyc/2-10)ns.min, $t_{h(RD-CS)}$=(tcyc/2-10)ns.min
$t_{ac3(RD-DB)}$=(tcyc/2 x m-35)ns.max (m=3 and 5 when 2 wait and 3 wait, respectively.)
$t_{ac3(AD-DB)}$=(tcyc/2 x n-35)ns.max (n=5 and 7 when 2 wait and 3 wait, respectively.)

Write Timing



*2:It depends on operation frequency.
$t_{d(AD-ALE)}$=(tcyc/2-23)ns.min
$t_{h(ALE-AD)}$=(tcyc/2-10)ns.min, $t_{h(WR-AD)}$=(tcyc/2-10)ns.min
$t_{h(WR-CS)}$=(tcyc/2-10)ns.min, $t_{h(WR-DB)}$=(tcyc/2-10)ns.min
$t_{d(DB-WR)}$=(tcyc/2 x m-25)ns.min
(m=3 and 5 when 2 wait and 3 wait, respectively.)

Measuring conditions
- V_{CC}=5V±10%
- Input timing voltage
  :Determined with V_{IH}=2.5V, V_{IL}=0.8V
- Output timing voltage
  :Determined with V_{OH}=2.0V, V_{OL}=0.8V

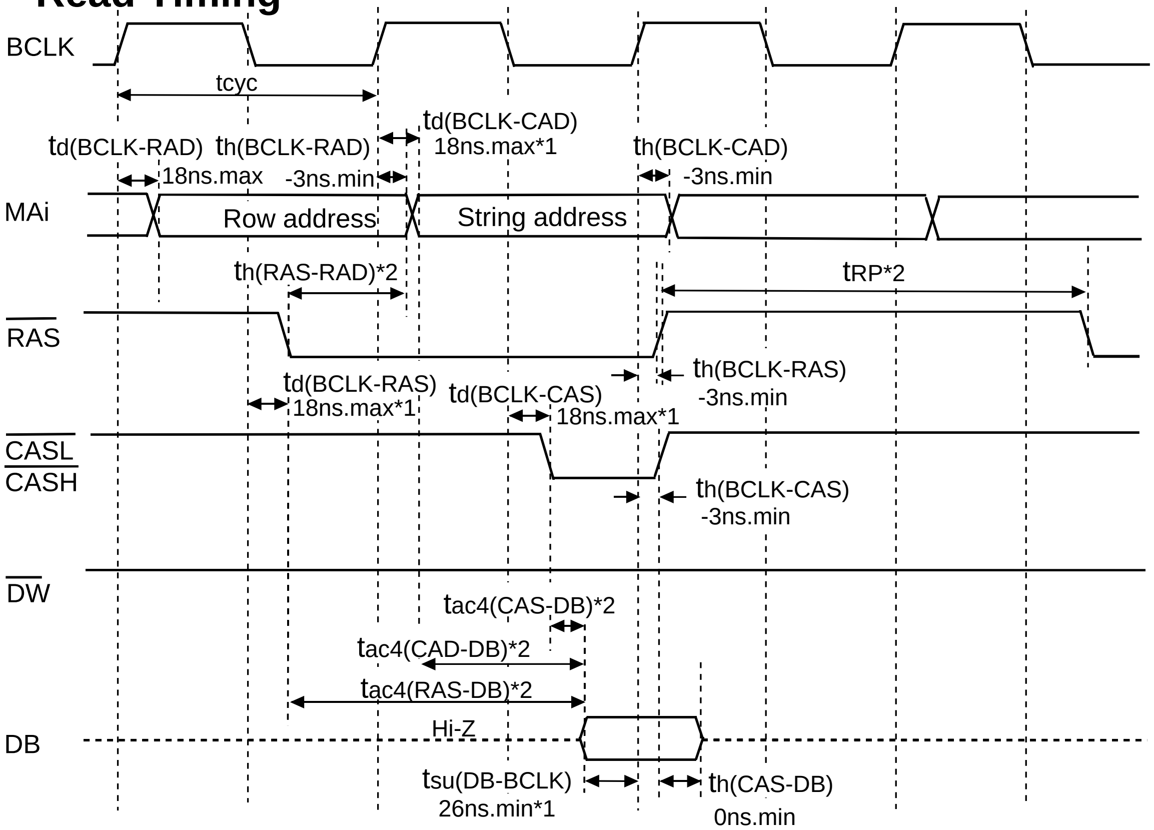
Figure 1.28.7. V_{CC}=5V timing diagram (6)

Vcc=5V

Memory expansion Mode and Microprocessor Mode

(When accessing DRAM area with 1 wait)

Read Timing



*1:It is a guarantee value with being alone. 35ns.max garantees as follows:

$t_{d(BCLK-RAS)} + t_{su(DB-BCLK)}$

$t_{d(BCLK-CAS)} + t_{su(DB-BCLK)}$

$t_{d(BCLK-CAD)} + t_{su(DB-BCLK)}$

*2:It depends on operation frequency.

$t_{ac4(RAS-DB)}$=(tcyc/2 x m-35)ns.max (m=3 and 5 when 1 wait and 2 wait, respectively.)

$t_{ac4(CAS-DB)}$=(tcyc/2 x n-35)ns.max (n=1 and 3 when 1 wait and 2 wait, respectively.)

$t_{ac4(CAD-DB)}$=(tcyc x l-35)ns.max (l=1 and 2 when 1 wait and 2 wait, respectively.)

$t_{h(RAS-RAD)}$=(tcyc/2-13)ns.min

t_{RP}=(tcyc/2 x 3-20)ns.min

Measuring conditions

- V_{CC}=5V±10%
- Input timing voltage
  :Determined with V_{IH}=2.5V, V_{IL}=0.8V
- Output timing voltage
  :Determined with V_{OH}=2.0V, V_{OL}=0.8V

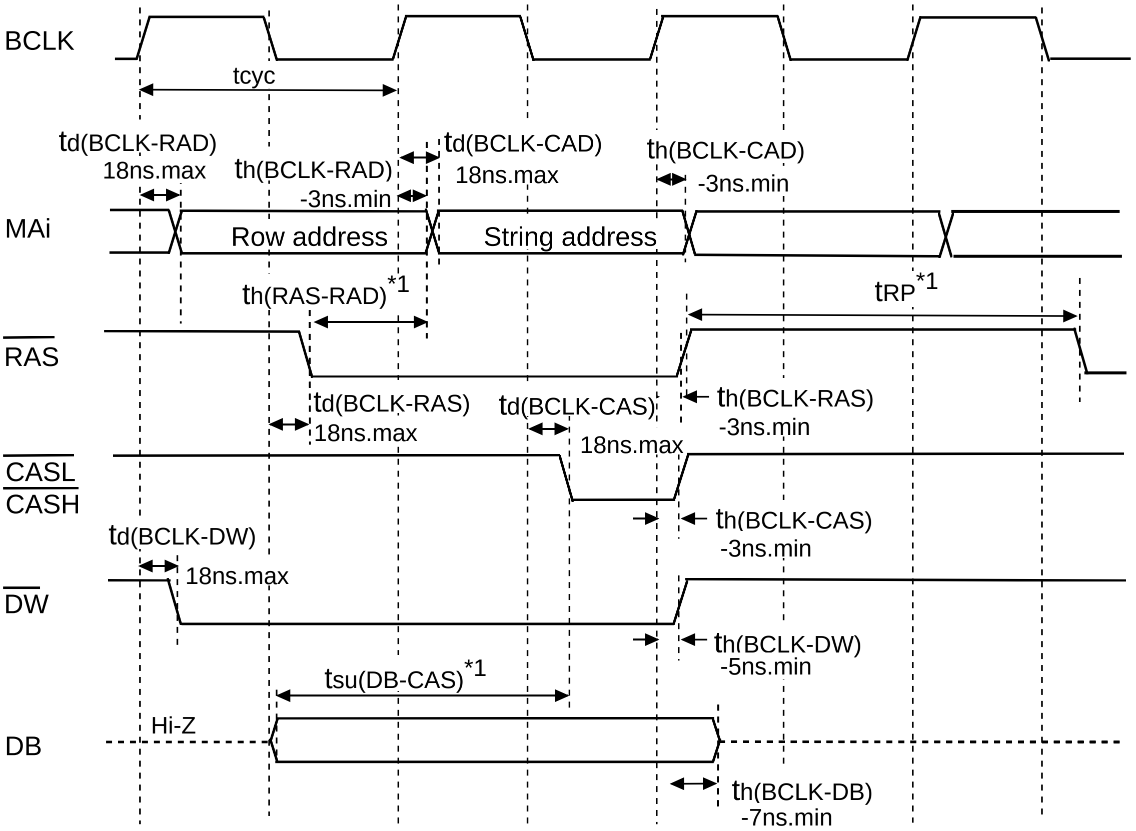
Figure 1.28.8. Vcc=5V timing diagram (7)

Vcc=5V

Memory expansion Mode and Microprocessor Mode

(When accessing DRAM area with 1 wait)

Write Timing



*1:It depends on operation frequency.

$t_{h(RAS-RAD)}$=(tcyc/2-13)ns.min

t_{RP}=(tcyc/2 x 3-20)ns.min

$t_{su(DB-CAS)}$=(tcyc-20)ns.min

Measuring conditions

- V_{CC}=5V±10%
- Input timing voltage
  :Determined with V_{IH}=2.5V, V_{IL}=0.8V
- Output timing voltage
  :Determined with V_{OH}=2.0V, V_{OL}=0.8V

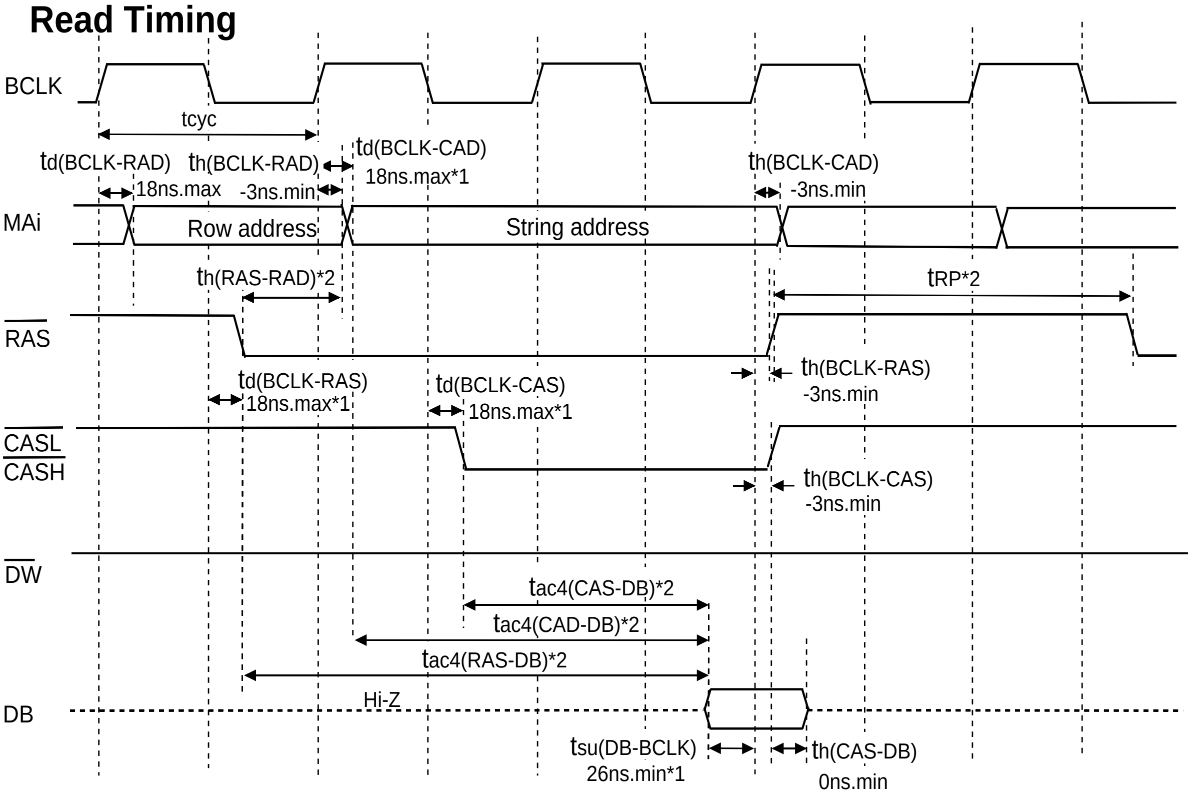
Figure 1.28.9. Vcc=5V timing diagram (8)

Memory expansion Mode and Microprocessor Mode

Vcc=5V

(When accessing DRAM area with 2 wait)

Read Timing



*1:It is a guarantee value with being alone. 35ns.max garantees as follows:
td(BCLK-RAS) + tsu(DB-BCLK)
td(BCLK-CAS) + tsu(DB-BCLK)
td(BCLK-CAD) + tsu(DB-BCLK)

*2:It depends on operation frequency.
tac4(RAS-DB)=(tcyc/2 x m-35)ns.max (m=3 and 5 when 1 wait and 2 wait, respectively.)
tac4(CAS-DB)=(tcyc/2 x n-35)ns.max (n=1 and 3 when 1 wait and 2 wait, respectively.)
tac4(CAD-DB)=(tcyc x l-35)ns.max (l=1 and 2 when 1 wait and 2 wait, respectively.)
th(RAS-RAD)=(tcyc/2-13)ns.min
tRP=(tcyc/2 x 3-20)ns.min

Measuring conditions
- Vcc=5V±10%
- Input timing voltage
  :Determined with VIH=2.5V, VIL=0.8V
- Output timing voltage
  :Determined with VOH=2.0V, VOL=0.8V

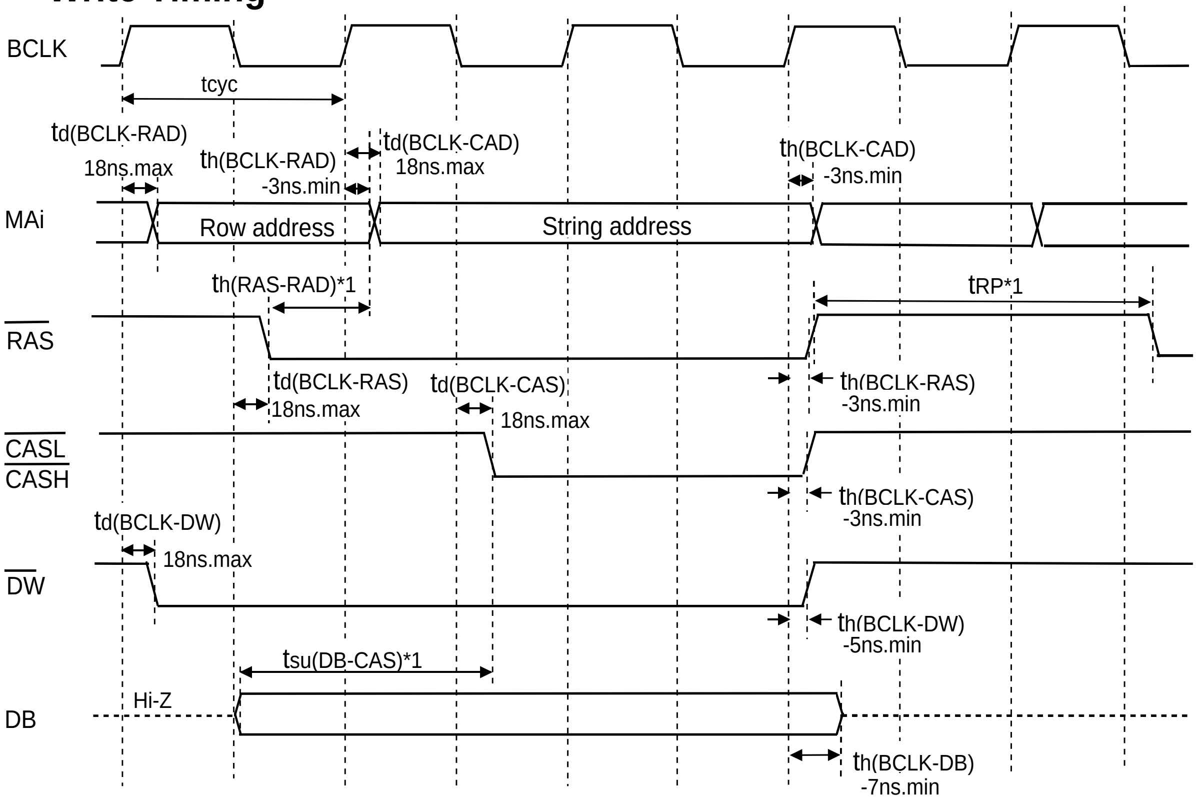
Figure 1.28.10. VCC=5V timing diagram (9)

Memory expansion Mode and Microprocessor Mode

Vcc=5V

(When accessing DRAM area with 2 wait)

Write Timing



*1:It depends on operation frequency.

$t_{h(RAS-RAD)}$=(tcyc/2-13)ns.min

t_{RP}=(tcyc/2 x 3-20)ns.min

$t_{su(DB-CAS)}$=(tcyc-20)ns.min

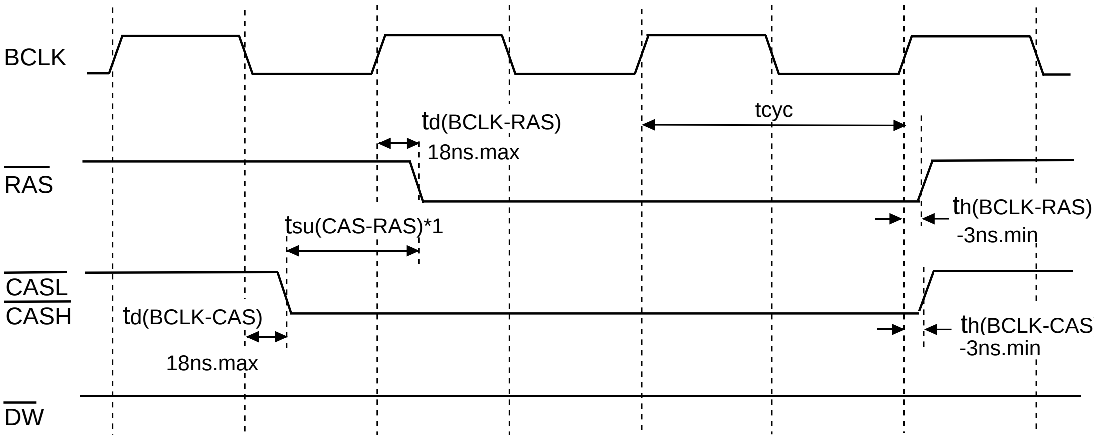
Measuring conditions

- V_{CC}=5V±10%
- Input timing voltage
  :Determined with V_{IH}=2.5V, V_{IL}=0.8V
- Output timing voltage
  :Determined with V_{OH}=2.0V, V_{OL}=0.8V

Figure 1.28.11. VCC=5V timing diagram (10)

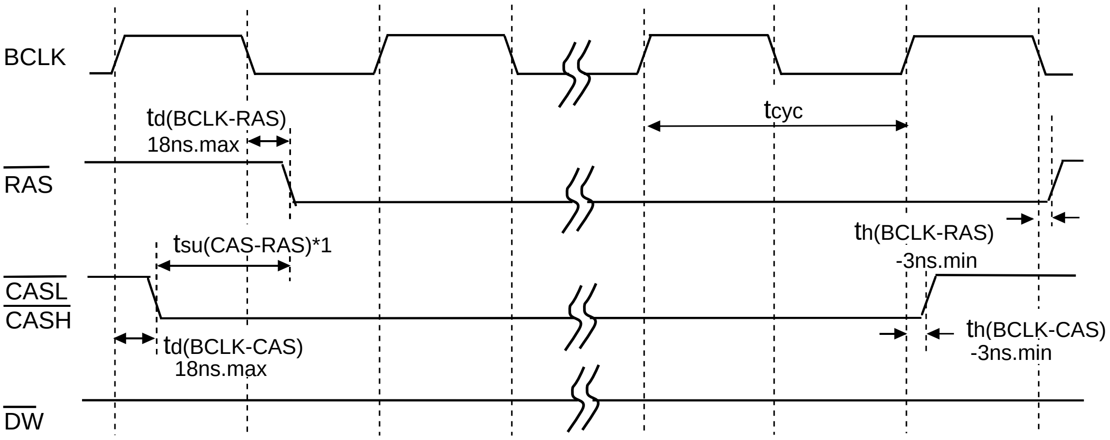
Memory expansion Mode and Microprocessor Mode Refresh Timing (CAS before RAS refresh)

Vcc=5V



*1:It depends on operation frequency.
tsu(CAS-RAS)=(tcyc/2-13)ns.min

Refresh Timing (Self-refresh)



*1:It depends on operation frequency.
tsu(CAS-RAS)=(tcyc/2-13)ns.min

Measuring conditions
- V_{CC}=5V±10%
- Input timing voltage
  :Determined with V_{IH}=2.5V, V_{IL}=0.8V
- Output timing voltage
  :Determined with V_{OH}=2.0V, V_{OL}=0.8V

Figure 1.28.12. Vcc=5V timing diagram (11)

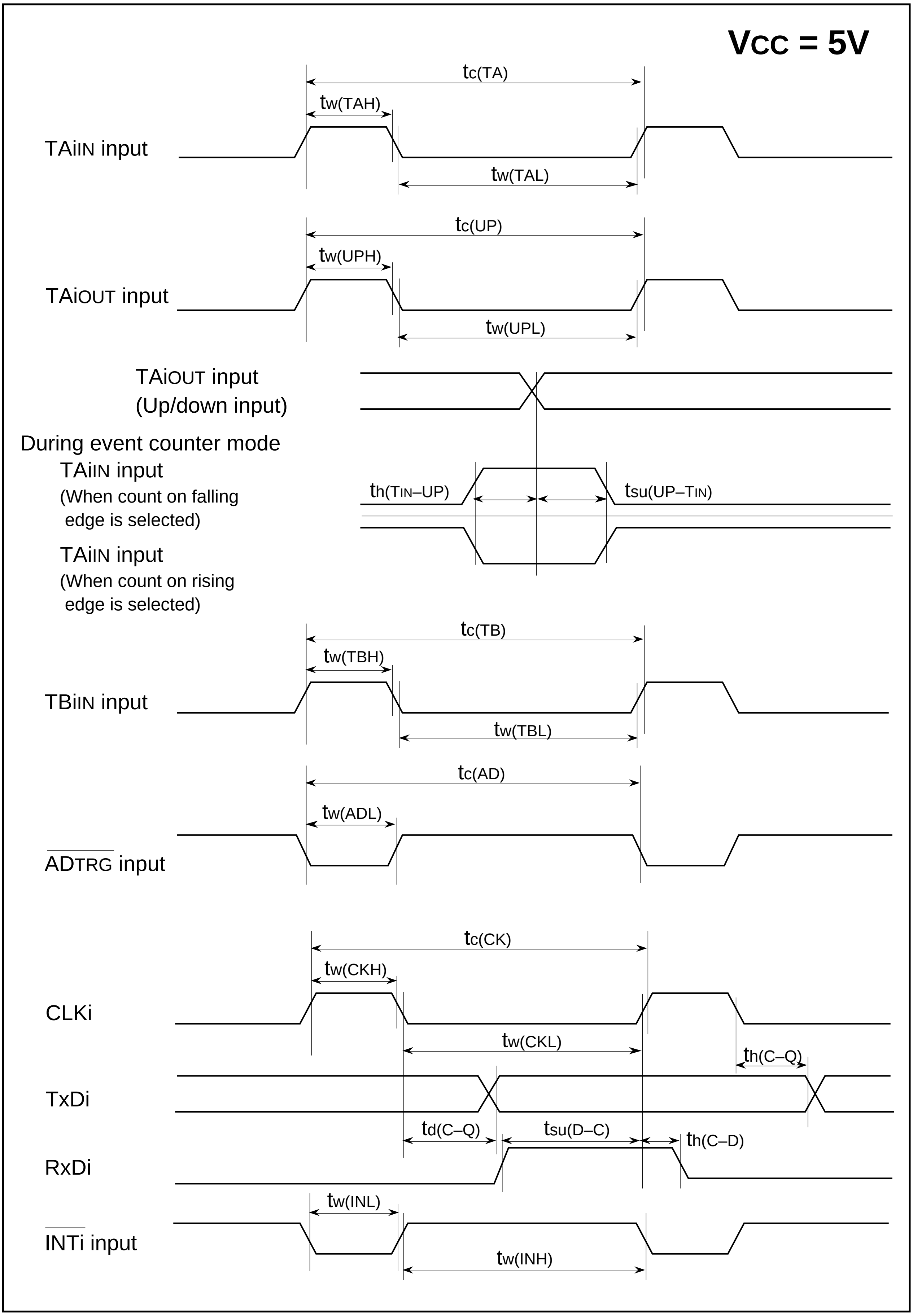
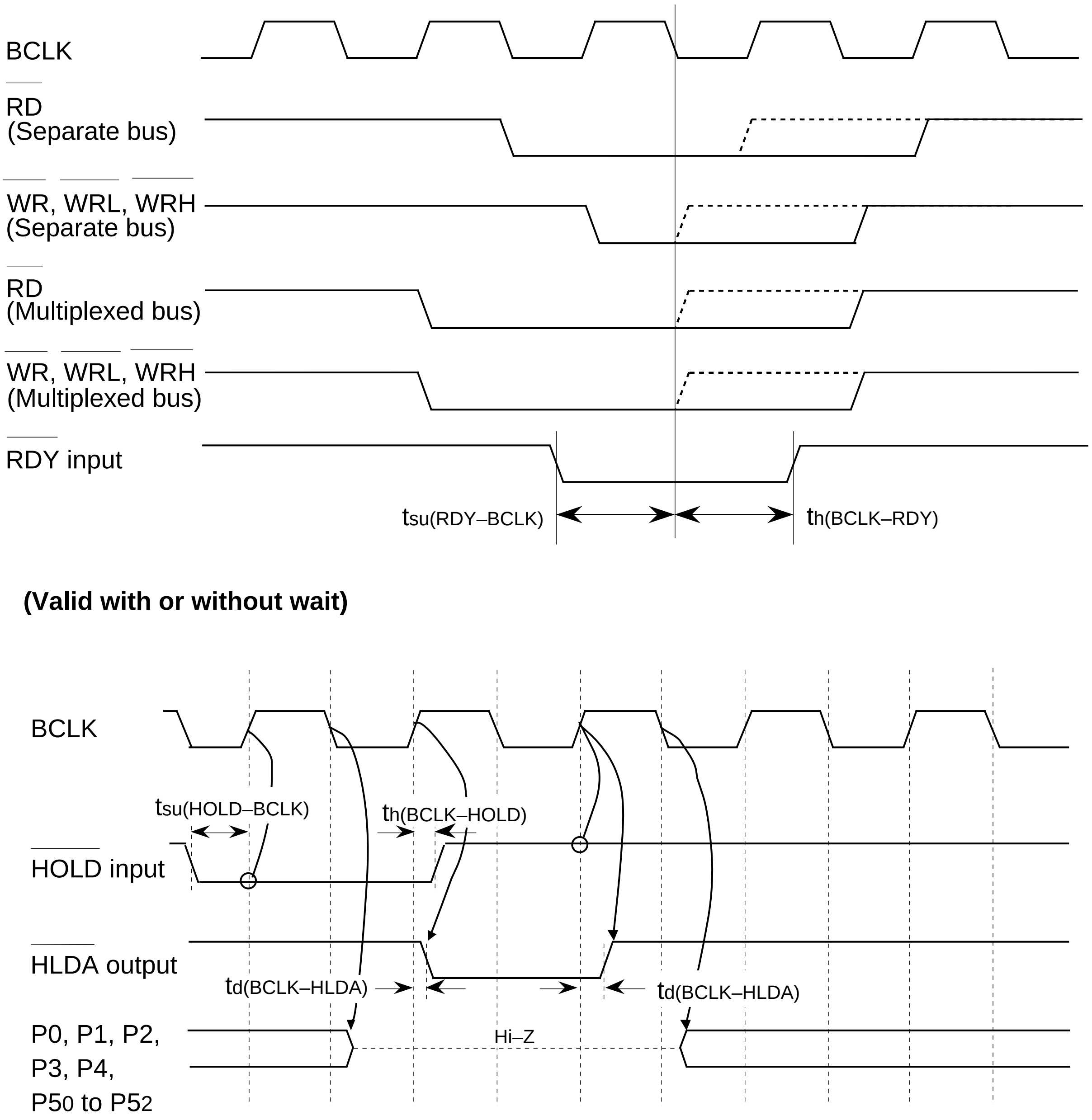


Figure 1.28.13. Vcc=5V timing diagram (12)

VCC = 5V

Memory Expansion Mode and Microprocessor Mode



(Valid only with wait)

(Valid with or without wait)

Measuring conditions :

- V_{CC}=5V±10%
- Input timing voltage : Determined with V_{IL}=1.0V, V_{IH}=4.0V
- Output timing voltage : Determined with V_{OL}=2.5V, V_{OH}=2.5V

Figure 1.28.14. VCC=5V timing diagram (13)

Electrical characteristics (Vcc = 3V)

VCC = 3V

Table 1.28.23. Electrical characteristics (referenced to V_{CC} = 3V, V_{SS} = 0V at Topr = 25^{o}C, f(X_{IN}) = 10MHz with wait)

Symbol	Parameter			Measuring condition	Standard			Unit
					Min	Typ.	Max.	
V_{OH}	HIGH output voltage	$P0_0$-$P0_7$,$P1_0$-$P1_7$,$P2_0$-$P2_7$, $P3_0$-$P3_7$,$P4_0$-$P4_7$,$P5_0$-$P5_7$, $P6_0$-$P6_7$,$P7_2$-$P7_7$,$P8_0$-$P8_4$, $P8_6$,$P8_7$,$P9_0$-$P9_7$,$P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$,$P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$ (Note 1)		I_{OH}= - 1mA	2.5			V
V_{OH}	HIGH output voltage	X_{OUT}	HIGHPOWER	I_{OH}= - 0.1 mA	2.5			V
			LOWPOWER	I_{OH}= - 50 µA	2.5			
	HIGH output voltage	X_{COUT}	HIGHPOWER	With no load applied		3.0		V
			LOWPOWER	With no load applied		1.6		
V_{OL}	LOW output voltage	$P0_0$-$P0_7$,$P1_0$-$P1_7$,$P2_0$-$P2_7$, $P3_0$-$P3_7$,$P4_0$-$P4_7$,$P5_0$-$P5_7$, $P6_0$-$P6_7$,$P7_0$-$P7_7$,$P8_0$-$P8_4$, $P8_6$,$P8_7$,$P9_0$-$P9_7$,$P10_0$-$P10_7$ $P11_0$-$P11_4$, $P12_0$-$P12_7$,$P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$ (Note 1)		I_{OL}=1mA			0.5	V
V_{OL}	LOW output voltage	X_{OUT}	HIGHPOWER	I_{OL}=0.1mA			0.5	V
			LOWPOWER	I_{OL}=50µA			0.5	
	LOW output voltage	X_{COUT}	HIGHPOWER	With no load applied		0		V
			LOWPOWER	With no load applied		0		
V_{T+}-V_{T-}	Hysteresis	$\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$-$TA4_{IN}$, $TB0_{IN}$-$TB2_{IN}$, $\overline{INT_0}$-$\overline{INT_5}$, $\overline{AD_{TRG}}$, $\overline{CTS_0}$-$\overline{CTS_4}$,CLK_0-CLK_4,$TA2_{OUT}$-$TA4_{OUT}$, $\overline{NMI}$, $\overline{KI_0}$-$\overline{KI_3}$, RxD0-RxD4, SCL_2-SCL_4, SDA_2-SDA_4			0.2		1.0	V
V_{T+}-V_{T-}	Hysteresis	$\overline{RESET}$			0.2		1.8	V
I_{IH}	HIGH input current	$P0_0$-$P0_7$,$P1_0$-$P1_7$,$P2_0$-$P2_7$, $P3_0$-$P3_7$,$P4_0$-$P4_7$,$P5_0$-$P5_7$, $P6_0$-$P6_7$,$P7_0$-$P7_7$,$P8_0$-$P8_7$, $P9_0$-$P9_7$,$P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$,$P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$ (Note 1) X_{IN}, RESET, CNVss, BYTE		V_I=3V			4.0	µA
I_{IL}	LOW input current	$P0_0$-$P0_7$,$P1_0$-$P1_7$,$P2_0$-$P2_7$, $P3_0$-$P3_7$,$P4_0$-$P4_7$,$P5_0$-$P5_7$, $P6_0$-$P6_7$,$P7_0$-$P7_7$,$P8_0$-$P8_7$, $P9_0$-$P9_7$,$P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$,$P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$ (Note 1) X_{IN}, RESET, CNVss, BYTE		V_I=0V			- 4.0	µA
R_{PULLUP}	Pull-up resistance	$P0_0$-$P0_7$,$P1_0$-$P1_7$,$P2_0$-$P2_7$, $P3_0$-$P3_7$,$P4_0$-$P4_7$,$P5_0$-$P5_7$, $P6_0$-$P6_7$,$P7_2$-$P7_7$,$P8_0$-$P8_4$, $P8_6$,$P8_7$,$P9_0$-$P9_7$,$P10_0$-$P10_7$ $P11_0$-$P11_4$, $P12_0$-$P12_7$,$P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$ (Note 1)		V_I=0V	66.0	120.0	500.0	kΩ
R_{fXIN}	Feedback resistance	X_{IN}				3.0		MΩ
R_{fXCIN}	Feedback resistance	X_{CIN}				10.0		MΩ
V_{RAM}	RAM retention voltage			When clock is stopped	2.0			V
I_{CC}	Power supply current	In single-chip mode, the output pins are open and other pins are V_{SS}	f(X_{IN})=10MHz Square wave, no division	Mask ROM 128 KB version ROMless RAM 10 KB version (Note 2)		12.0	20.0	mA
				Mask ROM 256 KB version ROMless RAM 24 KB version (Note 2)		14.0	23.0	
				Flash memory version		14.0	23.0	
			f(X_{CIN})=32kHz Square wave	Mask ROM 128 KB version ROMless RAM 10 KB version (Note 2)		45.0		µA
				Mask ROM 256 KB version ROMless RAM 24 KB version (Note 2)		60.0		
				Flash memory version		3.5		mA
			f(XCIN)=32kHz When a WAIT instruction is executed. Oscillation drive capacity is High.			3.0		µA
			f(XCIN)=32kHz When a WAIT instruction is executed. Oscillation drive capacity is Low.			1.5		µA
			Topr=25°C, when clock is stopped	Mask ROM 128 KB version ROMless RAM 10 KB version (Note 2)			1.0	µA
				Mask ROM 256 KB version ROMless RAM 24 KB version (Note 2)			1.0	
				Flash memory version			1.0	
			Topr=85°C, when clock is stopped				20.0	

Note 1: Ports P11 to P15 exist in 144-pin version.

Note 2: ROMless version exists in 144-pin version.

V_{CC} = 3V

Table 1.28.24. A-D conversion characteristics (referenced to V_{CC} = AV_{CC} = V_{REF} = 3V, V_{SS} = AV_{SS} = 0V at Topr = 25^{o}C, f(X_{IN}) = 10MHz unless otherwise specified)

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max	
-	Resolution		V_{REF} = V_{CC}			10	Bits
-	Absolute accuracy	Sample & hold function not available (8 bit)	V_{REF} = V_{CC} = 3V, ϕ_{AD} = f_{AD}/2			±2	LSB
R_{LADDER}	Ladder resistance		V_{REF} = V_{CC}	10		40	kΩ
t_{CONV}	Conversion time(8bit)			9.8			µs
V_{REF}	Reference voltage			2.7		V_{CC}	V
V_{IA}	Analog input voltage			0		V_{REF}	V

Table 1.28.25. D-A conversion characteristics (referenced to V_{CC} = 3V, V_{SS} = AV_{SS} = 0V, V_{REF} = 3V at Topr = 25^{o}C, f(X_{IN}) = 10MHz unless otherwise specified)

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max	
-	Resolution				8	Bits
-	Absolute accuracy				1.0	%
t_{su}	Setup time				3	µs
R_{O}	Output resistance		4	10	20	kΩ
I_{VREF}	Reference power supply input current	(Note)			1.0	mA

Note :This applies when using one D-A converter, with the D-A register for the unused D-A converter set to “00_{16}”. The A-D converter's ladder resistance is not included.
Also, when the contents of D-A register is except "00_{16}" and the Vref is unconnected at the A-D control register 1, I_{VREF} is sent.

