

About M32C/83 Group

The M32C/83 group of single-chip microcomputers are built using a high-performance silicon gate CMOS process uses a M32C/80 Series CPU core and are packaged in a 144-pin and 100-pin plastic molded QFP. 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.

Applications

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

Index

About M32C/83 Group	1	Three-phase motor control timers' functions	155
Central Processing Unit (CPU)	20	Serial I/O	168
Reset	24	CAN Module	198
SFR	37	Intelligent I/O	235
Software Reset	48	Base timer (group 0 to 3)	240
Processor Mode	48	Time measurement (group 0 and 1)	247
Bus Settings	52	WG function (group 0 to 3)	252
Bus Control	55	Serial I/O (group 0 to 2)	264
System Clock	65	A-D Converter	281
Power Saving	76	D-A Converter	296
Protection	81	CRC Calculation Circuit	298
Interrupt Outline	83	X-Y Converter	300
$\overline{\text{INT}}$ Interrupts	98	DRAM Controller	303
$\overline{\text{NMI}}$ Interrupt	99	Programmable I/O Ports	310
Key Input Interrupt	99	VDC	334
Address Match Interrupt	100	Usage Precaution	335
Intelligent I/O and CAN Interrupt	101	Electrical characteristics	344
Precautions for Interrupts	104	Outline Performance	381
Watchdog Timer	106	Flash Memory	383
DMAC	109	CPU Rewrite Mode	384
DMAC II	121	Outline Performance of CPU Rewrite Mode	384
Timer	129	Inhibit Rewriting Flash Memory Version	397
Timer A	131	Parallel I/O Mode	399
Timer B	147	Standard serial I/O mode	400

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.

Performance Outline

Table 1.1.1 and 1.1.2 are performance outline of M32C/83 group.

Table 1.1.1. Performance outline of M32C/83 group (144-pin version) (1/2)

Item		Performance
CPU	Number of basic instructions	108 instructions
	Shortest instruction execution time	33 ns($f(XIN)=30MHz$)
	Operation mode	Single-chip, memory expansion and microprocessor modes
	Memory space	16 M bytes
	Memory capacity	See ROM/RAM expansion figure.
Peripheral function		
	I/O port	123 pins (P0 to P15 except P85)
	Input port	1 pin (P85)
Multifunction timer	Output	16 bits x 5 (TA0, TA1, TA2, TA3, TA4)
	Input	16 bits x 6 (TB0, TB1, TB2, TB3, TB4, TB5)
Intelligent I/O		4 groups
	Time measurement	8 channels (group 0) + 4 channels (group 1)
	Waveform generation	4 channels (group 0) + 8 channels X 3 (group 1, 2 and 3)
	Bit-modulation PWM	8 channels X 2 (group 2 and 3)
	Real time port	8 channels X 2 (group 2 and 3)
	Communication function	<ul style="list-style-type: none"> • Clock synchronous serial I/O, UART (group 0 and 1) • HDLC data process (group 0 and 1) • Clock synchronous variable length serial I/O (group 2) • IE bus ^(Note 1) (group 2)
Serial I/O		5 channels (UART0 to UART4) IE Bus ^(Note 1, 3) , I ² C Bus ^(Note 2, 3)
CAN module		1 channel, 2.0B specification
A-D converter		10-bit A-D x 2 circuits, standard 18 inputs, max 34 inputs
D-A converter		8-bit D-A x 2 circuits
DMAC		4 channels
DMAC II		Start by all variable vector interrupt factor Immediate transfer, operation transfer and chain transfer function
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		42 internal and 8 external sources, 5 software sources, interrupt priority level 7 levels
Clock generating circuit		3 built-in clock generation circuits <ul style="list-style-type: none"> • Main/sub-clock generating circuit :built-in feedback resistance, and external ceramic or quartz oscillator • Ring oscillator for detecting main clock oscillation stop

Table 1.1.1. Performance outline of M32C/83 group (144-pin version) (2/2)

Electric characteristics	
Supply voltage	4.2 to 5.5V (f(XIN)=30MHz without wait), 3.0 to 3.6V (f(XIN)=20MHz without wait)
Power consumption	26mA (f(XIN)=20MHz without software wait, Vcc=5V) 38mA (f(XIN)=30MHz without software wait, Vcc=5V)
I/O characteristics	I/O withstand voltage :5V I/O current :5mA
Operating ambient temperature	-40 to 85°C
Device configuration	CMOS high performance silicon gate
Package	144-pin plastic mold QFP

Note 1 :IE Bus is a trademark of NEC corporation.

Note 2 :I²C Bus is a registered trademark of Philips.

Note 3 :This function is executed by using software and hardware.

Table 1.1.2. Performance outline of M32C/83 group (100-pin version) (1/2)

Item		Performance
CPU	Number of basic instructions	108 instructions
	Shortest instruction execution time	33 ns (f(XIN)=30MHz)
	Operation mode	Single-chip, memory expansion and microprocessor modes
	Memory space	16 M bytes
	Memory capacity	See ROM/RAM expansion figure.
Peripheral function		
I/O port		87 pins (P0 to P10 except P8s)
Input port		1 pin (P8s)
Multifunction timer	Output	16 bits x 5 (TA0, TA1, TA2, TA3, TA4)
	Input	16 bits x 6 (TB0, TB1, TB2, TB3, TB4, TB5)
Intelligent I/O		4 groups
Time measurement		3 channels (group 0) + 2 channels (group 1)
Waveform generation		2 channels X 2 (group 0 and 3) + 3 channels X 2 (group 1 and 2)
Bit-modulation PWM		3 channels (group 2) + 2 channels (group 3)
Real time port		3 channels (group 2) + 2 channels (group 3)
Communication function		<ul style="list-style-type: none"> • Clock synchronous serial I/O, UART (group 0 and 1) • HDLC data process (group 0 and 1) • Clock synchronous variable length serial I/O (group 2) • IE bus (Note 1) (group 2)
Serial I/O		5 channels (UART0 to UART4) IE Bus (Note 1, 3), I ² C Bus (Note 2, 3)
CAN module		1 channel, 2.0B specification
A-D converter		10 bits A-Dx 2 circuits, standard 10 inputs, max 26 inputs
D-A converter		8 bits D-A x 2 circuits
DMAC		4 channels
DMAC II		Start by all variable vector interrupt factor Immediate transfer, operation function and chain transfer function
DRAM controller		CAS before RAS refresh, self-refresh, EDO, FP
CRC calculation circuit		CRC-CCITT

Table 1.1.2. Performance outline of M32C/83 group (100-pin version) (2/2)

	X-Y converter	16 bits X 16 bits
	Watchdog timer	15 bits x 1 (with prescaler)
	Interrupt	42 internal and 8 external sources, 5 software sources, interrupt priority level 7 levels
	Clock generating circuit	3 built-in clock generation circuits <ul style="list-style-type: none"> • Main/sub-clock generating circuit :built-in feedback resistance, and external ceramic or quartz oscillator • Ring oscillator for detecting main clock oscillation stop
Electric characteristics		
	Supply voltage	4.2 to 5.5V (f(XIN)=30MHz without wait), 3.0 to 3.6V (f(XIN)=20MHz without wait)
	Power consumption	26mA (f(XIN)=20MHz without software wait, Vcc=5V) 38mA (f(XIN)=30MHz without software wait, Vcc=5V)
	I/O characteristics	I/O withstand voltage :5V I/O current :5mA
Operating ambient temperature		-40 to 85°C
Device configuration		CMOS high performance silicon gate
Package		100-pin plastic mold QFP

Note 1 :IE Bus is a trademark of NEC corporation.

Note 2 :I²C Bus is a registered trademark of Philips.

Note 3 :This function is executed by using software and hardware.

Mitsubishi plans to release the following products in the M32C/83 group:

- (1) Support for mask 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)

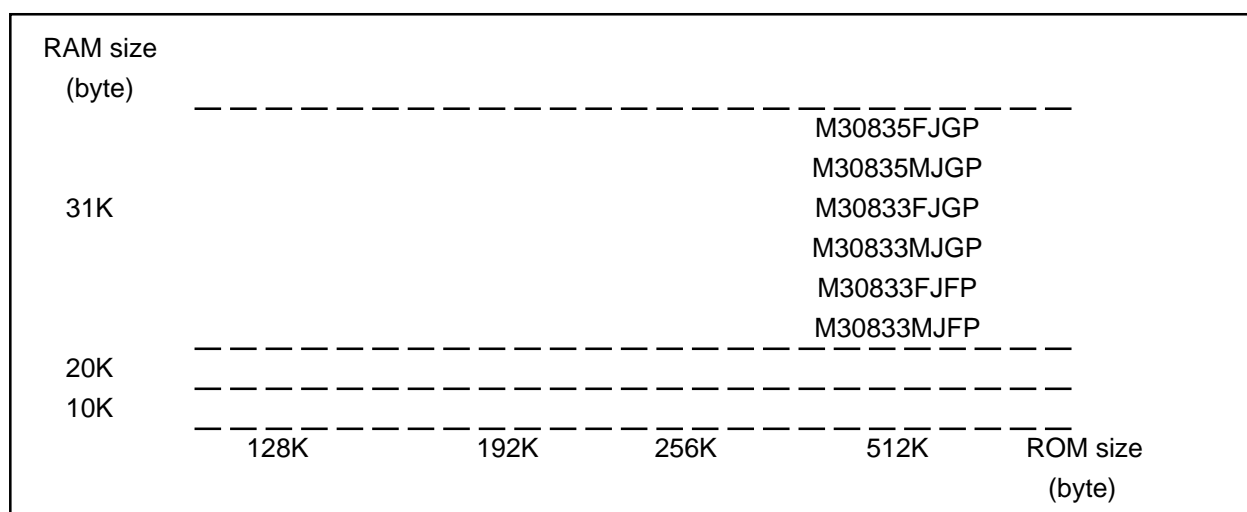


Figure 1.1.1. ROM expansion

Description

The M32C/83 group products currently supported are listed in Table 1.1.3.

Table 1.1.3. M32C/83 group

As of Nov. 2001

Type No	ROM capacity	RAM capacity	Package type	Remarks
M30835MJGP ***	512K	31K	144P6Q-A	Mask ROM version
M30833MJGP ***			100P6Q-A	
M30833MJFP ***			100P6S-A	
M30835FJGP **			144P6Q-A	Flash memory version
M30833FJGP **			100P6Q-A	
M30833FJFP **			100P6S-A	

** :Under development

*** :Under planning

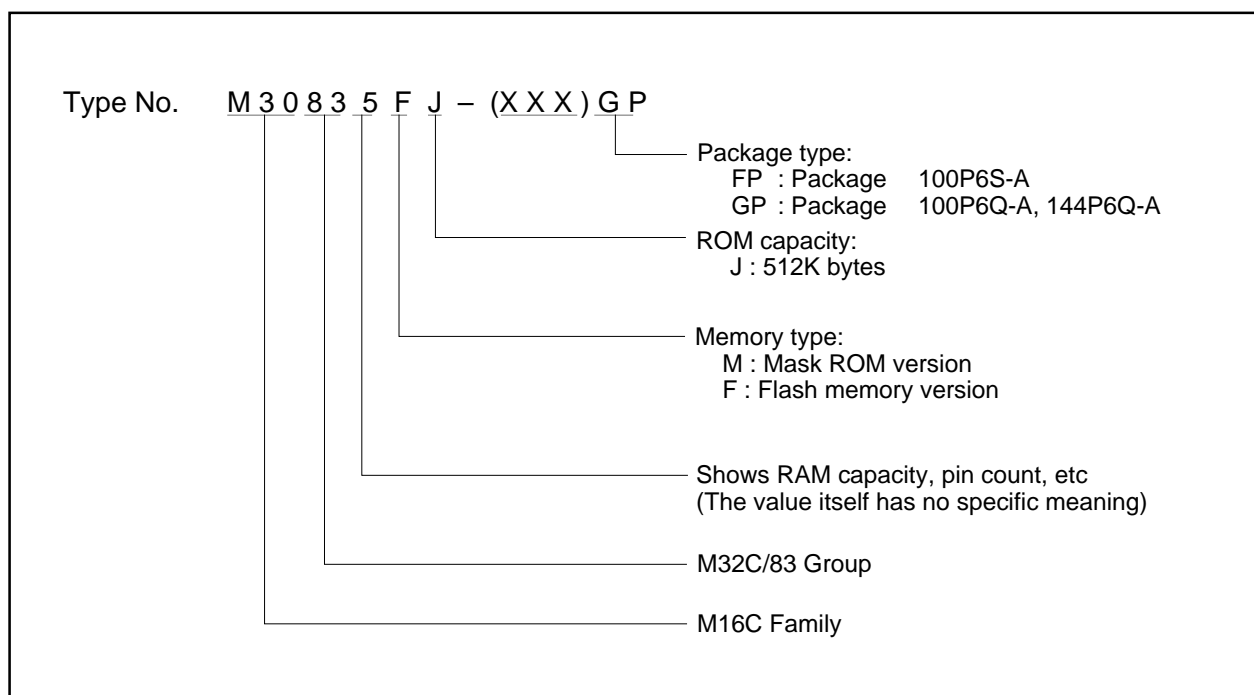


Figure 1.1.2. Type No., memory size, and package

Description

Table 1.1.4. 144-pin version pin description (1/3)

Pin No	Control	Port	Interrupt	Timer	UART/CAN	Intelligent I/O	Analog	Bus control
1		P96			TxD4/SDA4/SRxD4		ANEX1	
2		P95			CLK4		ANEX0	
3		P94		TB4IN	CTS4/RTS4/SS4		DA1	
4		P93		TB3IN	CTS3/RTS3/SS3		DA0	
5		P92		TB2IN	TxD3/SDA3/SRxD3	OUTC20/IEOUT		
6		P91		TB1IN	RxD3/SCL3/STxD3	IEIN		
7		P90		TB0IN	CLK3			
8		P146						
9		P145						
10		P144						
11		P143				INPC17/OUTC17		
12		P142				INPC16/OUTC16		
13		P141				OUTC15		
14		P140				OUTC14		
15	BYTE							
16	CNVss							
17	Xcin/Vcont	P87						
18	Xcout	P86						
19	RESET							
20	Xout							
21	Vss							
22	Xin							
23	Vcc							
24		P85	NMI					
25		P84	INT2					
26		P83	INT1		CANin			
27		P82	INT0		CANout	OUTC32		
28		P81		TA4in/U		OUTC30		
29		P80		TA4out/U		INPC02/ISRxD0/BE0in		
30		P77		TA3in	CANin	INPC01/OUTC01/ISCLK0		
31		P76		TA3out	CANout	INPC00/OUTC00/ISTxD0/BE0out		
32		P75		TA2in/W		INPC12/OUTC12/ISRxD1/BE1in		
33		P74		TA2out/W		INPC11/OUTC11/ISCLK1		
34		P73		TA1in/V	CTS2/RTS2/SS2	OUTC10/ISTxD1/BE1out		
35		P72		TA1out/V	CLK2			
36		P71		TB5in/TA0in	RxD2/SCL2/STxD2	OUTC22/ISRxD2/IEin		
37		P70		TA0out	TxD2/SDA2/SRxD2	OUTC20/ISTxD2/IEout		
38		P67			TxD1/SDA1/SRxD1			
39	Vcc							
40		P66			RxD1/SCL1/STxD1			
41	Vss							
42		P65			CLK1			
43		P64			CTS1/RTS1/SS1	OUTC21/ISCLK2		
44		P63			TxD0/SDA0/SRxD0			
45		P62			RxD0/SCL0/STxD0			
46		P61			CLK0			
47		P60			CTS0/RTS0/SS0			
48		P137				OUTC27		

Description

Table 1.1.5. 144-pin version pin description (2/3)

Pin No	Control	Port	Interrupt	Timer	UART/CAN	Intelligent I/O	Analog	Bus control
49		P136				OUTC21/ISCLK2		
50		P135				OUTC22/ISRxD2/IEIN		
51		P134				OUTC20/ISTxD2/IEOUT		
52		P57						RDY
53		P56						ALE/RAS
54		P55						HOLD
55		P54						HLDA/ALE
56		P133				OUTC23		
57	Vss							
58		P132				OUTC26		
59	Vcc							
60		P131				OUTC25		
61		P130				OUTC24		
62		P53						CLKOUT/BCLK/ALE
63		P52						RD/DW
64		P51						WRH/BHE/CASH
65		P50						WRL/WR/CASL
66		P127				OUTC37		
67		P126				OUTC36		
68		P125				OUTC35		
69		P47						CS0/A23
70		P46						CS1/A22
71		P45						CS2/A21
72		P44						CS3/A20(MA12)
73		P43						A19(MA11)
74	Vcc							
75		P42						A18(MA10)
76	Vss							
77		P41						A17(MA9)
78		P40						A16(MA8)
79		P37						A15(MA7)/(D15)
80		P36						A14(MA6)/(D14)
81		P35						A13(MA5)/(D13)
82		P34						A12(MA4)/(D12)
83		P33						A11(MA3)/(D11)
84		P32						A10(MA2)/(D10)
85		P31						A9(MA1)/(D9)
86		P124				OUTC34		
87		P123				OUTC33		
88		P122				OUTC32		
89		P121				OUTC31		
90		P120				OUTC30		
91	Vcc							
92		P30						A8(MA0)/(D8)
93	Vss							
94		P27					AN37	A7/(D7)
95		P26					AN36	A6/(D6)
96		P25					AN35	A5/(D5)