VCC = 3V

Timing requirements (referenced to VCC = 3V, VSS = 0V at Topr = 25°C unless otherwise specified)

Table 1.28.26. External clock input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t_c	External clock input cycle time	100		ns
$t_{w(H)}$	External clock input HIGH pulse width	40		ns
$t_{w(L)}$	External clock input LOW pulse width	40		ns
t_r	External clock rise time		18	ns
t_f	External clock fall time		18	ns

Table 1.28.27. Memory expansion and microprocessor modes

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{ac1(RD-DB)}$	Data input access time (RD standard, no wait)		(Note)	ns
$t_{ac1(AD-DB)}$	Data input access time (AD standard, CS standard, no wait)		(Note)	ns
$t_{ac2(RD-DB)}$	Data input access time (RD standard, with wait)		(Note)	ns
$t_{ac2(AD-DB)}$	Data input access time (AD standard, CS standard, with wait)		(Note)	ns
$t_{ac3(RD-DB)}$	Data input access time (RD standard, when accessing multiplex bus area)		(Note)	ns
$t_{ac3(AD-DB)}$	Data input access time (AD standard, CS standard, when accessing multiplex bus area)		(Note)	ns
$t_{ac4(RAS-DB)}$	Data input access time (RAS standard, DRAM access)		(Note)	ns
$t_{ac4(CAS-DB)}$	Data input access time (CAS standard, DRAM access)		(Note)	ns
$t_{ac4(CAD-DB)}$	Data input access time (CAD standard, DRAM access)		(Note)	ns
$t_{su(DB-BCLK)}$	Data input setup time	40		ns
$t_{su(RDY-BCLK)}$	$\overline{\text{RDY}}$ input setup time	60		ns
$t_{su(HOLD-BCLK)}$	$\overline{\text{HOLD}}$ input setup time	80		ns
$t_{h(RD-DB)}$	Data input hold time	0		ns
$t_{h(CAS-DB)}$	Data input hold time	0		ns
$t_{h(BCLK-RDY)}$	$\overline{\text{RDY}}$ input hold time	0		ns
$t_{h(BCLK-HOLD)}$	$\overline{\text{HOLD}}$ input hold time	0		ns
$t_{d(BCLK-HLDA)}$	$\overline{\text{HLDA}}$ output delay time		100	ns

Note: Calculated according to the BCLK frequency as follows:
Note that inserting wait or using lower operation frequency f(BCLK) is needed when calculated value is negative.

$$t_{ac1(RD-DB)} = \frac{10^9}{f_{(BCLK)} \times 2} - 42 \quad \text{[ns]}$$

$$t_{ac1(AD-DB)} = \frac{10^9}{f_{(BCLK)}} - 55 \quad \text{[ns]}$$

$$t_{ac2(RD-DB)} = \frac{10^9 \times m}{f_{(BCLK)} \times 2} - 42 \quad \text{[ns] (m=3, 5 and 7 when 1 wait, 2 wait and 3 wait, respectively)}$$

$$t_{ac2(AD-DB)} = \frac{10^9 \times n}{f_{(BCLK)}} - 55 \quad \text{[ns] (n=2, 3 and 4 when 1 wait, 2 wait and 3 wait, respectively)}$$

$$t_{ac3(RD-DB)} = \frac{10^9 \times m}{f_{(BCLK)} \times 2} - 55 \quad \text{[ns] (m=3 and 5 when 2 wait and 3 wait, respectively)}$$

$$t_{ac3(AD-DB)} = \frac{10^9 \times n}{f_{(BCLK)} \times 2} - 55 \quad \text{[ns] (n=5 and 7 when 2 wait and 3 wait, respectively)}$$

$$t_{ac4(RAS-DB)} = \frac{10^9 \times m}{f_{(BCLK)} \times 2} - 55 \quad \text{[ns] (m=3 and 5 when 1 wait and 2 wait, respectively)}$$

$$t_{ac4(CAS-DB)} = \frac{10^9 \times n}{f_{(BCLK)} \times 2} - 55 \quad \text{[ns] (n=1 and 3 when 1 wait and 2 wait, respectively)}$$

$$t_{ac4(CAD-DB)} = \frac{10^9 \times l}{f_{(BCLK)}} - 55 \quad \text{[ns] (l=1 and 2 when 1 wait and 2 wait, respectively)}$$

V_{CC} = 3V

Timing requirements (referenced to V_{CC} = 3V, V_{SS} = 0V at Topr = 25°C unless otherwise specified)

Table 1.28.28. Timer A input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	150		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	60		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	60		ns

Table 1.28.29. Timer A input (gating input in timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	600		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	300		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	300		ns

Table 1.28.30. Timer A input (external trigger input in one-shot timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TA)}$	TAiIN input cycle time	300		ns
$t_{w(TAH)}$	TAiIN input HIGH pulse width	150		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	150		ns

Table 1.28.31. Timer A input (external trigger input in pulse width modulation mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(TAH)}$	TAiIN input HIGH pulse width	150		ns
$t_{w(TAL)}$	TAiIN input LOW pulse width	150		ns

Table 1.28.32. Timer A input (up/down input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(UP)}$	TAiOUT input cycle time	3000		ns
$t_{w(UPH)}$	TAiOUT input HIGH pulse width	1500		ns
$t_{w(UPL)}$	TAiOUT input LOW pulse width	1500		ns
$t_{su(UP-TIN)}$	TAiOUT input setup time	600		ns
$t_{h(TIN-UP)}$	TAiOUT input hold time	600		ns

V$_{CC}$ = 3V

Timing requirements (referenced to V$_{CC}$ = 3V, V$_{SS}$ = 0V at Topr = 25°C unless otherwise specified)

Table 1.28.33. Timer B input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBi$_{IN}$ input cycle time (counted on one edge)	150		ns
$t_{w(TBH)}$	TBi$_{IN}$ input HIGH pulse width (counted on one edge)	60		ns
$t_{w(TBL)}$	TBi$_{IN}$ input LOW pulse width (counted on one edge)	60		ns
$t_{c(TB)}$	TBi$_{IN}$ input cycle time (counted on both edges)	300		ns
$t_{w(TBH)}$	TBi$_{IN}$ input HIGH pulse width (counted on both edges)	160		ns
$t_{w(TBL)}$	TBi$_{IN}$ input LOW pulse width (counted on both edges)	160		ns

Table 1.28.34. Timer B input (pulse period measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBi$_{IN}$ input cycle time	600		ns
$t_{w(TBH)}$	TBi$_{IN}$ input HIGH pulse width	300		ns
$t_{w(TBL)}$	TBi$_{IN}$ input LOW pulse width	300		ns

Table 1.28.35. Timer B input (pulse width measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(TB)}$	TBi$_{IN}$ input cycle time	600		ns
$t_{w(TBH)}$	TBi$_{IN}$ input HIGH pulse width	300		ns
$t_{w(TBL)}$	TBi$_{IN}$ input LOW pulse width	300		ns

Table 1.28.36. A-D trigger input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(AD)}$	$\overline{AD_{TRG}}$ input cycle time (trigger able minimum)	1500		ns
$t_{w(ADL)}$	$\overline{AD_{TRG}}$ input LOW pulse width	200		ns

Table 1.28.37. Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{c(CK)}$	CLKi input cycle time	300		ns
$t_{w(CKH)}$	CLKi input HIGH pulse width	150		ns
$t_{w(CKL)}$	CLKi input LOW pulse width	150		ns
$t_{d(C-Q)}$	TxDi output delay time		160	ns
$t_{h(C-Q)}$	TxDi hold time	0		ns
$t_{su(D-C)}$	RxDi input setup time	50		ns
$t_{h(C-D)}$	RxDi input hold time	90		ns

Table 1.28.38. External interrupt $\overline{INTi}$ inputs

Symbol	Parameter	Standard		Unit
		Min.	Max.	
$t_{w(INH)}$	$\overline{INTi}$ input HIGH pulse width	380		ns
$t_{w(INL)}$	$\overline{INTi}$ input LOW pulse width	380		ns

VCC = 3V

Switching characteristics (referenced to VCC = 3V, VSS = 0V at Topr = 25°C unless otherwise specified)

Table 1.28.39. Memory expansion and microprocessor modes (with no wait)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.28.1		25	ns
$t_{h(BCLK-AD)}$	Address output hold time (BCLK standard)		0		ns
$t_{h(RD-AD)}$	Address output hold time (RD standard)		0		ns
$t_{h(WR-AD)}$	Address output hold time (WR standard)		(Note)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			25	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (BCLK standard)		0		ns
$t_{h(RD-CS)}$	Chip select output hold time (RD standard)		0		ns
$t_{h(WR-CS)}$	Chip select output hold time (WR standard)		(Note)		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time			25	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time		– 2		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			10	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		– 3		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			25	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(DB-WR)}$	Data output delay time (WR standard)		(Note)		ns
$t_{h(WR-DB)}$	Data output hold time (WR standard)		(Note)		ns
$t_{w(WR)}$	WR signal width		(Note)		ns

Note: Calculated according to the BCLK frequency as follows:

$$t_{d(DB-WR)} = \frac{10^9}{f_{(BCLK)}} - 40 \quad [ns]$$

$$t_{h(WR-DB)} = \frac{10^9}{f_{(BCLK)} \times 2} - 20 \quad [ns]$$

$$t_{h(WR-AD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 20 \quad [ns]$$

$$t_{h(WR-CS)} = \frac{10^9}{f_{(BCLK)} \times 2} - 20 \quad [ns]$$

$$t_{w(WR)} = \frac{10^9}{f_{(BCLK)} \times 2} - 20 \quad [ns]$$

VCC = 3V

Switching characteristics (referenced to VCC = 3V, VSS = 0V at Topr = 25°C unless otherwise specified)

Table 1.28.40. Memory expansion and microprocessor modes (with wait, accessing external memory)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.28.1		25	ns
$t_{h(BCLK-AD)}$	Address output hold time (BCLK standard)		0		ns
$t_{h(RD-AD)}$	Address output hold time (RD standard)		0		ns
$t_{h(WR-AD)}$	Address output hold time (WR standard)		(Note)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			25	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (BCLK standard)		0		ns
$t_{h(RD-CS)}$	Chip select output hold time (RD standard)		0		ns
$t_{h(WR-CS)}$	Chip select output hold time (WR standard)		(Note)		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time			25	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time		– 2		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			10	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		– 3		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			25	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(DB-WR)}$	Data output delay time (WR standard)		(Note)		ns
$t_{h(WR-DB)}$	Data output hold time (WR standard)		(Note)		ns
$t_{w(WR)}$	WR signal width		(Note)		ns

Note: Calculated according to the BCLK frequency as follows:

$$t_{d(DB-WR)} = \frac{10^{9} \times n}{f_{(BCLK)}} - 40$$ [ns] (n=1, 2 and 3 when 1 wait, 2 wait and 3 wait, respectively)

$$t_{h(WR-DB)} = \frac{10^{9}}{f_{(BCLK)} \times 2} - 20$$ [ns]

$$t_{h(WR-AD)} = \frac{10^{9}}{f_{(BCLK)} \times 2} - 20$$ [ns]

$$t_{h(WR-CS)} = \frac{10^{9}}{f_{(BCLK)} \times 2} - 20$$ [ns]

$$t_{w(WR)} = \frac{10^{9} \times n}{f_{(BCLK)} \times 2} - 20$$ [ns] (n=1, 3 and 5 when 1 wait, 2 wait and 3 wait, respectively)

VCC = 3V

Switching characteristics (referenced to VCC = 3V, VSS = 0V at Topr = 25°C, CM15 = “1” unless otherwise specified)

Table 1.28.41. Memory expansion and microprocessor modes (with wait, accessing external memory, multiplex bus area selected)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.28.1		25	ns
$t_{h(BCLK-AD)}$	Address output hold time (BCLK standard)		0		ns
$t_{h(RD-AD)}$	Address output hold time (RD standard)		(Note)		ns
$t_{h(WR-AD)}$	Address output hold time (WR standard)		(Note)		ns
$t_{d(BCLK-CS)}$	Chip select output delay time			25	ns
$t_{h(BCLK-CS)}$	Chip select output hold time (BCLK standard)		0		ns
$t_{h(RD-CS)}$	Chip select output hold time (RD standard)		(Note)		ns
$t_{h(WR-CS)}$	Chip select output hold time (WR standard)		(Note)		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			25	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		– 3		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			25	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(DB-WR)}$	Data output delay time (WR standard)		(Note)		ns
$t_{h(WR-DB)}$	Data output hold time (WR standard)		(Note)		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time (BCLK standard)			25	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time (BCLK standard)		– 2		ns
$t_{d(AD-ALE)}$	ALE signal output delay time (address standard)		(Note)		ns
$t_{h(ALE-AD)}$	ALE signal output hold time (address standard)		(Note)		ns
$t_{dz(RD-AD)}$	Address output flowting start time			8	ns
$t_{h(BCLK-DB)}$	DB signal output hold time (BCLK standard)		0		ns

Note: Calculated according to the BCLK frequency as follows:

$$t_{h(RD-AD)} = \frac{10^9}{f_{(BCLK)} \text{ X } 2} - 20 \quad \text{[ns]}$$

$$t_{h(WR-AD)} = \frac{10^9}{f_{(BCLK)} \text{ X } 2} - 20 \quad \text{[ns]}$$

$$t_{h(RD-CS)} = \frac{10^9}{f_{(BCLK)} \text{ X } 2} - 20 \quad \text{[ns]}$$

$$t_{h(WR-CS)} = \frac{10^9}{f_{(BCLK)} \text{ X } 2} - 20 \quad \text{[ns]}$$

$$t_{d(DB-WR)} = \frac{10^9 \text{ X } m}{f_{(BCLK)} \text{ X } 2} - 40 \quad \text{[ns]}$$ (m=3 and 5 when 2 wait and 3 wait, respectively)

$$t_{h(WR-DB)} = \frac{10^9}{f_{(BCLK)} \text{ X } 2} - 20 \quad \text{[ns]}$$

$$t_{d(AD-ALE)} = \frac{10^9}{f_{(BCLK)} \text{ X } 2} - 27 \quad \text{[ns]}$$

$$t_{h(ALE-AD)} = \frac{10^9}{f_{(BCLK)} \text{ X } 2} - 20 \quad \text{[ns]}$$

VCC = 3V

Switching characteristics (referenced to VCC = 3V, VSS = 0V at Topr = 25°C unless otherwise specified)

Table 1.28.42. Memory expansion and microprocessor modes (with wait, accessing external memory, DRAM area selected)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
$t_{d(BCLK-RAD)}$	Row address output delay time	Figure 1.28.1		25	ns
$t_{h(BCLK-RAD)}$	Row address output hold time (BCLK standard)		0		ns
$t_{d(BCLK-CAD)}$	String address output delay time			25	ns
$t_{h(BCLK-CAD)}$	String address output hold time (BCLK standard)		0		ns
$t_{h(RAS-RAD)}$	Row address output hold time after RAS output		(Note)		ns
$t_{d(BCLK-RAS)}$	RAS output delay time (BCLK standard)			25	ns
$t_{h(BCLK-RAS)}$	RAS output hold time (BCLK standard)		0		ns
t_{RP}	RAS "H" hold time		(Note)		ns
$t_{d(BCLK-CAS)}$	CAS output delay time (BCLK standard)			25	ns
$t_{h(BCLK-CAS)}$	CAS output hold time (BCLK standard)		0		ns
$t_{d(BCLK-DW)}$	Data output delay time (BCLK standard)			25	ns
$t_{h(BCLK-DW)}$	Data output hold time (BCLK standard)		– 3		ns
$t_{su(DB-CAS)}$	CAS after DB output setup time		(Note)		ns
$t_{h(BCLK-DB)}$	DB signal output hold time (BCLK standard)		– 7		ns
$t_{su(CAS-RAS)}$	CAS before RAS setup time (refresh)		(Note)		ns

Note: Calculated according to the BCLK frequency as follows:

$$t_{h(RAS-RAD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 25 \quad [ns]$$

$$t_{RP} = \frac{10^9 \times 3}{f_{(BCLK)} \times 2} - 40 \quad [ns]$$

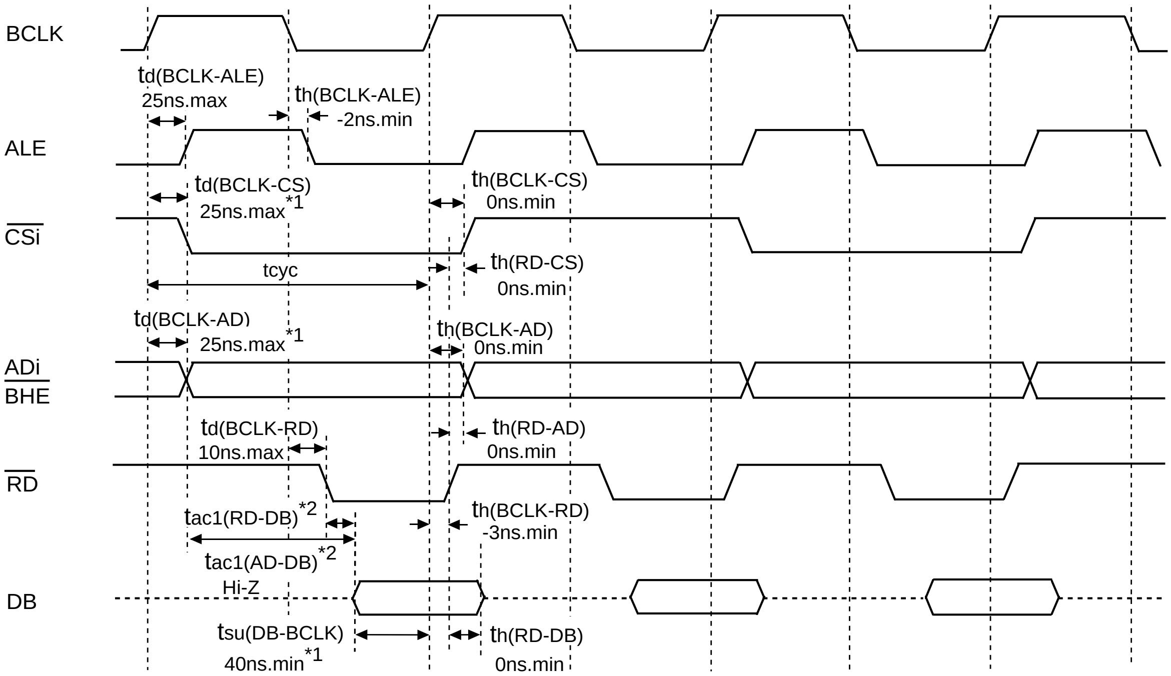
$$t_{su(DB-CAS)} = \frac{10^9}{f_{(BCLK)}} - 40 \quad [ns]$$

$$t_{su(CAS-RAS)} = \frac{10^9}{f_{(BCLK)} \times 2} - 25 \quad [ns]$$

Vcc=3V

Memory expansion Mode and Microprocessor Mode (without wait)

Read Timing



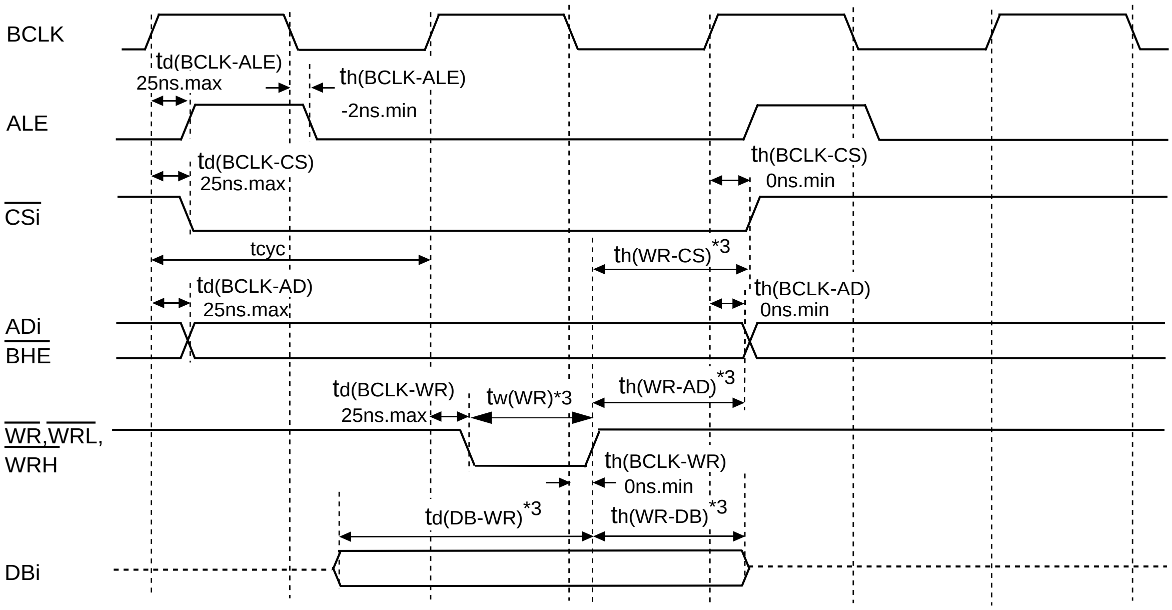
*1:It is a guarantee value with being alone. 55ns.max garantees as $t_{d(BCLK-AD)}+t_{su(DB-BCLK)}$.

*2:It depends on operation frequency.

$t_{ac1(RD-DB)}$=(tcyc/2-42)ns.max

$t_{ac1(AD-DB)}$=(tcyc-55)ns.max

Write Timing (Written by 2 cycles in selecting no wait)



*3:It depends on operation frequency.

$t_{d(DB-WR)}$=(tcyc-40)ns.min

$t_{h(WR-DB)}$=(tcyc/2-20)ns.min

$t_{h(WR-AD)}$=(tcyc/2-20)ns.min

$t_{h(WR-CS)}$=(tcyc/2-20)ns.min

$t_{w(WR)}$=(tcyc/2-20)ns.min

Measuring conditions

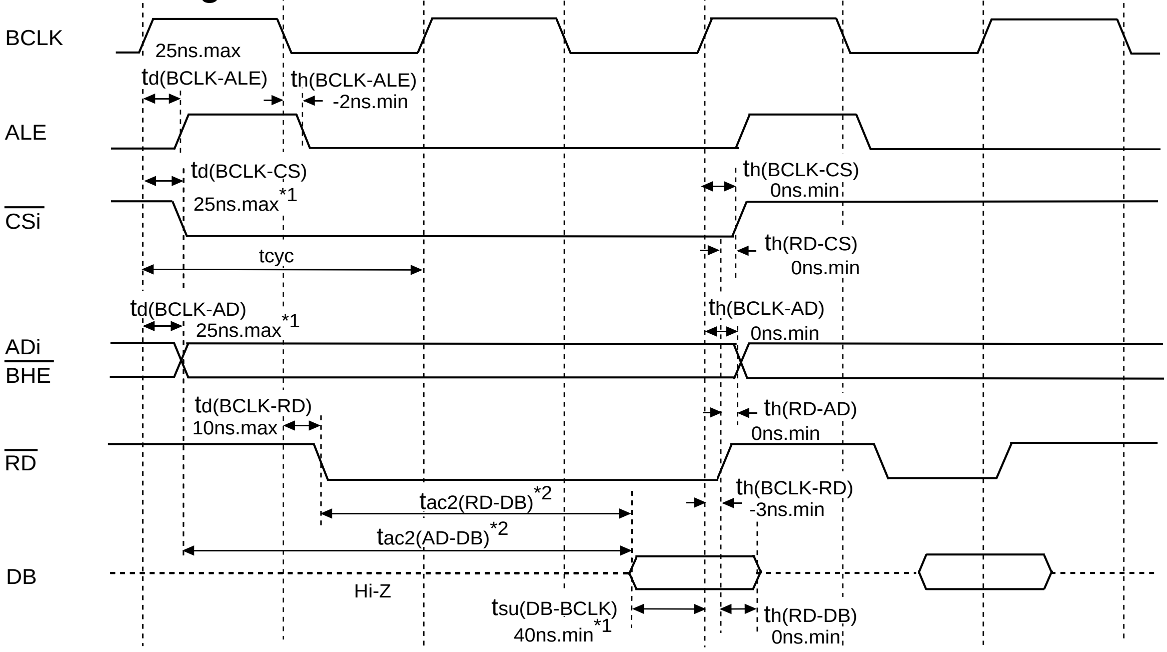
- V_{CC}=3V±10%
- Input timing voltage :Determined with V_{IH}=1.5V, V_{IL}=0.5V
- Output timing voltage :Determined with V_{OH}=1.5V, V_{OL}=1.5V

Figure 1.28.15. V_{CC}=3V timing diagram (1)

Vcc=3V

Memory expansion Mode and Microprocessor Mode (with 1 wait)

Read Timing



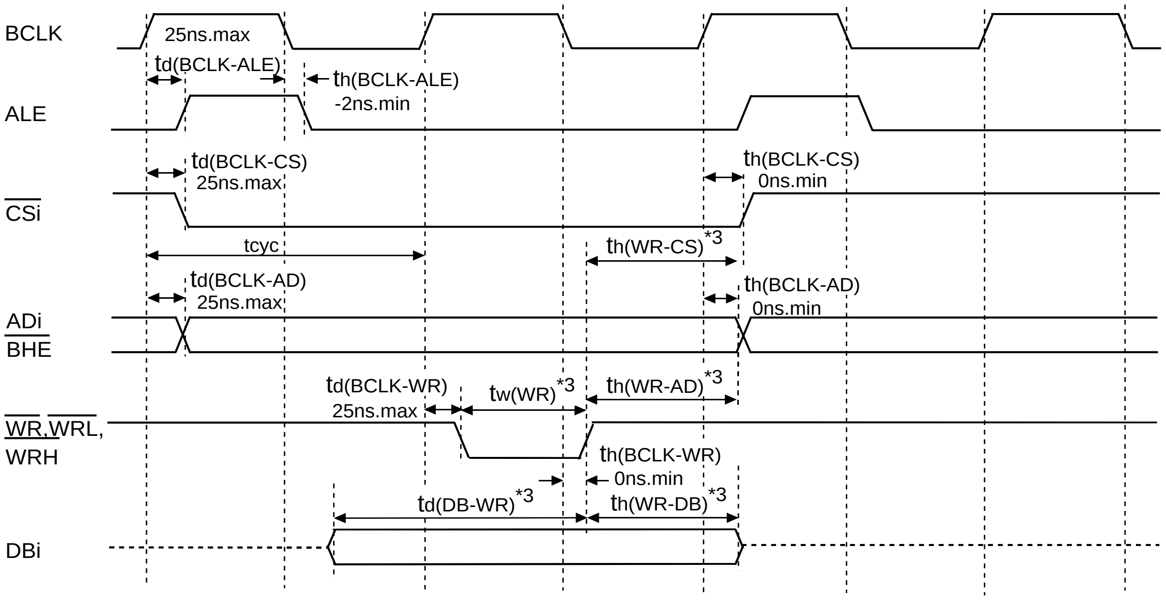
*1:It is a guarantee value with being alone. 55ns.max garantees as $t_{d(BCLK-AD)}$+$t_{su(DB-BCLK)}$.

*2:It depends on operation frequency.

$t_{ac2(RD-DB)}$=(tcyc/2 x m-42)ns.max (m=3, 5 and 7 when 1 wait, 2 wait and 3 wait, respectively.)

$t_{ac2(AD-DB)}$=(tcyc x n-55)ns.max (n=2, 3 and 4 when 1 wait, 2 wait and 3 wait, respectively.)

Write Timing



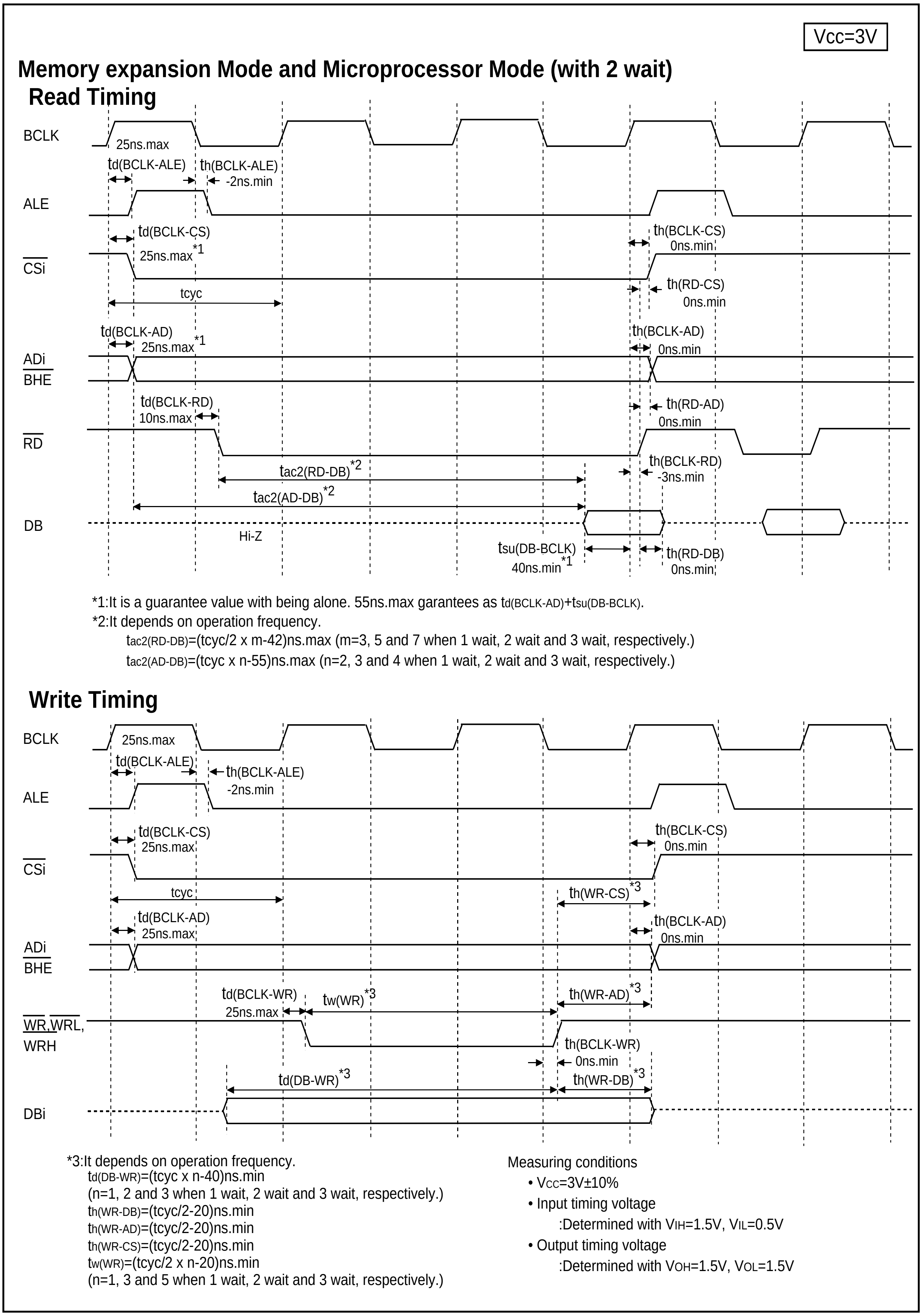
*3:It depends on operation frequency.

$t_{d(DB-WR)}$=(tcyc x n-40)ns.min
(n=1, 2 and 3 when 1 wait, 2 wait and 3 wait, respectively.)
$t_{h(WR-DB)}$=(tcyc/2-20)ns.min
$t_{h(WR-AD)}$=(tcyc/2-20)ns.min
$t_{h(WR-CS)}$=(tcyc/2-20)ns.min
$t_{w(WR)}$=(tcyc/2 x n-20)ns.min
(n=1, 3 and 5 when 1 wait, 2 wait and 3 wait, respectively.)

Measuring conditions

- V_{CC}=3V±10%
- Input timing voltage
  :Determined with V_{IH}=1.5V, V_{IL}=0.5V
- Output timing voltage
  :Determined with V_{OH}=1.5V, V_{OL}=1.5V

Figure 1.28.16. V_{CC}=3V timing diagram (2)



*1:It is a guarantee value with being alone. 55ns.max garantees as $t_{d(BCLK-AD)}$+$t_{su(DB-BCLK)}$.
*2:It depends on operation frequency.
$t_{ac2(RD-DB)}$=(tcyc/2 x m-42)ns.max (m=3, 5 and 7 when 1 wait, 2 wait and 3 wait, respectively.)
$t_{ac2(AD-DB)}$=(tcyc x n-55)ns.max (n=2, 3 and 4 when 1 wait, 2 wait and 3 wait, respectively.)

*3:It depends on operation frequency.
$t_{d(DB-WR)}$=(tcyc x n-40)ns.min
(n=1, 2 and 3 when 1 wait, 2 wait and 3 wait, respectively.)
$t_{h(WR-DB)}$=(tcyc/2-20)ns.min
$t_{h(WR-AD)}$=(tcyc/2-20)ns.min
$t_{h(WR-CS)}$=(tcyc/2-20)ns.min
$t_{w(WR)}$=(tcyc/2 x n-20)ns.min
(n=1, 3 and 5 when 1 wait, 2 wait and 3 wait, respectively.)

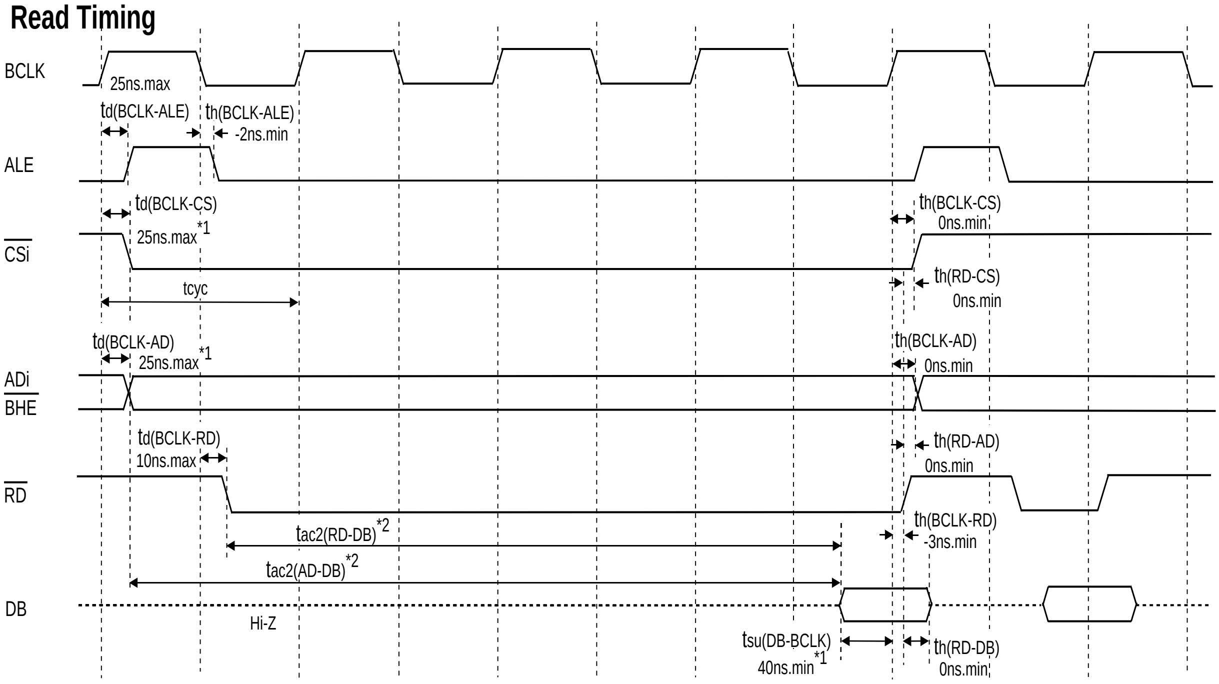
Measuring conditions
- V_{CC}=3V±10%
- Input timing voltage
  :Determined with V_{IH}=1.5V, V_{IL}=0.5V
- Output timing voltage
  :Determined with V_{OH}=1.5V, V_{OL}=1.5V

Figure 1.28.17. V_{CC}=3V timing diagram (3)

Memory expansion Mode and Microprocessor Mode (with 3 wait)

Vcc=3V

Read Timing



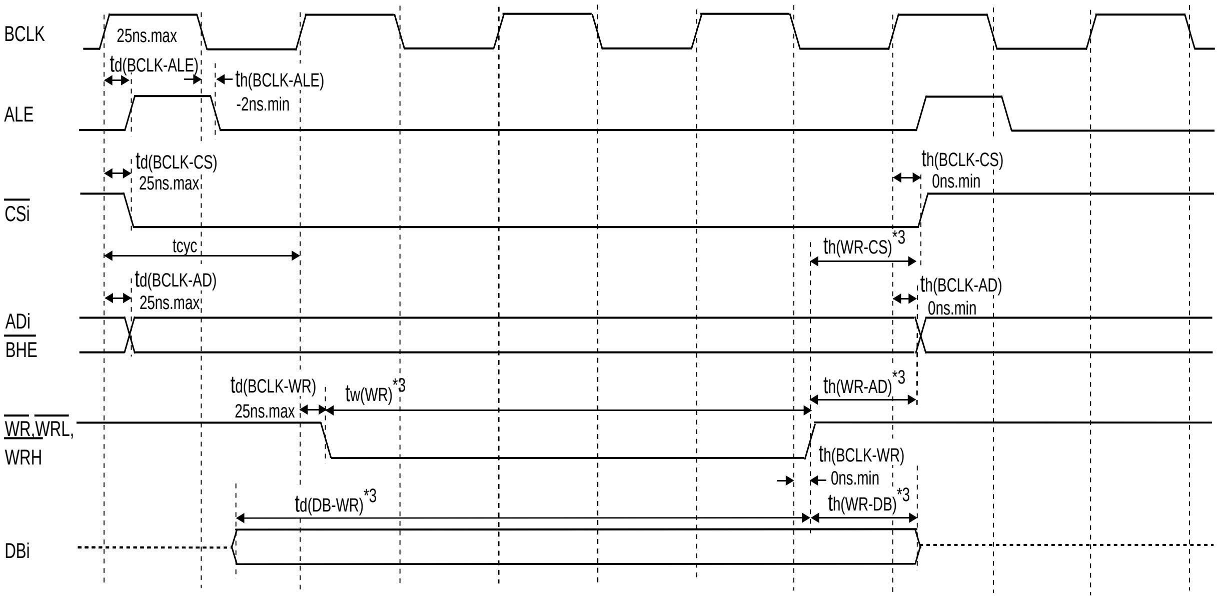
*1:It is a guarantee value with being alone. 55ns.max garantees as $t_{d(BCLK-AD)}$+$t_{su(DB-BCLK)}$.

*2:It depends on operation frequency.

$t_{ac2(RD-DB)}$=(tcyc/2 x m-42)ns.max (m=3, 5 and 7 when 1 wait, 2 wait and 3 wait, respectively.)

$t_{ac2(AD-DB)}$=(tcyc x n-55)ns.max (n=2, 3 and 4 when 1 wait, 2 wait and 3 wait, respectively.)

Write Timing



*3:It depends on operation frequency.

$t_{d(DB-WR)}$=(tcyc x n-40)ns.min

(n=1, 2 and 3 when 1 wait, 2 wait and 3 wait, respectively.)

$t_{h(WR-DB)}$=(tcyc/2-20)ns.min

$t_{h(WR-AD)}$=(tcyc/2-20)ns.min

$t_{h(WR-CS)}$=(tcyc/2-20)ns.min

$t_{w(WR)}$=(tcyc/2 x n-20)ns.min

(n=1, 3 and 5 when 1 wait, 2 wait and 3 wait, respectively.)

Measuring conditions

- V_{CC}=3V±10%
- Input timing voltage
  :Determined with V_{IH}=1.5V, V_{IL}=0.5V
- Output timing voltage
  :Determined with V_{OH}=1.5V, V_{OL}=1.5V

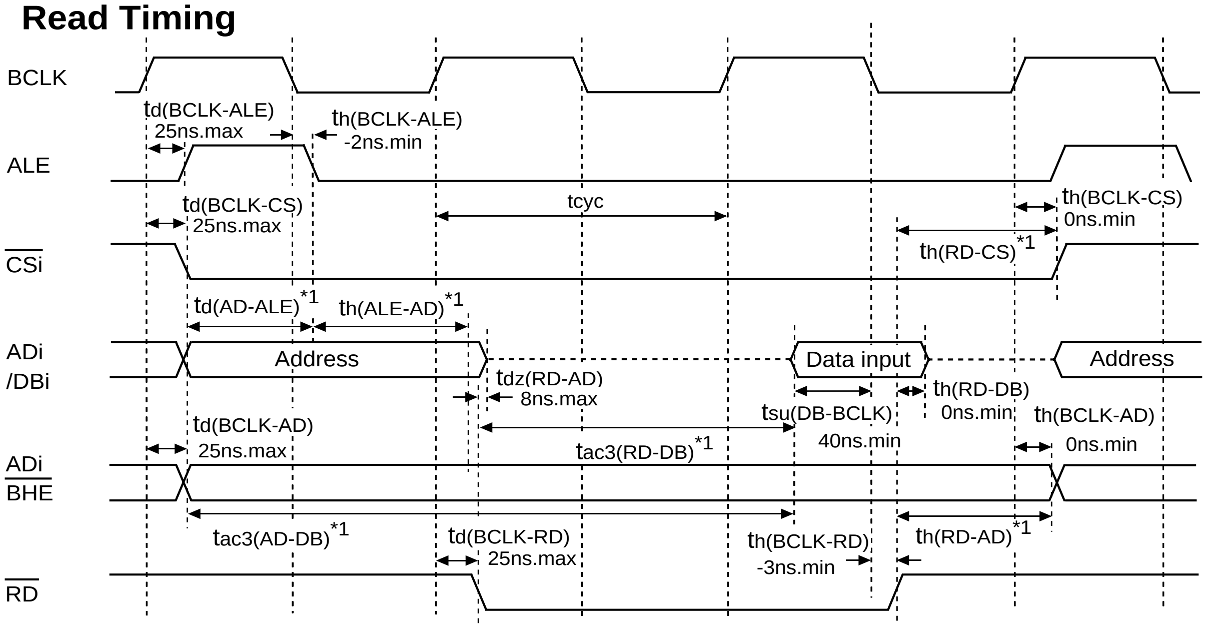
Figure 1.28.18. V_{CC}=3V timing diagram (4)

Memory expansion Mode and Microprocessor Mode

Vcc=3V

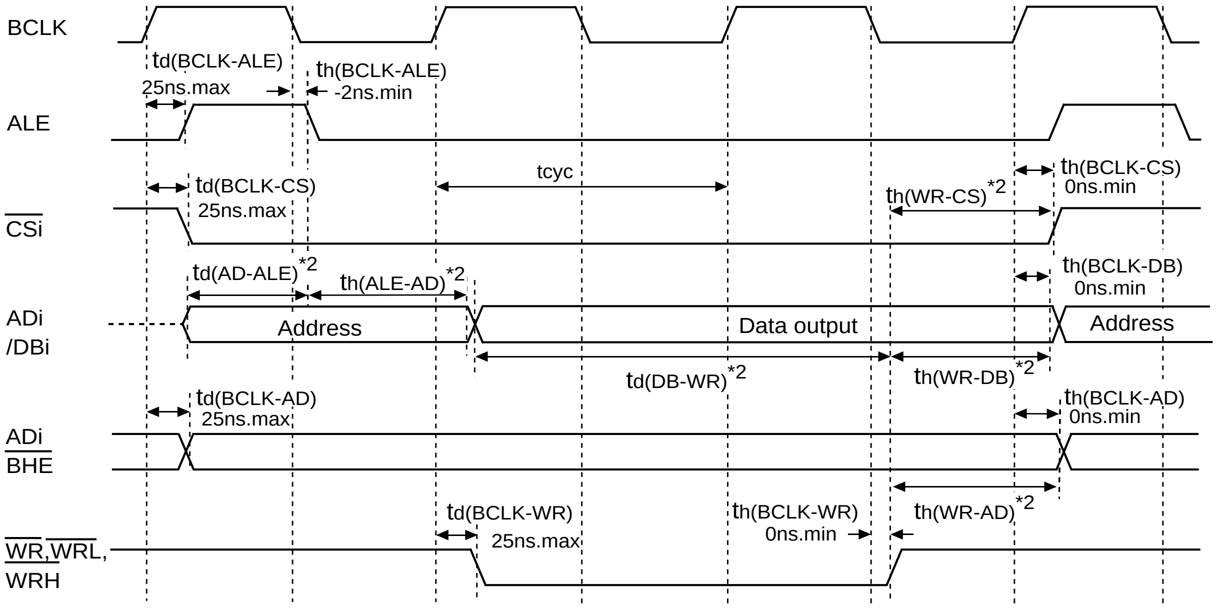
(When accessing external memory area with 2 wait, and select multiplexed bus)

Read Timing



*1:It depends on operation frequency.
$t_{d(AD-ALE)}$=(tcyc/2-27)ns.min
$t_{h(ALE-AD)}$=(tcyc/2-20)ns.min, $t_{h(RD-AD)}$=(tcyc/2-20)ns.min, $t_{h(RD-CS)}$=(tcyc/2-20)ns.min
$t_{ac3(RD-DB)}$=(tcyc/2 x m-55)ns.max (m=3 and 5 when 2 wait and 3 wait, respectively.)
$t_{ac3(AD-DB)}$=(tcyc/2 x n-55)ns.max (n=5 and 7 when 2 wait and 3 wait, respectively.)

Write Timing



*2:It depends on operation frequency.
$t_{d(AD-ALE)}$=(tcyc/2-27)ns.min
$t_{h(ALE-AD)}$=(tcyc/2-20)ns.min, $t_{h(WR-AD)}$=(tcyc/2-20)ns.min
$t_{h(WR-CS)}$=(tcyc/2-20)ns.min, $t_{h(WR-DB)}$=(tcyc/2-20)ns.min
$t_{d(DB-WR)}$=(tcyc/2 x m-40)ns.min
(m=3 and 5 when 2 wait and 3 wait, respectively.)

Measuring conditions
- V_{CC}=3V±10%
- Input timing voltage
  :Determined with V_{IH}=1.5V, V_{IL}=0.5V
- Output timing voltage
  :Determined with V_{OH}=1.5V, V_{OL}=1.5V

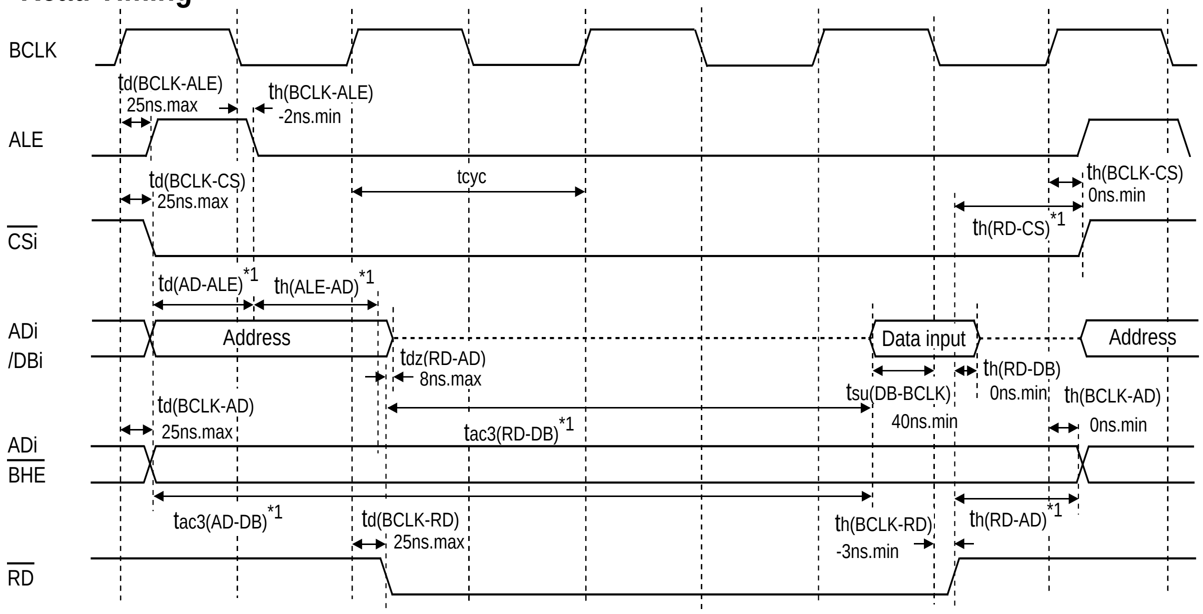
Figure 1.28.19. V_{CC}=3V timing diagram (5)

Memory expansion Mode and Microprocessor Mode

Vcc=3V

(When accessing external memory area with 3 wait, and select multiplexed bus)

Read Timing



*1:It depends on operation frequency.

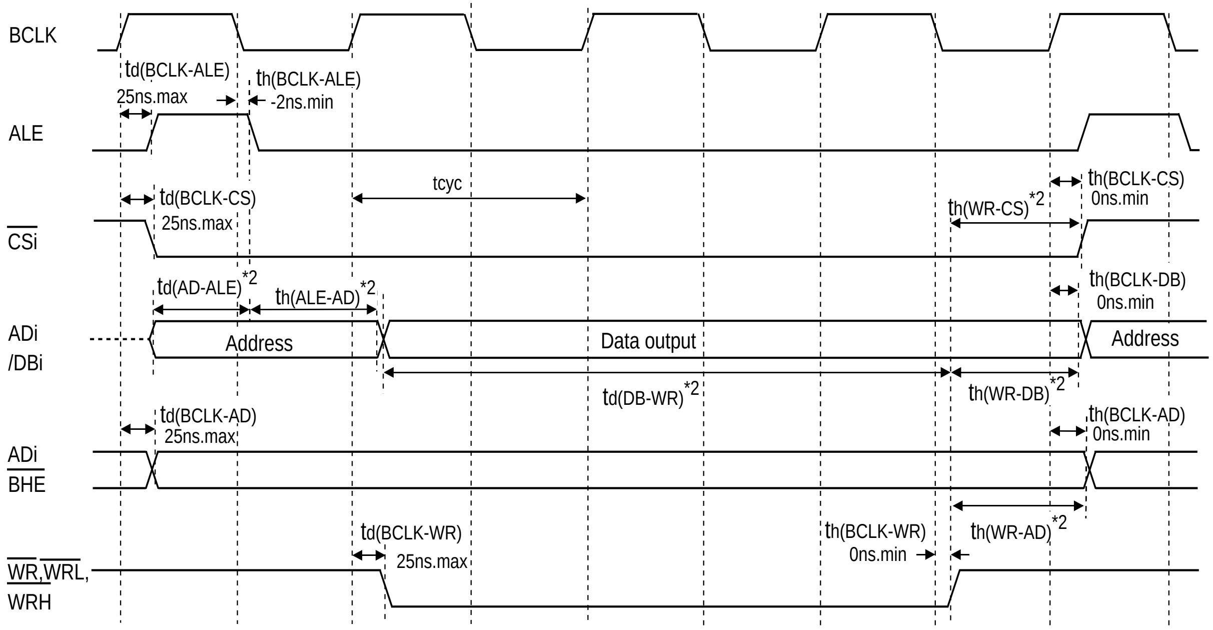
$t_{d(AD-ALE)}$=(tcyc/2-27)ns.min

$t_{h(ALE-AD)}$=(tcyc/2-20)ns.min, $t_{h(RD-AD)}$=(tcyc/2-20)ns.min, $t_{h(RD-CS)}$=(tcyc/2-20)ns.min

$t_{ac3(RD-DB)}$=(tcyc/2 x m-55)ns.max (m=3 and 5 when 2 wait and 3 wait, respectively.)

$t_{ac3(AD-DB)}$=(tcyc/2 x n-55)ns.max (n=5 and 7 when 2 wait and 3 wait, respectively.)

Write Timing



*2:It depends on operation frequency.

$t_{d(AD-ALE)}$=(tcyc/2-27)ns.min

$t_{h(ALE-AD)}$=(tcyc/2-20)ns.min, $t_{h(WR-AD)}$=(tcyc/2-20)ns.min

$t_{h(WR-CS)}$=(tcyc/2-20)ns.min, $t_{h(WR-DB)}$=(tcyc/2-20)ns.min

$t_{d(DB-WR)}$=(tcyc/2 x m-40)ns.min

(m=3 and 5 when 2 wait and 3 wait, respectively.)

Measuring conditions

- V_{CC}=3V±10%
- Input timing voltage
  :Determined with V_{IH}=1.5V, V_{IL}=0.5V
- Output timing voltage
  :Determined with V_{OH}=1.5V, V_{OL}=1.5V

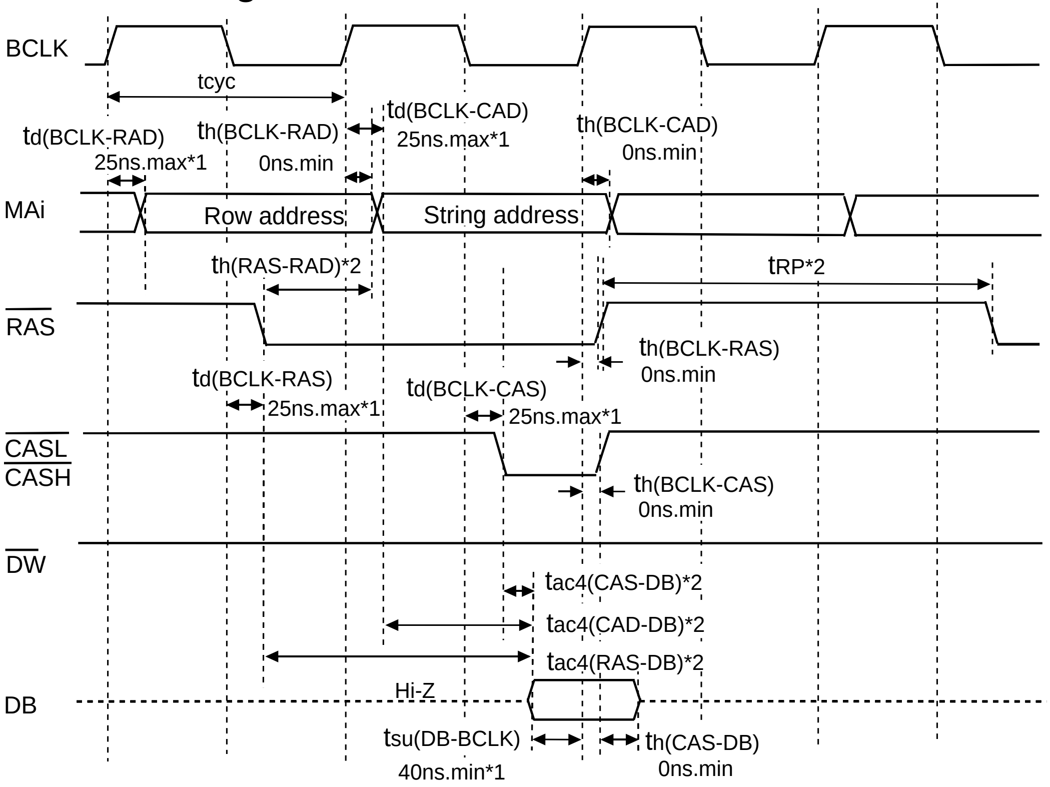
Figure 1.28.20. V_{CC}=3V timing diagram (6)

Memory expansion Mode and Microprocessor Mode

Vcc=3V

(When accessing DRAM area with 1 wait)

Read Timing



*1:It is a guarantee value with being alone. 55ns.max garantees as follows:
td(BCLK-RAS) + tsu(DB-BCLK)
td(BCLK-CAS) + tsu(DB-BCLK)
td(BCLK-CAD) + tsu(DB-BCLK)

*2:It depends on operation frequency.
tac4(RAS-DB)=(tcyc/2 x m-55)ns.max (m=3 and 5 when 1 wait and 2 wait, respectively.)
tac4(CAS-DB)=(tcyc/2 x n-55)ns.max (n=1 and 3 when 1 wait and 2 wait, respectively.)
tac4(CAD-DB)=(tcyc x l-55)ns.max (l=1 and 2 when 1 wait and 2 wait, respectively.)
th(RAS-RAD)=(tcyc/2-25)ns.min
tRP=(tcyc/2 x 3-40)ns.min

Measuring conditions
- V_{CC}=3V±10%
- Input timing voltage
  :Determined with V_{IH}=1.5V, V_{IL}=0.5V
- Output timing voltage
  :Determined with V_{OH}=1.5V, V_{OL}=1.5V

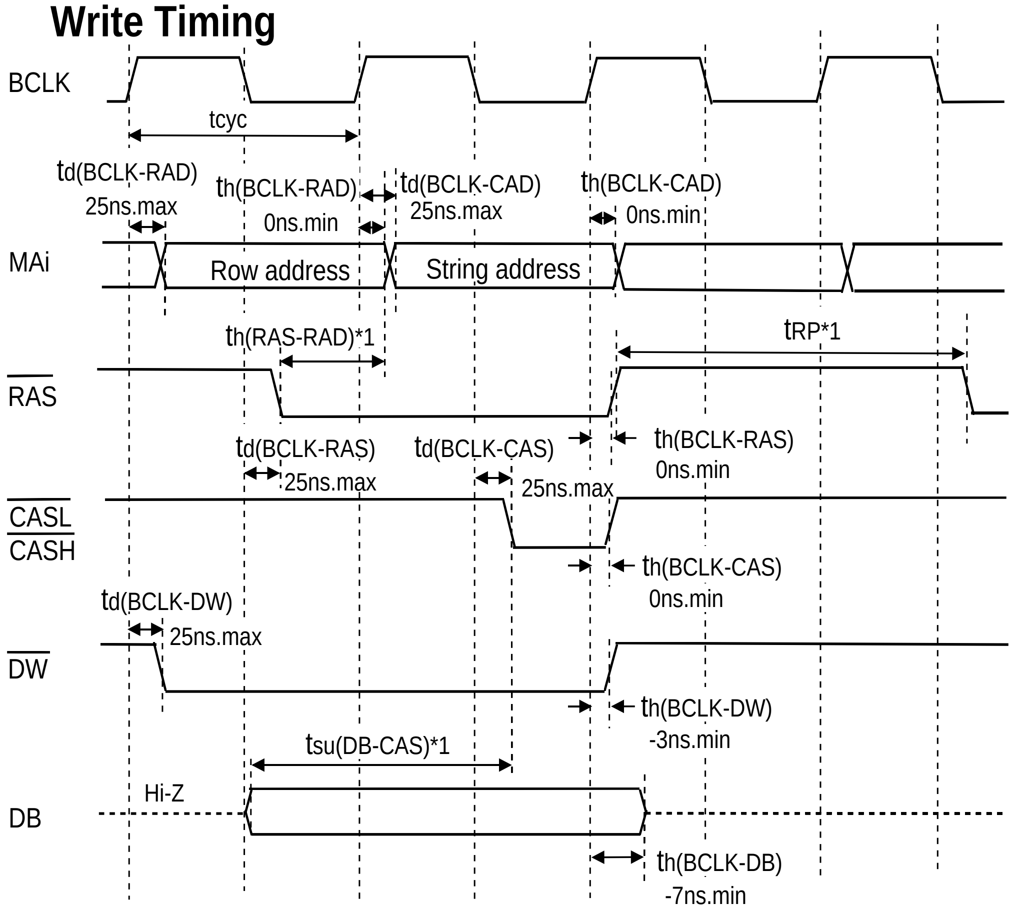
Figure 1.28.21. V_{CC}=3V timing diagram (7)

Memory expansion Mode and Microprocessor Mode

Vcc=3V

(When accessing DRAM area with 1 wait)

Write Timing



*1:It depends on operation frequency.
th(RAS-RAD)=(tcyc/2-25)ns.min
tRP=(tcyc/2 x 3-40)ns.min
tsu(DB-CAS)=(tcyc-40)ns.min

Measuring conditions

- V_{CC}=3V±10%
- Input timing voltage
  :Determined with V_{IH}=1.5V, V_{IL}=0.5V
- Output timing voltage
  :Determined with V_{OH}=1.5V, V_{OL}=1.5V

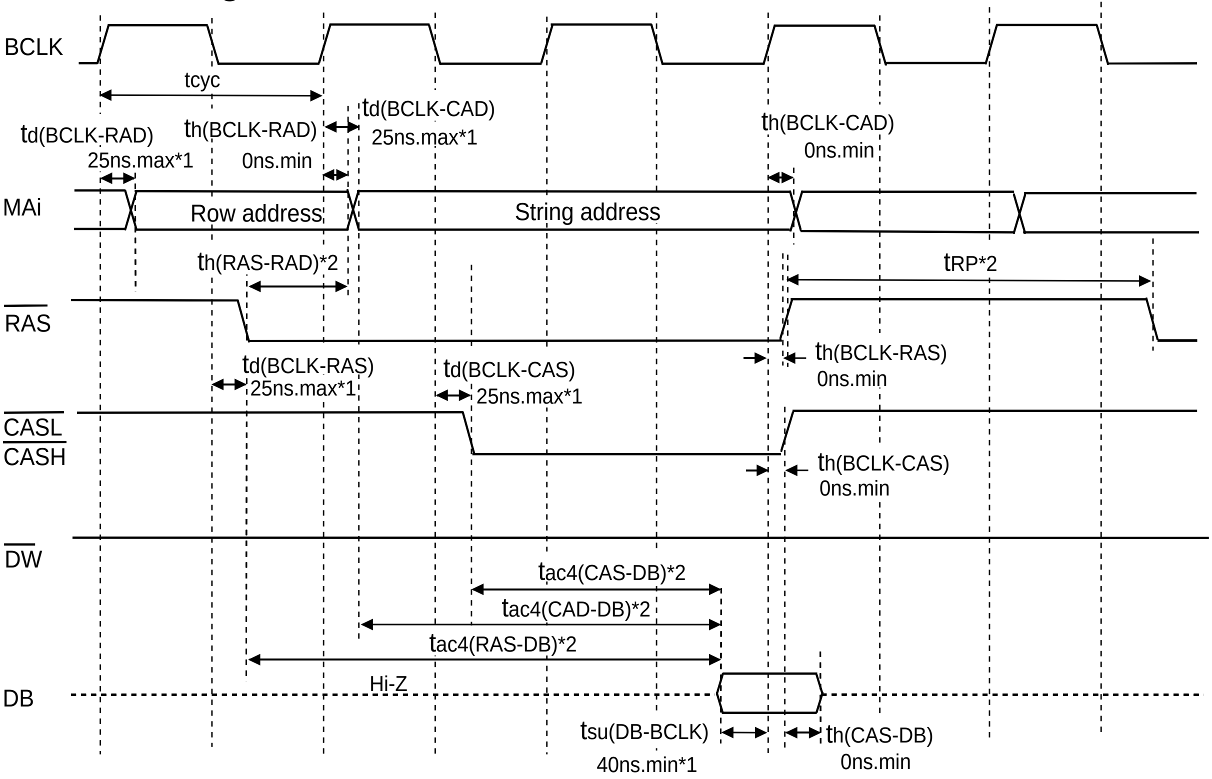
Figure 1.28.22. V_{CC}=3V timing diagram (8)

Memory expansion Mode and Microprocessor Mode

(When accessing DRAM area with 2 wait)

Vcc=3V

Read Timing



*1:It is a guarantee value with being alone. 55ns.max garantees as follows:

td(BCLK-RAS) + tsu(DB-BCLK)

td(BCLK-CAS) + tsu(DB-BCLK)

td(BCLK-CAD) + tsu(DB-BCLK)

*2:It depends on operation frequency.

tac4(RAS-DB)=(tcyc/2 x m-55)ns.max (m=3 and 5 when 1 wait and 2 wait, respectively.)

tac4(CAS-DB)=(tcyc/2 x n-55)ns.max (n=1 and 3 when 1 wait and 2 wait, respectively.)

tac4(CAD-DB)=(tcyc x l-55)ns.max (l=1 and 2 when 1 wait and 2 wait, respectively.)

th(RAS-RAD)=(tcyc/2-25)ns.min

tRP=(tcyc/2 x 3-40)ns.min

Measuring conditions

- VCC=3V±10%
- Input timing voltage
  :Determined with VIH=1.5V, VIL=0.5V
- Output timing voltage
  :Determined with VOH=1.5V, VOL=1.5V

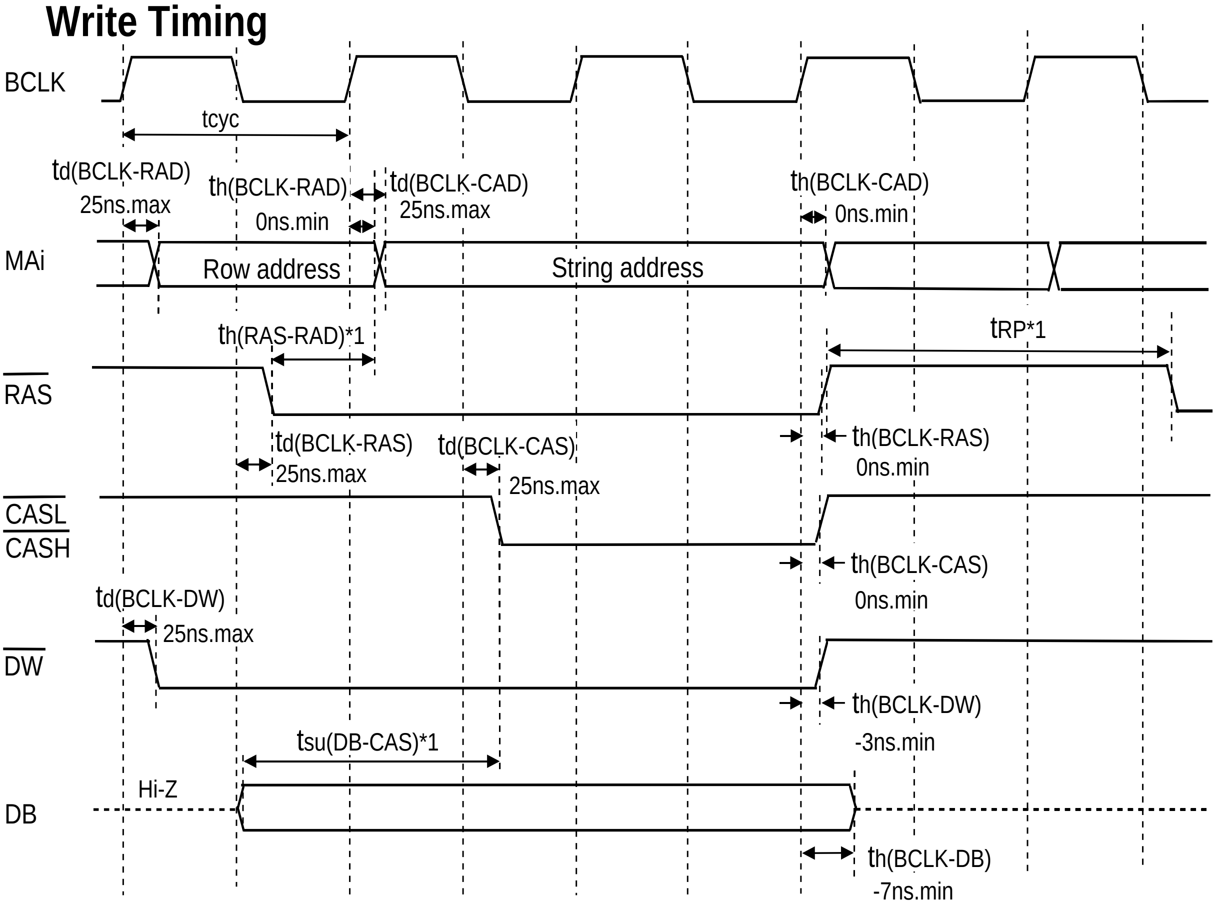
Figure 1.28.23. VCC=3V timing diagram (9)

Memory expansion Mode and Microprocessor Mode

Vcc=3V

(When accessing DRAM area with 2 wait)

Write Timing



*1:It depends on operation frequency.
$t_{h(RAS-RAD)}$=(tcyc/2-25)ns.min
t_{RP}=(tcyc/2 x 3-40)ns.min
$t_{su(DB-CAS)}$=(tcyc-40)ns.min

Measuring conditions

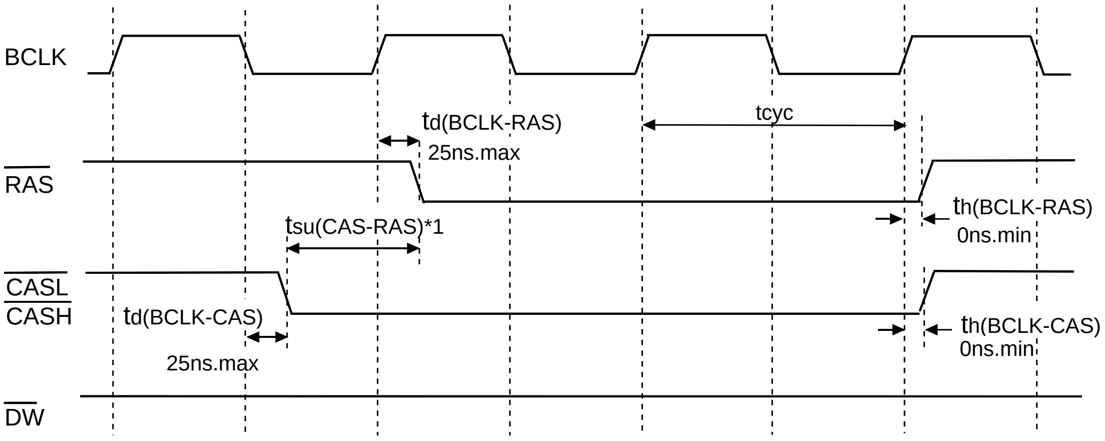
- V_{CC}=3V±10%
- Input timing voltage
  :Determined with V_{IH}=1.5V, V_{IL}=0.5V
- Output timing voltage
  :Determined with V_{OH}=1.5V, V_{OL}=1.5V

Figure 1.28.24. V_{CC}=3V timing diagram (10)

Memory expansion Mode and Microprocessor Mode

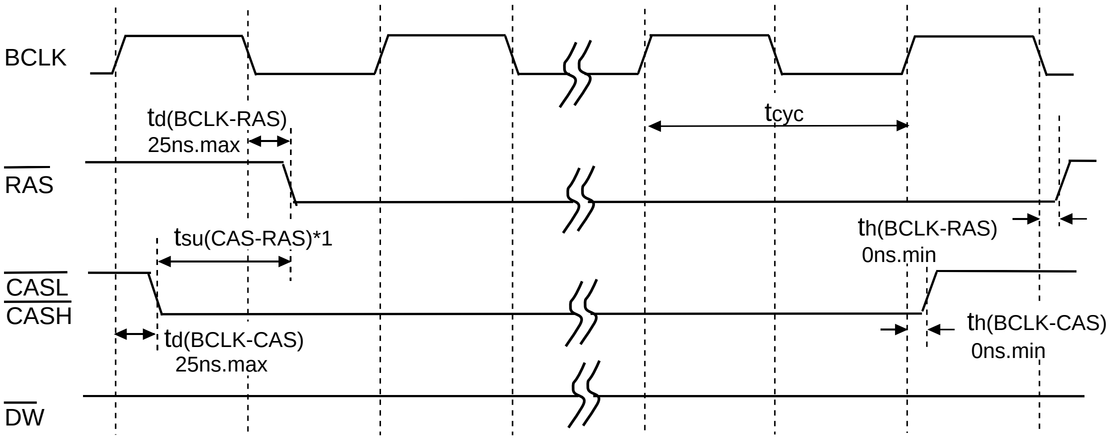
Vcc=3V

Refresh Timing (CAS before RAS refresh)



*1:It depends on operation frequency.
$t_{su(CAS-RAS)}$=(tcyc/2-25)ns.min

Refresh Timing (Self-refresh)



*1:It depends on operation frequency.
$t_{su(CAS-RAS)}$=(tcyc/2-25)ns.min

Measuring conditions

- V_{CC}=3V±10%
- Input timing voltage
  :Determined with V_{IH}=1.5V, V_{IL}=0.5V
- Output timing voltage
  :Determined with V_{OH}=1.5V, V_{OL}=1.5V

Figure 1.28.25. V_{CC}=3V timing diagram (11)

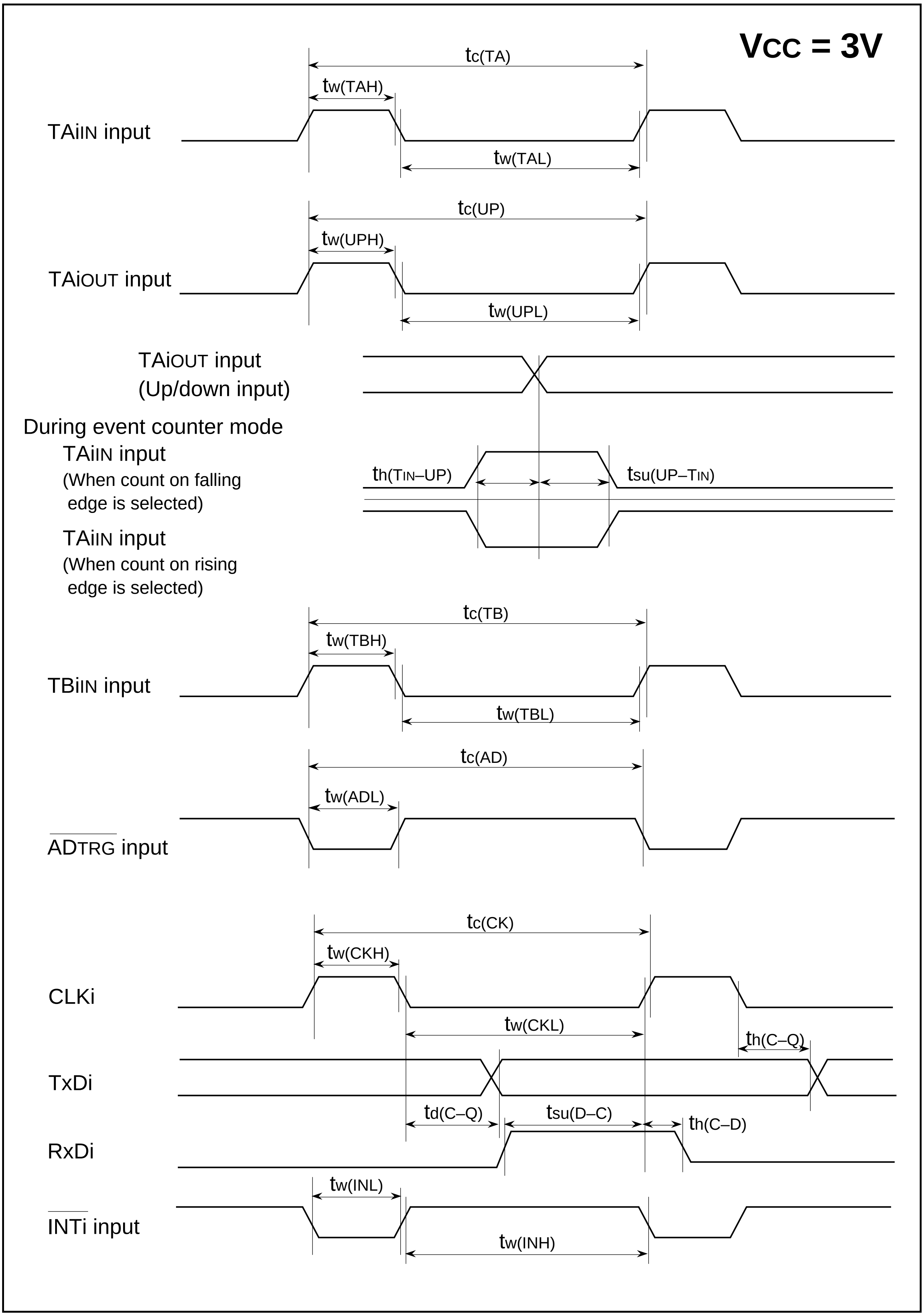
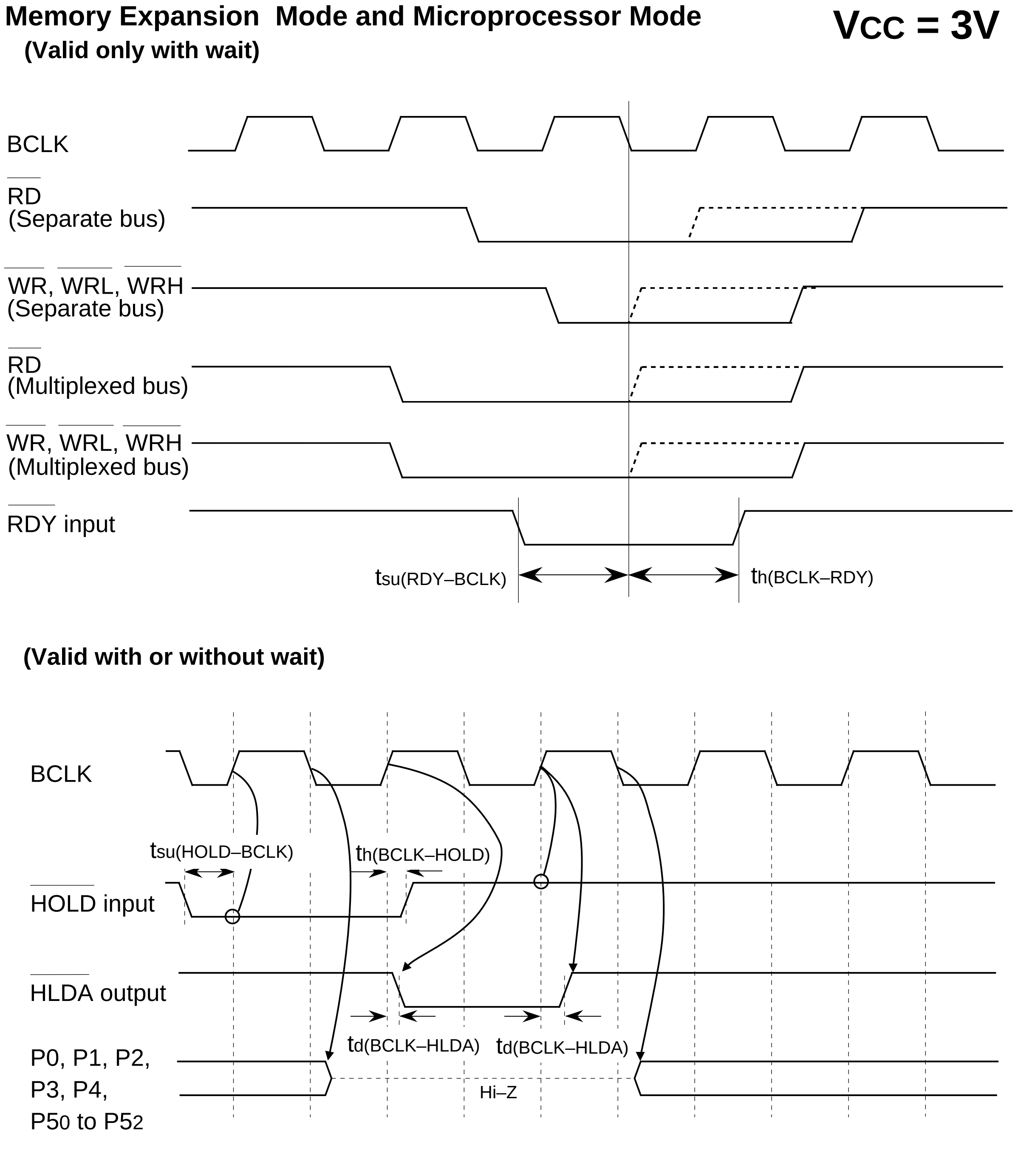


Figure 1.28.26. V_{CC}**=3V timing diagram (12)**



Measuring conditions :

- V_{CC}=3V±10%
- Input timing voltage : Determined with V_{IH}=2.4V, V_{IL}=0.6V
- Output timing voltage : Determined with V_{OH}=1.5V, V_{OL}=1.5V

Figure 1.28.27. V_{cc}=3V timing diagram (13)

Outline Performance

Table 1.29.1 shows the outline performance of the M16C/80 (flash memory version).

Table 1.29.1. Outline Performance of the M16C/80 (flash memory version)

Item		Performance
Power supply voltage		5V version: f(XIN)=20MHz, without wait, 4.2V to 5.5V f(XIN)=10MHz, without wait, 2.7V to 5.5V
Program/erase voltage		5V version: 4.2V to 5.5 V f(BCLK)=12.5MHz, with one wait f(BCLK)=6.25MHz, without wait
Flash memory operation mode		Three modes (parallel I/O, standard serial I/O, CPU rewrite)
Erase block division	User ROM area	See Figure 1.29.3
	Boot ROM area	One division (8 Kbytes) (Note 1)
Program method		In units of pages (in units of 256 bytes)
Erase method		Collective erase/block erase
Program/erase control method		Program/erase control by software command
Protect method		Protected for each block by lock bit
Number of commands		8 commands
Program/erase count		100 times
Data holding		10 years
ROM code protect		Parallel I/O and standard serial modes are supported.

Note: The boot ROM area contains a standard serial I/O mode control program which is stored in it when shipped from the factory. This area can be erased and programmed in only parallel I/O mode.

The following shows Mitsubishi plans to develop a line of M16C/80 products (flash memory version).

(1) ROM capacity

(2) Package 100P6S-A ... Plastic molded QFP
100P6Q-A ... Plastic molded QFP
144P6Q-A ... Plastic molded QFP

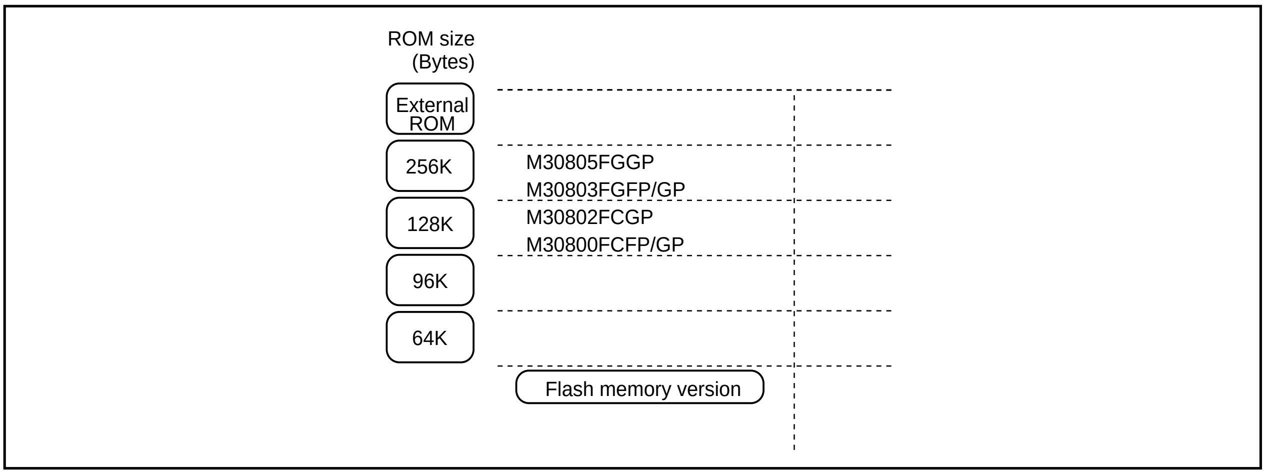


Figure 1.29.1. ROM Expansion

The following lists the M16C/80 products to be supported in the future.

Table 1.29.2. Product List

As of July, 2000

Type No		ROM capacity	RAM capacity	Package type	Remarks
M30800FCFP	*	128 Kbytes	10 Kbytes	100P6S-A	
M30800FCGP	*			100P6Q-A	
M30803FGFP	*	256 Kbytes	20 Kbytes	100P6S-A	
M30803FGGP	*			100P6Q-A	
M30802FCGP	*	128 Kbytes	10 Kbytes	144P6Q-A	
M30805FGGP	*	256 Kbytes	20 Kbytes		

* : Under development

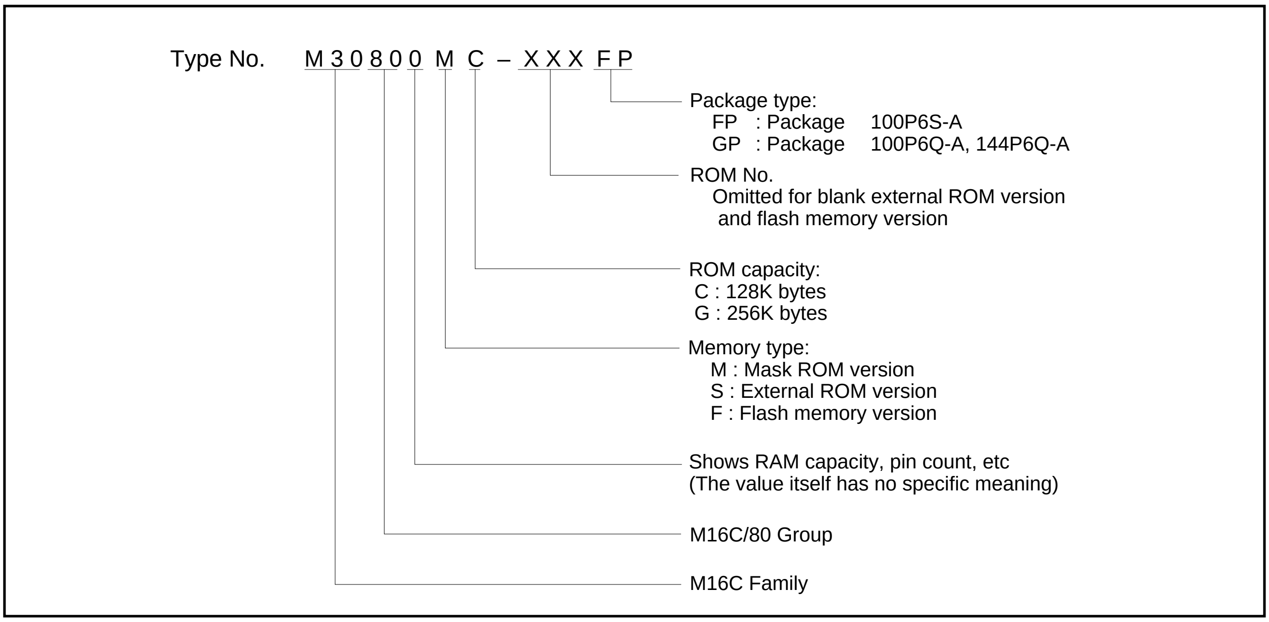

Type No. M 3 0 8 0 0 M C – X X X F P
Package type:
FP : Package 100P6S-A
GP : Package 100P6Q-A, 144P6Q-A
ROM No.
Omitted for blank external ROM version and flash memory version
ROM capacity:
C : 128K bytes
G : 256K bytes
Memory type:
M : Mask ROM version
S : External ROM version
F : Flash memory version
Shows RAM capacity, pin count, etc
(The value itself has no specific meaning)
M16C/80 Group
M16C Family

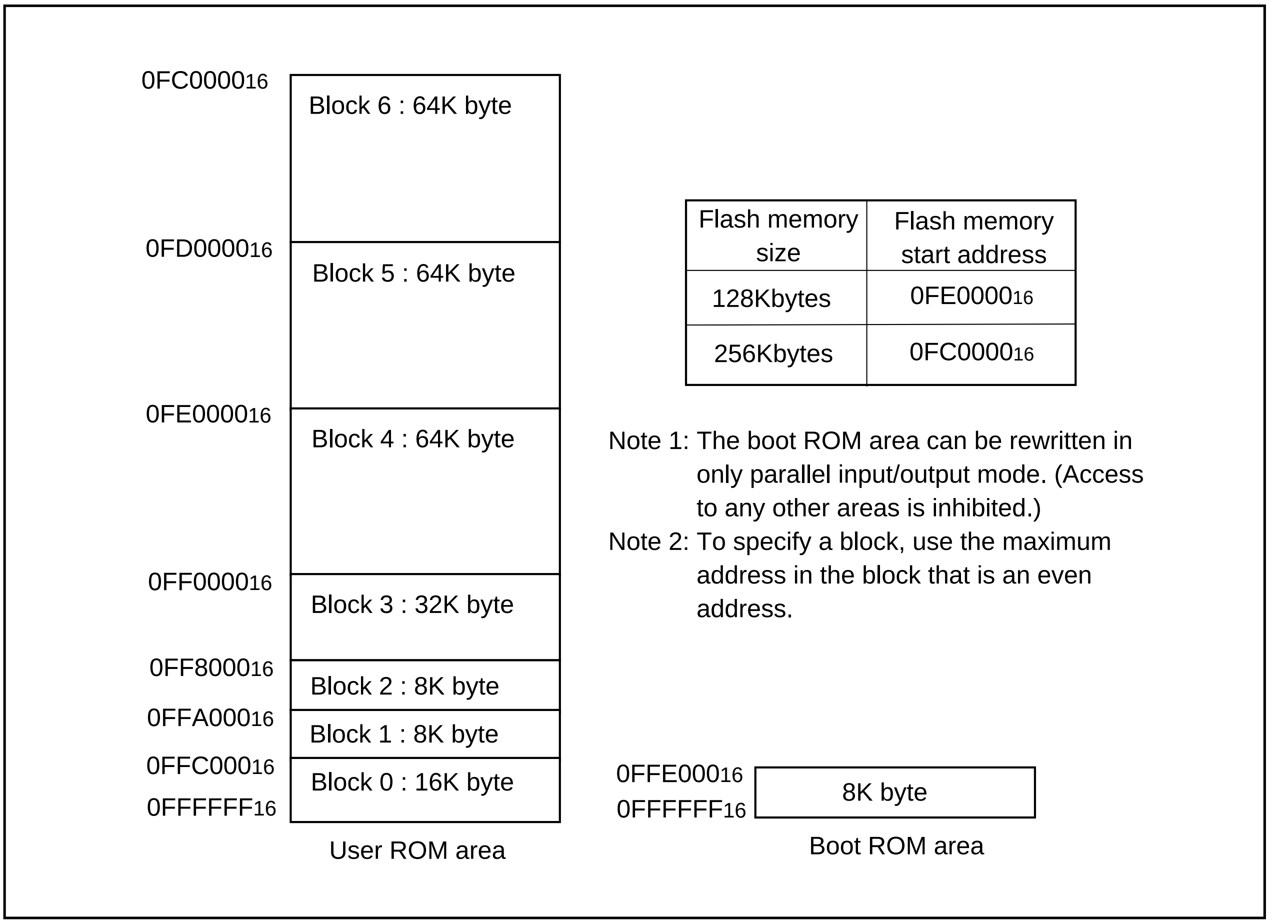
Figure 1.29.2. Type No., memory size, and package

Flash Memory

The M16C/80 (flash memory version) contains the flash memory that can be rewritten with a single voltage of 5 V. For this flash memory, three flash memory modes are available in which to read, program, and erase: parallel I/O and standard serial I/O modes in which the flash memory can be manipulated using a programmer and a CPU rewrite mode in which the flash memory can be manipulated by the Central Processing Unit (CPU). Each mode is detailed in the pages to follow.