Description

Table 1.1.6. 144-pin version pin description (3/3)

Pin No	Control	Port	Interrupt	Timer	UART/CAN	Intelligent I/O	Analog	Bus control
97		P24					AN24	A4(/D4)
98		P23					AN23	A3(/D3)
99		P22					AN22	A2(/D2)
100		P21					AN21	A1(/D1)
101		P20					AN20	A0(/D0)
102		P17	INT5					D15
103		P16	INT4					D14
104		P15	INT3					D13
105		P14						D12
106		P13						D11
107		P12						D10
108		P11						D9
109		P10						D8
110		P07					AN07	D7
111		P06					AN06	D6
112		P05					AN05	D5
113		P04					AN04	D4
114		P114						
115		P113				OUTC13		
116		P112				INPC12/OUTC12/ISRxD1/BE1IN		
117		P111				INPC11/OUTC11/ISCLK1		
118		P110				OUTC10/ISTxD1/BE1oUT		
119		P03					AN03	D3
120		P02					AN02	D2
121		P01					AN01	D1
122		P00					AN00	D0
123		P157				INPC07	AN157	
124		P156				INPC06	AN156	
125		P155				INPC05/OUTC05	AN155	
126		P154				INPC04/OUTC04	AN154	
127		P153				INPC03	AN153	
128		P152				INPC02/ISRxD0/BE0IN	AN152	
129		P151				INPC01/OUTC01/ISCLK0	AN151	
130	Vss							
131		P150				INPC00/OUTC00/ISTxD0/BE0oUT	AN150	
132	Vcc							
133		P107	KI3				AN7	
134		P106	KI2				AN6	
135		P105	KI1				AN5	
136		P104	KI0				AN4	
137		P103					AN3	
138		P102					AN2	
139		P101					AN1	
140	AVss							
141		P100					AN0	
142	VREF							
143	AVcc							
144		P97			RxD4/SCL4/STxD4		ADTRG	

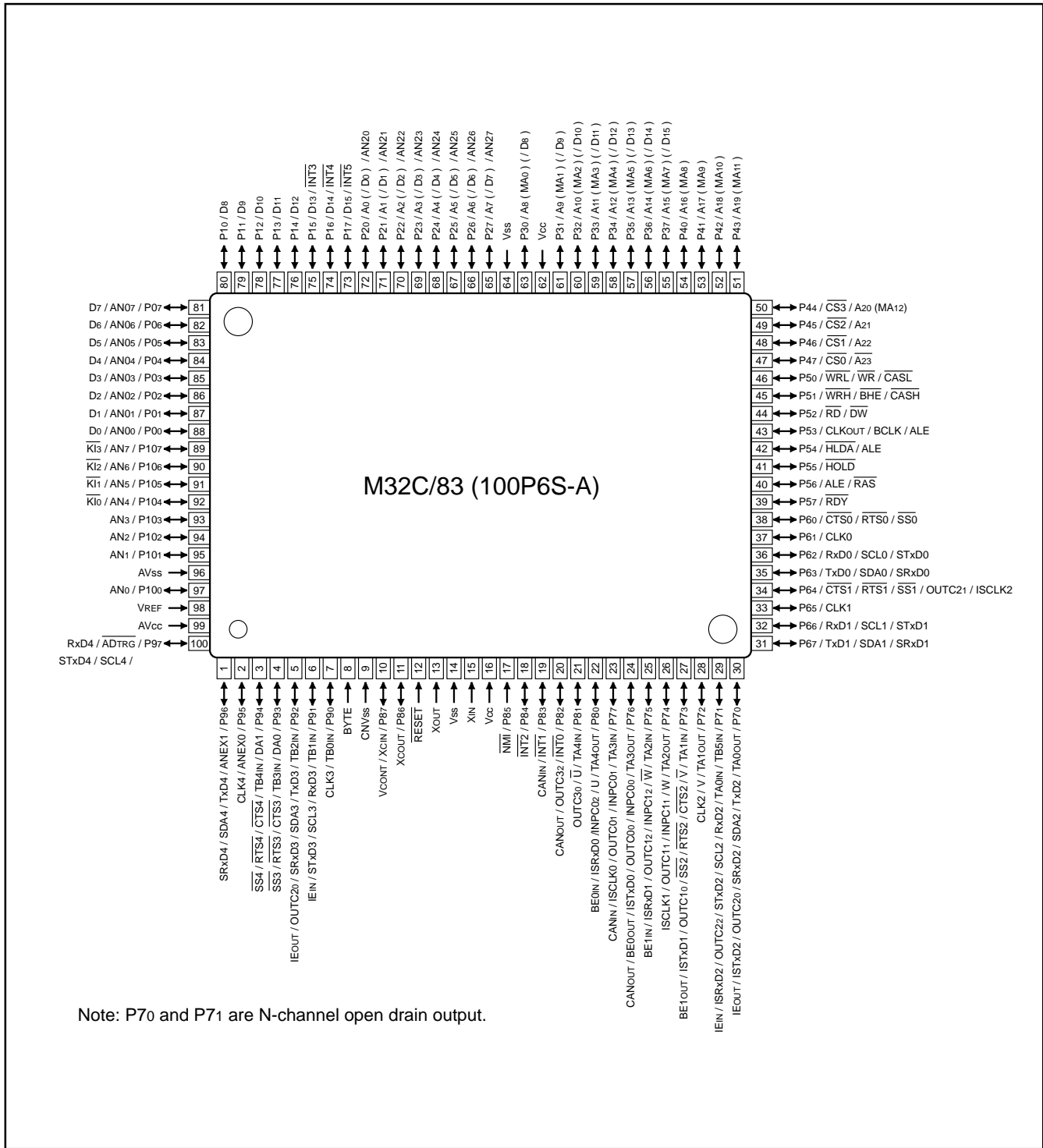


Figure 1.1.4. 100-pin version pin configuration (top view)