The flash memory is divided into several blocks as shown in Figure 1.29.3, so that memory can be erased one block at a time. Each block has a lock bit to enable or disable execution of an erase or program operation, allowing for data in each block to be protected.

In addition to the ordinary user ROM area to store a microcomputer operation control program, the flash memory has a boot ROM area that is used to store a program to control rewriting in CPU rewrite and standard serial I/O modes. This boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the factory. However, the user can write a rewrite control program in this area that suits the user's application system. This boot ROM area can be rewritten in only parallel I/O mode.



Flash memory size	Flash memory start address
128Kbytes	0FE0000$_{16}$
256Kbytes	0FC0000$_{16}$

Note 1: The boot ROM area can be rewritten in only parallel input/output mode. (Access to any other areas is inhibited.)

Note 2: To specify a block, use the maximum address in the block that is an even address.

Figure 1.29.3. Block diagram of flash memory version

CPU Rewrite Mode

In CPU rewrite mode, the on-chip flash memory can be operated on (read, program, or erase) under control of the Central Processing Unit (CPU).

In CPU rewrite mode, only the user ROM area shown in Figure 1.29.3 can be rewritten; the boot ROM area cannot be rewritten. Make sure the program and block erase commands are issued for only the user ROM area and each block area.

The control program for CPU rewrite mode can be stored in either user ROM or boot ROM area. In the CPU rewrite mode, because the flash memory cannot be read from the CPU, the rewrite control program must be transferred to any area other than the internal flash memory before it can be executed.

Microcomputer Mode and Boot Mode

The control program for CPU rewrite mode must be written into the user ROM or boot ROM area in parallel I/O mode beforehand. (If the control program is written into the boot ROM area, the standard serial I/O mode becomes unusable.)

See Figure 1.29.3 for details about the boot ROM area.

Normal microcomputer mode is entered when the microcomputer is reset with pulling CNV$_{SS}$ pin low. In this case, the CPU starts operating using the control program in the user ROM area.

When the microcomputer is reset by pulling the P5$_5$ pin low, the CNV$_{SS}$ pin high, and the P5$_0$ pin high, the CPU starts operating using the control program in the boot ROM area. This mode is called the “boot” mode. The control program in the boot ROM area can also be used to rewrite the user ROM area.

Block Address

Block addresses refer to the maximum even address of each block. These addresses are used in the block erase command, lock bit program command, and read lock status command.

Outline Performance (CPU Rewrite Mode)

In the CPU rewrite mode, the CPU erases, programs and reads the internal flash memory as instructed by software commands. Operations must be executed from a memory other than the internal flash memory, such as the internal RAM.

When the CPU rewrite mode select bit (bit 1 at address 0377_{16}) is set to “1”, transition to CPU rewrite mode occurs and software commands can be accepted.

In the CPU rewrite mode, write to and read from software commands and data into even-numbered address (“0” for byte address A0) in 16-bit units. Always write 8-bit software commands into even-numbered address. Commands are ignored with odd-numbered addresses.

Use software commands to control program and erase operations. Whether a program or erase operation has terminated normally or in error can be verified by reading the status register.

Figure 1.30.1 shows the flash memory control register 0 and the flash memory control register 1.

Bit 0 of the flash memory control register 0 is the RY/$\overline{\text{BY}}$ status flag used exclusively to read the operating status of the flash memory. During programming and erase operations, it is “0”. Otherwise, it is “1”.

Bit 1 of the flash memory control register 0 is the CPU rewrite mode select bit. The CPU rewrite mode is entered by setting this bit to “1”, so that software commands become acceptable. In CPU rewrite mode, the CPU becomes unable to access the internal flash memory directly. Therefore, write bit 1 in an area other than the internal flash memory. To set this bit to “1”, it is necessary to write “0” and then write “1” in succession when NMI pin is "H" level. The bit can be set to “0” by only writing a “0” .

Bit 2 of the flash memory control register 0 is a lock bit disable bit. By setting this bit to “1”, it is possible to disable erase and write protect (block lock) effectuated by the lock bit data. The lock bit disable select bit only disables the lock bit function; it does not change the lock data bit value. However, if an erase operation is performed when this bit =“1”, the lock bit data that is “0” (locked) is set to “1” (unlocked) after erasure. To set this bit to “1”, it is necessary to write “0” and then write “1” in succession. This bit can be manipulated only when the CPU rewrite mode select bit = “1”.

Bit 3 of the flash memory control register 0 is the flash memory reset bit used to reset the control circuit of the internal flash memory. This bit is used when exiting CPU rewrite mode and when flash memory access has failed. When the CPU rewrite mode select bit is “1”, writing “1” for this bit resets the control circuit. To release the reset, it is necessary to set this bit to “0”.

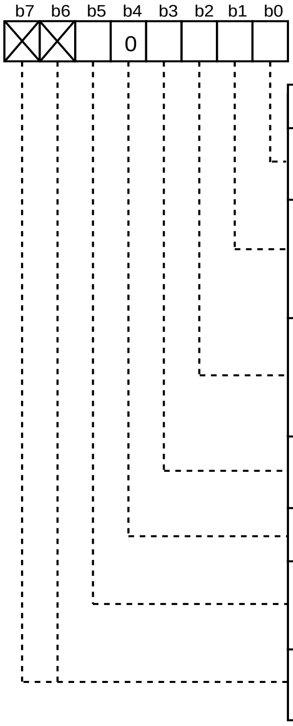
Bit 5 of the flash memory control register 0 is a user ROM area select bit which is effective in only boot mode. If this bit is set to “1” in boot mode, the area to be accessed is switched from the boot ROM area to the user ROM area. When the CPU rewrite mode needs to be used in boot mode, set this bit to “1”. Note that if the microcomputer is booted from the user ROM area, it is always the user ROM area that can be accessed and this bit has no effect. When in boot mode, the function of this bit is effective regardless of whether the CPU rewrite mode is on or off. Use the control program except in the internal flash memory to rewrite this bit.

Bit 3 of the flash memory control register 1 turns power supply to the internal flash memory on/off. When this bit is set to “1”, power is not supplied to the internal flash memory, thus power consumption can be reduced. However, in this state, the internal flash memory cannot be accessed. To set this bit to “1”, it is necessary to write “0” and then write “1” in succession. Use this bit mainly in the low speed mode (when X_{CIN} is the block count source of BCLK).

When the CPU is shifted to the stop or wait modes, power to the internal flash memory is automatically shut off. It is reconnected automatically when CPU operation is restored. Therefore, it is not particularly necessary to set flash memory control register 1.

Figure 1.30.2 shows a flowchart for setting/releasing the CPU rewrite mode. Figure 1.30.3 shows a flowchart for shifting to the low speed mode. Always perform operation as indicated in these flowcharts.

Flash memory control register 0



Symbol	Address	When reset
FMR0	0377_{16}	XX000001$_2$

Bit symbol	Bit name	Function	R	W
FMR00	RY/$\overline{\text{BY}}$ status flag	0: Busy (being written or erased) 1: Ready	○	×
FMR01	CPU rewrite mode select bit (Note 1)	0: Normal mode (Software commands invalid) 1: CPU rewrite mode (Software commands acceptable)	○	○
FMR02	Lock bit disable bit (Note 2)	0: Block lock by lock bit data is enabled 1: Block lock by lock bit data is disabled	○	○
FMR03	Flash memory reset bit (Note 3)	0: Normal operation 1: Reset	○	○
Reserved bit		Must always be set to “0”	○	○
FMR05	User ROM area select bit (Note 4) (Effective in only boot mode)	0: Boot ROM area is accessed 1: User ROM area is accessed	○	○
Nothing is assigned. When write, set "0". When read, values are indeterminate.			—	—

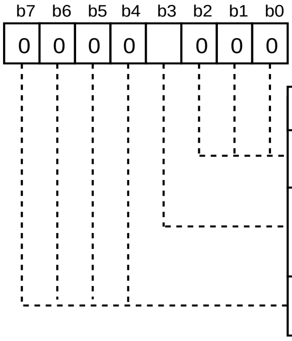
Note 1: For this bit to be set to “1”, the user needs to write a “0” and then a “1” to it in succession. When it is not this procedure, it is not enacted in “1”. This is necessary to ensure that no interrupt or DMA transfer will be executed during the interval. Use the control program except in the internal flash memory for write to this bit. Also write to this bit when NMI pin is "H" level.

Note 2: For this bit to be set to “1”, the user needs to write a “0” and then a “1” to it in succession when the CPU rewrite mode select bit = “1”. When it is not this procedure, it is not enacted in “1”. This is necessary to ensure that no interrupt or DMA transfer will be executed during the interval.

Note 3: Effective only when the CPU rewrite mode select bit = 1. Set this bit to 0 subsequently after setting it to 1 (reset).

Note 4: Use the control program except in the internal flash memory for write to this bit.

Flash memory control register 1

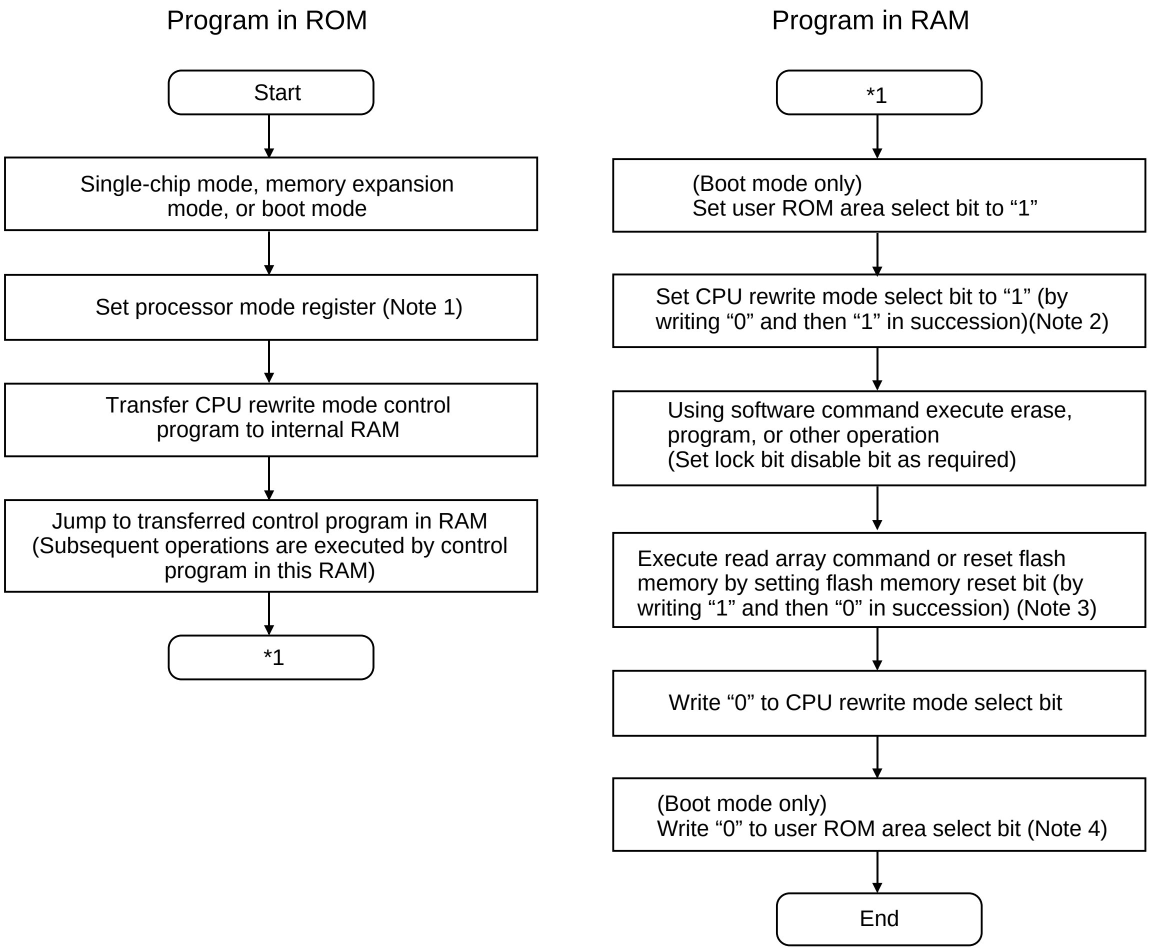


Symbol	Address	When reset
FMR1	0376_{16}	XXXX0XXX$_2$

Bit symbol	Bit name	Function	R	W
Reserved bit		Must always be set to “0”	—	○
FMR13	Flash memory power supply-OFF bit (Note)	0: Flash memory power supply is connected 1: Flash memory power supply-off	○	○
Reserved bit		Must always be set to “0”	—	○

Note : For this bit to be set to “1”, the user needs to write a “0” and then a “1” to it in succession. When it is not this procedure, it is not enacted in “1”. This is necessary to ensure that no interrupt or DMA transfer will be executed during the interval. Use the control program except in the internal flash memory for write to this bit.
During parallel I/O mode,programming,erase or read of flash memory is not controlled by this bit,only by external pins.

Figure 1.30.1. Flash memory control registers


Program in ROM
Start
Single-chip mode, memory expansion mode, or boot mode
Set processor mode register (Note 1)
Transfer CPU rewrite mode control program to internal RAM
Jump to transferred control program in RAM (Subsequent operations are executed by control program in this RAM)
*1
Program in RAM
*1
(Boot mode only) Set user ROM area select bit to “1”
Set CPU rewrite mode select bit to “1” (by writing “0” and then “1” in succession)(Note 2)
Using software command execute erase, program, or other operation (Set lock bit disable bit as required)
Execute read array command or reset flash memory by setting flash memory reset bit (by writing “1” and then “0” in succession) (Note 3)
Write “0” to CPU rewrite mode select bit
(Boot mode only) Write “0” to user ROM area select bit (Note 4)
End

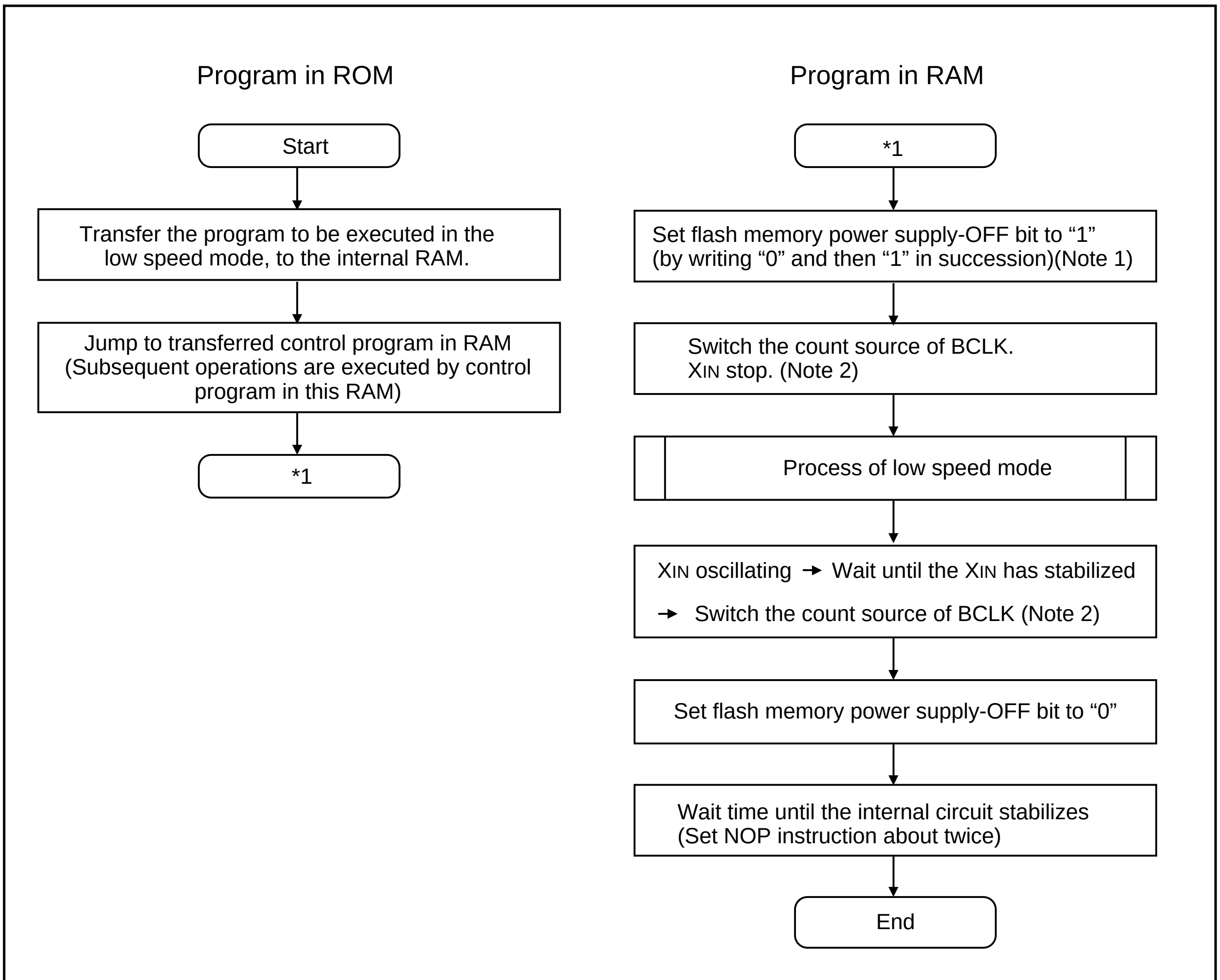
Note 1: During CPU rewrite mode, set the main clock frequency as shown below using the main clock division register (address $000C_{16}$):
6.25 MHz or less when wait bit (bit 2 at address 0005_{16}) = “0” (without internal access wait state)
12.5 MHz or less when wait bit (bit 2 at address 0005_{16}) = “1” (with internal access wait state)

Note 2: For CPU rewrite mode select bit to be set to “1”, the user needs to write a “0” and then a “1” to it in succession. When it is not this procedure, it is not enacted in “1”. This is necessary to ensure that no interrupt or DMA transfer will be executed during the interval. Use the program except in the internal flash memory for write to this bit. Also write to this bit when NMI pin is "H" level.

Note 3: Before exiting the CPU rewrite mode after completing erase or program operation, always be sure to execute a read array command or reset the flash memory.

Note 4: “1” can be set. However, when this bit is “1”, user ROM area is accessed.

Figure 1.30.2. CPU rewrite mode set/reset flowchart


Program in ROM
Start
Transfer the program to be executed in the low speed mode, to the internal RAM.
Jump to transferred control program in RAM (Subsequent operations are executed by control program in this RAM)
*1
Program in RAM
*1
Set flash memory power supply-OFF bit to “1” (by writing “0” and then “1” in succession)(Note 1)
Switch the count source of BCLK. XIN stop. (Note 2)
Process of low speed mode
XIN oscillating → Wait until the XIN has stabilized → Switch the count source of BCLK (Note 2)
Set flash memory power supply-OFF bit to “0”
Wait time until the internal circuit stabilizes (Set NOP instruction about twice)
End

Note 1: For flash memory power supply-OFF bit to be set to “1”, the user needs to write a “0” and then a “1” to it in succession. When it is not this procedure, it is not enacted in “1”. This is necessary to ensure that no interrupt or DMA transfer will be executed during the interval.

Note 2: Before the count source for BCLK can be changed from X_{IN} to X_{CIN} or vice versa, the clock to which the count source is going to be switched must be oscillating stably.

Figure 1.30.3. Shifting to the low speed mode flowchart

Precautions on CPU Rewrite Mode

Described below are the precautions to be observed when rewriting the flash memory in CPU rewrite mode.

(1) Operation speed

During CPU rewrite mode, set the main clock frequency as shown below using the main clock division register (address $000C_{16}$):
6.25 MHz or less when wait bit (bit 2 at address 0005_{16}) = 0 (without internal access wait state)
12.5 MHz or less when wait bit (bit 2 at address 0005_{16}) = 1 (with internal access wait state)

(2) Instructions inhibited against use

The instructions listed below cannot be used during CPU rewrite mode because they refer to the internal data of the flash memory:
UND instruction, INTO instruction, JMPS instruction, JSRS instruction, and BRK instruction

(3) Interrupts inhibited against use

The address match interrupt cannot be used during CPU rewrite mode because they refer to the internal data of the flash memory. If interrupts have their vector in the variable vector table, they can be used by transferring the vector into the RAM area. The $\overline{\text{NMI}}$ and watchdog timer interrupts each can be used to change the CPU rewrite mode select bit forcibly to normal mode (FMR01="0") upon occurrence of the interrupt. Since the rewrite operation is halted when the $\overline{\text{NMI}}$ and watchdog timer interrupts occur, set the CPU rewite mode select bit to "1" and the erase/program operation needs to be performed over again.

(4) Reset

Reset input is always accepted.

(5) Access disable

Write CPU rewrite mode select bit, flash memory power supply-OFF bit and user ROM area select bit in an area other than the internal flash memory.

(6) How to access

For CPU rewrite mode select bit, lock bit disable bit, and flash memory power supply-OFF bit to be set to “1”, the user needs to write a “0” and then a “1” to it in succession. When it is not this procedure, it is not enacted in “1”. This is necessary to ensure that no interrupt or DMA transfer will be executed during the interval.
Write to the CPU rewrite mode select bit when NMI pin is "H" level.

(7)Writing in the user ROM area

If power is lost while rewriting blocks that contain the flash rewrite program with the CPU rewrite mode, those blocks may not be correctly rewritten and it is possible that the flash memory can no longer be rewritten after that. Therefore, it is recommended to use the standard serial I/O mode or parallel I/O mode to rewrite these blocks.

(8)Using the lock bit

To use the CPU rewrite mode, use a boot program that can set and cancel the lock command.

Software Commands

Table 1.30.1 lists the software commands available with the M16C/62A (flash memory version).
After setting the CPU rewrite mode select bit to 1, write a software command to specify an erase or program operation. Note that when entering a software command, the upper byte (D_8 to D_{15}) is ignored.
The content of each software command is explained below.

Table 1.30.1. List of software commands (CPU rewrite mode)

Command	First bus cycle			Second bus cycle			Third bus cycle		
	Mode	Address	Data (D_0 to D_7)	Mode	Address	Data (D_0 to D_7)	Mode	Address	Data (D_0 to D_7)
Read array	Write	X (Note 6)	FF_{16}						
Read status register	Write	X	70_{16}	Read	X	SRD (Note 2)			
Clear status register	Write	X	50_{16}						
Page program (Note 3)	Write	X	41_{16}	Write	WA0 (Note 3)	WD0 (Note 3)	Write	WA1	WD1
Block erase	Write	X	20_{16}	Write	BA (Note 4)	$D0_{16}$			
Erase all unlock block	Write	X	$A7_{16}$	Write	X	$D0_{16}$			
Lock bit program	Write	X	77_{16}	Write	BA	$D0_{16}$			
Read lock bit status	Write	X	71_{16}	Read	BA	D_6 (Note 5)			

Note 1: When a software command is input, the high-order byte of data (D_8 to D_{15}) is ignored.
Note 2: SRD = Status Register Data
Note 3: WA = Write Address, WD = Write Data
WA and WD must be set sequentially from 00_{16} to FE_{16} (byte address; however, an even address). The page size is 256 bytes.
Note 4: BA = Block Address (Enter the maximum address of each block that is an even address.)
Note 5: D_6 corresponds to the block lock status. Block not locked when D_6 = 1, block locked when D_6 = 0.
Note 6: X denotes a given address in the user ROM area (that is an even address).

Read Array Command (FF_{16})

The read array mode is entered by writing the command code "FF_{16}" in the first bus cycle. When an even address to be read is input in one of the bus cycles that follow, the content of the specified address is read out at the data bus (D_0–D_{15}), 16 bits at a time.
The read array mode is retained intact until another command is written.

Read Status Register Command (70_{16})

When the command code "70_{16}" is written in the first bus cycle, the content of the status register is read out at the data bus (D_0–D_7) by a read in the second bus cycle.
The status register is explained in the next section.

Clear Status Register Command (50_{16})

This command is used to clear the bits SR3 to 5 of the status register after they have been set. These bits indicate that operation has ended in an error. To use this command, write the command code "50_{16}" in the first bus cycle.

Page Program Command (41_{16})

Page program allows for high-speed programming in units of 256 bytes. Page program operation starts when the command code "41_{16}" is written in the first bus cycle. In the second bus cycle through the 129th bus cycle, the write data is sequentially written 16 bits at a time. At this time, the addresses A_0-A_7 need to be incremented by 2 from "00_{16}" to "FE_{16}." When the system finishes loading the data, it starts an auto write operation (data program and verify operation).

Whether the auto write operation is completed can be confirmed by reading the status register or the flash memory control register 0. At the same time the auto write operation starts, the read status register mode is automatically entered, so the content of the status register can be read out. The status register bit 7 (SR7) is set to 0 at the same time the auto write operation starts and is returned to 1 upon completion of the auto write operation. In this case, the read status register mode remains active until the Read Array command (FF_{16}) or Read Lock Bit Status command (71_{16}) is written or the flash memory is reset using its reset bit.

The RY/$\overline{\text{BY}}$ status flag of the flash memory control register 0 is 0 during auto write operation and 1 when the auto write operation is completed as is the status register bit 7.

After the auto write operation is completed, the status register can be read out to know the result of the auto write operation. For details, refer to the section where the status register is detailed.

Figure 1.30.4 shows an example of a page program flowchart.

Each block of the flash memory can be write protected by using a lock bit. For details, refer to the section where the data protect function is detailed.

Additional writes to the already programmed pages are prohibited.

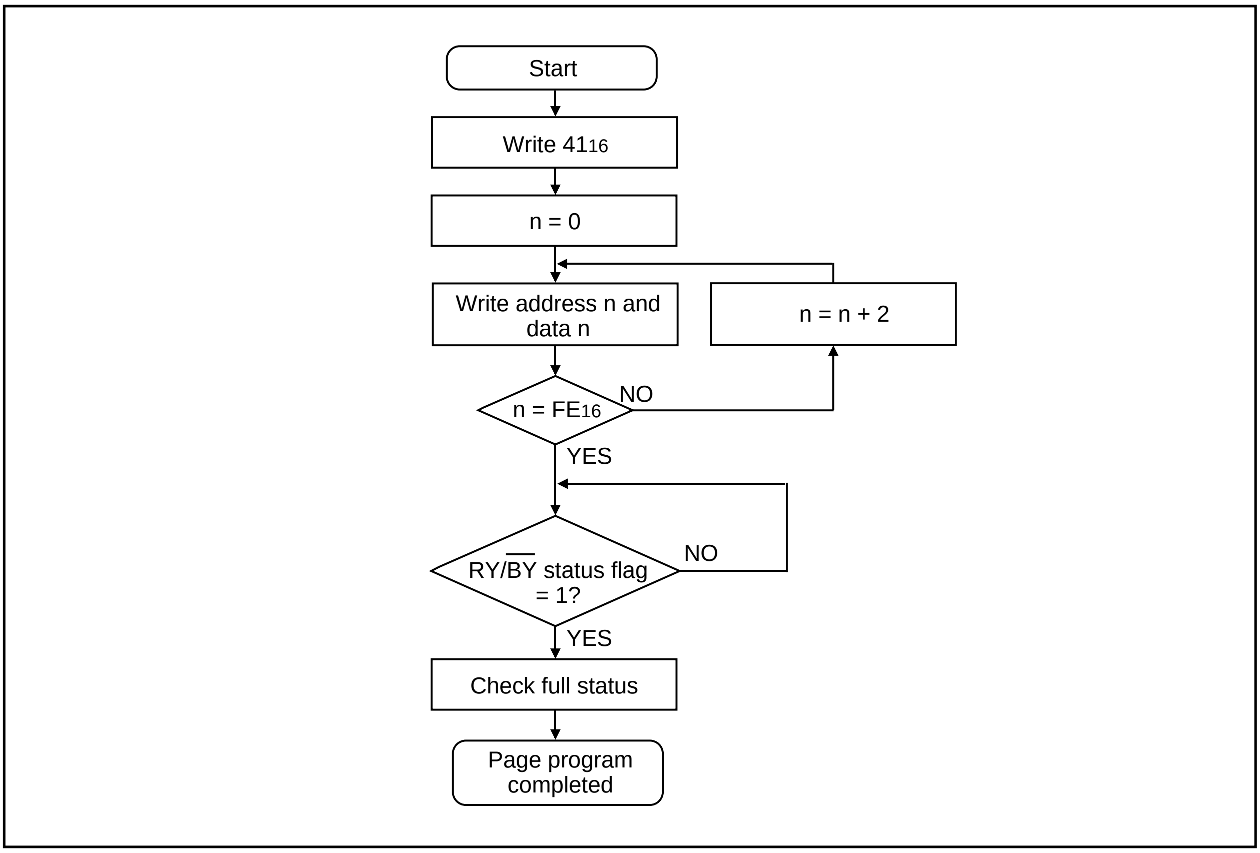


Figure 1.30.4. Page program flowchart

Block Erase Command (20_{16}/$D0_{16}$)

By writing the command code "20_{16}" in the first bus cycle and the confirmation command code "$D0_{16}$" in the second bus cycle that follows to the block address of a flash memory block, the system initiates an auto erase (erase and erase verify) operation.

Whether the auto erase operation is completed can be confirmed by reading the status register or the flash memory control register 0. At the same time the auto erase operation starts, the read status register mode is automatically entered, so the content of the status register can be read out. The status register bit 7 (SR7) is set to 0 at the same time the auto erase operation starts and is returned to 1 upon completion of the auto erase operation. In this case, the read status register mode remains active until the Read Array command (FF_{16}) or Read Lock Bit Status command (71_{16}) is written or the flash memory is reset using its reset bit.

The RY/$\overline{\text{BY}}$ status flag of the flash memory control register 0 is 0 during auto erase operation and 1 when the auto erase operation is completed as is the status register bit 7.

After the auto erase operation is completed, the status register can be read out to know the result of the auto erase operation. For details, refer to the section where the status register is detailed.

Figure 1.30.5 shows an example of a block erase flowchart.

Each block of the flash memory can be protected against erasure by using a lock bit. For details, refer to the section where the data protect function is detailed.

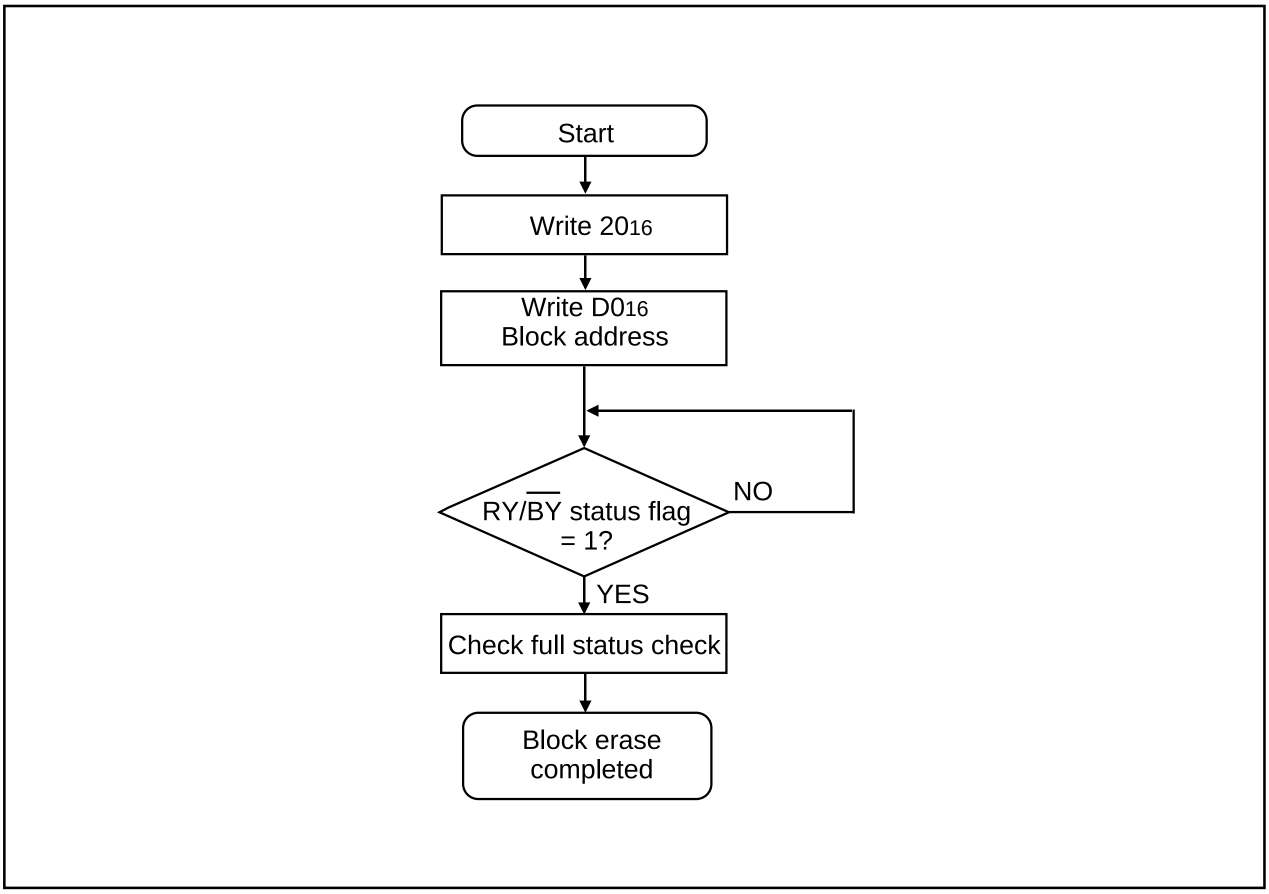


Figure 1.30.5. Block erase flowchart

Erase All Unlock Blocks Command ($A7_{16}$/$D0_{16}$)

By writing the command code “$A7_{16}$” in the first bus cycle and the confirmation command code “$D0_{16}$” in the second bus cycle that follows, the system starts erasing blocks successively.

Whether the erase all unlock blocks command is terminated can be confirmed by reading the status register or the flash memory control register 0, in the same way as for block erase. Also, the status register can be read out to know the result of the auto erase operation.

When the lock bit disable bit of the flash memory control register 0 = 1, all blocks are erased no matter how the lock bit is set. On the other hand, when the lock bit disable bit = 0, the function of the lock bit is effective and only nonlocked blocks (where lock bit data = 1) are erased.

Lock Bit Program Command (77_{16}/$D0_{16}$)

By writing the command code “77_{16}” in the first bus cycle and the confirmation command code “$D0_{16}$” in the second bus cycle that follows to the block address of a flash memory block, the system sets the lock bit for the specified block to 0 (locked).

Figure 1.30.6 shows an example of a lock bit program flowchart. The status of the lock bit (lock bit data) can be read out by a read lock bit status command.

Whether the lock bit program command is terminated can be confirmed by reading the status register or the flash memory control register 0, in the same way as for page program.

For details about the function of the lock bit and how to reset the lock bit, refer to the section where the data protect function is detailed.

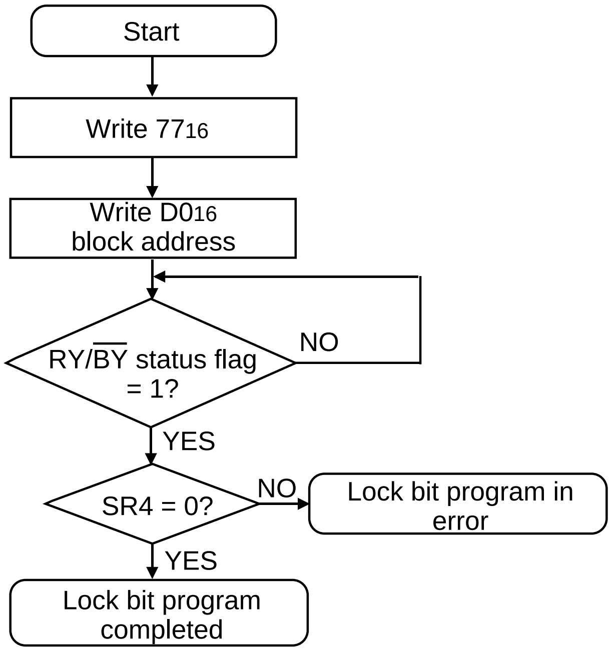

Start
Write 77_{16}
Write $D0_{16}$
block address
$RY/\overline{BY}$ status flag
= 1?
NO
YES
SR4 = 0?
NO
Lock bit program in error
YES
Lock bit program completed

Figure 1.30.6. Lock bit program flowchart

Read Lock Bit Status Command (71_{16})

By writing the command code "71_{16}" in the first bus cycle and then the block address of a flash memory block in the second bus cycle that follows, the system reads out the status of the lock bit of the specified block on to the data (D6).

Figure 1.30.7 shows an example of a read lock bit program flowchart.

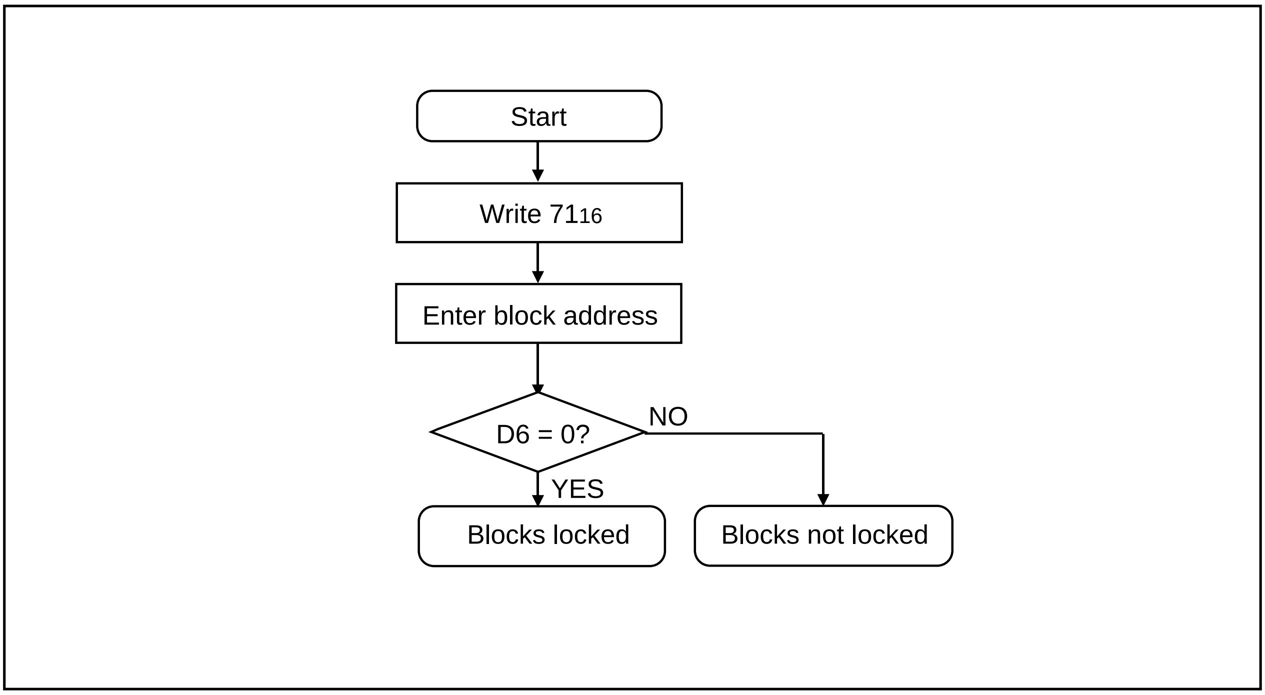


Figure 1.30.7. Read lock bit status flowchart

Data Protect Function (Block Lock)

Each block in Figure 1.29.3 has a nonvolatile lock bit to specify that the block be protected (locked) against erase/write. The lock bit program command is used to set the lock bit to 0 (locked). The lock bit of each block can be read out using the read lock bit status command.

Whether block lock is enabled or disabled is determined by the status of the lock bit and how the flash memory control register 0's lock bit disable bit is set.

(1) When the lock bit disable bit = 0, a specified block can be locked or unlocked by the lock bit status (lock bit data). Blocks whose lock bit data = 0 are locked, so they are disabled against erase/write. On the other hand, the blocks whose lock bit data = 1 are not locked, so they are enabled for erase/ write.

(2) When the lock bit disable bit = 1, all blocks are nonlocked regardless of the lock bit data, so they are enabled for erase/write. In this case, the lock bit data that is 0 (locked) is set to 1 (nonlocked) after erasure, so that the lock bit-actuated lock is removed.

Status Register

The status register indicates the operating status of the flash memory and whether an erase or program operation has terminated normally or in an error. The content of this register can be read out by only writing the read status register command (70_{16}). Table 1.30.2 details the status register.

The status register is cleared by writing the Clear Status Register command (50_{16}).

After a reset, the status register is set to "80_{16}."

Each bit in this register is explained below.

Write state machine (WSM) status (SR7)

After power-on, the write state machine (WSM) status is set to 1.

The write state machine (WSM) status indicates the operating status of the device, as for output on the RY/$\overline{\text{BY}}$ pin. This status bit is set to 0 during auto write or auto erase operation and is set to 1 upon completion of these operations.

Erase status (SR5)

The erase status informs the operating status of auto erase operation to the CPU. When an erase error occurs, it is set to 1.

The erase status is reset to 0 when cleared.

Program status (SR4)

The program status informs the operating status of auto write operation to the CPU. When a write error occurs, it is set to 1.

The program status is reset to 0 when cleared.

When an erase command is in error (which occurs if the command entered after the block erase command (20_{16}) is not the confirmation command ($D0_{16}$), both the program status and erase status (SR5) are set to 1.

When the program status or erase status = 1, the following commands entered by command write are not accepted.

Also, in one of the following cases, both SR4 and SR5 are set to 1 (command sequence error):

(1) When the valid command is not entered correctly

(2) When the data entered in the second bus cycle of lock bit program (77_{16}/$D0_{16}$), block erase (20_{16}/$D0_{16}$), or erase all unlock blocks ($A7_{16}$/$D0_{16}$) is not the $D0_{16}$ or FF_{16}. However, if FF_{16} is entered, read array is assumed and the command that has been set up in the first bus cycle is canceled.