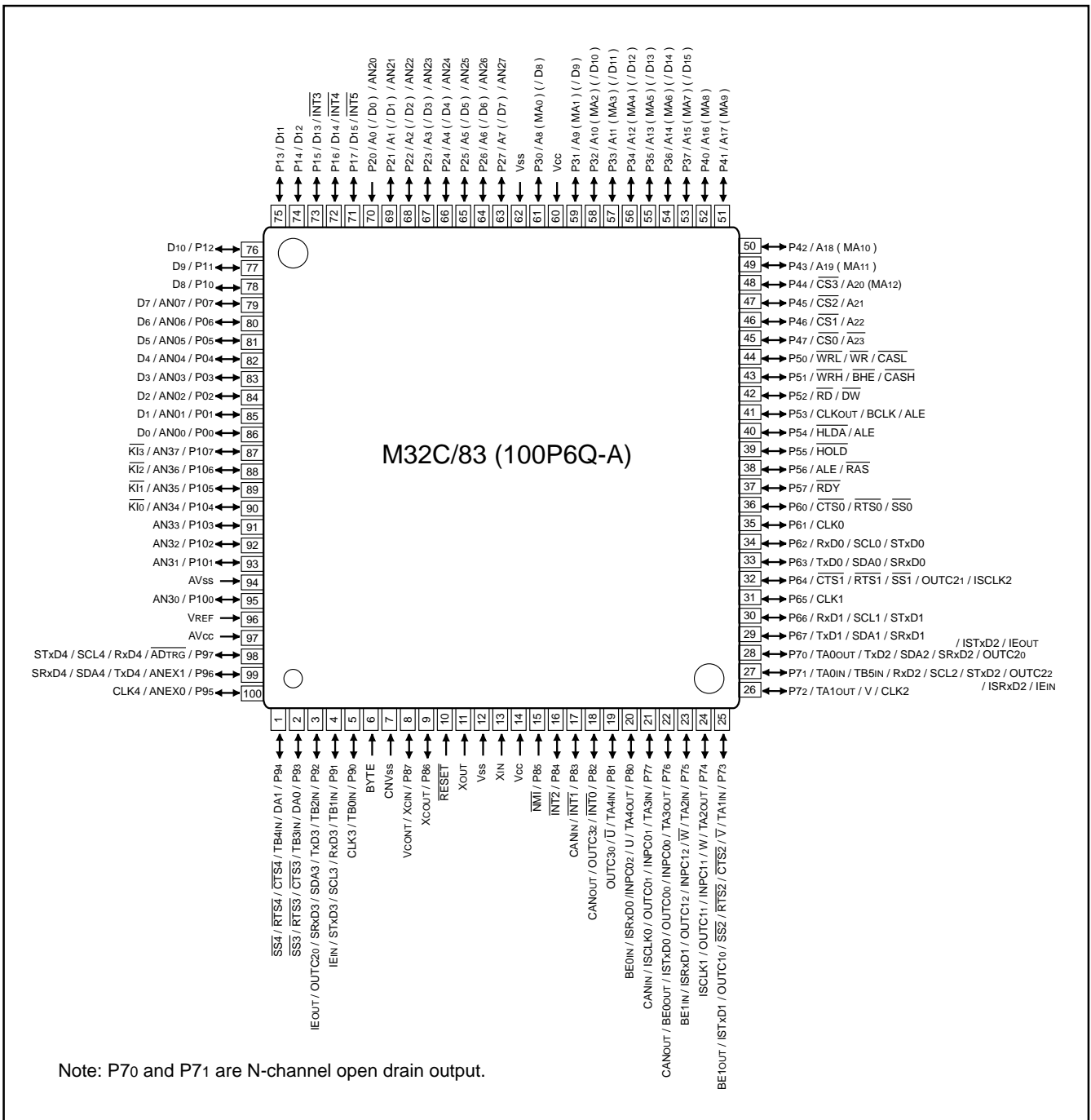


Figure 1.1.5. 100-pin version pin configuration (top view)

Table 1.1.7. 100-pin version pin description (1/2)

Package Pin No		Control	Port	Interrupt	Timer	UART/CAN	Intelligent I/O	Analog	Bus control
FP	GP								
1	99		P9 ₆			TxD4/SDA4/SRx4D4		ANEX1	
2	100		P9 ₅		TB4IN	CLK4		ANEX0	
3	1		P9 ₄		TB3IN	CTS4/RTS4/SS4		DA1	
4	2		P9 ₃		TB2IN	CTS3/RTS3/SS3		DA0	
5	3		P9 ₂		TB1IN	TxD3/SDA3/SRx3D3	OUTC2 ₀ /IEOUT		
6	4		P9 ₁		TB0IN	RxD3/SCL3/STxD3	IEIN		
7	5		P9 ₀			CLK3			
8	6	BYTE							
9	7	CNV _{SS}							
10	8	X _{CIN} /V _{CONT}	P8 ₇						
11	9	X _{COU} T	P8 ₆						
12	10	RESET							
13	11	X _{OUT}							
14	12	V _{SS}							
15	13	X _{IN}							
16	14	V _{CC}							
17	15		P8 ₅	NMI					
18	16		P8 ₄	INT ₂					
19	17		P8 ₃	INT ₁		CANIN			
20	18		P8 ₂	INT ₀	TA4IN/U	CANOUT	OUTC3 ₂		
21	19		P8 ₁		TA4OUT/ \bar{U}		OUTC3 ₀		
22	20		P8 ₀		TA3IN		INPC0 ₂ /ISRxD0/BE0IN		
23	21		P7 ₇		TA3OUT	CANIN	INPC0 ₁ /OUTC0 ₁ /ISCLK0		
24	22		P7 ₆		TA2IN/W	CANOUT	INPC0 ₀ /OUTC0 ₀ /ISTxD0/BE0OUT		
25	23		P7 ₅		TA2OUT/ \bar{W}		INPC1 ₂ /OUTC1 ₂ /ISRxD1/BE1IN		
26	24		P7 ₄		TA1IN/V		INPC1 ₁ /OUTC1 ₁ /ISCLK1		
27	25		P7 ₃		TA1OUT/ \bar{V}	CTS2/RTS2/SS2	OUTC1 ₀ /ISTxD1/BE1OUT		
28	26		P7 ₂		TB5IN/TA0IN	CLK2			
29	27		P7 ₁		TA0OUT	RxD2/SCL2/STxD2	OUTC2 ₂ /ISRxD2/IEIN		
30	28		P7 ₀			TxD2/SDA2/SRx2D2	OUTC2 ₀ /ISTxD2/IEOUT		
31	29		P6 ₇			TxD1/SDA1/SRx1D1			
32	30		P6 ₆			RxD1/SCL1/STxD1			
33	31		P6 ₅			CLK1			
34	32		P6 ₄			CTS1/RTS1/SS1	OUTC2 ₁ /ISCLK2		
35	33		P6 ₃			TxD0/SDA0/SRx0D0			
36	34		P6 ₂			RxD0/SCL0/STxD0			
37	35		P6 ₁			CLK0			
38	36		P6 ₀			CTS0/RTS0/SS0			
39	37		P5 ₇						RDY
40	38		P5 ₆						ALE/RAS
41	39		P5 ₅						HOLD
42	40		P5 ₄						HLDA/ALE
43	41		P5 ₃						CLKOUT/BCLK/ALE
44	42		P5 ₂						RD/DW
45	43		P5 ₁						WRH/BHE/CASH
46	44		P5 ₀						WRL/WR/CASL
47	45		P4 ₇						CS0/A23
48	46		P4 ₆						CS1/A22
49	47		P4 ₅						CS2/A21
50	48		P4 ₄						CS3/A20(MA12)

Description

Table 1.1.8. 100-pin version pin description (2/2)

Package pin No		Control	Port	Interrupt	Timer	UART/CAN	Intelligent I/O	Analog	Bus control
FP	GP								
51	49		P4 ₃						A19(MA11)
52	50		P4 ₂						A18(MA10)
53	51		P4 ₁						A17(MA9)
54	52		P4 ₀						A16(MA8)
55	53		P3 ₇						A15(MA7)/(D15)
56	54		P3 ₆						A14(MA6)/(D14)
57	55		P3 ₅						A13(MA5)/(D13)
58	56		P3 ₄						A12(MA4)/(D12)
59	57		P3 ₃						A11(MA3)/(D11)
60	58		P3 ₂						A10(MA2)/(D10)
61	59		P3 ₁						A9(MA1)/(D9)
62	60	Vcc							
63	61		P3 ₀						A8(MA0)/(D8)
64	62	Vss							
65	63		P2 ₇					AN27	A7(/D7)
66	64		P2 ₆					AN26	A6(/D6)
67	65		P2 ₅					AN25	A5(/D5)
68	66		P2 ₄					AN24	A4(/D4)
69	67		P2 ₃					AN23	A3(/D3)
70	68		P2 ₂					AN22	A2(/D2)
71	69		P2 ₁					AN21	A1(/D1)
72	70		P2 ₀					AN20	A0(/D0)
73	71		P1 ₇	INT5					D15
74	72		P1 ₆	INT4					D14
75	73		P1 ₅	INT3					D13
76	74		P1 ₄						D12
77	75		P1 ₃						D11
78	76		P1 ₂						D10
79	77		P1 ₁						D9
80	78		P1 ₀						D8
81	79		P0 ₇					AN07	D7
82	80		P0 ₆					AN06	D6
83	81		P0 ₅					AN05	D5
84	82		P0 ₄					AN04	D4
85	83		P0 ₃					AN03	D3
86	84		P0 ₂					AN02	D2
87	85		P0 ₁					AN01	D1
88	86		P0 ₀					AN00	D0
89	87		P10 ₇	KI ₃				AN7	
90	88		P10 ₆	KI ₂				AN6	
91	89		P10 ₅	KI ₁				AN5	
92	90		P10 ₄	KI ₀				AN4	
93	91		P10 ₃					AN3	
94	92		P10 ₂					AN2	
95	93		P10 ₁					AN1	
96	94	AVss							
97	95		P10 ₀					AN0	
98	96	VREF							
99	97	AVCC							
100	98		P9 ₇			RxD4/SCL4/STxD4		ADTRG	

Table 1.1.9. Pin description (1/4)