Block status after program (SR3)

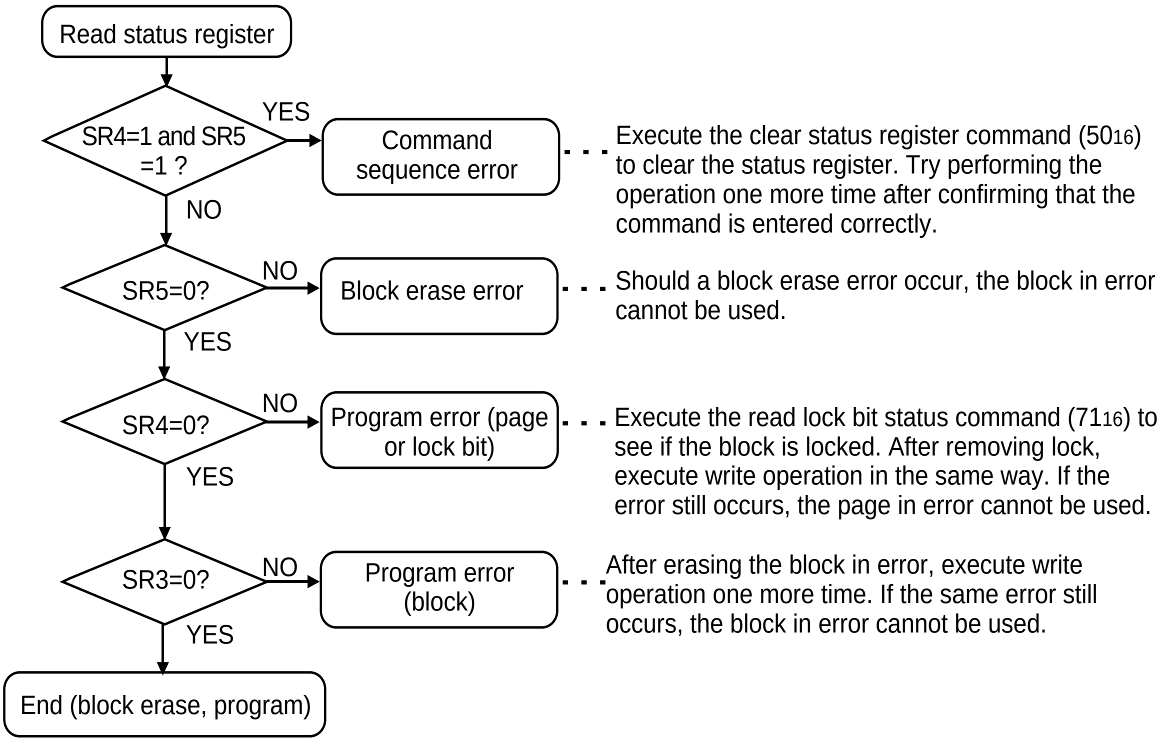
If excessive data is written (phenomenon whereby the memory cell becomes depressed which results in data not being read correctly), “1” is set for the program status after-program at the end of the page write operation. In other words, when writing ends successfully, “80_{16}” is output; when writing fails, “90_{16}” is output; and when excessive data is written, “88_{16}” is output.

Table 1.30.2. Definition of each bit in status register

Each bit of SRD	Status name	Definition	
		"1"	"0"
SR7 (bit7)	Write state machine (WSM) status	Ready	Busy
SR6 (bit6)	Reserved	-	-
SR5 (bit5)	Erase status	Terminated in error	Terminated normally
SR4 (bit4)	Program status	Terminated in error	Terminated normally
SR3 (bit3)	Block status after program	Terminated in error	Terminated normally
SR2 (bit2)	Reserved	-	-
SR1 (bit1)	Reserved	-	-
SR0 (bit0)	Reserved	-	-

Full Status Check

By performing full status check, it is possible to know the execution results of erase and program operations. Figure 1.30.8 shows a full status check flowchart and the action to be taken when each error occurs.



Command sequence error . . . Execute the clear status register command (50_{16}) to clear the status register. Try performing the operation one more time after confirming that the command is entered correctly.

Block erase error . . . Should a block erase error occur, the block in error cannot be used.

Program error (page or lock bit) . . . Execute the read lock bit status command (71_{16}) to see if the block is locked. After removing lock, execute write operation in the same way. If the error still occurs, the page in error cannot be used.

Program error (block) . . . After erasing the block in error, execute write operation one more time. If the same error still occurs, the block in error cannot be used.

Note: When one of SR5 to SR3 is set to 1, none of the page program, block erase, erase all unlock blocks and lock bit program commands is accepted. Execute the clear status register command (50_{16}) before executing these commands.

Figure 1.30.8. Full status check flowchart and remedial procedure for errors

Functions To Inhibit Rewriting Flash Memory Version

To prevent the contents of the flash memory version from being read out or rewritten easily, the device incorporates a ROM code protect function for use in parallel I/O mode and an ID code check function for use in standard serial I/O mode.

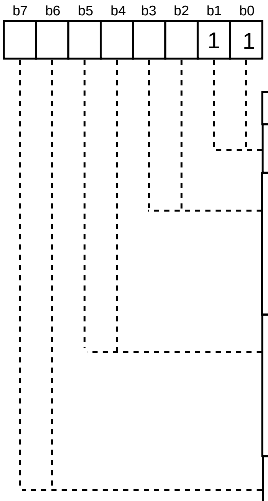
ROM code protect function

The ROM code protect function reading out or modifying the contents of the flash memory version by using the ROM code protect control address (0FFFFFF16) during parallel I/O mode. Figure 1.30.9 shows the ROM code protect control address (0FFFFFF16). (This address exists in the user ROM area.)

If one of the pair of ROM code protect bits is set to 0, ROM code protect is turned on, so that the contents of the flash memory version are protected against readout and modification. ROM code protect is implemented in two levels. If level 2 is selected, the flash memory is protected even against readout by a shipment inspection LSI tester, etc. When an attempt is made to select both level 1 and level 2, level 2 is selected by default.

If both of the two ROM code protect reset bits are set to “00,” ROM code protect is turned off, so that the contents of the flash memory version can be read out or modified. Once ROM code protect is turned on, the contents of the ROM code protect reset bits cannot be modified in parallel I/O mode. Use the serial I/O or some other mode to rewrite the contents of the ROM code protect reset bits.

ROM code protect control address



Symbol	Address	When reset
ROMCP	0FFFFFF16	FF16

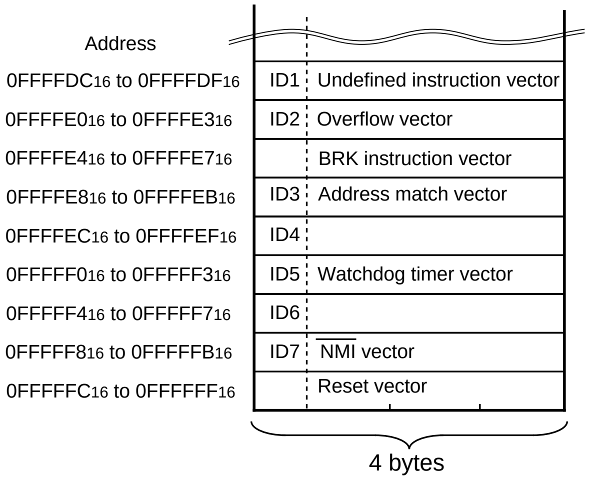
Bit symbol	Bit name	Function
Reserved bit		Always set this bit to 1.
ROMCP2	ROM code protect level 2 set bit (Note 1, 2)	b3 b2 00: Protect enabled 01: Protect enabled 10: Protect enabled 11: Protect disabled
ROMCR	ROM code protect reset bit (Note 3)	b5 b4 00: Protect removed 01: Protect set bit effective 10: Protect set bit effective 11: Protect set bit effective
ROMCP1	ROM code protect level 1 set bit (Note 1)	b7 b6 00: Protect enabled 01: Protect enabled 10: Protect enabled 11: Protect disabled

Note 1: When ROM code protect is turned on, the on-chip flash memory is protected against readout or modification in parallel input/output mode.
Note 2: When ROM code protect level 2 is turned on, ROM code readout by a shipment inspection LSI tester, etc. also is inhibited.
Note 3: The ROM code protect reset bits can be used to turn off ROM code protect level 1 and ROM code protect level 2. However, since these bits cannot be changed in parallel input/output mode, they need to be rewritten in serial input/output or some other mode.

Figure 1.30.9. ROM code protect control address

ID Code Check Function

Use this function in standard serial I/O mode. When the contents of the flash memory are not blank, the ID code sent from the peripheral unit is compared with the ID code written in the flash memory to see if they match. If the ID codes do not match, the commands sent from the peripheral unit are not accepted. The ID code consists of 8-bit data, the areas of which, beginning with the first byte, are $0FFFFDF_{16}$, $0FFFFE3_{16}$, $0FFFFEB_{16}$, $0FFFFEF_{16}$, $0FFFFF3_{16}$, $0FFFFF7_{16}$, and $0FFFFFB_{16}$. Write a program which has had the ID code preset at these addresses to the flash memory.



Address		
$0FFFFDC_{16}$ to $0FFFFDF_{16}$	ID1	Undefined instruction vector
$0FFFFE0_{16}$ to $0FFFFE3_{16}$	ID2	Overflow vector
$0FFFFE4_{16}$ to $0FFFFE7_{16}$		BRK instruction vector
$0FFFFE8_{16}$ to $0FFFFEB_{16}$	ID3	Address match vector
$0FFFFEC_{16}$ to $0FFFFEF_{16}$	ID4	
$0FFFFF0_{16}$ to $0FFFFF3_{16}$	ID5	Watchdog timer vector
$0FFFFF4_{16}$ to $0FFFFF7_{16}$	ID6	
$0FFFFF8_{16}$ to $0FFFFFB_{16}$	ID7	$\overline{\text{NMI}}$ vector
$0FFFFFC_{16}$ to $0FFFFFF_{16}$		Reset vector

Figure 1.30.10. ID code store addresses

Parallel I/O Mode

In this mode, the M16C/80 (flash memory version) operates in a manner similar to the flash memory M5M29FB/T800 from Mitsubishi. Since there are some differences with regard to the functions not available with the microcomputer and matters related to memory capacity, the M16C/80 cannot be programed by a programer for the flash memory.

Use an exclusive programer supporting M16C/80 (flash memory version).

Refer to the instruction manual of each programer maker for the details of use.

User ROM and Boot ROM Areas

In parallel I/O mode, the user ROM and boot ROM areas shown in Figure 1.29.3 can be rewritten. Both areas of flash memory can be operated on in the same way.

Program and block erase operations can be performed in the user ROM area. The user ROM area and its blocks are shown in Figure 1.29.3.

The boot ROM area is 8 Kbytes in size. In parallel I/O mode, it is located at addresses $0FFE000_{16}$ through $0FFFFFF_{16}$. Make sure program and block erase operations are always performed within this address range. (Access to any location outside this address range is prohibited.)

In the boot ROM area, an erase block operation is applied to only one 8 Kbyte block. The boot ROM area has had a standard serial I/O mode control program stored in it when shipped from the Mitsubishi factory. Therefore, using the device in standard serial input/output mode, you do not need to write to the boot ROM area.

Pin functions (Flash memory standard serial I/O mode)

Pin	Name	I/O	Description
V_{CC},V_{SS}	Power input		Apply 4.2V to 5.5V to Vcc pin and 0 V to Vss pin.
CNV_{SS}	CNV_{SS}	I	Connect to Vcc pin.
$\overline{\text{RESET}}$	Reset input	I	Reset input pin. While reset is "L" level, a 20 cycle or longer clock must be input to XIN pin.
X_{IN}	Clock input	I	Connect a ceramic resonator or crystal oscillator between X_{IN} and X_{OUT} pins. To input an externally generated clock, input it to X_{IN} pin and open X_{OUT} pin.
X_{OUT}	Clock output	O	
BYTE	BYTE	I	Connect this pin to Vcc or Vss.
AV_{CC}, AV_{SS}	Analog power supply input	I	Connect AVSS to Vss and AVcc to Vcc, respectively.
V_{REF}	Reference voltage input	I	Enter the reference voltage for A-D converter from this pin.
$P0_0$ to $P0_7$	Input port P0	I	Input "H" or "L" level signal or open.
$P1_0$ to $P1_7$	Input port P1	I	Input "H" or "L" level signal or open.
$P2_0$ to $P2_7$	Input port P2	I	Input "H" or "L" level signal or open.
$P3_0$ to $P3_7$	Input port P3	I	Input "H" or "L" level signal or open.
$P4_0$ to $P4_7$	Input port P4	I	Input "H" or "L" level signal or open.
$P5_1$ to $P5_4$, $P5_6$, $P5_7$	Input port P5	I	Input "H" or "L" level signal or open.
$P5_0$	$\overline{\text{CE}}$ input	I	Input "H" level signal.
$P5_5$	$\overline{\text{EPM}}$ input	I	Input "L" level signal.
$P6_0$ to $P6_3$	Input port P6	I	Input "H" or "L" level signal or open.
$P6_4$	BUSY output	O	Standard serial mode 1: BUSY signal output pin Standard serial mode 2: Monitors the program operation check
$P6_5$	SCLK input	I	Standard serial mode 1: Serial clock input pin Standard serial mode 2: Input "L" level signal.
$P6_6$	RxD input	I	Serial data input pin
$P6_7$	TxD output	O	Serial data output pin
$P7_0$ to $P7_7$	Input port P7	I	Input "H" or "L" level signal or open.
$P8_0$ to $P8_4$, $P8_6$, $P8_7$	Input port P8	I	Input "H" or "L" level signal or open.
$P8_5$	$\overline{\text{NMI}}$ input	I	Connect this pin to Vcc.
$P9_0$ to $P9_7$	Input port P9	I	Input "H" or "L" level signal or open.
$P10_0$ to $P10_7$	Input port P10	I	Input "H" or "L" level signal or open.
$P11_0$ to $P11_4$	Input port P11	I	Input "H" or "L" level signal or open. (Note)
$P12_0$ to $P12_7$	Input port P12	I	Input "H" or "L" level signal or open. (Note)
$P13_0$ to $P13_7$	Input port P13	I	Input "H" or "L" level signal or open. (Note)
$P14_0$ to $P14_6$	Input port P14	I	Input "H" or "L" level signal or open. (Note)
$P15_0$ to $P15_7$	Input port P15	I	Input "H" or "L" level signal or open. (Note)

Note: Port P11 to P15 exist in 144-pin version.

Mode setting

Signal	Value
CNVss	Vcc
EPM	Vss
RESET	Vss >> Vcc
CE	Vcc

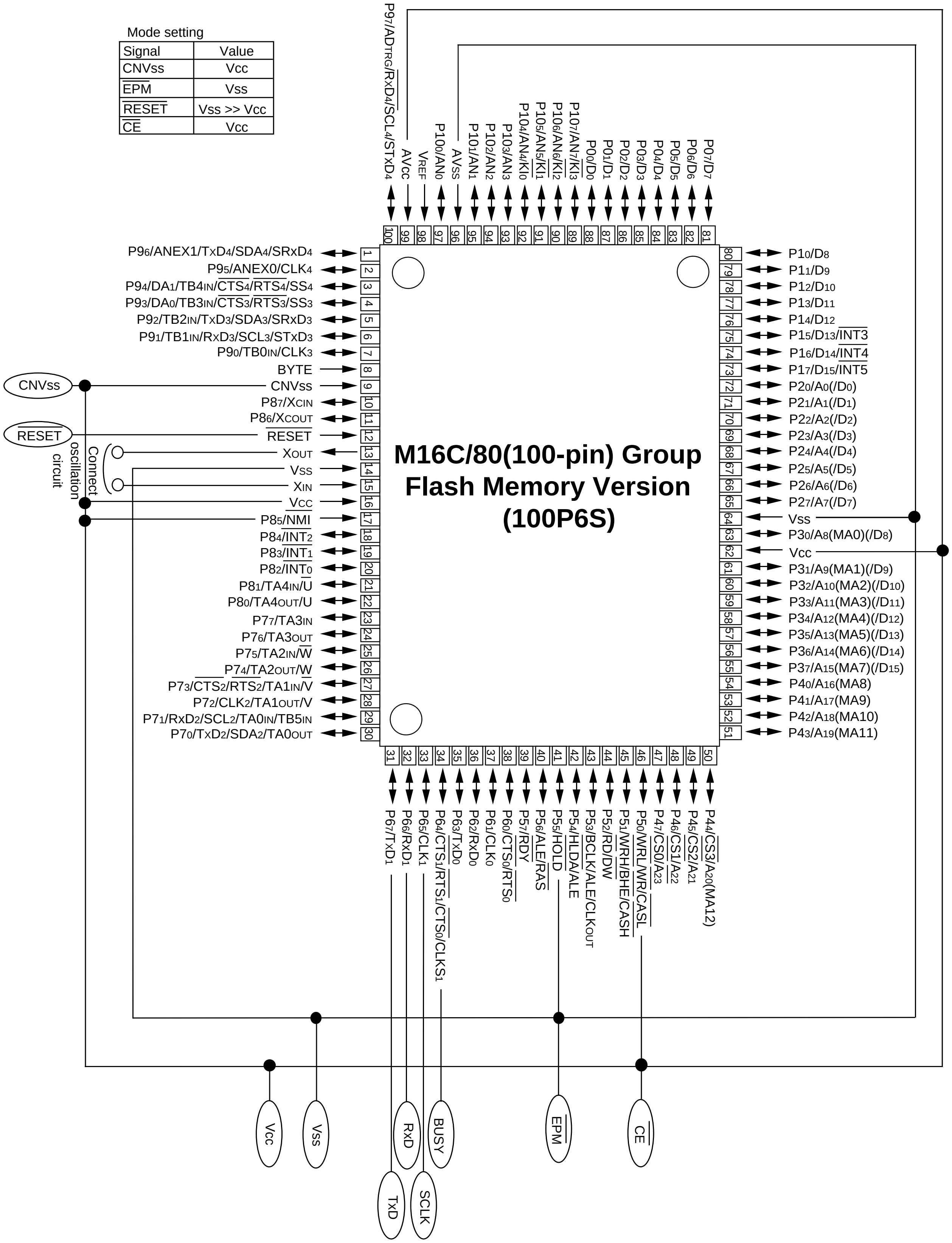
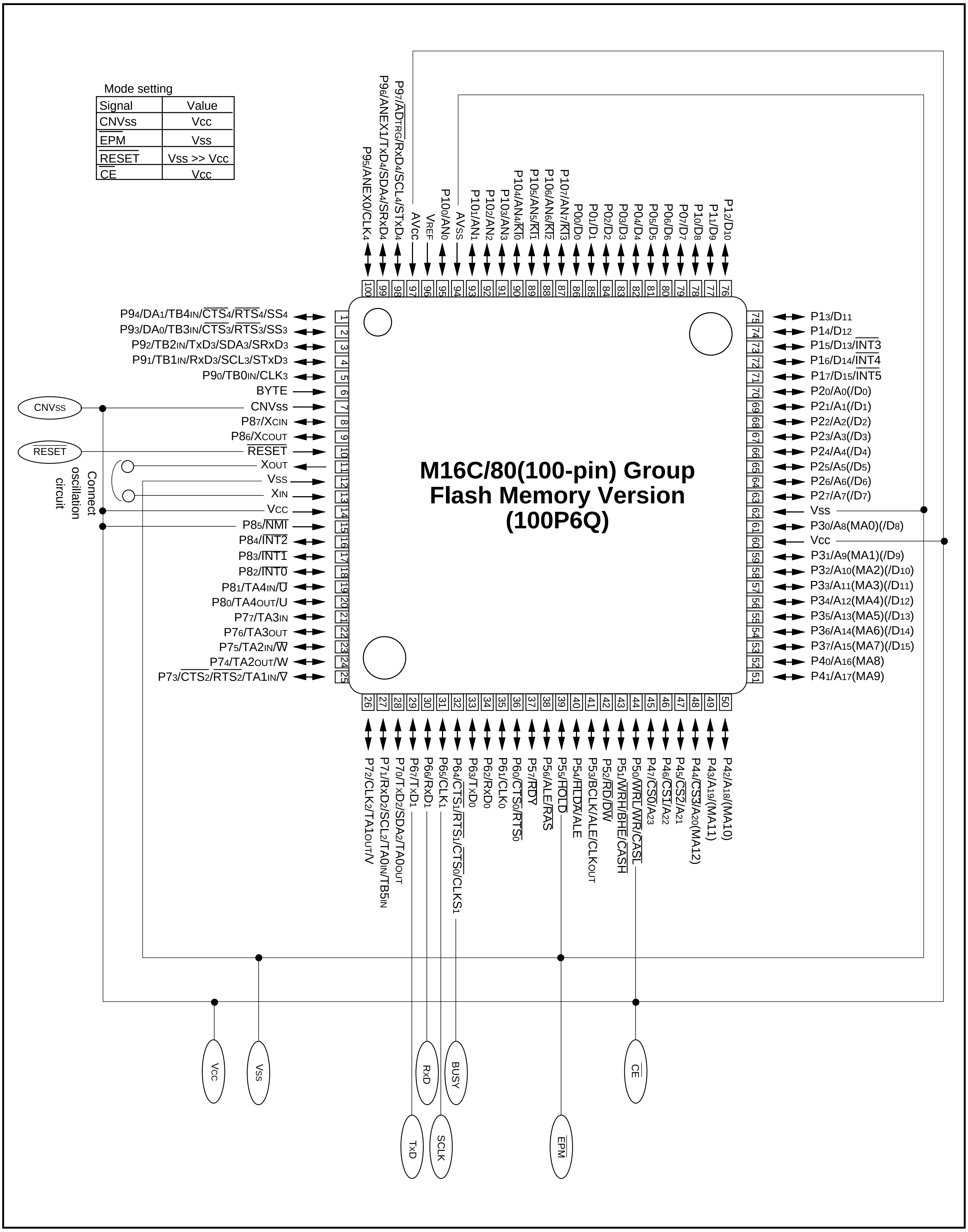


Figure 1.31.1. Pin connections for standard serial I/O mode (1)



Mode setting

Signal	Value
CNVss	Vcc
EPM	Vss
RESET	Vss >> Vcc
CE	Vcc

Figure 1.31.2. Pin connections for standard serial I/O mode (2)

Mode setting

Signal	Value
CNVss	Vcc
EPM	Vss
RESET	Vss >> Vcc
CE	Vcc

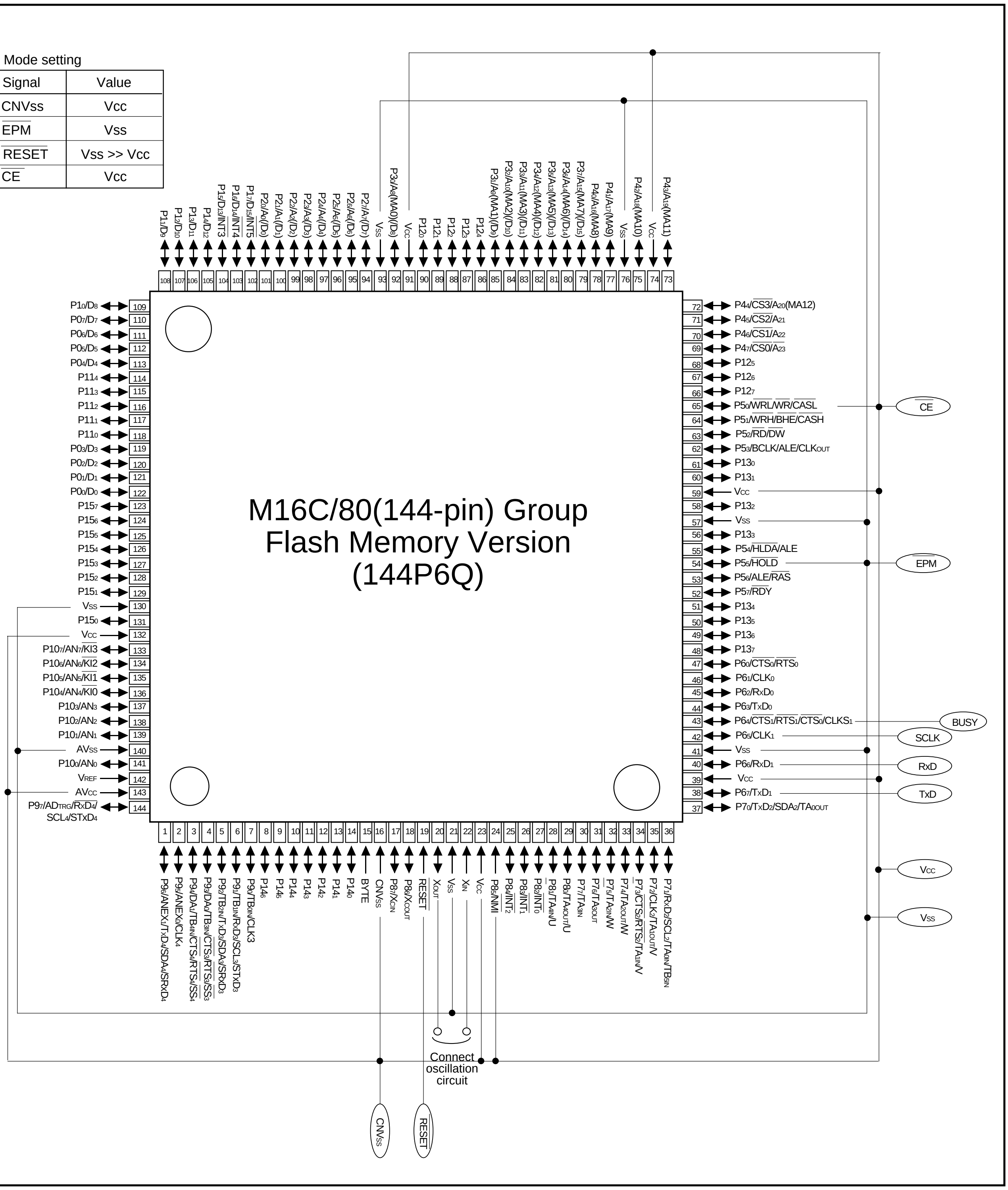


Figure 1.31.3. Pin connections for standard serial I/O mode (3)

Standard serial I/O mode

The standard serial I/O mode inputs and outputs the software commands, addresses and data needed to operate (read, program, erase, etc.) the internal flash memory. This I/O is serial. There are actually two standard serial I/O modes: mode 1, which is clock synchronized, and mode 2, which is asynchronized. Both modes require a purpose-specific peripheral unit.

The standard serial I/O mode is different from the parallel I/O mode in that the CPU controls flash memory rewrite (uses the CPU's rewrite mode), rewrite data input and so forth. It is started when the reset is released, which is done when the $P5_0$ ($\overline{CE}$) pin is "H" level, the $P5_5$ ($\overline{EPM}$) pin "L" level and the CNVss pin "H" level. (In the ordinary command mode, set CNVss pin to "L" level.)
This control program is written in the boot ROM area when the product is shipped from Mitsubishi. Accordingly, make note of the fact that the standard serial I/O mode cannot be used if the boot ROM area is rewritten in the parallel I/O mode. Figures 1.31.1 and 1.31.3 show the pin connections for the standard serial I/O mode. Serial data I/O uses UART1 and transfers the data serially in 8-bit units. Standard serial I/O switches between mode 1 (clock synchronized) and mode 2 (clock asynchronized) according to the level of CLK_1 pin when the reset is released.

To use standard serial I/O mode 1 (clock synchronized), set the CLK_1 pin to "H" level and release the reset. The operation uses the four UART1 pins CLK_1, RxD_1, TxD_1 and RTS_1 (BUSY). The CLK_1 pin is the transfer clock input pin through which an external transfer clock is input. The TxD_1 pin is for CMOS output. The RTS_1 (BUSY) pin outputs an "L" level when ready for reception and an "H" level when reception starts.

To use standard serial I/O mode 2 (clock asynchronized), set the CLK_1 pin to "L" level and release the reset. The operation uses the two UART1 pins RxD_1 and TxD_1.

In the standard serial I/O mode, only the user ROM area indicated in Figure 1.31.20 can be rewritten. The boot ROM cannot.
In the standard serial I/O mode, a 7-byte ID code is used. When there is data in the flash memory, commands sent from the peripheral unit (programmer) are not accepted unless the ID code matches.

Overview of standard serial I/O mode 1 (clock synchronized)

In standard serial I/O mode 1, software commands, addresses and data are input and output between the MCU and peripheral units (serial programer, etc.) using 4-wire clock-synchronized serial I/O (UART1). Standard serial I/O mode 1 is engaged by releasing the reset with the $P6_5$ (CLK_1) pin "H" level.

In reception, software commands, addresses and program data are synchronized with the rise of the transfer clock that is input to the CLK_1 pin, and are then input to the MCU via the RxD_1 pin. In transmission, the read data and status are synchronized with the fall of the transfer clock, and output from the TxD_1 pin.

The TxD_1 pin is for CMOS output. Transfer is in 8-bit units with LSB first.

When busy, such as during transmission, reception, erasing or program execution, the RTS_1 (BUSY) pin is "H" level. Accordingly, always start the next transfer after the RST_1 (BUSY) pin is "L" level.

Also, data and status registers in memory can be read after inputting software commands. Status, such as the operating state of the flash memory or whether a program or erase operation ended successfully or not, can be checked by reading the status register. Here following are explained software commands, status registers, etc.

Software Commands

Table 1.31.1 lists software commands. In the standard serial I/O mode 1, erase operations, programs and reading are controlled by transferring software commands via the RxD$_1$ pin. Software commands are explained here below.

Table 1.31.1. Software commands (Standard serial I/O mode 1)

	Control command	1st byte transfer	2nd byte	3rd byte	4th byte	5th byte	6th byte		When ID is not verified
1	Page read	FF$_{16}$	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
2	Page program	41$_{16}$	Address (middle)	Address (high)	Data input	Data input	Data input	Data input to 259th byte	Not acceptable
3	Block erase	20$_{16}$	Address (middle)	Address (high)	D0$_{16}$				Not acceptable
4	Erase all unlocked blocks	A7$_{16}$	D0$_{16}$						Not acceptable
5	Read status register	70$_{16}$	SRD output	SRD1 output					Acceptable
6	Clear status register	50$_{16}$							Not acceptable
7	Read lock bit status	71$_{16}$	Address (middle)	Address (high)	Lock bit data output				Not acceptable
8	Lock bit program	77$_{16}$	Address (middle)	Address (high)	D0$_{16}$				Not acceptable
9	Lock bit enable	7A$_{16}$							Not acceptable
10	Lock bit disable	75$_{16}$							Not acceptable
11	Code processing function	F5$_{16}$	Address (low)	Address (middle)	Address (high)	ID size	ID1	To ID7	Acceptable
12	Download function	FA$_{16}$	Size (low)	Size (high)	Check-sum	Data input	To required number of times		Not acceptable
13	Version data output function	FB$_{16}$	Version data output	Version data output	Version data output	Version data output	Version data output	Version data output to 9th byte	Acceptable
14	Boot ROM area output function	FC$_{16}$	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
15	Read check data	FD$_{16}$	Check data (low)	Check data (high)					Not acceptable

Note 1: Shading indicates transfer from flash memory microcomputer to peripheral unit. All other data is transferred from the peripheral unit to the flash memory microcomputer.

Note 2: SRD refers to status register data. SRD1 refers to status register data1 .

Note 3: All commands can be accepted when the flash memory is totally blank.

Page Read Command

This command reads the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page read command as explained here following.

(1) Transfer the "FF16" command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, data (D_0–D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} will be output sequentially from the smallest address first in sync with the rise of the clock.

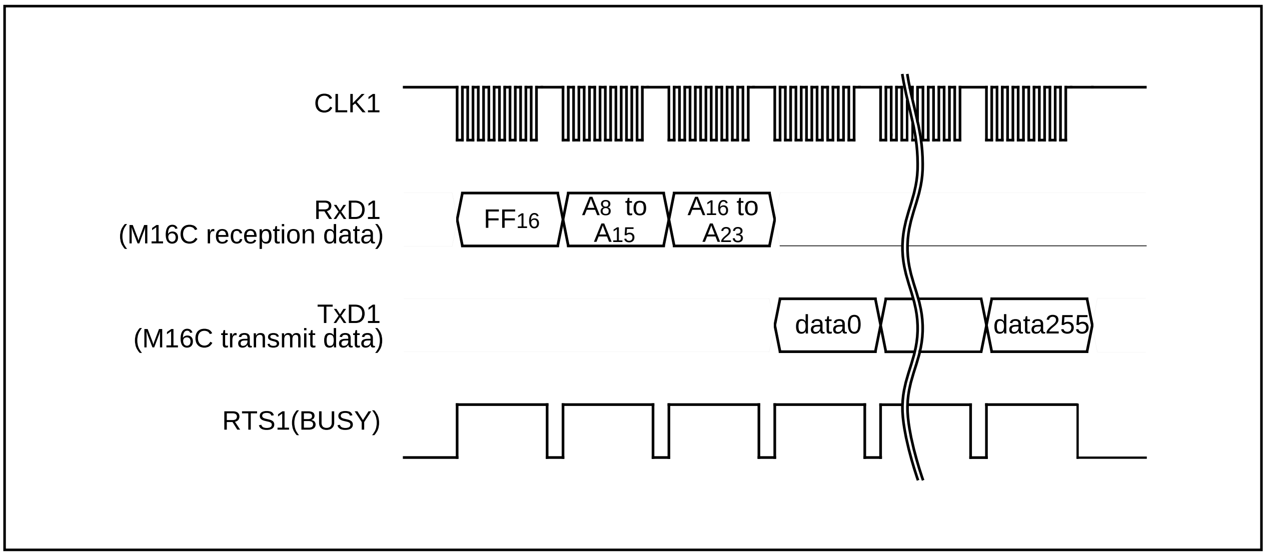


Figure 1.31.4. Timing for page read

Read Status Register Command

This command reads status information. When the "7016" command code is sent with the 1st byte, the contents of the status register (SRD) specified with the 2nd byte and the contents of status register 1 (SRD1) specified with the 3rd byte are read.

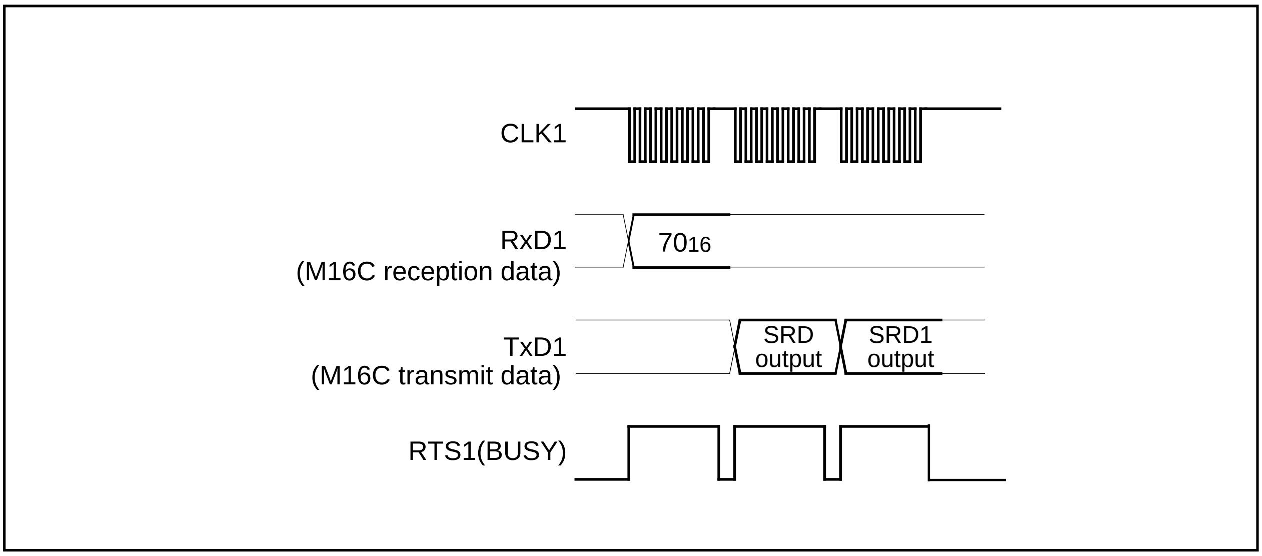


Figure 1.31.5. Timing for reading the status register

Clear Status Register Command

This command clears the bits (SR3–SR5) which are set when the status register operation ends in error. When the "50_{16}" command code is sent with the 1st byte, the aforementioned bits are cleared. When the clear status register operation ends, the RTS_1 (BUSY) signal changes from the "H" to the "L" level.

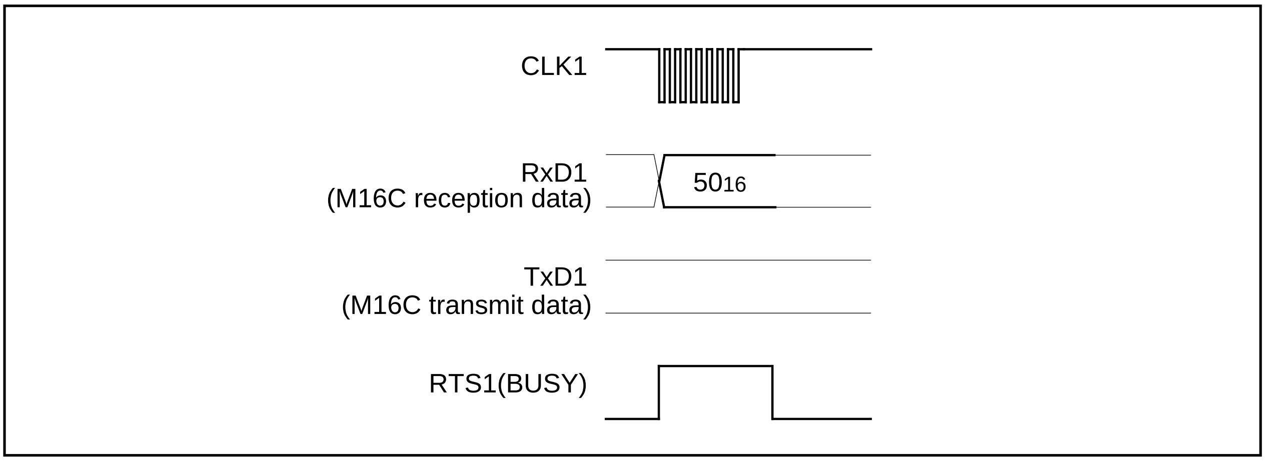


Figure 1.31.6. Timing for clearing the status register

Page Program Command

This command writes the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page program command as explained here following.

(1) Transfer the "41_{16}" command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, as write data (D_0–D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} is input sequentially from the smallest address first, that page is automatically written.

When reception setup for the next 256 bytes ends, the RTS_1 (BUSY) signal changes from the "H" to the "L" level. The result of the page program can be known by reading the status register. For more information, see the section on the status register.

Each block can be write-protected with the lock bit. For more information, see the section on the data protection function. Additional writing is not allowed with already programmed pages.

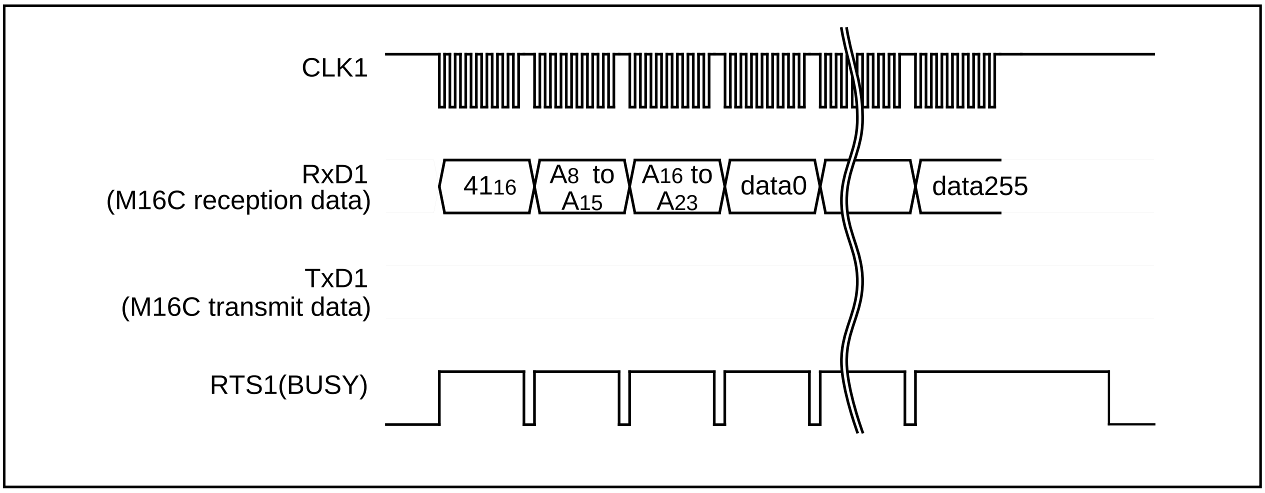


Figure 1.31.7. Timing for the page program

Block Erase Command

This command erases the data in the specified block. Execute the block erase command as explained here following.

(1) Transfer the “20_{16}” command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) Transfer the verify command code “$D0_{16}$” with the 4th byte. With the verify command code, the erase operation will start for the specified block in the flash memory. Write the highest address of the specified block for addresses A_{16} to A_{23}.

When block erasing ends, the RTS_1 (BUSY) signal changes from the “H” to the “L” level. After block erase ends, the result of the block erase operation can be known by reading the status register. For more information, see the section on the status register.

Each block can be erase-protected with the lock bit. For more information, see the section on the data protection function.