Port	Function	Pin name	I/O type	Description
	Power supply input	VCC VSS	I I	4.2 to 5.5 V or 3.0V to 3.6V. 0 V.
	CPU mode switch	CNVSS	I	Connect it to VSS : Single-chip or memory expansion mode Connect it to VCC : Microprocessor mode
	External data bus width select input	BYTE	I	Selects the width of the data bus for external memory. Connect it to VSS : A 16-bit width Connect it to VCC : An 8-bit width
	Reset input	RESET	I	A "L" on this input resets the microcomputer.
	Clock input Clock output	XIN XOUT	I O	These pins are provided for the main clock generating circuit. Connect a ceramic resonator or crystal between the XIN and the XOUT pins. To use an externally derived clock, input it to the XIN pin and leave the XOUT pin open.
	Analog power supply input	AVCC AVSS	I I	Connect this pin to VCC. Connect this pin to VSS.
	Reference voltage input	VREF	I	This pin is a reference voltage input for the A-D converter.
P0	I/O port	P00 to P07	I/O	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. The user can specify in units of four bits via software whether or not they are tied to a pull-up resistor.
	Data bus	D0 to D7	I/O	When set as a separate bus, these pins input and output 8 low-order data bits.
	Analog input port	AN00 to AN07	I	P00 to P07 are analog input ports for the A-D converter.
P1	I/O port	P10 to P17	I/O	This is an 8-bit I/O port equivalent to P0.
	External interrupt input port	INT3 to INT5	I	P15 to P17 function as external interrupt pins.
	Data bus	D8 to D15	I/O	When set as a separate bus, these pins input and output 8 high-order data bits.
P2	I/O port	P20 to P27	I/O	This is an 8-bit I/O port equivalent to P0.
	Address bus	A0 to A7	O	These pins output 8 low-order address bits.
	Address bus/data bus	A0/D0 to A7/D7	I/O	If a multiplexed bus is set, these pins input and output data and output 8 low-order address bits separated in time by multiplexing.
	Analog input port	AN20 to AN27	I	P20 to P27 are analog input ports for the A-D converter.
P3	I/O port	P30 to P37	I/O	This is an 8-bit I/O port equivalent to P0.
	Address bus	A8 to A15	O	These pins output 8 middle-order address bits.
	Address bus/data bus	A8/D8 to A15/D15	I/O	If the external bus is set as a 16-bit wide multiplexed bus, these pins output 8 middle-order address bits, and input and output 8 middle-order data separated in time by multiplexing.
	Address bus	MA0 to MA7	O	If accessing to DRAM area, these pins output row address and column address separated in time by multiplexing.
P4	I/O port	P40 to P47	I/O	This is an 8-bit I/O port equivalent to P0.
	Address bus	A16 to A22 A23	O	These pins output 8 high-order address bits. Highest address bit (A23) outputs inversely.
	Chip select	CS0 to CS3	O	P40 to P47 are chip select output pins to specify access area.
	Address bus	MA8 to MA12	O	If accessing to DRAM area, these pins output row address and column address separated in time by multiplexing.

Description

Table 1.1.10. Pin description (2/4)

Port	Function	Pin name	I/O type	Description
P5	I/O port	P50 to P57	I/O	This is an 8-bit I/O port equivalent to P0.
	Clock output	CLKOUT	I/O	P53 in this port outputs a divide-by-8 or divide-by-32 clock of XIN or a clock of the same frequency as XCIN.
	Bus control	\overline{WRL} / \overline{WR} , \overline{WRH} / \overline{BHE} , RD	O O O	Output \overline{WRL} , \overline{WRH} and \overline{RD} , or \overline{WR} , \overline{BHE} and \overline{RD} bus control signals. ■ \overline{WRL} , \overline{WRH} , and \overline{RD} selected In 16-bit data bus, data is written to even addresses when the \overline{WRL} signal is "L". Data is written to odd addresses when the \overline{WRH} signal is "L". Data is read when \overline{RD} is "L". ■ \overline{WR} , \overline{BHE} , and \overline{RD} selected Data is written when \overline{WR} is "L". Data is read when \overline{RD} is "L". Odd addresses are accessed when \overline{BHE} is "L". Even addresses are accessed when \overline{BHE} is "H". Use \overline{WR} , \overline{BHE} , and \overline{RD} when all external memory is an 8-bit data bus.
		\overline{BCLK} , \overline{HOLD} ,	O I	Output operation clock for CPU. While the input level at the \overline{HOLD} pin is "L", the microcomputer is placed in the hold state.
		\overline{HLDA} ALE, RDY	O O I	While in the hold state, \overline{HLDA} outputs a "L" level. ALE is used to latch the address. While the input level of the \overline{RDY} pin is "L", the microcomputer is in the ready state.
Bus control for DRAM	\overline{DW} , \overline{CASL} , \overline{CASH} , RAS	O O O O	When \overline{DW} signal is "L", write to DRAM. Timing signal when latching to line address of even address. Timing signal when latching to line address of odd address. Timing signal when latching to row address.	
P6	I/O port	P60 to P67	I/O	This is an 8-bit I/O port equivalent to P0.
	UART port	CTS/RTS/SS CLK RxD/SCL/STxD TxD/SDA/SRxD	I/O	P60 to P63 are I/O ports for UART0. P64 to P67 are I/O ports for UART1.
	Intelligent I/O port	OUTC/ISCLK	I/O	ISCLK is a clock I/O port for intelligent I/O communication. OUTC is an output port for waveform generation function.
P7	I/O port	P70 to P77	I/O	This is an 8-bit I/O port equivalent to P0. However, P70 and P71 are N-channel open drain outputs.
	Timer A port	TAOUT TAIN	O I	P70 to P77 are I/O ports for timers A0–A3.
	Timer B port	TBIN	I	P71 is an input port for timer B5.
	Three phase motor control output port	\overline{V} , \overline{V} \overline{W} , \overline{W}	O	P72 and P73 are V phase outputs. P74 and P75 are W phase outputs.
	UART port	CTS/RTS/SS CLK RxD/SCL/STxD TxD/SDA/SRxD	I/O	P70 to P73 are I/O ports for UART2.
	Intelligent I/O port	INPC/OUTC ISCLK/ISTxD/ ISRxD IEOUT/IEIN BEOUT/BEIN	I/O	INPC is an input port for time measurement function. OUTC is an output port for waveform generation function. ISCLK is a clock I/O port for intelligent I/O communication. ISTxD/IEOUT/BEOUT is transmit data output port for intelligent I/O communication. ISRxD/IEIN/BEIN is receive data input port for intelligent I/O communication.
	CANOUT CANIN	CAN	O I	P76 and P77 are I/O ports for CAN communication function.

Table 1.1.11. Pin description (3/4)

Port	Function	Pin name	I/O type	Description
P8	I/O port	P80-P84, P86, P87	I/O	This is a 7-bit I/O port equivalent to P0.
	Sub clock input	XCIN	I	P86 and P87 function as I/O ports for the sub clock generating circuit by software. Connect a crystal between the XCIN and the XCOUT pins.
	Sub clock output	XCOUT	O	
	Low-pass filter connect pin for PLL frequency synthesizer	VCOUT	O	When using PLL frequency synthesizer, connect P87 to a low-pass filter. To stabilize PLL frequency, connect P86 to Vss.
	Timer A port	TA4OUT TA4IN	O I	P80 to P81 are I/O ports for timer A4.
	Three phase motor control output port	U, \bar{U}	O	P80 and P81 are U phase output ports.
	External interrupt input port	INT0 to INT2	I	P82 to P84 are external interrupt input ports.
	Intelligent I/O port	INPC/ISRxD/BEIN	I	INPC is an input port for time measurement function. ISRxD/BEIN is receive data input port for intelligent I/O communication.
Input port	P85/ $\overline{\text{NMI}}$	I	Input port and input ports for NMI interrupt.	
P9	I/O port	P90 to P97	I/O	This is an 8-bit I/O port equivalent to P0.
	Timer B port	TB0IN to TB4IN	I	P90 to P94 are input port for timer B4.
	UART port	CTS/RTS/SS CLK RxD/SCL/STxD TxD/SDA/SRxD	I/O I/O I/O I/O	P90 to P93 are I/O ports for UART3. P94 to P97 are I/O ports for UART4.
	D-A output port	DA0, DA1	O	P93 and P94 are D-A output ports.
	A-D related port	ANEX1, ANEX2 ADTRG	I I	P95 to P96 are expanded input port for A-D converter. P97 is A-D trigger input port.
	Intelligent I/O port	OUTC/IEOUT IEIN	I/O I	OUTC is an output port for waveform generation function. IEOUT is transmit data output port for intelligent I/O communication. IEIN is receive data input port for intelligent I/O communication.
The protect register prevents a false write to P9 direction register and function select register A3.				
P10	I/O port	P100 to P107	I/O	This is an 8-bit I/O port equivalent to P0.
	Key input interrupt port	KI0 to KI3	I	P104 to P107 are key input interrupt ports.
	Analog input port	AN0 to AN7	I	P100 to P107 are analog input ports for A-D convertor.