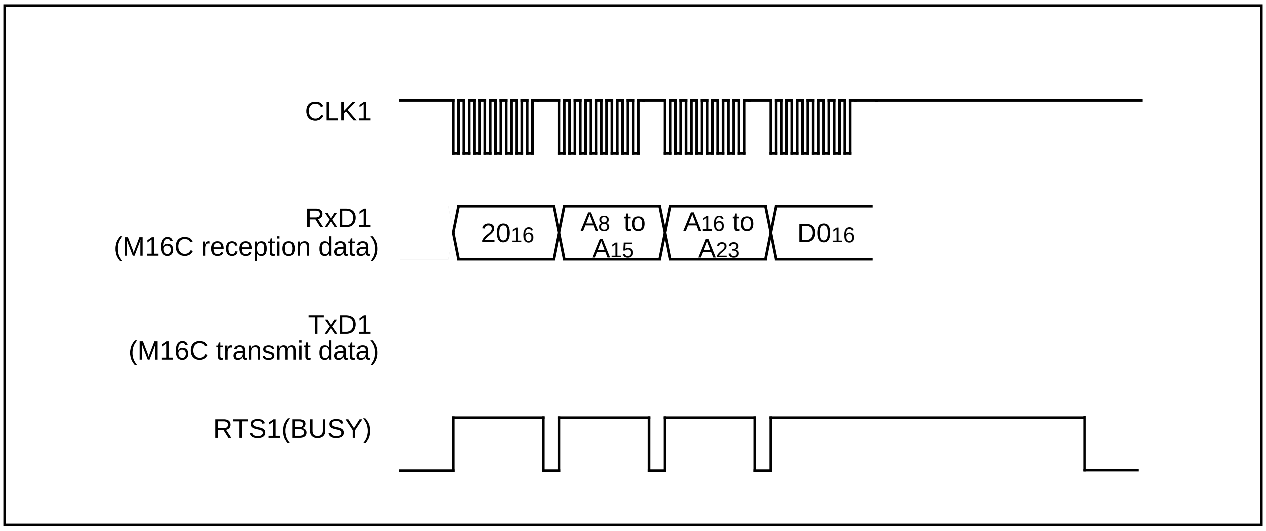


Figure 1.31.8. Timing for block erasing

Erase All Unlocked Blocks Command

This command erases the content of all blocks. Execute the erase all unlocked blocks command as explained here following.

(1) Transfer the "A7$_{16}$" command code with the 1st byte.

(2) Transfer the verify command code "D0$_{16}$" with the 2nd byte. With the verify command code, the erase operation will start and continue for all blocks in the flash memory.

When block erasing ends, the RTS$_1$ (BUSY) signal changes from the "H" to the "L" level. The result of the erase operation can be known by reading the status register. Each block can be erase-protected with the lock bit. For more information, see the section on the data protection function.

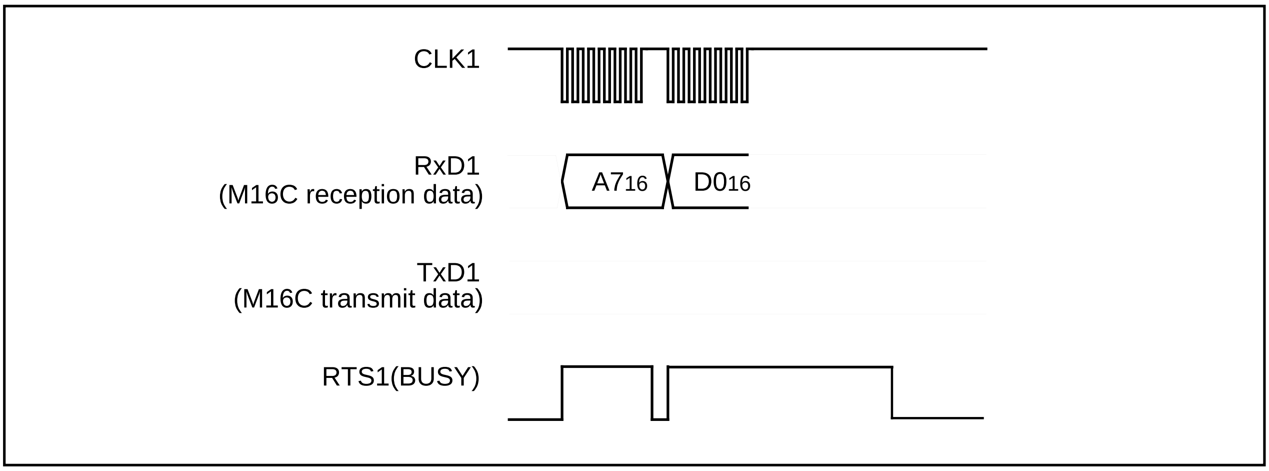


Figure 1.31.9. Timing for erasing all unlocked blocks

Lock Bit Program Command

This command writes "0" (lock) for the lock bit of the specified block. Execute the lock bit program command as explained here following.

(1) Transfer the "77$_{16}$" command code with the 1st byte.

(2) Transfer addresses A$_8$ to A$_{15}$ and A$_{16}$ to A$_{23}$ with the 2nd and 3rd bytes respectively.

(3) Transfer the verify command code "D0$_{16}$" with the 4th byte. With the verify command code, "0" is written for the lock bit of the specified block. Write the highest address of the specified block for addresses A$_8$ to A$_{23}$.

When writing ends, the RTS$_1$ (BUSY) signal changes from the "H" to the "L" level. Lock bit status can be read with the read lock bit status command. For information on the lock bit function, reset procedure and so on, see the section on the data protection function.

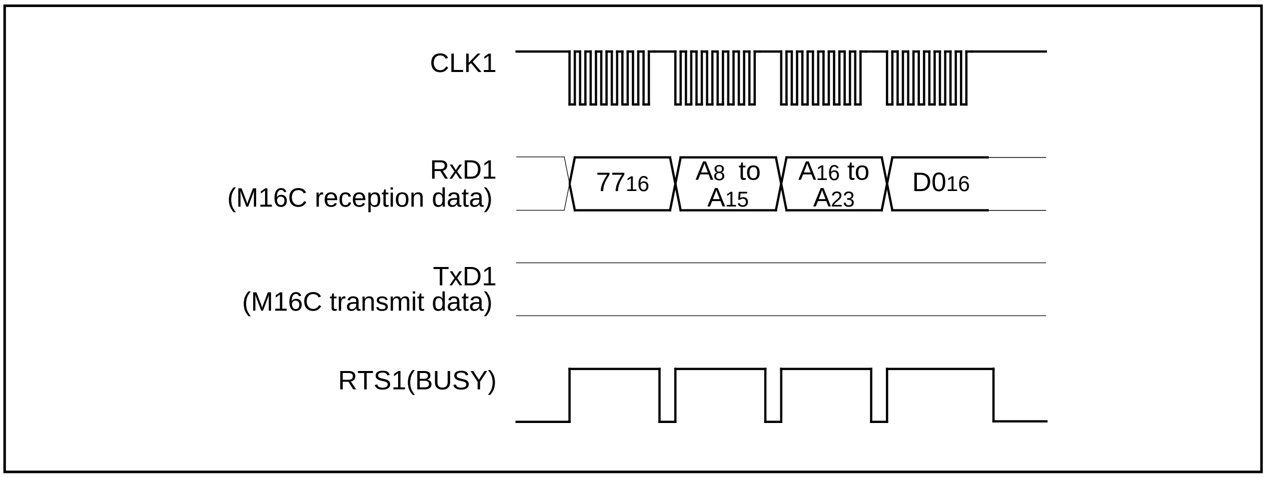


Figure 1.31.10. Timing for the lock bit program

Read Lock Bit Status Command

This command reads the lock bit status of the specified block. Execute the read lock bit status command as explained here following.

(1) Transfer the “71_{16}” command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) The lock bit data of the specified block is output with the 4th byte. The 6th bit (D6) of output data is the lock bit data. Write the highest address of the specified block for addresses A_8 to A_{23}.

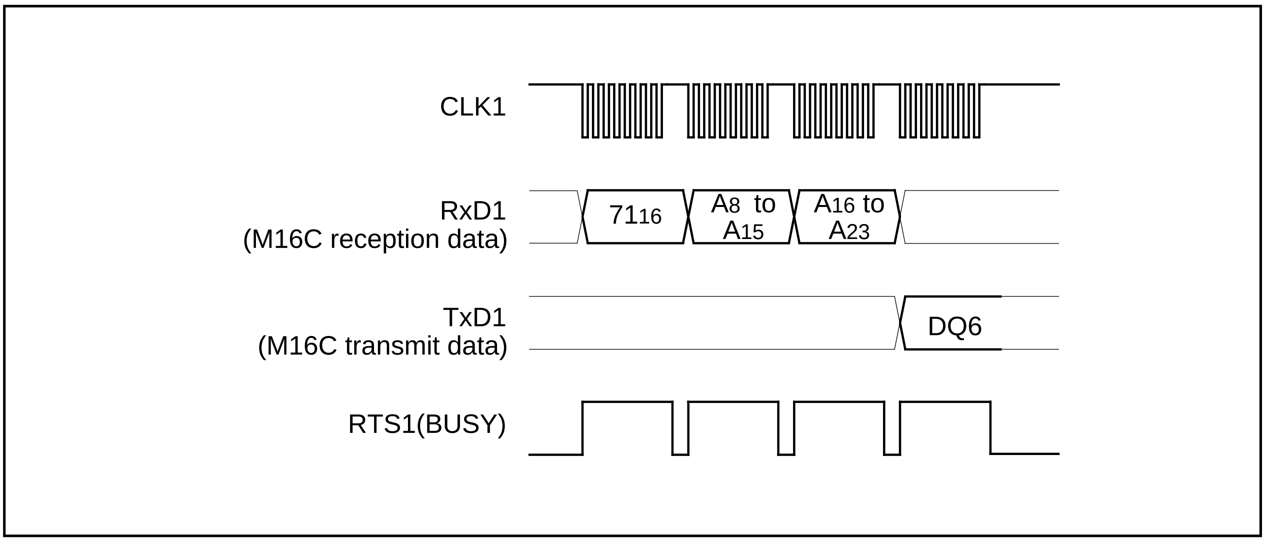


Figure 1.31.11. Timing for reading lock bit status

Lock Bit Enable Command

This command enables the lock bit in blocks whose bit was disabled with the lock bit disable command. The command code “$7A_{16}$” is sent with the 1st byte of the serial transmission. This command only enables the lock bit function; it does not set the lock bit itself.

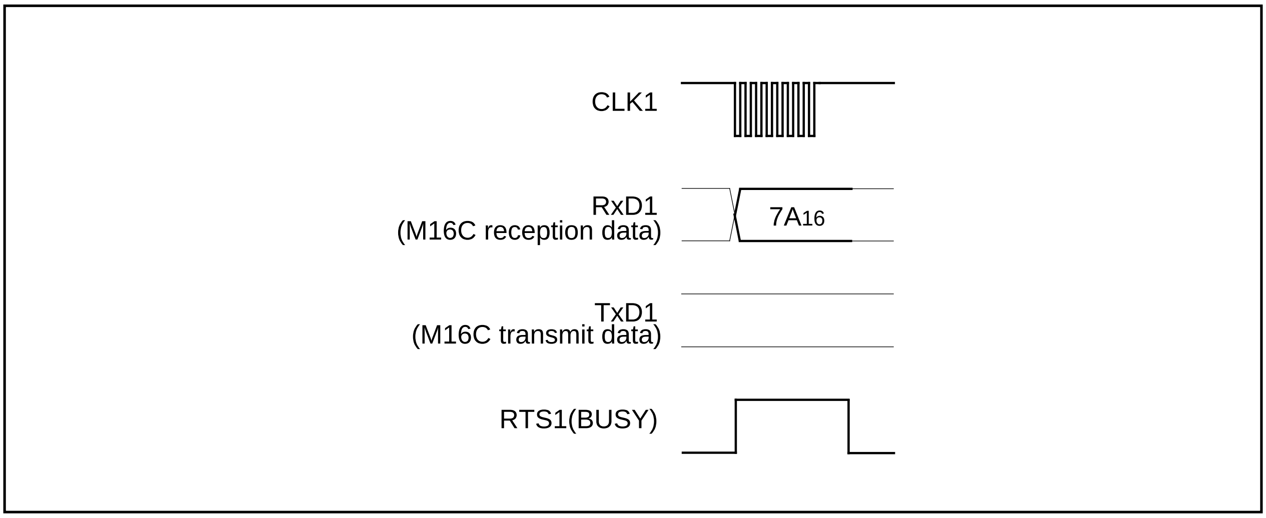


Figure 1.31.12. Timing for enabling the lock bit

Lock Bit Disable Command

This command disables the lock bit. The command code "75$_{16}$" is sent with the 1st byte of the serial transmission. This command only disables the lock bit function; it does not set the lock bit itself. However, if an erase command is executed after executing the lock bit disable command, "0" (locked) lock bit data is set to "1" (unlocked) after the erase operation ends. In any case, after the reset is cancelled, the lock bit is enabled.

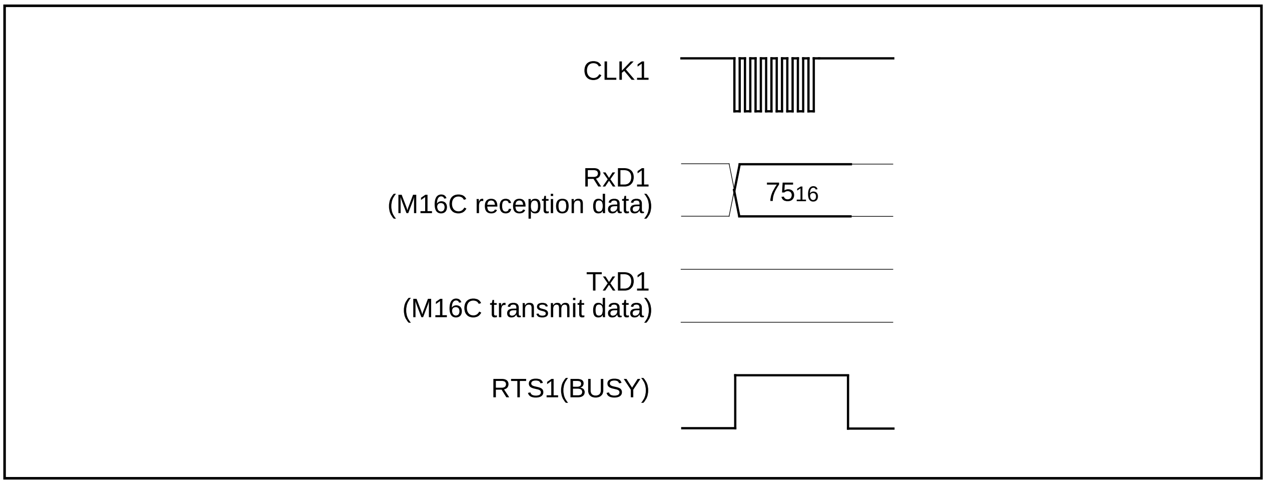


Figure 1.31.13. Timing for disabling the lock bit

Download Command

This command downloads a program to the RAM for execution. Execute the download command as explained here following.

(1) Transfer the "FA$_{16}$" command code with the 1st byte.

(2) Transfer the program size with the 2nd and 3rd bytes.

(3) Transfer the check sum with the 4th byte. The check sum is added to all data sent with the 5th byte onward.

(4) The program to execute is sent with the 5th byte onward.

When all data has been transmitted, if the check sum matches, the downloaded program is executed. The size of the program will vary according to the internal RAM.

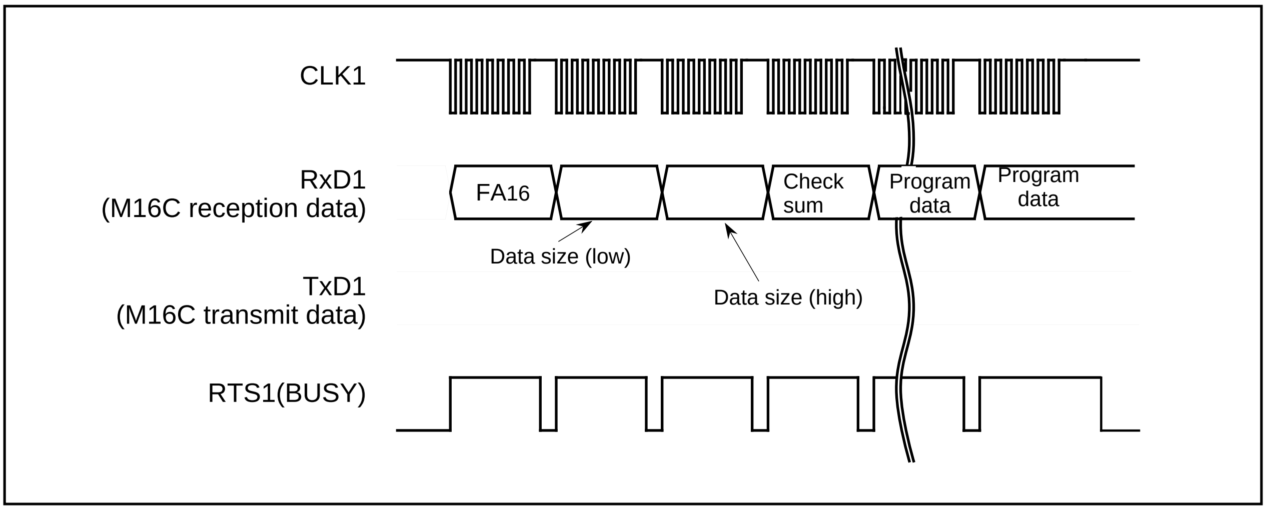


Figure 1.31.14. Timing for download

Version Information Output Command

This command outputs the version information of the control program stored in the boot area. Execute the version information output command as explained here following.

(1) Transfer the “FB16” command code with the 1st byte.

(2) The version information will be output from the 2nd byte onward. This data is composed of 8 ASCII code characters.

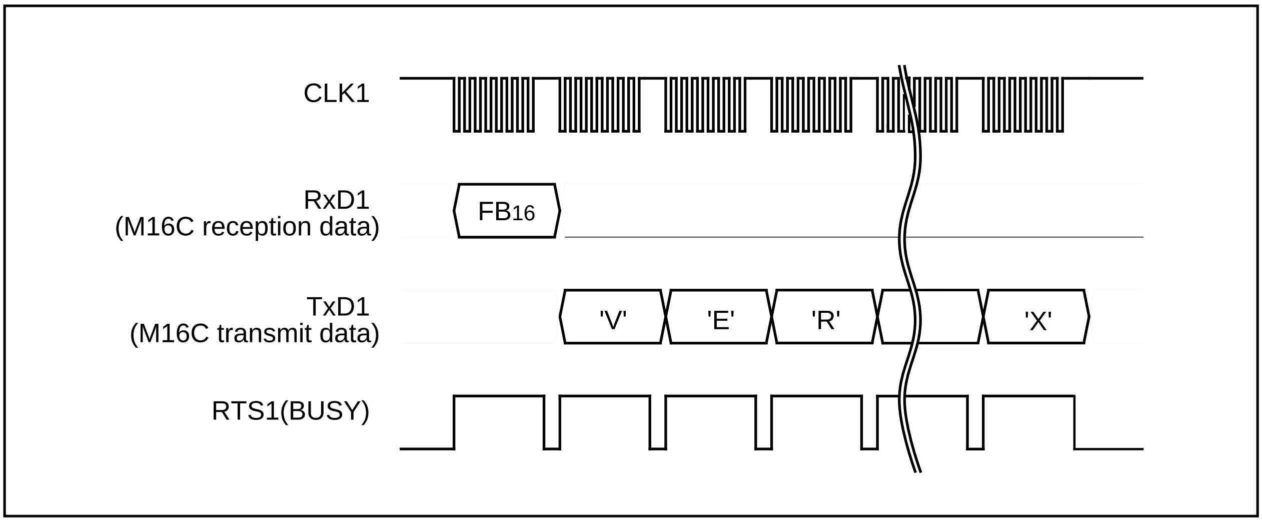


Figure 1.31.15. Timing for version information output

Boot ROM Area Output Command

This command outputs the control program stored in the boot ROM area in one page blocks (256 bytes). Execute the boot ROM area output command as explained here following.

(1) Transfer the “FC16” command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, data (D_0–D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} will be output sequentially from the smallest address first, in sync with the rise of the clock.

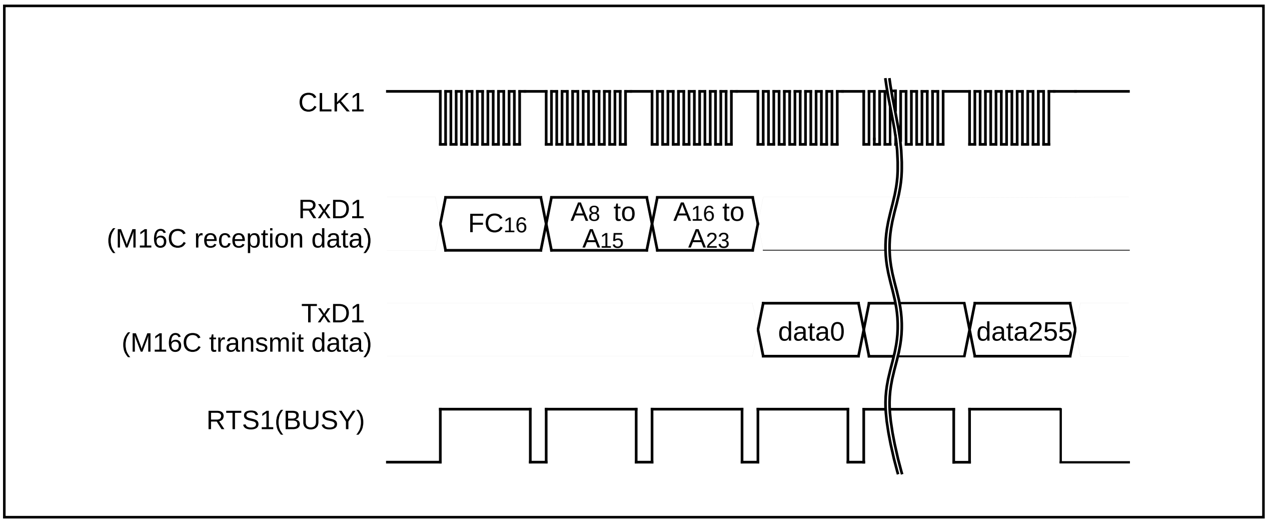


Figure 1.31.16. Timing for boot ROM area output

ID Check

This command checks the ID code. Execute the boot ID check command as explained here following.

(1) Transfer the “F5$_{16}$” command code with the 1st byte.

(2) Transfer addresses A$_0$ to A$_7$, A$_8$ to A$_{15}$ and A$_{16}$ to A$_{23}$ of the 1st byte of the ID code with the 2nd, 3rd and 4th bytes respectively.

(3) Transfer the number of data sets of the ID code with the 5th byte.

(4) The ID code is sent with the 6th byte onward, starting with the 1st byte of the code.

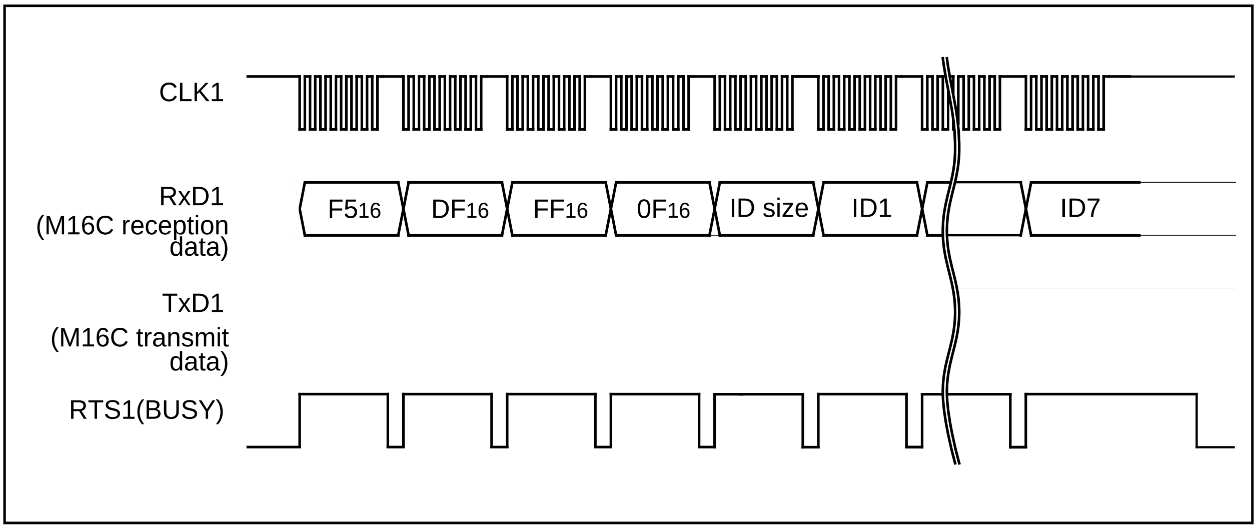
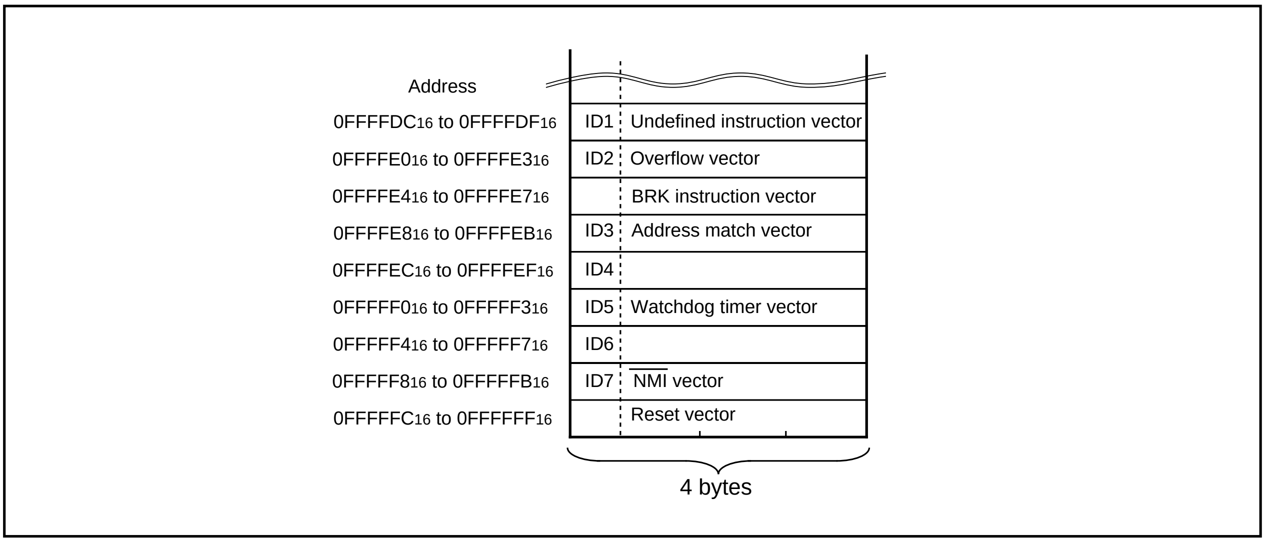


Figure 1.31.17. Timing for the ID check

ID Code

When the flash memory is not blank, the ID code sent from the peripheral units and the ID code written in the flash memory are compared to see if they match. If the codes do not match, the command sent from the peripheral units is not accepted. An ID code contains 8 bits of data. Area is, from the 1st byte, addresses 0FFFFDF$_{16}$, 0FFFFE3$_{16}$, 0FFFFEB$_{16}$, 0FFFFEF$_{16}$, 0FFFFF3$_{16}$, 0FFFFF7$_{16}$ and 0FFFFFB$_{16}$. Write a program into the flash memory, which already has the ID code set for these addresses.



Address		
0FFFFDC$_{16}$ to 0FFFFDF$_{16}$	ID1	Undefined instruction vector
0FFFFE0$_{16}$ to 0FFFFE3$_{16}$	ID2	Overflow vector
0FFFFE4$_{16}$ to 0FFFFE7$_{16}$		BRK instruction vector
0FFFFE8$_{16}$ to 0FFFFEB$_{16}$	ID3	Address match vector
0FFFFEC$_{16}$ to 0FFFFEF$_{16}$	ID4	
0FFFFF0$_{16}$ to 0FFFFF3$_{16}$	ID5	Watchdog timer vector
0FFFFF4$_{16}$ to 0FFFFF7$_{16}$	ID6	
0FFFFF8$_{16}$ to 0FFFFFB$_{16}$	ID7	$\overline{\text{NMI}}$ vector
0FFFFFC$_{16}$ to 0FFFFFF$_{16}$		Reset vector

Figure 1.31.18. ID code storage addresses

Read Check Data

This command reads the check data that confirms that the write data, which was sent with the page program command, was successfully received.

(1) Transfer the "FD_{16}" command code with the 1st byte.

(2) The check data (low) is received with the 2nd byte and the check data (high) with the 3rd.

To use this read check data command, first execute the command and then initialize the check data. Next, execute the page program command the required number of times. After that, when the read check command is executed again, the check data for all of the read data that was sent with the page program command during this time is read. The check data is the result of CRC operation of write data.

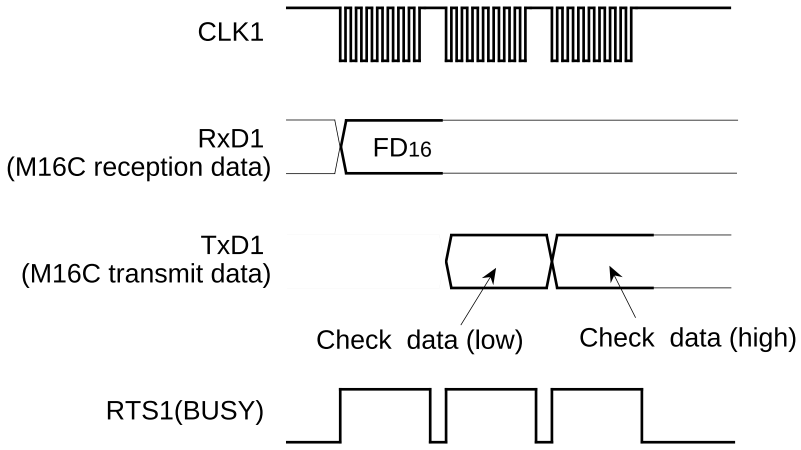


Figure 1.31.19. Timing for the read check data

Data Protection (Block Lock)

Each of the blocks in Figure 1.31.20 have a nonvolatile lock bit that specifies protection (block lock) against erasing/writing. A block is locked (writing “0” for the lock bit) with the lock bit program command. Also, the lock bit of any block can be read with the read lock bit status command.

Block lock disable/enable is determined by the status of the lock bit itself and execution status of the lock bit disable and lock enable bit commands.

(1) After the reset has been cancelled and the lock bit enable command executed, the specified block can be locked/unlocked using the lock bit (lock bit data). Blocks with a “0” lock bit data are locked and cannot be erased or written in. On the other hand, blocks with a “1” lock bit data are unlocked and can be erased or written in.

(2) After the lock bit enable command has been executed, all blocks are unlocked regardless of lock bit data status and can be erased or written in. In this case, lock bit data that was “0” before the block was erased is set to “1” (unlocked) after erasing, therefore the block is actually unlocked with the lock bit.

Flash memory size	Flash memory start address
128 Kbytes	0FE0000$_{16}$
256 Kbytes	0FC0000$_{16}$

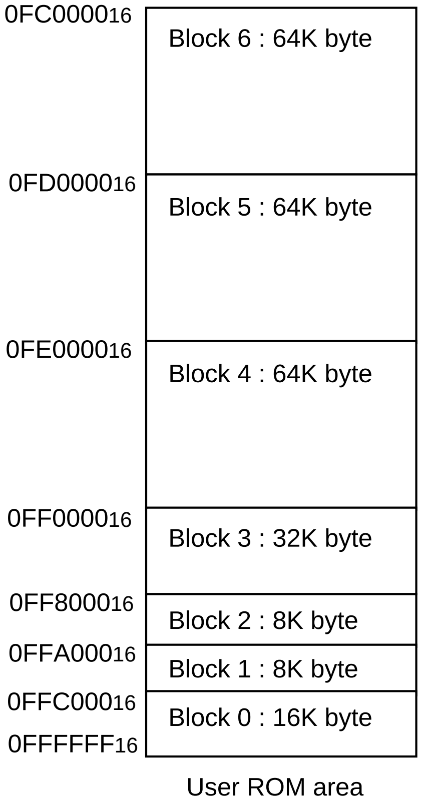


Figure 1.31.20. Blocks in the user area

Status Register (SRD)

The status register indicates operating status of the flash memory and status such as whether an erase operation or a program ended successfully or in error. It can be read by writing the read status register command (70_{16}). Also, the status register is cleared by writing the clear status register command (50_{16}). Table 1.31.2 gives the definition of each status register bit. After clearing the reset, the status register outputs “80_{16}”.

Table 1.31.2. Status register (SRD)

SRD0 bits	Status name	Definition	
		"1"	"0"
SR7 (bit7)	Write state machine (WSM) status	Ready	Busy
SR6 (bit6)	Reserved	-	-
SR5 (bit5)	Erase status	Terminated in error	Terminated normally
SR4 (bit4)	Program status	Terminated in error	Terminated normally
SR3 (bit3)	Block status after program	Terminated in error	Terminated normally
SR2 (bit2)	Reserved	-	-
SR1 (bit1)	Reserved	-	-
SR0 (bit0)	Reserved	-	-

Write State Machine (WSM) Status (SR7)

The write state machine (WSM) status indicates the operating status of the flash memory. When power is turned on, “1” (ready) is set for it. The bit is set to “0” (busy) during an auto write or auto erase operation, but it is set back to “1” when the operation ends.

Erase Status (SR5)

The erase status reports the operating status of the auto erase operation. If an erase error occurs, it is set to “1”. When the erase status is cleared, it is set to “0”.

Program Status (SR4)

The program status reports the operating status of the auto write operation. If a write error occurs, it is set to “1”. When the program status is cleared, it is set to “0”.

Program Status After Program (SR3)

If excessive data is written (phenomenon whereby the memory cell becomes depressed which results in data not being read correctly), “1” is set for the program status after-program at the end of the page write operation. In other words, when writing ends successfully, “80_{16}” is output; when writing fails, “90_{16}” is output; and when excessive data is written, “88_{16}” is output.

If “1” is written for any of the SR5, SR4 or SR3 bits, the page program, block erase, erase all unlocked blocks and lock bit program commands are not accepted. Before executing these commands, execute the clear status register command (50_{16}) and clear the status register.

Status Register 1 (SRD1)

Status register 1 indicates the status of serial communications, results from ID checks and results from check sum comparisons. It can be read after the SRD by writing the read status register command (70_{16}). Also, status register 1 is cleared by writing the clear status register command (50_{16}).

Table 1.31.3 gives the definition of each status register 1 bit. “00_{16}” is output when power is turned ON and the flag status is maintained even after the reset.

Table 1.31.3. Status register 1 (SRD1)

SRD1 bits	Status name	Definition	
		"1"	"0"
SR15 (bit7)	Boot update completed bit	Update completed	Not update
SR14 (bit6)	Reserved	-	-
SR13 (bit5)	Reserved	-	-
SR12 (bit4)	Checksum match bit	Match	Mismatch
SR11 (bit3) SR10 (bit2)	ID check completed bits	00 Not verified 01 Verification mismatch 10 Reserved 11 Verified	
SR9 (bit1)	Data receive time out	Time out	Normal operation
SR8 (bit0)	Reserved	-	-

Boot Update Completed Bit (SR15)

This flag indicates whether the control program was downloaded to the RAM or not, using the download function.

Check Sum Consistency Bit (SR12)

This flag indicates whether the check sum matches or not when a program, is downloaded for execution using the download function.

ID Check Completed Bits (SR11 and SR10)

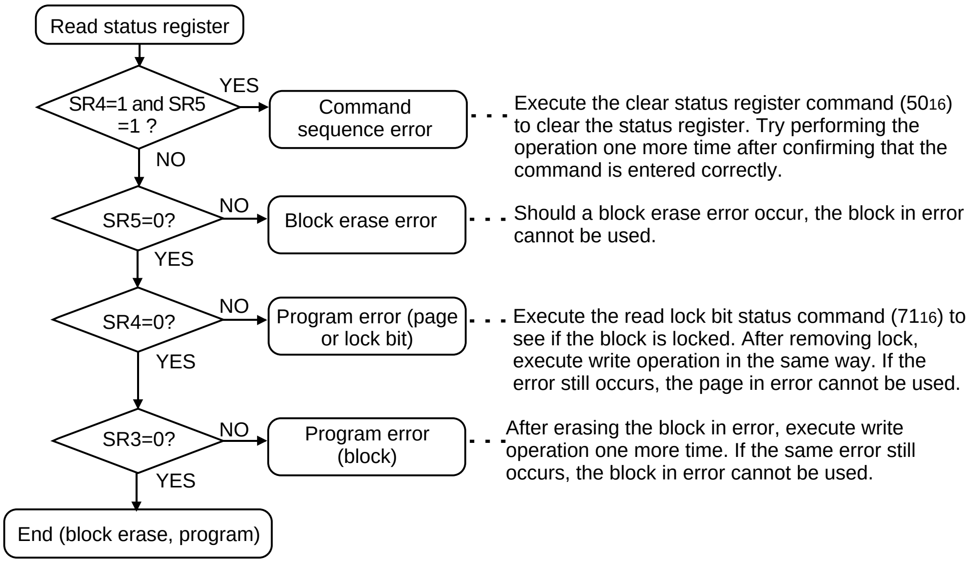
These flags indicate the result of ID checks. Some commands cannot be accepted without an ID check.

Data Reception Time Out (SR9)

This flag indicates when a time out error is generated during data reception. If this flag is attached during data reception, the received data is discarded and the microcomputer returns to the command wait state.

Full Status Check

Results from executed erase and program operations can be known by running a full status check. Figure 1.31.21 shows a flowchart of the full status check and explains how to remedy errors which occur.

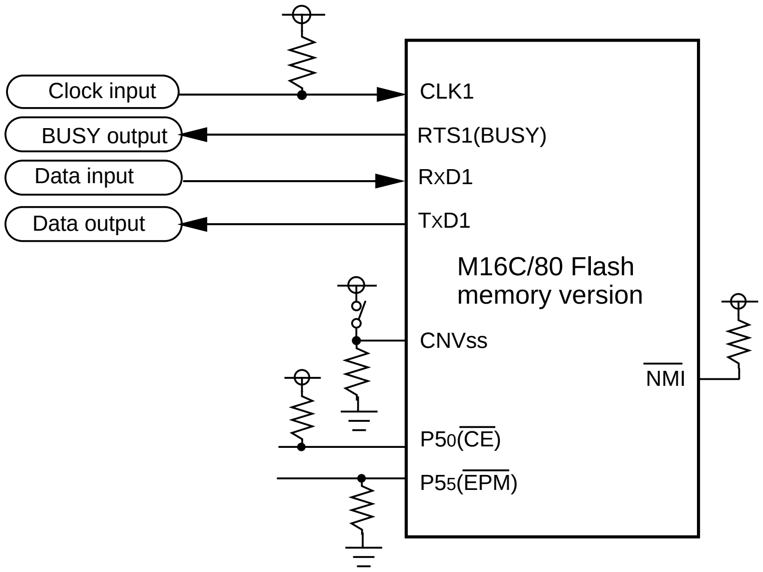

Read status register
SR4=1 and SR5 =1 ?
YES
NO
Command sequence error
Execute the clear status register command (50_{16}) to clear the status register. Try performing the operation one more time after confirming that the command is entered correctly.
SR5=0?
NO
YES
Block erase error
Should a block erase error occur, the block in error cannot be used.
SR4=0?
NO
YES
Program error (page or lock bit)
Execute the read lock bit status command (71_{16}) to see if the block is locked. After removing lock, execute write operation in the same way. If the error still occurs, the page in error cannot be used.
SR3=0?
NO
YES
Program error (block)
After erasing the block in error, execute write operation one more time. If the same error still occurs, the block in error cannot be used.
End (block erase, program)

Note: When one of SR5 to SR3 is set to 1, none of the page program, block erase, erase all unlock blocks and lock bit program commands is accepted. Execute the clear status register command (50_{16}) before executing these commands.

Figure 1.31.21. Full status check flowchart and remedial procedure for errors

Example Circuit Application for The Standard Serial I/O Mode 1

The below figure shows a circuit application for the standard serial I/O mode 1. Control pins will vary according to peripheral unit (programmer), therefore see the peripheral unit (programmer) manual for more information.



(1) Control pins and external circuitry will vary according to peripheral unit (programmer). For more information, see the peripheral unit (programmer) manual.
(2) In this example, the microprocessor mode and standard serial I/O mode are switched via a switch.

Figure 1.31.22. Example circuit application for the standard serial I/O mode 1

Overview of standard serial I/O mode 2 (clock asynchronized)

In standard serial I/O mode 2, software commands, addresses and data are input and output between the MCU and peripheral units (serial programer, etc.) using 2-wire clock-asynchronized serial I/O (UART1). Standard serial I/O mode 2 is engaged by releasing the reset with the $P6_5$ (CLK_1) pin "L" level.
The TxD_1 pin is for CMOS output. Data transfer is in 8-bit units with LSB first, 1 stop bit and parity OFF.
After the reset is released, connections can be established at 9,600 bps when initial communications (Figure 1.31.23) are made with a peripheral unit. However, this requires a main clock with a minimum 2 MHz input oscillation frequency. Baud rate can also be changed from 9,600 bps to 19,200, 38,400, 57,600 or 115,200 bps by executing software commands. However, communication errors may occur because of the oscillation frequency of the main clock. If errors occur, change the main clock's oscillation frequency and the baud rate.
After executing commands from a peripheral unit that requires time to erase and write data, as with erase and program commands, allow a sufficient time interval or execute the read status command and check how processing ended, before executing the next command.
Data and status registers in memory can be read after transmitting software commands. Status, such as the operating state of the flash memory or whether a program or erase operation ended successfully or not, can be checked by reading the status register. Here following are explained initial communications with peripheral units, how frequency is identified and software commands.

Initial communications with peripheral units

After the reset is released, the bit rate generator is adjusted to 9,600 bps to match the oscillation frequency of the main clock, by sending the code as prescribed by the protocol for initial communications with peripheral units (Figure 1.31.23).

(1) Transmit "00_{16}" from a peripheral unit 16 times. (The MCU with internal flash memory sets the bit rate generator so that "00_{16}" can be successfully received.)

(2) The MCU with internal flash memory outputs the "$B0_{16}$" check code and initial communications end successfully *[1]. Initial communications must be transmitted at a speed of 9,600 bps and a transfer interval of a minimum 15 ms. Also, the baud rate at the end of initial communications is 9,600 bps.

*1. If the peripheral unit cannot receive "$B0_{16}$" successfully, change the oscillation frequency of the main clock.