Description

Table 1.1.12. Pin description (4/4)

Port	Function	Pin name	I/O type	Description
P11 (Note)	I/O port	P110 to P114	I/O	This is an 5-bit I/O port equivalent to P0.
	Intelligent I/O port	INPC/OUTC ISCLK ISTxD/ISRxD BEOUT/BEIN	I/O I/O I/O	INPC is an input port for time measurement function. OUTC is an output port for waveform generation function. ISCLK is a clock I/O port for intelligent I/O communication. ISTxD/BEOUT is transmit data output port for intelligent I/O communication. ISRxD/BEIN is receive data input port for intelligent I/O communication.
P12 (Note)	I/O port	P120 to P127	I/O	This is an 8-bit I/O port equivalent to P0.
	Intelligent I/O port	OUTC	O	OUTC is an output port for waveform generation function.
P13 (Note)	I/O port	P130 to P137	I/O	This is an 8-bit I/O port equivalent to P0.
	Intelligent I/O port	OUTC ISCLK/ISTxD/ ISRxD IEOUT/IEIN	I/O I/O I/O I/O	OUTC is an output port for waveform generation function. ISCLK is a clock I/O port for intelligent I/O communication. ISTxD/IEOUT is transmit data output port for intelligent I/O communication. ISRxD/IEIN is receive data input port for intelligent I/O communication.
P14 (Note)	I/O port	P140 to P146	I/O	This is a 7-bit I/O port equivalent to P0.
	Intelligent I/O port	INPC/OUTC	I/O	INPC is an input port for time measurement function. OUTC is an output port for waveform generation function.
P15 (Note)	I/O port	P150 to P157	I/O	This is an 8-bit I/O port equivalent to P0.
	Intelligent I/O port	INPC/OUTC ISCLK/ISTxD/ ISRxD BEOUT/BEIN	I/O I/O I/O	INPC is an input port for time measurement function. OUTC is an output port for waveform generation function. ISCLK is a clock I/O port for intelligent I/O communication. ISTxD/BEOUT is transmit data output port for intelligent I/O communication. ISRxD/BEIN is receive data input port for intelligent I/O communication.
	Analog input port	AN150 to AN157	I	P150 to P157 are analog input ports for A-D convertor.

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

Block Diagram

The M32C/83 group includes the following devices in a single-chip. ROM and RAM for code instructions and data, storage, CPU for executing operation and peripheral functions such as timer, serial I/O, D-A converter, DMAC, CRC operation circuit, A-D converter, DRAM controller, intelligent I/O and I/O ports.

Figure 1.1.6 is a block diagram of the M32C/83 group (144-pin version).

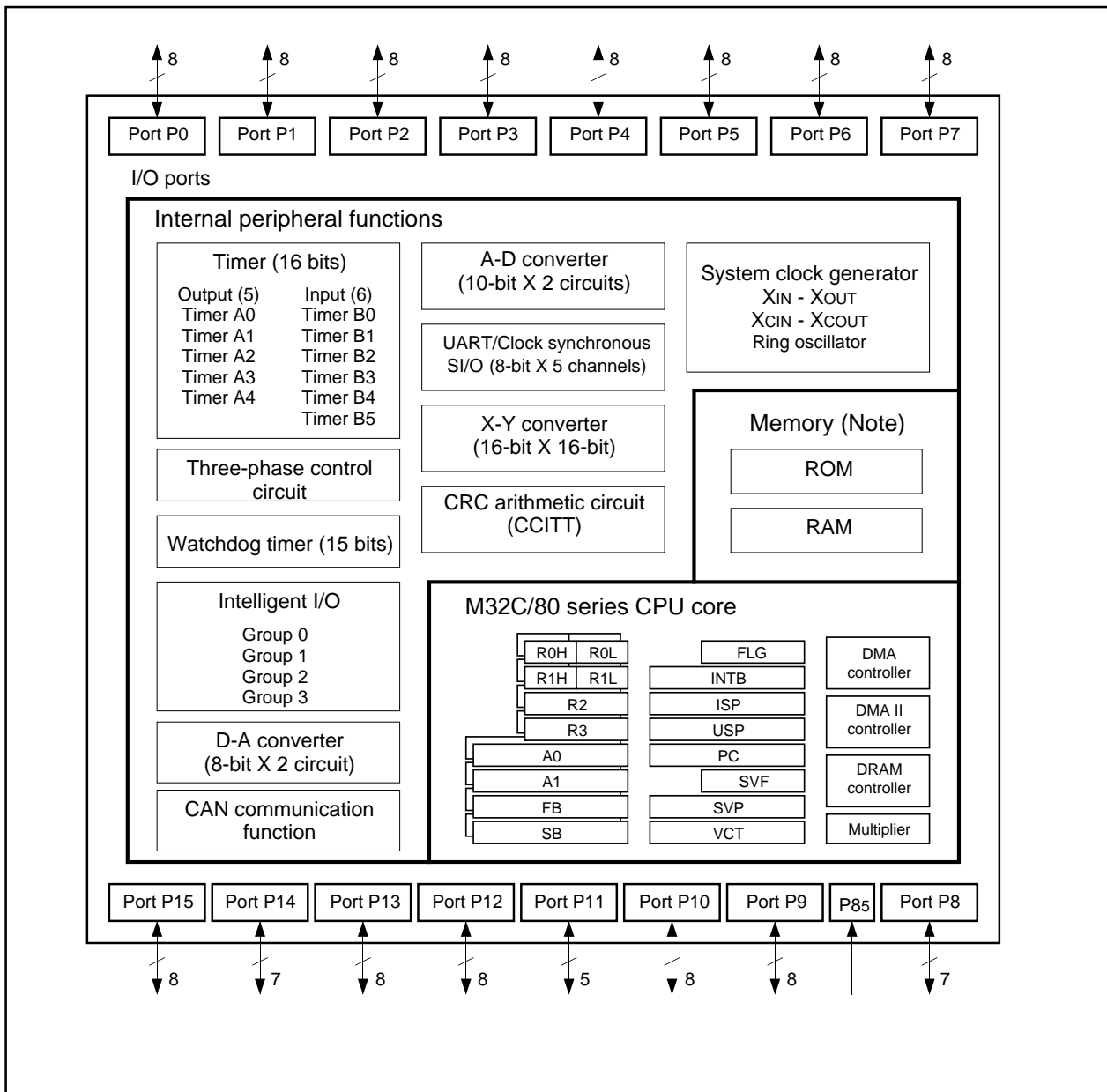


Figure 1.1.6. Block diagram of the M32C/83 group (144-pin version)

Memory

Figure 1.2.1 is a memory map of the M32C/83 group. The address space extends 16 Mbytes from address 000000₁₆ to FFFFFFF₁₆. From FFFFFFF₁₆ down is ROM. For example, in the M30835FJGP, there are 512K bytes of internal ROM from F80000₁₆ to FFFFFFF₁₆. The vector table for fixed interrupts such as the reset and NMI are mapped to FFFFDC₁₆ to FFFFFFF₁₆. 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 000400₁₆ up is RAM. For example, in the M30835FJGP, 31 Kbytes of internal RAM are mapped to the space from 000400₁₆ to 007FFF₁₆. 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 from 000000₁₆ to 0003FF₁₆. This area accommodates the control registers for peripheral devices such as I/O ports, A-D converter, serial I/O, and timers, etc. Any part of the SFR area that is not occupied is reserved and cannot be used for any other purpose.

The special page vector table is mapped from FFFE00₁₆ to FFFFDB₁₆. 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.

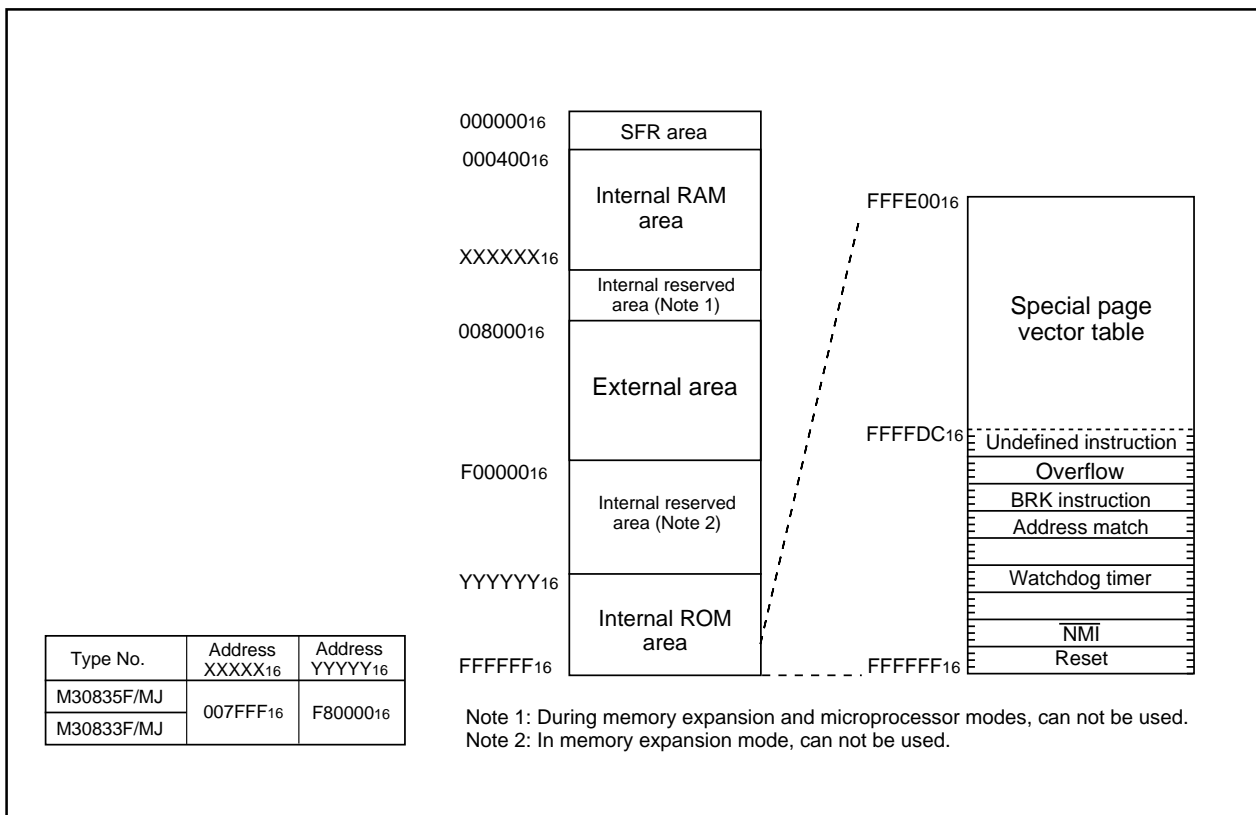


Figure 1.2.1. Memory map

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.

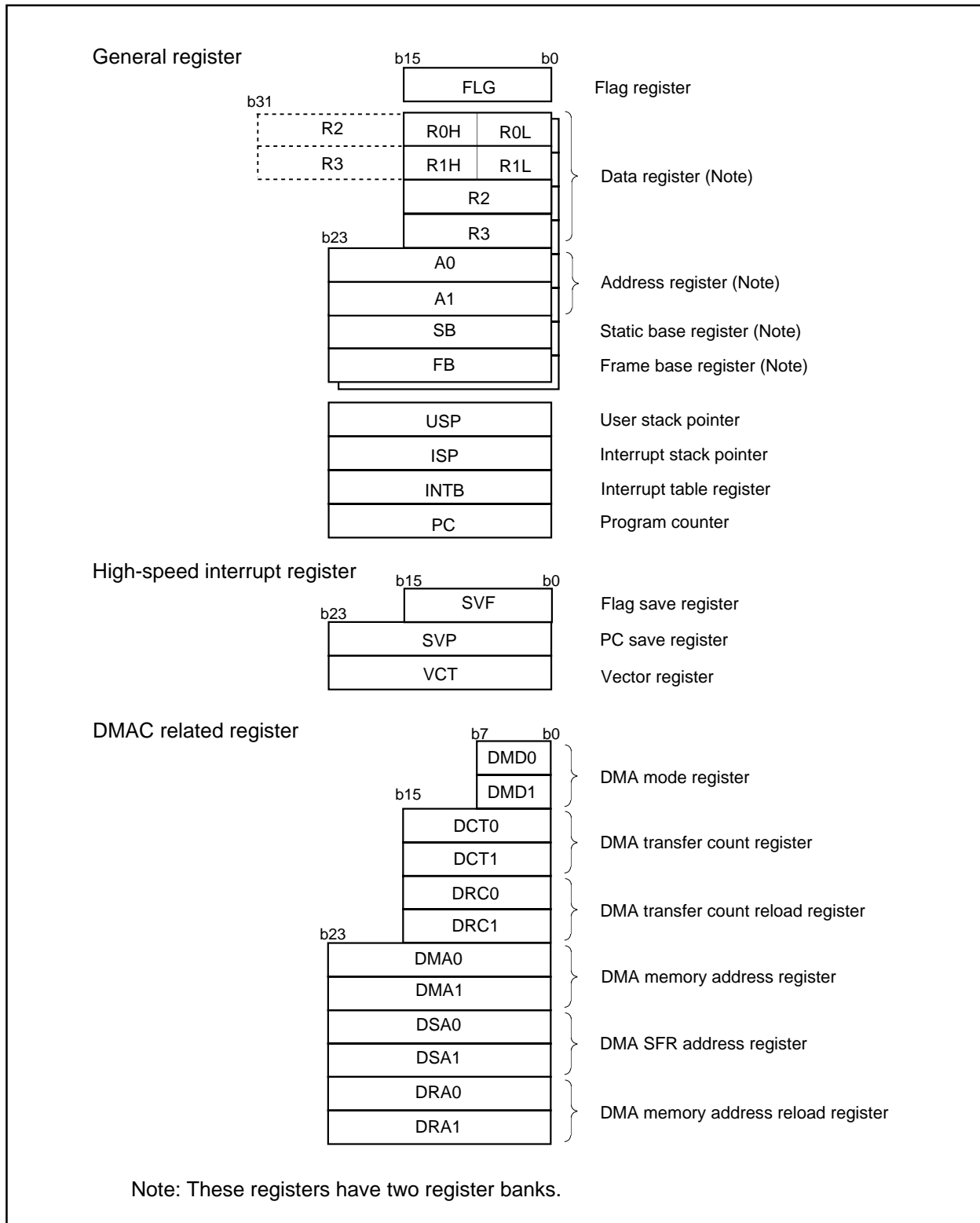


Figure 1.3.1. Central processing unit register

(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 function 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 function as 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.

The desired type of stack pointer (USP or ISP) can be selected by a stack pointer select flag (U flag). This flag is located at bit 7 in the flag register (FLG).

To execute efficiently set USP and ISP to an even number.

(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.

This register consist of 24 bits and is used to indicate a jump address 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)

This flag retains a carry, borrow, or shift-out bit that has occurred in the arithmetic/logic unit.

• Bit 1: Debug flag (D)

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)

This flag is set to "1" when an arithmetic operation resulted in 0; otherwise, cleared to "0".

• Bit 3: Sign flag (S)

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)**

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)**

This flag is set to “1” when an arithmetic operation resulted in overflow; otherwise, cleared to “0”.