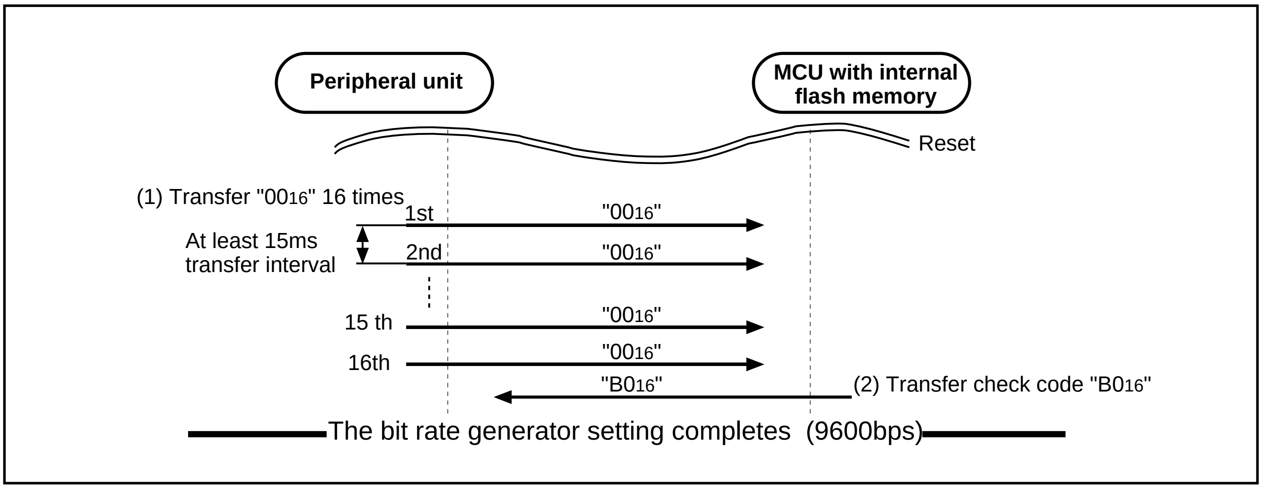

Peripheral unit
MCU with internal flash memory
Reset
(1) Transfer "0016" 16 times
1st
"0016"
At least 15ms transfer interval
2nd
"0016"
15 th
"0016"
16th
"0016"
"B016"
(2) Transfer check code "B016"
The bit rate generator setting completes (9600bps)

Figure 1.31.23. Peripheral unit and initial communication

How frequency is identified

When "00₁₆" data is received 16 times from a peripheral unit at a baud rate of 9,600 bps, the value of the bit rate generator is set to match the operating frequency (2 - 20 MHz). The highest speed is taken from the first 8 transmissions and the lowest from the last 8. These values are then used to calculate the bit rate generator value for a baud rate of 9,600 bps.

Baud rate cannot be attained with some operating frequencies. Table 1.31.4 gives the operation frequency and the baud rate that can be attained for.

Table 1.31.4 Operation frequency and the baud rate

Operation frequency (MHz)	Baud rate 9,600bps	Baud rate 19,200bps	Baud rate 38,400bps	Baud rate 57,600bps	Baud rate 115,200bps
20MHz	√	√	√	√	√
16MHz	√	√	√	√	–
12MHz	√	√	√	√	–
11MHz	√	√	√	√	–
10MHz	√	√	√	√	–
8MHz	√	√	√	√	–
7.3728MHz	√	√	√	√	–
6MHz	√	√	√	–	–
5MHz	√	√	√	–	–
4.5MHz	√	√	√	√	–
4.194304MHz	√	√	√	–	–
4MHz	√	√	–	–	–
3.58MHz	√	√	√	√	–
3MHz	√	√	√	–	–
2MHz	√	–	–	–	–

√ : Communications possible
– : Communications not possible

Software Commands

Table 1.31.5 lists software commands. In the standard serial I/O mode 2, erase operations, programs and reading are controlled by transferring software commands via the RxD$_1$ pin. Standard serial I/O mode 2 adds five transmission speed commands - 9,600, 19,200, 38,400, 57,600 and 115,200 bps - to the software commands of standard serial I/O mode 1. Software commands are explained here below.

Table 1.31.5. Software commands (Standard serial I/O mode 2)

	Control command	1st byte transfer	2nd byte	3rd byte	4th byte	5th byte	6th byte		When ID is not verified
1	Page read	FF_{16}	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
2	Page program	41_{16}	Address (middle)	Address (high)	Data input	Data input	Data input	Data input to 259th byte	Not acceptable
3	Block erase	20_{16}	Address (middle)	Address (high)	$D0_{16}$				Not acceptable
4	Erase all unlocked blocks	$A7_{16}$	$D0_{16}$						Not acceptable
5	Read status register	70_{16}	SRD output	SRD1 output					Acceptable
6	Clear status register	50_{16}							Not acceptable
7	Read lock bit status	71_{16}	Address (middle)	Address (high)	Lock bit data output				Not acceptable
8	Lock bit program	77_{16}	Address (middle)	Address (high)	$D0_{16}$				Not acceptable
9	Lock bit enable	$7A_{16}$							Not acceptable
10	Lock bit disable	75_{16}							Not acceptable
11	Code processing function	$F5_{16}$	Address (low)	Address (middle)	Address (high)	ID size	ID1	To ID7	Acceptable
12	Download function	FA_{16}	Size (low)	Size (high)	Check-sum	Data input	To required number of times		Not acceptable
13	Version data output function	FB_{16}	Version data output	Version data output	Version data output	Version data output	Version data output	Version data output to 9th byte	Acceptable
14	Boot ROM area output function	FC_{16}	Address (middle)	Address (high)	Data output	Data output	Data output	Data output to 259th byte	Not acceptable
15	Read check data	FD_{16}	Check data (low)	Check data (high)					Not acceptable
16	Baud rate 9600	$B0_{16}$	$B0_{16}$						Acceptable
17	Baud rate 19200	$B1_{16}$	$B1_{16}$						Acceptable
18	Baud rate 38400	$B2_{16}$	$B2_{16}$						Acceptable
19	Baud rate 57600	$B3_{16}$	$B3_{16}$						Acceptable
20	Baud rate 115200	$B4_{16}$	$B4_{16}$						Acceptable

Note 1: Shading indicates transfer from flash memory microcomputer to peripheral unit. All other data is transferred from the peripheral unit to the flash memory microcomputer.

Note 2: SRD refers to status register data. SRD1 refers to status register data 1.

Note 3: All commands can be accepted when the flash memory is totally blank.

Page Read Command

This command reads the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page read command as explained here following.

(1) Transfer the “FF_{16}” command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, data (D_0–D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} will be output sequentially from the smallest address first in sync with the rise of the clock.

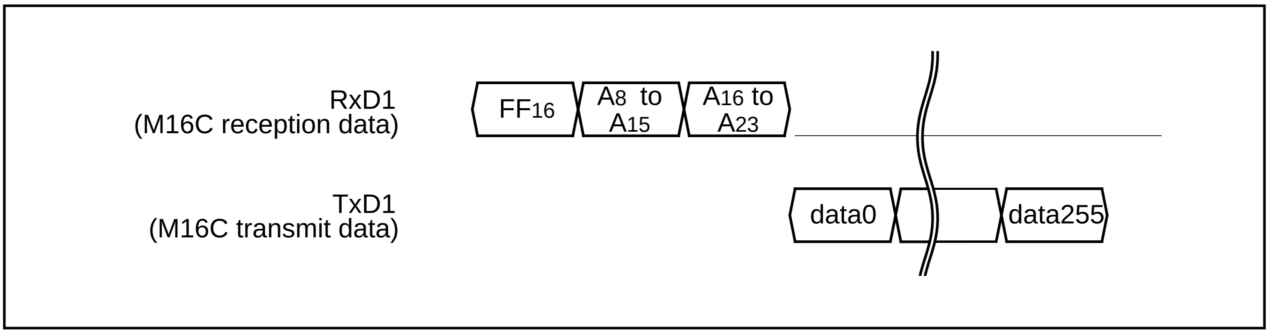


Figure 1.31.24. Timing for page read

Read Status Register Command

This command reads status information. When the “70_{16}” command code is sent with the 1st byte, the contents of the status register (SRD) specified with the 2nd byte and the contents of status register 1 (SRD1) specified with the 3rd byte are read.

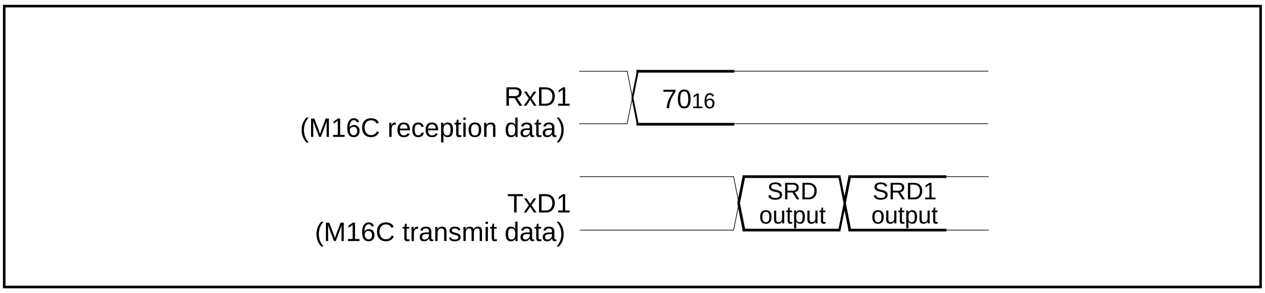


Figure 1.31.25. Timing for reading the status register

Clear Status Register Command

This command clears the bits (SR3–SR5) which are set when the status register operation ends in error. When the “50_{16}” command code is sent with the 1st byte, the aforementioned bits are cleared.

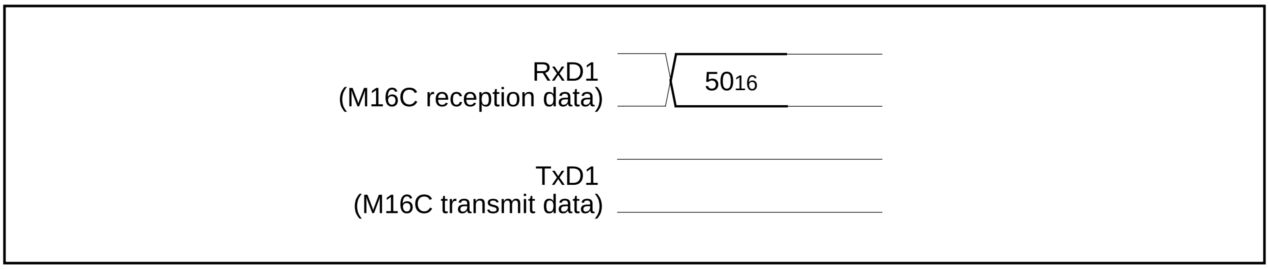


Figure 1.31.26. Timing for clearing the status register

Page Program Command

This command writes the specified page (256 bytes) in the flash memory sequentially one byte at a time. Execute the page program command as explained here following.

(1) Transfer the “41_{16}” command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, as write data (D_0–D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} is input sequentially from the smallest address first, that page is automatically written.

The result of the page program can be known by reading the status register. For more information, see the section on the status register.
Each block can be write-protected with the lock bit. For more information, see the section on the data protection function. Additional writing is not allowed with already programmed pages.

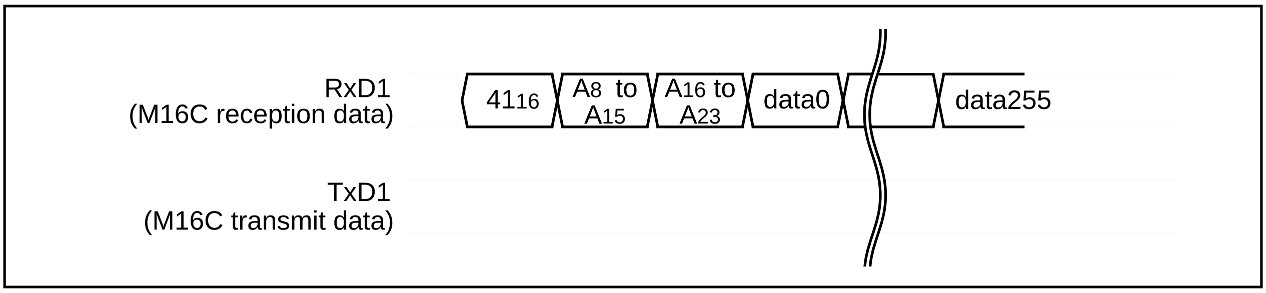


Figure 1.31.27. Timing for the page program

Block Erase Command

This command erases the data in the specified block. Execute the block erase command as explained here following.

(1) Transfer the “20_{16}” command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) Transfer the verify command code “$D0_{16}$” with the 4th byte. With the verify command code, the erase operation will start for the specified block in the flash memory. Write the highest address of the specified block for addresses A_{16} to A_{23}.

After block erase ends, the result of the block erase operation can be known by reading the status register. For more information, see the section on the status register.
Each block can be erase-protected with the lock bit. For more information, see the section on the data protection function.

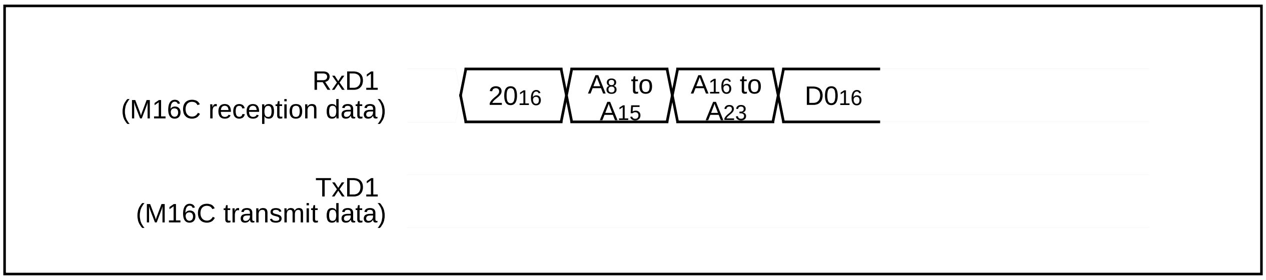


Figure 1.31.28. Timing for block erasing

Erase All Unlocked Blocks Command

This command erases the content of all blocks. Execute the erase all unlocked blocks command as explained here following.

(1) Transfer the "$A7_{16}$" command code with the 1st byte.

(2) Transfer the verify command code "$D0_{16}$" with the 2nd byte. With the verify command code, the erase operation will start and continue for all blocks in the flash memory.

The result of the erase operation can be known by reading the status register. Each block can be erase-protected with the lock bit. For more information, see the section on the data protection function.

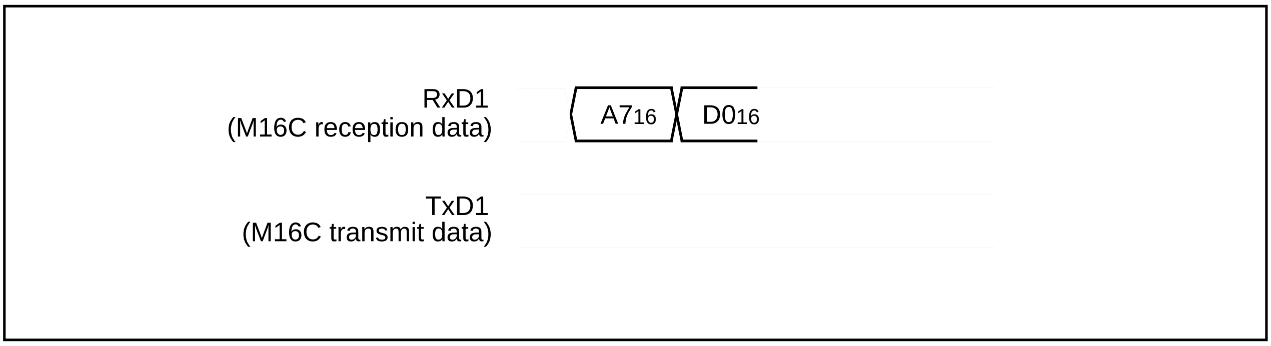


Figure 1.31.29. Timing for erasing all unlocked blocks

Lock Bit Program Command

This command writes "0" (lock) for the lock bit of the specified block. Execute the lock bit program command as explained here following.

(1) Transfer the "77_{16}" command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) Transfer the verify command code "$D0_{16}$" with the 4th byte. With the verify command code, "0" is written for the lock bit of the specified block. Write the highest address of the specified block for addresses A_8 to A_{23}.

Lock bit status can be read with the read lock bit status command. For information on the lock bit function, reset procedure and so on, see the section on the data protection function.

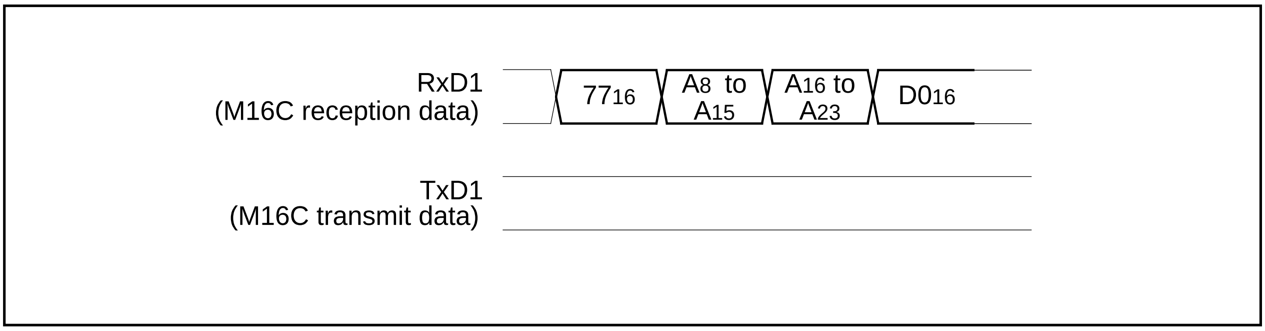


Figure 1.31.30. Timing for the lock bit program

Read Lock Bit Status Command

This command reads the lock bit status of the specified block. Execute the read lock bit status command as explained here following.

(1) Transfer the “71_{16}” command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) The lock bit data of the specified block is output with the 4th byte. Write the highest address of the specified block for addresses A_8 to A_{23}.

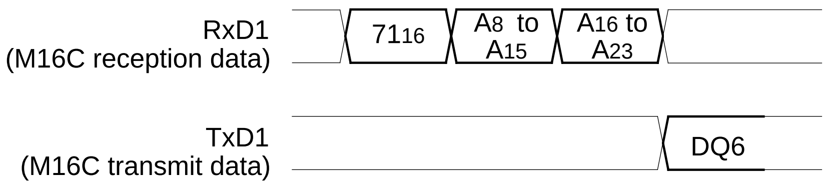


Figure 1.31.31. Timing for reading lock bit status

Lock Bit Enable Command

This command enables the lock bit in blocks whose bit was disabled with the lock bit disable command. The command code “$7A_{16}$” is sent with the 1st byte of the serial transmission. This command only enables the lock bit function; it does not set the lock bit itself.

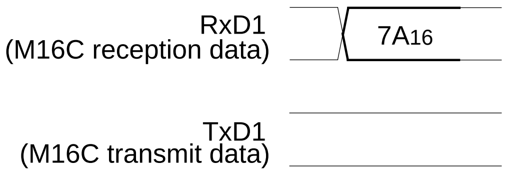


Figure 1.31.32. Timing for enabling the lock bit

Lock Bit Disable Command

This command disables the lock bit. The command code “75_{16}” is sent with the 1st byte of the serial transmission. This command only disables the lock bit function; it does not set the lock bit itself. However, if an erase command is executed after executing the lock bit disable command, “0” (locked) lock bit data is set to “1” (unlocked) after the erase operation ends. In any case, after the reset is cancelled, the lock bit is enabled.

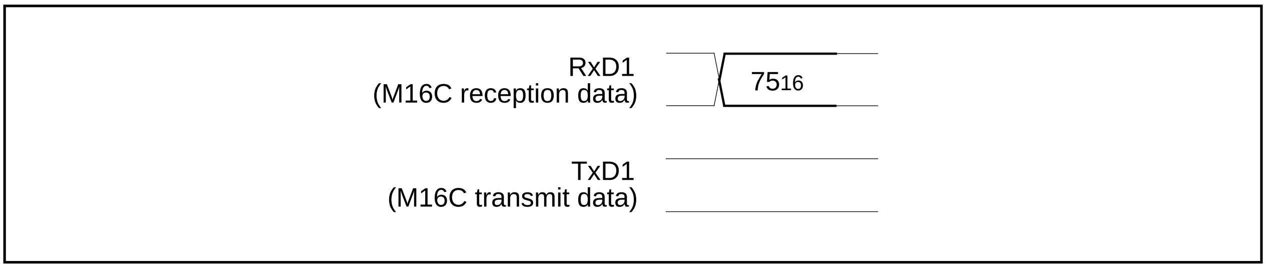


Figure 1.31.33. Timing for disabling the lock bit

Download Command

This command downloads a program to the RAM for execution. Execute the download command as explained here following.

(1) Transfer the “FA_{16}” command code with the 1st byte.
(2) Transfer the program size with the 2nd and 3rd bytes.
(3) Transfer the check sum with the 4th byte. The check sum is added to all data sent with the 5th byte onward.
(4) The program to execute is sent with the 5th byte onward.

When all data has been transmitted, if the check sum matches, the downloaded program is executed. The size of the program will vary according to the internal RAM.

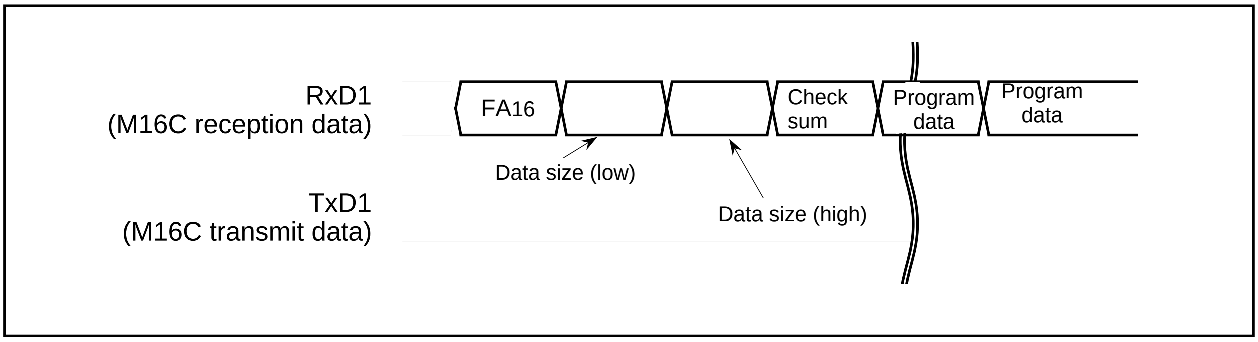


Figure 1.31.34. Timing for download

Version Information Output Command

This command outputs the version information of the control program stored in the boot area. Execute the version information output command as explained here following.

(1) Transfer the "FB$_{16}$" command code with the 1st byte.

(2) The version information will be output from the 2nd byte onward. This data is composed of 8 ASCII code characters.

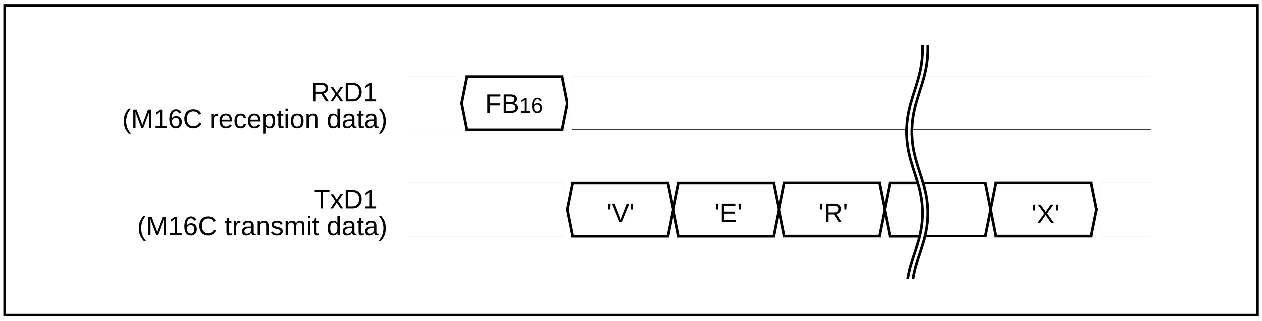


Figure 1.31.35. Timing for version information output

Boot ROM Area Output Command

This command outputs the control program stored in the boot ROM area in one page blocks (256 bytes). Execute the boot ROM area output command as explained here following.

(1) Transfer the "FC$_{16}$" command code with the 1st byte.

(2) Transfer addresses A_8 to A_{15} and A_{16} to A_{23} with the 2nd and 3rd bytes respectively.

(3) From the 4th byte onward, data (D_0–D_7) for the page (256 bytes) specified with addresses A_8 to A_{23} will be output sequentially from the smallest address first, in sync with the rise of the clock.

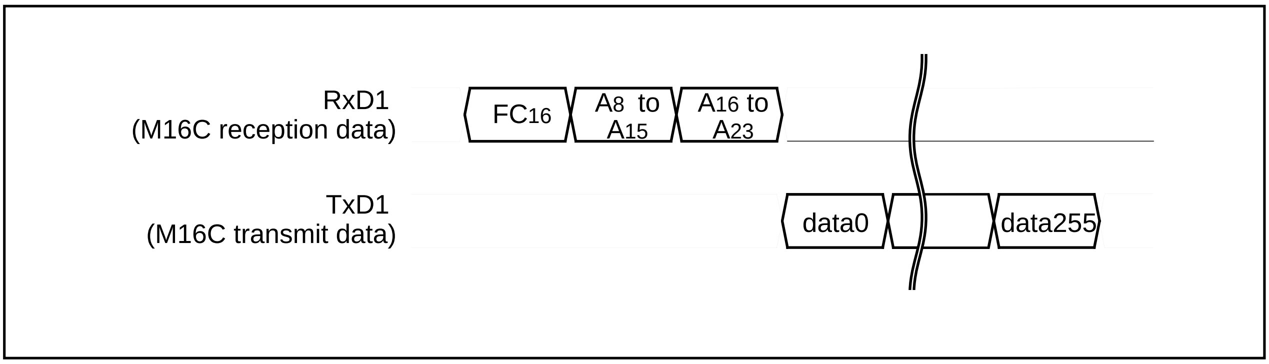


Figure 1.31.36. Timing for boot ROM area output

ID Check

This command checks the ID code. Execute the boot ID check command as explained here following.

(1) Transfer the “$F5_{16}$” command code with the 1st byte.

(2) Transfer addresses A_0 to A_7, A_8 to A_{15} and A_{16} to A_{23} of the 1st byte of the ID code with the 2nd, 3rd and 4th bytes respectively.

(3) Transfer the number of data sets of the ID code with the 5th byte.

(4) The ID code is sent with the 6th byte onward, starting with the 1st byte of the code.

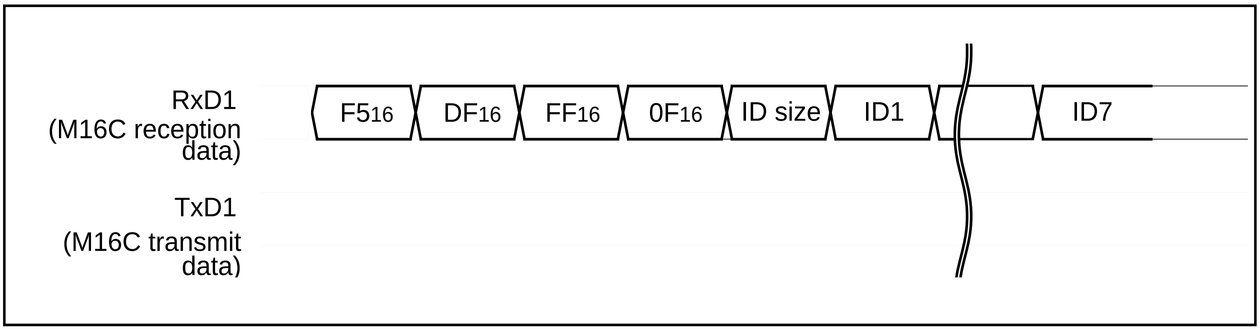
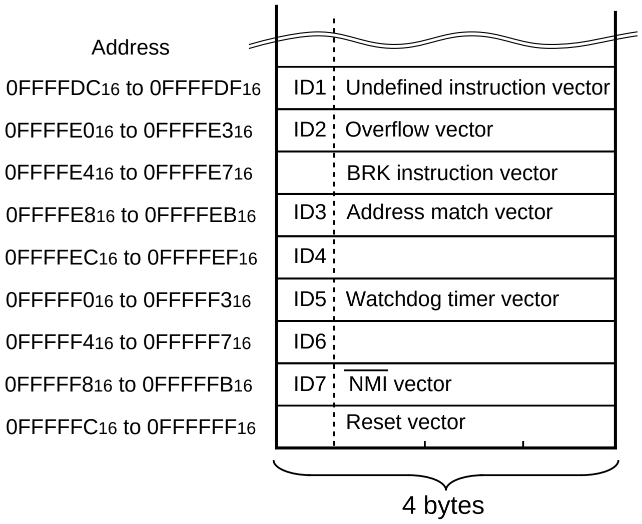


Figure 1.31.37. Timing for the ID check

ID Code

When the flash memory is not blank, the ID code sent from the peripheral units and the ID code written in the flash memory are compared to see if they match. If the codes do not match, the command sent from the peripheral units is not accepted. An ID code contains 8 bits of data. Area is, from the 1st byte, addresses $0FFFFDF_{16}$, $0FFFFE3_{16}$, $0FFFFEB_{16}$, $0FFFFEF_{16}$, $0FFFFF3_{16}$, $0FFFFF7_{16}$ and $0FFFFFB_{16}$. Write a program into the flash memory, which already has the ID code set for these addresses.



Address		
$0FFFFDC_{16}$ to $0FFFFDF_{16}$	ID1	Undefined instruction vector
$0FFFFE0_{16}$ to $0FFFFE3_{16}$	ID2	Overflow vector
$0FFFFE4_{16}$ to $0FFFFE7_{16}$		BRK instruction vector
$0FFFFE8_{16}$ to $0FFFFEB_{16}$	ID3	Address match vector
$0FFFFEC_{16}$ to $0FFFFEF_{16}$	ID4	
$0FFFFF0_{16}$ to $0FFFFF3_{16}$	ID5	Watchdog timer vector
$0FFFFF4_{16}$ to $0FFFFF7_{16}$	ID6	
$0FFFFF8_{16}$ to $0FFFFFB_{16}$	ID7	$\overline{\text{NMI}}$ vector
$0FFFFFC_{16}$ to $0FFFFFF_{16}$		Reset vector

Figure 1.31.38. ID code storage addresses

Read Check Data

This command reads the check data that confirms that the write data, which was sent with the page program command, was successfully received.

(1) Transfer the "FD_{16}" command code with the 1st byte.

(2) The check data (low) is received with the 2nd byte and the check data (high) with the 3rd.

To use this read check data command, first execute the command and then initialize the check data. Next, execute the page program command the required number of times. After that, when the read check command is executed again, the check data for all of the read data that was sent with the page program command during this time is read. The check data is the result of CRC operation of write data.

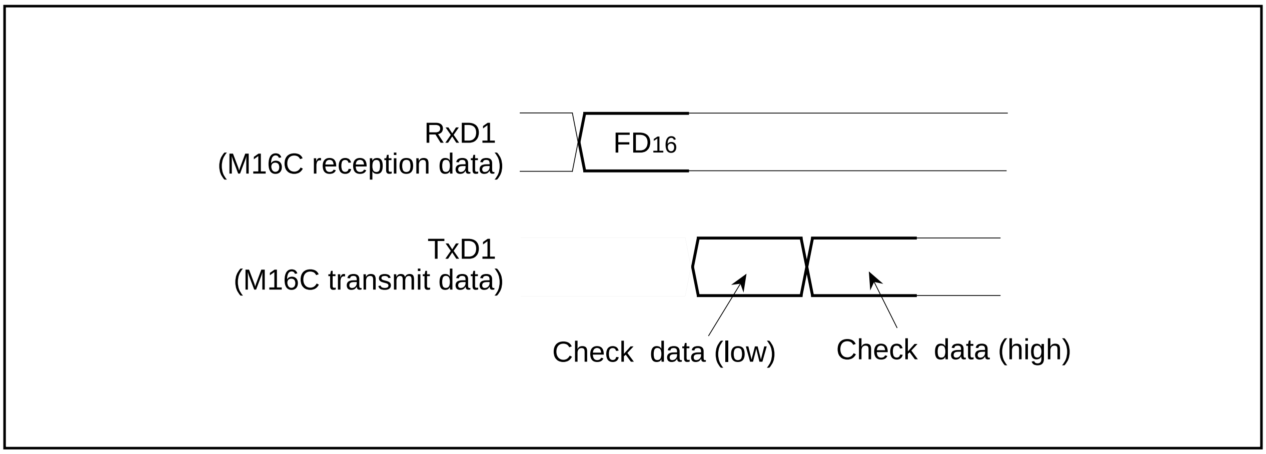


Figure 1.31.39. Timing for the read check data

Baud Rate 9600

This command changes baud rate to 9,600 bps. Execute it as follows.

(1) Transfer the "$B0_{16}$" command code with the 1st byte.

(2) After the "$B0_{16}$" check code is output with the 2nd byte, change the baud rate to 9,600 bps.

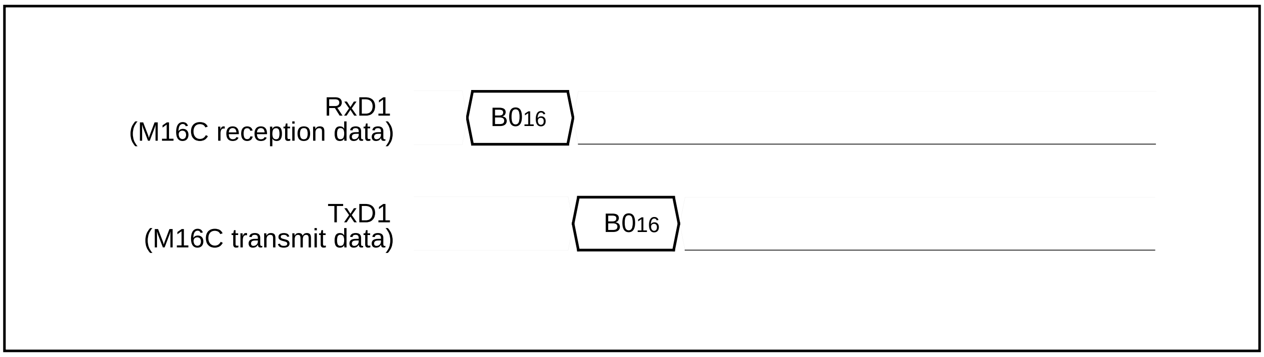


Figure 1.31.40. Timing of baud rate 9600

Baud Rate 19200

This command changes baud rate to 19,200 bps. Execute it as follows.

(1) Transfer the "$B1_{16}$" command code with the 1st byte.

(2) After the "$B1_{16}$" check code is output with the 2nd byte, change the baud rate to 19,200 bps.

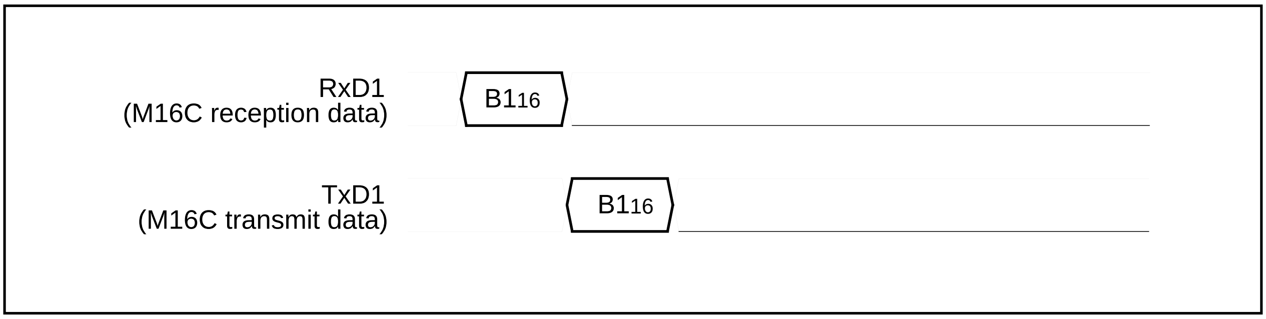


Figure 1.31.41. Timing of baud rate 19200

Baud Rate 38400

This command changes baud rate to 38,400 bps. Execute it as follows.

(1) Transfer the "$B2_{16}$" command code with the 1st byte.

(2) After the "$B2_{16}$" check code is output with the 2nd byte, change the baud rate to 38,400 bps.

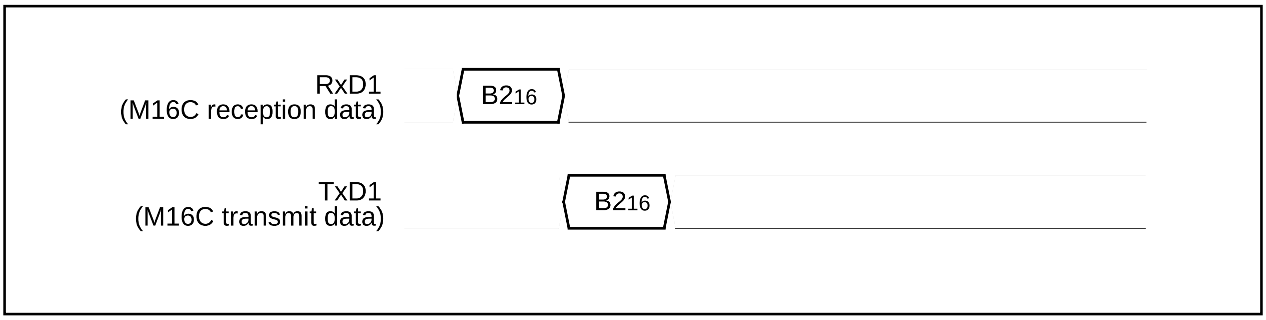


Figure 1.31.42. Timing of baud rate 38400

Baud Rate 57600

This command changes baud rate to 57,600 bps. Execute it as follows.

(1) Transfer the "$B3_{16}$" command code with the 1st byte.

(2) After the "$B3_{16}$" check code is output with the 2nd byte, change the baud rate to 57,600 bps.

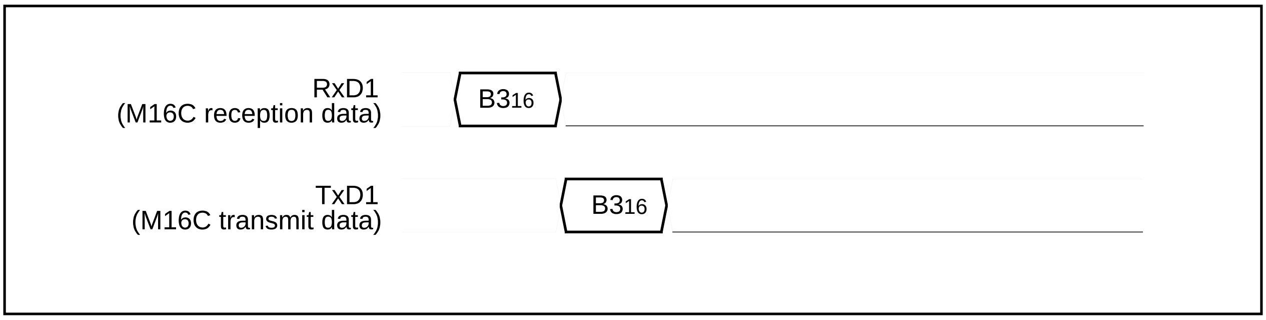


Figure 1.31.43. Timing of baud rate 57600

Baud Rate 115200

This command changes baud rate to 115,200 bps. Execute it as follows.

(1) Transfer the "B4₁₆" command code with the 1st byte.

(2) After the "B4₁₆" check code is output with the 2nd byte, change the baud rate to 19,200 bps.

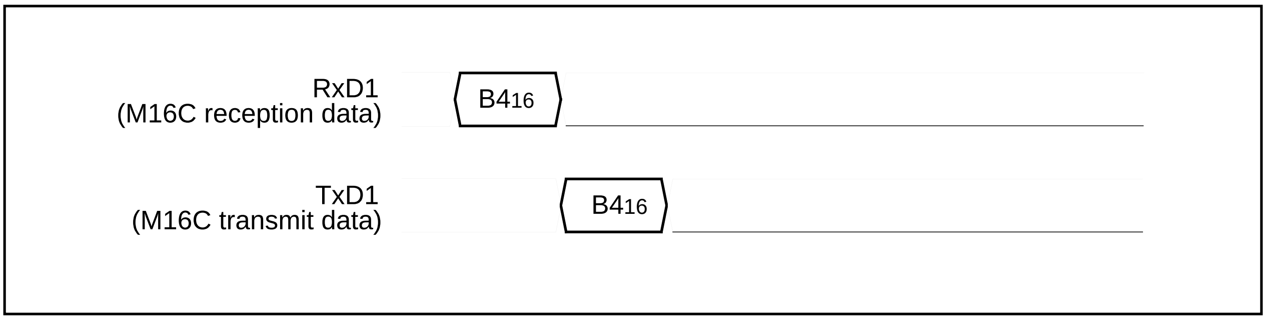
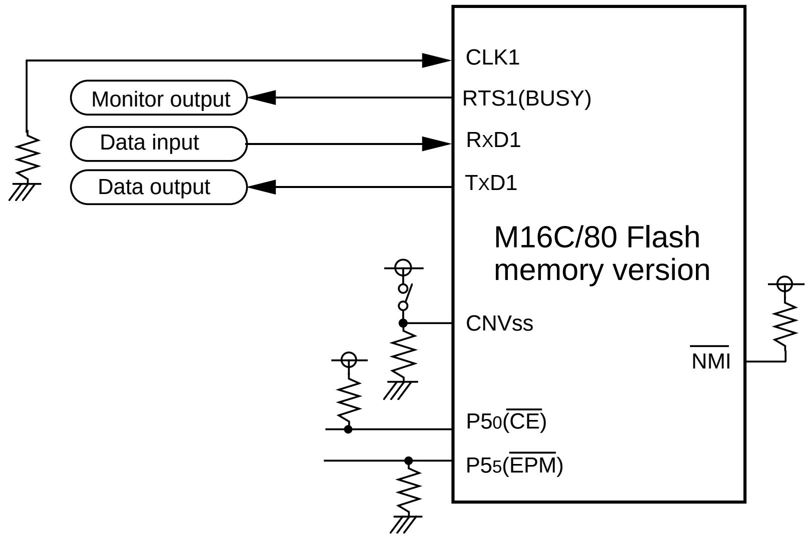


Figure 1.31.44. Timing of baud rate 115200

Example Circuit Application for The Standard Serial I/O Mode 2

The below figure shows a circuit application for the standard serial I/O mode 2.



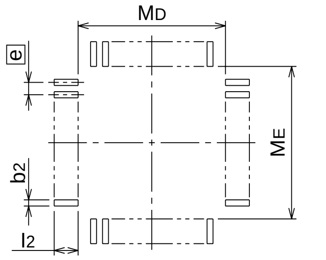
(1) In this example, the microprocessor mode and standard serial I/O mode are switched via a switch.

Figure 1.31.45. Example circuit application for the standard serial I/O mode 2

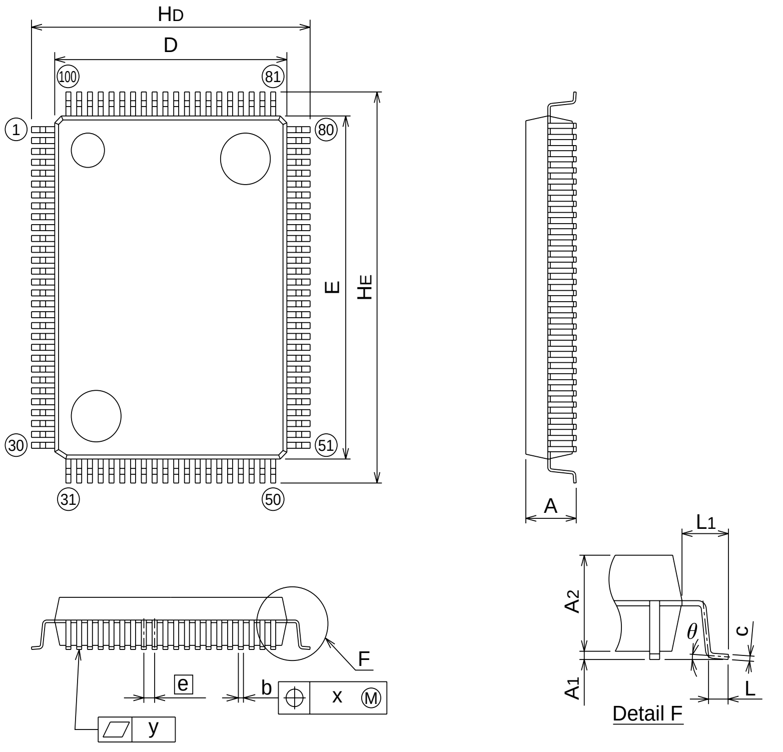
100P6S-A (MMP)

Plastic 100pin 14X20mm body QFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
QFP100-P-1420-0.65	–	1.58	Alloy 42



Recommended Mount Pad

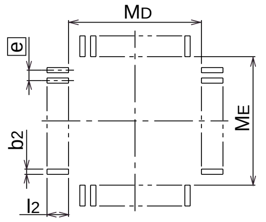


Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	3.05
A_1	0	0.1	0.2
A_2	–	2.8	–
b	0.25	0.3	0.4
c	0.13	0.15	0.2
D	13.8	14.0	14.2
E	19.8	20.0	20.2
e	–	0.65	–
H_D	16.5	16.8	17.1
H_E	22.5	22.8	23.1
L	0.4	0.6	0.8
L_1	–	1.4	–
x	–	–	0.13
y	–	–	0.1
θ	0°	–	10°
b_2	–	0.35	–
I_2	1.3	–	–
M_D	–	14.6	–
M_E	–	20.6	–

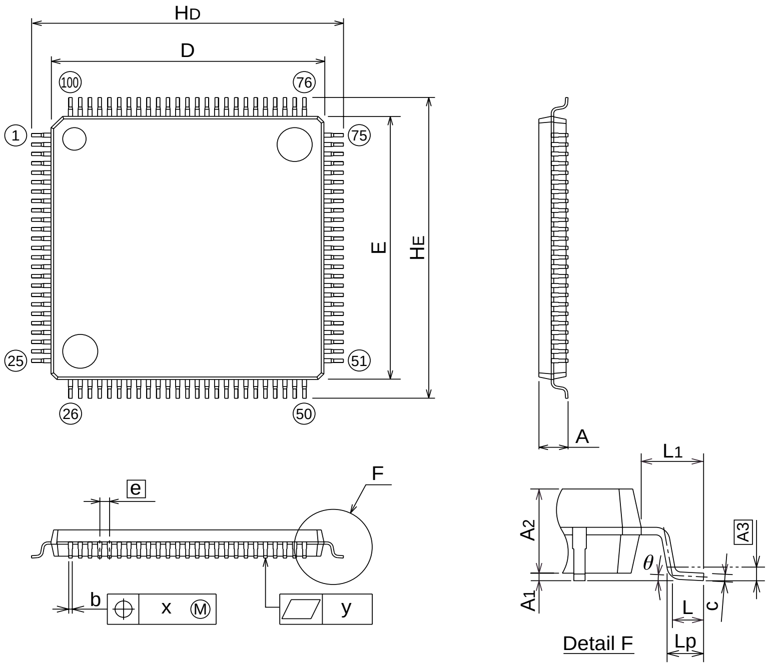
100P6Q-A (MMP)

Plastic 100pin 14X14mm body LQFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP100-P-1414-0.50	–	0.63	Cu Alloy



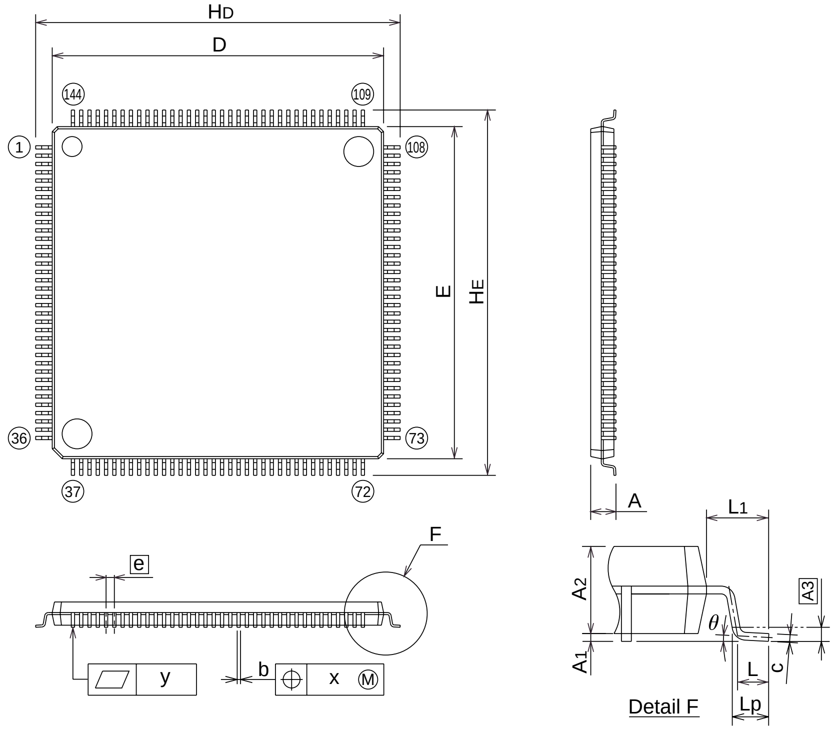
Recommended Mount Pad

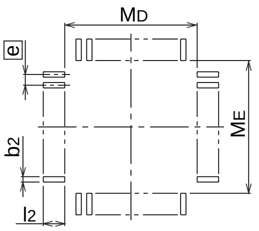


Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	1.7
A_1	0	0.1	0.2
A_2	–	1.4	–
b	0.13	0.18	0.28
c	0.105	0.125	0.175
D	13.9	14.0	14.1
E	13.9	14.0	14.1
e	–	0.5	–
H_D	15.8	16.0	16.2
H_E	15.8	16.0	16.2
L	0.3	0.5	0.7
L_1	–	1.0	–
Lp	0.45	0.6	0.75
A_3	–	0.25	–
x	–	–	0.08
y	–	–	0.1
θ	0°	–	10°
b_2	–	0.225	–
I_2	0.9	–	–
M_D	–	14.4	–
M_E	–	14.4	–

144P6Q-A (MMP) Plastic 144pin 20×20mm body LQFP

EIAJ Package Code	JEDEC Code	Weight(g)	Lead Material
LQFP144-P-2020-0.50	–	1.23	Cu Alloy





Recommended Mount Pad

Symbol	Dimension in Millimeters		
	Min	Nom	Max
A	–	–	1.7
A_1	0.05	0.125	0.2
A_2	–	1.4	–
b	0.17	0.22	0.27
c	0.105	0.125	0.175
D	19.9	20.0	20.1
E	19.9	20.0	20.1
$\boxed{e}$	–	0.5	–
H_D	21.8	22.0	22.2
H_E	21.8	22.0	22.2
L	0.35	0.5	0.65
L_1	–	1.0	–
Lp	0.45	0.6	0.75
$\boxed{A_3}$	–	0.25	–
x	–	–	0.08
y	–	–	0.1
θ	0°	–	8°
b_2	–	0.225	–
l_2	0.95	–	–
M_D	–	20.4	–
M_E	–	20.4	–

Revision History

Version	Contents for change	Revision date
REV.B	• Page 1 line 5 1 M byte --> 16 M bytes • Page 1 line 15 10 MHz with software one wait --> 10 MHz : under planning • Page 1 line 16 35 mW (f(XIN)=20MHz, without software wait, Vcc=5V; M30800MC-XXXFP target value) --> 45 mA (M30800MC-XXXFP) • Page 1 X-Y converter ---- 1 circuit Addition • Page 4 line 28 35 mA --> 45 mA • Page 6 figure 1.1.4 • Page 18 figure 1.5.4 and corresponding pages (106) Peripheral subfunction select register --> Function select register C (107) Port function select register 0 --> Function select register A0 (108) Port function select register 1 --> Function select register A1 (109) Peripheral function select register 0 --> Function select register B0 (110) Peripheral function select register 1 --> Function select register B1 (111) Port function select register 2 --> Function select register A2 (112) Port function select register 3 --> Function select register A3 (113) Peripheral function select register 2 --> Function select register B2 • Page 21 figure 1.6.3 Register name change same as figure 1.5.4 • Page 24 figure 1.8.1 Processor mode register 0 Note 6 --> Note 7, Note 6 Addition Processor mode register 1 Note 3 • Page 31 line 4 Addition: The ALE signal is occurred regardless of internal area and external area. • Page 31 table 1.10.4, Page 33 table 1.10.5 R/$\overline{W}$ --> $\overline{RD}$/$\overline{WR}$ • Page 42 table 1.11.4 System clock control register 0 Note 2 • Page 51 line 7, table 1.15.1 port function select register 3 (address 03B5$_{16}$) --> port function select register 3 (address 03B5$_{16}$) and D-A control register (address 039C$_{16}$) • Page 60 line 3 the interrupt occurs. --> the interrupt can be set to occur on input signal level and input signal edge. • Page 65 line 10 Set register --> When writing to DCT2, DCT3, DRC2, DRC3, DMA2 and DMA3, set register • Page 67 table 1.20.2 Addition: Note 5 • Page 86 line 1 successively when --> successively two times when • Page 93 line 16 Count source input --> Count source input (Set the corresponding function select register A to I/O port.) • Page 114 table 1.25.6, page 122 table 1.26.1 UARTi transmit/receive mode register Addition: Note 2 UARTi transmit/receive mode register Addition: Note 3 • Page 115 table 1.25.7 UARTi transmit/receive mode register 0 Delate: Note 3 • Page 120 line 13 Addition: -Set the corresponding function select register A to I/O port • Page 123 table 1.26.3 • Page 130 table 1.27.3 • Page 139 figure 1.29.2 • Page 142 line 2 0623$_{16}$ --> 0326$_{16}$ • Page 144 table 1.29.5 • Page 156 table 1.31.2 D-A control register (Note) Addition: Note • Page 164 figure 1.34.2 16-bit bus mode A9 --> $\overline{A9}$ • Page 165 line 5 f_{32} --> BCLK(frequency x 32)	'98. 10.19

Version	Contents for change	Revision date
REV.B	• Page 165 figure 1.34.3 operation clock --> BCLK • Page 169 they function as output regardless of the contents of the direction registers. When pins are to be used as the outputs for the D-A converter, do not set the direction registers to output mode. --> set the corresponding function select registers A, B and C. When pins are to be used as the outputs for the D-A converter, set the function select register of each pin to I/O port, and set the direction registers to input mode. Table 1.35.1 lists each port and peripheral function.	'98.10.19
REV.C	All page M30800MC-XXXFP --> M16C/80 (100-pin version) group Page 2 Figure 1 changed, GP package is added Page 3 Figure 3 Note 1 and Note 2 is added Page 5 Figure 4, Table 2 New type no. is added Page 6 Figure 5 GP is added Page 10 Line 2 18 registers --> 28 registers Page 11 (7) Set USP and ISP to an even number so that execution efficiency is increased. --> added Page 17 Figure 11 (54) UART4 special mode register 3 --> added Page 18 Figure 12 UART3 special mode register 3 --> added UART2 special mode register 3 --> added Function select register B3 --> added Page 20 Figure 14 UART4 special mode register 3 --> added UART3 special mode register 3 --> added UART2 special mode register 3 --> added Page 21 Figure 15 Function select register B3 --> added Page 24 Figure 23 PM1 Note 4 -->added Page 31 Figure 26 Page 45 Table 14, Page 46 Table 15 Note --> added Page 50 Figure 32-4 Changed Page 51 Line 6 port function select register 3 --> function select register A3 Page 52 Line 17 FFFFE4$_{16}$ to FFFFE7$_{16}$ are all --> FFFFE7$_{16}$ is Page 53 Table 17 BRK instruction If the vector is --> If the contents of FFFFE7$_{16}$ is Page 53 Table 18 Instruction fetch and DBC --> delated Page 58 Figure 36 IPL --> RLVL Page 61 Figure 38 004D$_{16}$ --> 0093$_{16}$ Page 67 Figure 44-1Note 3 and 6 --> added Page 68 Figure 45 memory --> memory (forward direction) Page 70 Figure 46-2DMAi memory address reload register Note: vector register (SVP) --> save PC register (SVP) Page 84 Figure 56 Note 4 addresses 0342$_{16}$ and 0343$_{16}$ --> address 0343$_{16}$ Page 93 Table 30 Count source: TBj overflow --> added Page 96 Figure 69 Three-phase PWM control register 0 Note 4:both bit 0 and 1 --> bit 1	'98.3.2

Version	Contents for change	Revision date
REV.C	Three-phase PWM control register 1 Page 100 Line 1 In three-phase --> In "L" active output polarity in three-phase Page 100 Line 26,31 the state of set by port direction register --> the high-impedance state Page 101 Figure 73 Right: INV14 --> added Page 103 Figure 74 Page 108 Table 32 UART4 LSB first/MSB first selection : Note 1 --> Note 2 Page 118 Figure 83 UART transmit/receive control register 2 Page 119 Figure UART 3,4 special mode register 3 --> added Page 126 Line 3 CLK and CLKS select bit (bits 4 and 5 at address 0370_{16}) --> port function select register (bits of related to-$P6_4$ and $P6_5$) Page 145 Page 176 Table 124 P91: STxD3 output --> added P97: STxD4 output --> added Page 178 Figure 125-2 Function select register A3 Page 179 Figure 125-3 Function select register B0 Page 180 Figure 125-4 Function select register B3 Page 187 A-D Converter (5) Page 188 DMAC	'98.3.2
	Page1 Supply voltage 4.0V-5.5V, Mask ROM version is added. Page 5 Table 1.1.1 DMAC 2 channels --> 4 channels Page 8 P0 description is changed Page 9 P6 description is changed P7, 8, 9, 10 equivalent to P0 --> P6 Page 10 Figure 1.2.1 M30800FC, M30803FG are added. Page 18 Figure 1.4.3 (15) DRAM control register 0XXX0000 --> ?XXX???? Page 19 Figure 1.4.4 Delate Note, (143)-(147) 00 --> ?? Page 20 Figure 1.5.1 Add Note Page 25 Figure 1.6.1 Processor mode register 1 When reset 00_{16} --> $C0_{16}$ Page 30 Line 15 ... output to A_9 to A_{20} --> A_8 to A_{20} Page 32 Figure 1.7.2 Page 35 (8) BCLK output Page 38 Figure 1.7.6 Page 39 Figure 1.7.7 Page 40 Figure 1.8.1 and 1.8.2 Note Page 42 Page 43 Figure 1.8.4 Page 44 Figure 1.8.5 Note 2, Line 6 Pin outputs "L" is delated. Page 45 Page 47 Line 15 ... as BCLK --> as BCLK from the interrupt routine Table 1.8.3 Page 48 Status Transition of BCLK Page 51 Figure 1.8.7 Page 52 Line 6, Figure 1.8.6 Delate D-A control register Page 56 Line 14	'98.4.12

Version	Contents for change	Revision date
REV.C	Page 58 Table 1.9.3 Software interrupt number 40,41, Add fault error, Add Note 2 Page 59 Interrupt control register Line 4 delate Page 64 Interrupt sequence (1) Page 66 Saving registers Last line added Page 67 Interrupt Priority *1 delated, Last line added Page 72 (2) Setting the stack pointer Last line added Page 74 Watchdog timer Line 2 A watchdog timer interrupt is generated when --> Whether a watchdog timer interrupt is generated or reset is selected when Last part :Watchdog timer function select bit is initialized only at reset. After reset, watchdog timer interrupt is selected. added Page 75 Figure 75 System clock control register 0 added Page 97 Figure 1.13.9 Count value Page 181 Figure 1.25.4 Page 182 Figure 1.25.5 Page 131 Figure 1.16.12 Both register Note2 added Page 135 Table 1.17.3 RxDi bit 1 and 6 at address 03C7$_{16}$ --> bit 1 and 7 ... Page 132 Table 1.18.3 RxDi bit 1 and 6 at address 03C7$_{16}$ --> bit 1 and 7 ... Page 147 Figure 1.19.1 Upper figure changed, note added Page 153 Bit 4 overflow --> underflow Page 154 Figure 1.20.3 overflow --> underflow Page 159 Clock phase setting UARTi transmission-reception control register 0 ..., whereas UARTi special mode register 3 ... --> Bit 6 of UARTi transmission-reception control register 0 ..., whereas bit 1 of UARTi special mode register 3 ... Line 15 ... output is high impedance. --> ... output is indeterminate. Page 171 Line 3 Set the function select register A to I/O port and the direction register to input mode. added Page 171 Figure 1.22.2 Note delate Page 176 Figure 1.24.3 added Page 178 Figure 1.25.1 When reset --> indeterminate, Note 4 is added. Page 200 Table 1.26.2 and 1.26.3 and Figure 1.26.14 CNVss is added Page 204- Electric characteristics added	
Rev.C1	Page 214 Table 1.28.22 $t_{h(BCLK-DW)}$ add Page 220 Figure 1.28.6 $t_{h(BCLK-CAS)}$ --> $t_{h(BCLK-DW)}$ Page 223 Figure 1.28.9 WR, WRL, WRH(sepalate bus) wave change	99.5.12
Rev.C2	Page 24 Line 3 A software reset has almost the same ... --> A software reset has the same ...	99.5.20
	Page 161 Note 2: When f(X$_{IN}$) is over 10 MHz, the f$_{AD}$ frequency must be under 10 MHz by dividing. --> addition	99.6.4
	Page 18 Figure 1.4.3 (60) Timer B3,4,5 count start flag value change Page 19 Figure 1.4.4 Flash memory control register 0 and 1 added Page 22 Figure 1.5.3 Flash memory control register 0 and 1 added	99.7.6

Version	Contents for change	Revision date
	Page 43 Figure 1.8.4 CM0 Note 5 delate Page 81 Figure 1.11.5 DMAi memory address reload register Address DRA2, DRA3 000000_{16} --> $XXXXXX_{16}$ Page 181, 182 Figures 1.25.4-1.25.5 D0-D15 waveform changed Page 185 (6) Pull up control register changed Page 208 Table 1.28.3 V_{T+}-V_{T-} $TB0_{IN}$-$TB2_{IN}$ --> $TB0_{IN}$-$TB5_{IN}$, $TA2_{OUT}$-$TA4_{OUT}$ --> $TA0_{OUT}$-$TA4_{OUT}$ Page 211 Table 1.28.19 Page 212 Table 1.28.20 Page 213 Table 1.28.21 Page 214 Table 1.28.22 Page 216 Figure 1.28.2 Page 217 Figure 1.28.3 Page 218 Figure 1.28.4 Page 219 Figure 1.28.5 Page 220 Figure 1.28.6 Page 221 Figure 1.28.7 Page 223 Figure 1.28.9	
Rev.C3	Flash memory ROM version added	99.9.24
	Page 2,3 Figure 1.1.1, 1.1.2 Japanese font change to English font	99.12.8
Rev.D	Page 1 • DMAC...4 channels (trigger: 24 sources) --> 31 sources • Supply voltage 4.2 to 5.5V (f(XIN)=20MHz) Flash memory version--> addition • Interrupt...4 software --> 5 software Page 1,5 Table 1.1.1 Feature • Memory capacity ROM 128 Kbytes --> (See ROM expansion figure.) RAM 10K --> 10/20 Kbytes Page 5 Table 1.1.1 Interrupt...4 software --> 5 software Page 2, 3 Figure 1.1.1, 1.1.2 Note 1 addition Page 6 Figure 1.1.4, Table 1.1.2 M30803MG-XXXFP/GP addition Page 7 Figure1.1.5 ROM capacity G:256 Kbytes addition Page 8 P00 to P07 However, it is possible to select pull-up resistance presence to the usable port as I/O port by setting. --> addition CNVss Connect it to the Vss pin when operating in single-chip or memory expansion mode. Connect it to the Vcc pin when in microprocessor mode. --> Connect it to the Vss pin when operating in single-chip or memory expansion mode after reset. Connect it to the Vcc pin when in microprocessor mode after reset. BYTE When operating in single-chip mode,connect this pin to VSS. --> When not using the external bus,connect this pin to VSS. Page 9 P50 to P57 In single chip mode, --> delate Page 10 Figure 1.2.1 M30803FG --> M30803MG/FG Page 13 Figure 1.4.3 (2) processor mode register $C0_{16}$ --> 00_{16} Page 20 to 23 Figure 1.5.1 to 1.5.4 Note addition Page 23 Figure 1.5.4 Note 2 addition	14/3/'00

Version	Contents for change	Revision date
	Page 25 Figure 1.6.1, 1.6.2 Figure 1.6.1 is divided to Figure 1.6.1and 1.6.2 Page 30 Table 1.7.4 Page 34 Figure 1.7.3 Note addition Page 36 Line 3 the chip select control register --> the wait control registe Page 38, 39 Figure 1.7.6, 1.7.7 Note change Page 42 Line 7 addition When the main clock is stoped (bit 5 at address 0006_{16} =1) or the mode is shifted to stop mode (bit 0 at address 0007_{16} =1), the main clock division register (address $000C_{16}$) is set to the divided-8 mode. Page 42 (3)BCLK When shifting to stop mode, --> When main clock is stoped or shifting to stop mode, Page 43 Figure 1.8.4 CM0 Note 6 change, Note 7, 8 addition, CM1 Note 4 addition Page 44 Figure 1.8.5 Note 2 change Page 48 Line 5 When shifting to stop mode and reset, --> When shifting to stop mode, reset or stopping main clock, (12) Low power dissipation mode addition When the main clock is stoped, the main clock division register (address $000C_{16}$) is set to the division by 8 mode. Page 51 Figure 1.8.7. Clock transition Note 3, 4 addition Page 52 Line 9 addition Page 54 Software Interrupts (2) Overflow interrupt, "CMPX" addition Page 55 (2) Peripheral I/O interrupts • Bus collision detection/start, stop condition (UART2, UART3, UART4) interrupts --> change Page 57 • Variable vector tables addition Set an even address to the start address of vector table setting in INTB so that operating efficiency is increased. Page 58 Table 1.9.3 Bus collision detection/start, stop condition interrupts --> Bus collision detection, start/ stop condition detection interrupts Page 58 Table 1.9.3, page 68 Figure 1.9.8 Software interrupt number 40, 41 fault errir --> addition Page 71 Address match interrupt Line 7 addition Page 72 (3) The $\overline{\text{NMI}}$ interrupt • Do not reset the CPU with the input to the $\overline{\text{NMI}}$ pin being in the “L” state. --> • Signal of "L" level width more than 1 clock of CPU operation clock (BCLK) is necessary for $\overline{\text{NMI}}$ pin. Page 72 (4) External interrupt Page 74 Figure 1.10.1 Page 76 Line 2 "DMAC is a function that to transmit 1 data of a source address (8 bits /16 bits) to a destination address when transmission request occurs. " addition. Page 76 Line 12 addition When writing to DSA2 and DSA3, set register bank select flag (B flag) to "1" and use LDC instruction to set SB and FB registers. Page 76 Figure 1.11.1 Page 77 Table 1.11.1 Transfer memory space (16 Mbyte space) --> addition	

Version	Contents for change	Revision date
	Page 78 Figure 1.11.2 Note :6 OR instruction --> OR instruction etc. Page 80 Figure 1.11.4 DRCi • Transfer counter --> • Transfer count register Page 81 Figure 1.11.5 DMAi, DSAi, DRAi Transfer count specification "(16 Mbytes area)" addition DRAi memory address counter --> memory address register Page 85 Line 9 addition (1) Internal factors, (2) External factors change Page 87 Fugure 1.12.1 "Timer B2 overflow" addition Page 88 Fugure 1.12.2 Timer A --> Timer B2 overflow (to timer A count source) Page 93 Table 1.13.2 Cout source • TB2 overflows, TAj overflows --> •TB2 overflows or underflows , TAj overflows or underflows Page 95 Figure 1.13.7 When using two-phase signal processing Note 3 --> addition Page 102 Figure 1.14.3 TBSR When reset 00_{16} --> 000XXXXX$_{16}$ b4-b0 When read, the value is "0" --> indeterminate Page 104 Table 1.14.2 Cout source • TBj overflows --> •TBj overflows or underflows Page 124 Figure 1.16.5 UiTB Note 1 delate Page 126-127 Figure 1.16.7 to 1.16.8 CRD change Page 130 Figure 1.16.11 SDHI Enabled <--> Disabled Page 144 (a) Separate $\overline{\text{CTS}}/\overline{\text{RTS}}$ pins function (UART0) Page 146 Table 1.19.1 Addition in "Other things" Page 147 Figure 1.19.1 è„Figure A "L" level returns from TxD due to the occurrence of a parity error. --> A "L" level returns from SIM card... Page 149 Figure 1.19.4 Note addition Page 150 Table 1.20.1 Note 1: LSB first --> MSB first, Note 3 Change Page 156 Figure 1.20.4 4 to 5 cycles --> 3 to 6 cycles Page 163, 165-169 Figure 1.21.2-Figure 1.21.8 ADCON1 Note 2-6 addition Page 170 Line 14,23 addition Page 171 Line 5 addition Page 172 Figure 1.22.3 Note :3 D-A control register --> D-A register Page 176 Figure 1.24.3 Page 178 Figure 1.25.1 Note 1 position change Page 178 Line 10 DRAM controler --> addition Page 179 Figure 1.25.2 Note 1 --> change Page 184 (1) Direction registers, (2) Port registers --> change Page 185 (4) Function select register B --> change Page 189 Figure 1.26.4 Port Pi direction register Note 2 addition Page 190 Figure 1.26.5 Port Pi register Note 1 and 2 addition Page 191 Table 1.26.1 Note addition Page 192 Figure 1.26.6 Function select register A1 Note 1 addition Page 194 Figure 1.26.8 Function select register B1 Note 2 addition Page 195 Figure 1.26.9 Function select register B3 Note 1 --> addition, PSL3_3-PSL3_6 change Page 196 Figure 1.26.13 Port control register Note 2 addition Page 197 Figure 1.26.4 Port Pi direction register Note 2 addition Page 200 Precaution on A-D converter (6) --> addition Page 203 Stop Mode and Wait Mode (2) all clock stop bits --> all clock stop control bits Page 203 Noise addition Page 203 Precaution on interrupt (1) line 7 --> addition	

Version	Contents for change	Revision date
	Page 204 Making power consumption electricity small --> addition Page 207 Table 1.28.3 V_{T+} – V_{T-} SCL$_2$-SCL$_4$, SDA$_2$-SDA$_4$ Addition Page 208 Table 1.28.5 Note Change Page 215 Table 1.28.22 t_{RP} expression change Page 217-220 Figure 1.28.2-1.28.5 $t_{w(WR)}$ addition, $t_{h(BCLK-DB)}$ delate Page 219, 220, 222, 223, 225 Figure 1.28.4, 1.28.5, 1.28.7, 1.28.8, 1.28.10 addition Page 225, 226 Figure 1.28.10, 1.28.11 $t_{h(BCLK-DB)}$ -5 ns.min --> -7 ns.min Page 227 Figure 1.28.12 Refresh timing (self refresh) RAS timing Page 230 3V of electric characteristics addition Page 246 Table 1.29.1 Data hold --> addition Page 247 Figure 1.29.2 Package type 144P6Q --> 144P6Q-A Page 248 Flash memory line 5 change Page 250 Function outline Line 24 (Parallel ... function) --> delate Page 269 Standard serial I/O mode Line 26 externl device --> external device (programmer) Page 285 Figure1.31.21 programer --> peripheral unit (programmer)	
Rev.D3	Page 43 Figure1.8.4 Note of the system clock control register 0-->addition Page 44 Line 4 Note-->addition Page 45 Table1.8.2 Note-->addition Page 71 Line 9 "Address match interrupt is not generated with a start instruction of interrupt routine."-->Delete Page 73 (6) Precaution of Address mach interrupt-->addition Page 79 Figure1.11.2 Note-->change Page 87 Precaution for DMAC-->addition Page 131 Figure1.16.11 Bit 7-->Must set to "1" in selecting IIC mode. Page 152 Figure1.20.1 Bit 7-->Must set to "1" in selecting IIC mode. Page 182 Addition Page 205 (3) Address match interrupt in Interrupt precautions-->addition Page 206 (2) DMAC-->addition Page 207 Precautions for using CLK$_{OUT}$ pin-->addition Page 210 Table1.28.3 Icc when clock stop Topr=25C^{o}-->change Page 212 Table1.28.6 External clock input HIGH and LOW pulse waidth 22-->20 External clock rise and fall time 10-->5 Page 215, 216 Table1.28.19, 20 $t_{h(BCLK-DB)}$-->delete, $t_{w(WR)}$-->addition Page 218 Table1.28.22 $t_{h(BCLK-DB)}$ -5ns --> -7ns Page 233 Table1.28.23 Icc when clock stop Topr=25C^{o}-->change Page 235 Table1.28.27 $t_{h(CAS-DB)}$-->addition Page 238, 239 Table1.28.39, 40 $t_{w(WR)}$ -->addition, $t_{h(BCLK-RD)}$ 0ns-->-3ns Page 240 Table1.28.41 $t_{d(AD-ALE)}$=10^{9}/(f$_{(BCLK)}$X2)-20 -->10^{9}/(f$_{(BCLK)}$X2)-27 Page 241 Table1.28.42 $t_{h(BCLK-CAS)}$ 0ns-->-3ns Page 242 Figure1.28.15 $t_{ac1(RD-DB)}$ min-->max, $t_{ac1(AD-DB)}$ min-->max Page 243 Figure1.28.16 $t_{ac2(RD-DB)}$ min-->max, $t_{ac2(AD-DB)}$ min-->max Page 244, 255 Figure1.28.17 2 wait, Figure1.28.18 3 wait-->addition Page 246 Figure1.28.19 $t_{ac3(AD-DB)}$-->addition, $t_{su(DB-RD)}$-->$t_{su(DB-BCLK)}$, $t_{h(BCLK-RD)}$ 0ns -->-3ns, $t_{d(AD-ALE)}$=(tcyc/2-20)ns--> ... -27)ns Page 247 Figure1.28.20 Addition Page 248, 249 Figure1.28.21, 1.28.22 -->addition	19/6/'00

Version	Contents for change	Revision date
	Page 250 Figure1.28.23 $t_{h(BCLK-DB)}$-->$t_{h(CAS-DB)}$ Page 251 Figure 1.28.24 $t_{d(DB-CAS)}$-->$t_{su(DB-CAS)}$, $t_{h(BCLK-CAS)}$-->$t_{h(BCLK-DB)}$ Page 252 Figure1.28.25 $t_{d(CAS-RAS)}$-->$t_{su(CAS-RAS)}$ Page 255 Table1.29.1 Power supply (under planning)-->delete, Program/erase voltage f(X_{IN})-->f($_{BCLK}$), 2.7V-5.5V-->delete	
Rev.E	144-pin version description addition Pages 1, 6 •Supply voltage --> external ROM version addition Page 7 (3) Package 144P6Q --> 144P6Q-A Page 21 Figure 1.4.4 (111) Function select register C 00_{16} --> 0XXXXXX0 (119) Function select register B3 ?0000??? --> 00000X0X Similarly, page 202 Figures 1.26.11 When reset ?0000??? --> 00000X0X Figures 1.26.12 When reset 00_{16} --> 0XXXXXX0 Page 28 Figure 1.6.2 ROMless version addition Page 29 Figure 1.6.3 External area 0 to 3 addition Page 34 Addition Page 37 Figure 1.7.4 Input $\overline{\text{RDY}}$ signal at i + 1 cycles for i wait --> $\overline{\text{RDY}}$ signal received timing for i wait: i +1 Page 46 Figure 1.8.4 System clock control register 0 CM0 --> contents of the Function changed, Notes 10, 11 addition Page 48 On the second line from the bottom, 'Although stop mode ... must be set to "1".' -->addition Page 49 Table 1.8.4 CS0 to CS3 --> $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\underline{\text{BHE}}}$ $\overline{\text{WR}}$, $\overline{\underline{\text{BHE}}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\underline{\text{W}}}$, $\overline{\text{CASL}}$ --> $\overline{\text{WR}}$, $\overline{\text{WRL}}$, $\overline{\text{WRH}}$, $\overline{\underline{\text{DW}}}$, $\overline{\text{CASL}}$ Page 52 Table 1.8.6 CM0i: Clock control register 0 (address 0006_{16}) bit i, MCDi: Main clock division register (address $000C_{16}$) bit i --> addition Page 60 • Vector table dedicated for emulator Interrupt vector address (address 000020_{16} to 000023_{16})-->... (address 000020_{16} to 000022_{16}) Page 69 Interrupt priority 'the interrupt that a request came to most in the first place is accepted at first, and then,' --> delete Page 75 (6) Explanation of No.1 and No. 2 are partly revised. Page 76, 77 From "• To return from an interrupt..." to the end of page 77 --> addition Page 78 "In the stop...released." on the third line from the bottom --> addition Page 79 Figure 1.10.2 Notes 10, 11 --> addition Page 87 Figure 1.11.6 is partly revised. Page 88 Table for "Coefficient j, k" is partly revised. Page 89 Figure 1.11.7 is partly revised. Page 90 Explanation of (3) is partly revised. Page 94 Figure 1.13.3 Timer Ai register -->Notes 2 to 4 addition, •Pulse width modulation mode (8-bit PWM) --> Values that can be set is changed, Up/down flag --> Note addition Page 97 Figure 1.13.6 --> change Page 98 Table 1.13.3 --> Note 2 addition, •Normal processing operation --> •Normal processing operation (Timer A2 and timer A3), •Multiply-by-4 processing operation --> •Multiply-by-4 processing operation (Timer A3 and timer A4) Page 99 Figure 1.13.7 Timer Ai mode register (When using two-phase pulse signal process-	09/02/'01

Version	Contents for change	Revision date
	ing) --> "Note 2 Timer A2 is fixed to ... multiply-by-4 processing operation." is added, note 3 change Page 109 Figure 1.14.6 Note 1 It is indeterminate when reset --> addition Page 112 Figure 1.15.2 Dead time timer-->Note 1 addition, Timer B2 interrupt occurrence frequency set counter-->Note 3 addition Page 113 Notes 2, 3 --> addition Page 114 three-phase waveform mode --> three phase PWM output mode Page 128 Figure 1.16.5 UARTi bit rate generator --> Note 2 addition Page 128 Figure 1.16.5 UARTi transmit buffer register, UARTi bit rate generator-->Note 1 addition Page 129 Figure 1.16.6 UARTi transmit/receive mode register-->Note 2 addition in CKDIR of UART mode Page 133 Figure 1.16.10 UART transmit/receive control register 2-->Note delete Page 136 Note 2, Page 143 Note 3 ... the UARTi receive interrupt request bit is not set to "1" --> ... the UARTi receive interrupt request bit will not change Page 145 Figure 1.18.1 UARTi transmit/receive mode register (i=0,1) --> Note 1 addition, UARTi transmit/receive mode register (i=2 to 4) --> Note 2 addition Page 157 On the 12th line, ... allocated to bit 3 in UART2 transmission buffer register 1 (address 033F$_{16}$) ... --> ... allocated to bit 11 in UART2 transmission buffer register (address 033E$_{16}$) ... Page 161 < Master Mode (TxDi and RxDi are selected, DINC = 0) > ..., and the STxDi, SRxDi and CLKi pins ...--> ..., and the TxDi, RxDi and CLKi pins ... Page 165 Table 1.21.1 Absolute precision --> change Page 170 Table 1.21.3 Reading of result of A-D converter --> (at any time) addition Page 173 Table 1.21.6 Input pin --> change to AN$_0$ to AN$_7$, With emphasis on the pin --> addition Page 182 On the second line from the bottom, ..., and dummy cycle for refresh ... --> ..., and processing necessary for dummy cycle to refresh DRAM ... Page 183 Figure 1.25.2 is partly revised. Page 189 On the 18th and 27th lines, page 194 Port Pi direction register Note 2, page 196 Port Pi register Note 1 ... for setting of bus control such as address bus and data bus is ... --> of pins A$_0$ to A$_{22}$, $\overline{\text{A}_{23}}$, D$_0$ to D$_{15}$, MA$_0$ to MA$_{12}$, $\overline{\text{CS0}}$ to $\overline{\text{CS 3}}$, $\overline{\text{WRL}}$/$\overline{\text{WR}}$/$\overline{\text{CASL}}$, $\overline{\text{WRH}}$/$\overline{\text{BHE}}$/$\overline{\text{CASH}}$, $\overline{\text{RD}}$/$\overline{\text{DW}}$, BCLK/ALE/CLK$_{\text{OUT}}$, $\overline{\text{HLDA}}$/ALE, $\overline{\text{HOLD}}$, ALE/$\overline{\text{RAS}}$, and $\overline{\text{RDY}}$ are ... Page 203 Figure 1.26.13 is partly revised. Page 207, 208 Timer A (event counter mode) --> (3) addition, Timer A (one-shot timer mode) --> (2) changes to (3), (2) and (4) addition Page 209 Timer B (pulse period/pulse width measurement mode) --> (3) addition Page 212 to 214 (2) $\overline{\text{NMI}}$ interrupt •The $\overline{\text{NMI}}$ pin also serves as P8$_5$, ... •Signal of "L" level ... --> addition (3) Address match interrupt From "• To return from an interrupt..." to "; Interrupt completed" on page 77 --> addition (4) External interrupt, (5) Rewrite the interrupt control register --> addition Page 215 Explanation of (3) is partly revised. Page 215 $\overline{\text{HOLD}}$ signal --> addition Page 216 DRAM controller --> addition	

Version	Contents for change	Revision date
	Page 217 Setting the registers, Notes on the microprocessor mode ... single-chip mode -->addition Page 217 Explanation of note on Flash memroy version is added. Page 219 Note 2 80mA --> –80mA Page 220 Table 1.28.3 Ta --> Topr, Note2 --> addition Pages 220, 243 Tables 1.28.3, 1.28.23 Icc Power supply current ROMless version --> addition, Ta --> Topr, Note 2 --> addition Page 227 Calculation for td(AD-ALE) is partly revised. Page 234, 235 Figures 1.28.6 and 1.28.7 Timing for td(AD-ALE) is partly revised. Page 243 Table 1.28.23 Topr=25°C, when clock is stopped: 2.0µA --> 1.0µA, Notes 1, 2 --> addition Page 250 Table 1.28.41 th(BCLK-RD) Min. 0 ns --> –3 ns Pages 251, 258 to 262 Table 1.28.42, figures 1.28.21 to 1.28.25 th(BCLK-CAS): –3 ns --> 0 ns Pages 251, 259, 261 Table 1.28,42, figures 1.28.22, 1.28.24 th(BCLK-DW): 0 ns --> –3 ns Page 266 Table 1.29.2 M30805FGGP RAM capacity 24 Kbytes --> 20 Kbytes Page 270 Figure 1.30.1 Flash memory control register 0 Note 1 Also write to this bit ... "H" level. --> addition Page 273 (3) Disabling erase or ... serial I/O mode --> delete, (7), (8) --> addition Page 285 Note --> addition Page 288 --> addition Pages 291, 307 Tables 1.31.1, 1.31.5 Note 2 ... status register 1 data --> ... status register data 1 Page 319 144P6Q-A version --> addition	
Rev.E1	Page 32 Table 1.7.3 --> change	16/03/'01
Rev.E2	Page 28 Figure 1.6.1 Note 8 --> addition Page 88 Table for "Coefficient j, k" is partly revised.	10/05/'01

Keep safety first in your circuit designs!

- Mitsubishi Electric Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage. Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of non-flammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

- These materials are intended as a reference to assist our customers in the selection of the Mitsubishi semiconductor product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Mitsubishi Electric Corporation or a third party.
- Mitsubishi Electric Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
- All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Mitsubishi Electric Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor for the latest product information before purchasing a product listed herein.

  The information described here may contain technical inaccuracies or typographical errors. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.

  Please also pay attention to information published by Mitsubishi Electric Corporation by various means, including the Mitsubishi Semiconductor home page (http://www.mitsubishichips.com).
- When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Mitsubishi Electric Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
- Mitsubishi Electric Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semiconductor product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
- The prior written approval of Mitsubishi Electric Corporation is necessary to reprint or reproduce in whole or in part these materials.
- If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.

  Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
- Please contact Mitsubishi Electric Corporation or an authorized Mitsubishi Semicon ductor product distributor for further details on these materials or the products con tained therein.

MITSUBISHI SEMICONDUCTORS

M16C/80 Group DATA SHEET REV. E2

May First Edition 2001

Editioned by
Committee of editing of Mitsubishi Semiconductor DATA SHEET

Published by
Mitsubishi Electric Corp., Kitaitami Works