- **Bit 6: Interrupt enable flag (I)**

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)**

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 Numbers. 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**

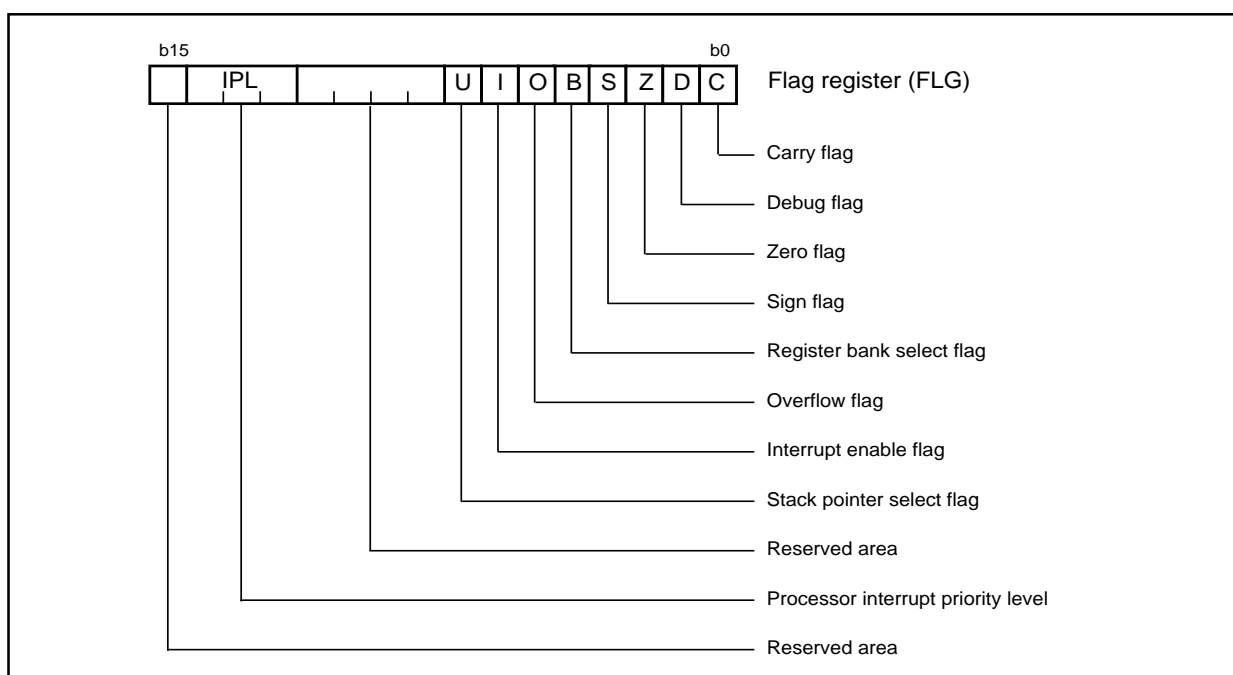


Figure 1.3.2. Flag register (FLG)

Reset

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 hardware resets.

When the supply voltage is in the range where operation is guaranteed, a reset is enabled by holding the reset pin Low (0.2V_{CC} max.) for at least 20 cycles. When the reset pin level is then returned to High while main clock is stable, the reset status is cancelled and program execution resumes from the address in the reset vector table.

Since the value of RAM is indeterminate when power is applied, the initial values must be set. Also, if a reset signal is input during write to RAM, the access to the RAM will be interrupted. Consequently, the value of the RAM being written may change to an unintended value due to the interruption.

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

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

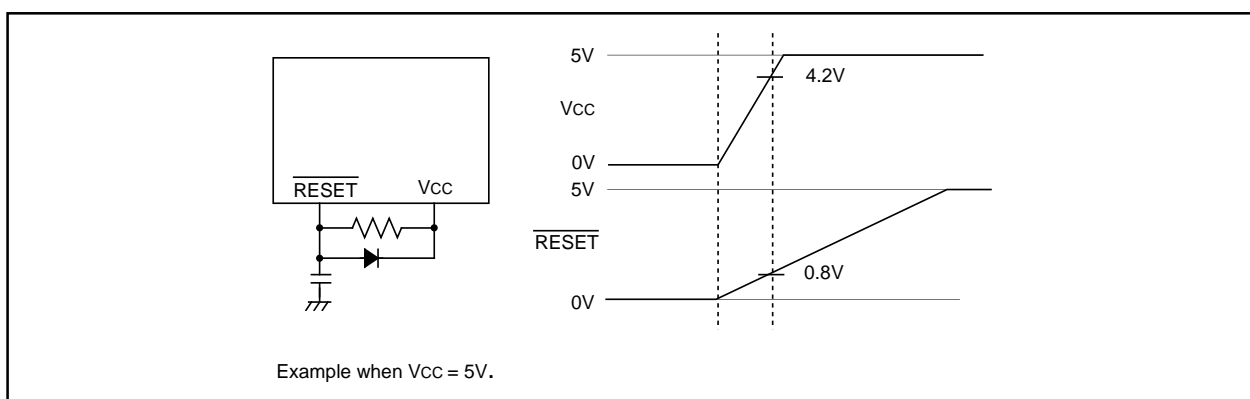


Figure 1.4.1. Example reset circuit

Reset

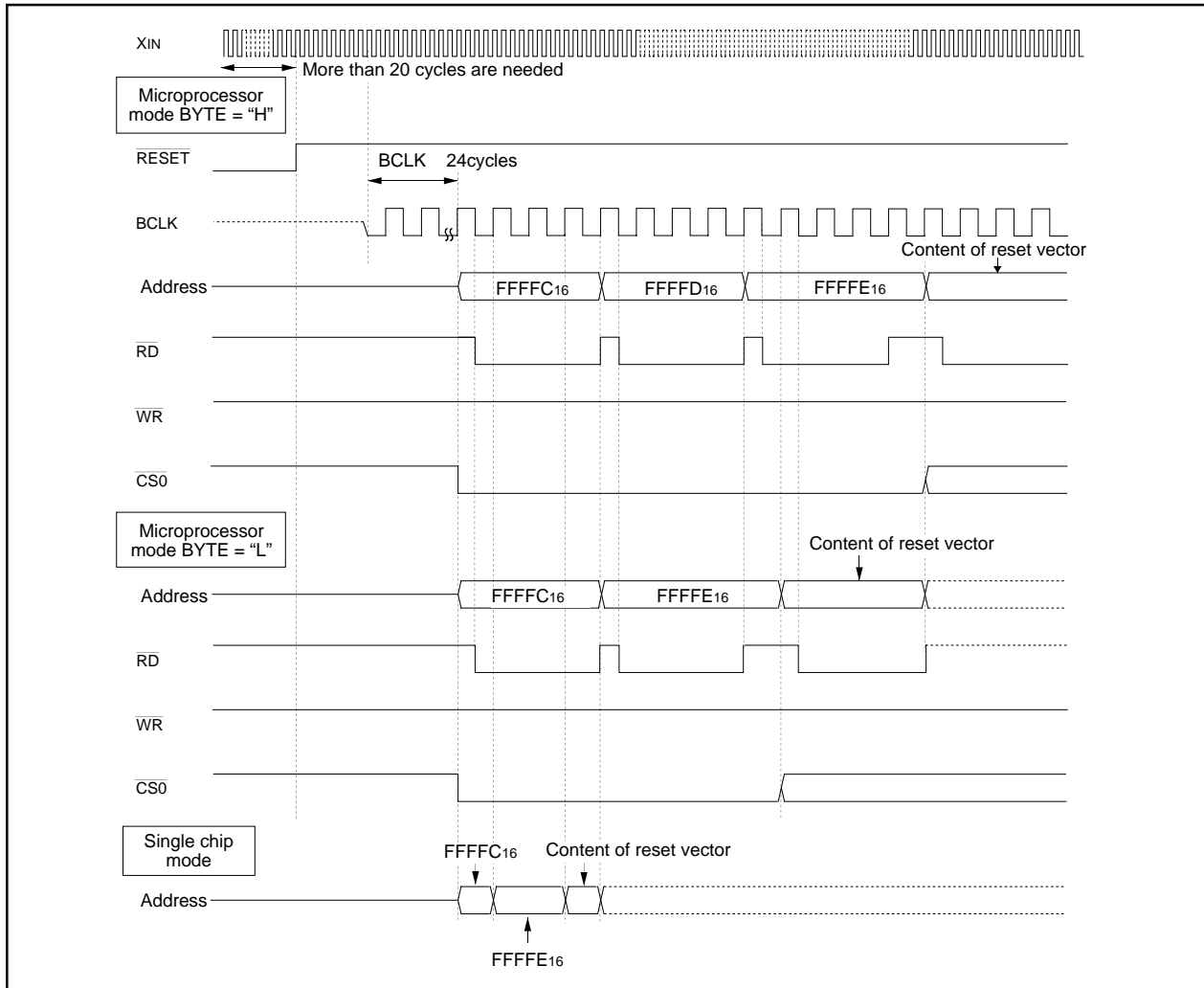


Figure 1.4.2. Reset sequence

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)	
P1	Input port (floating)	Data input (floating)	Input port (floating)
P2, P3, P4	Input port (floating)	Address output (undefined)	
P50	Input port (floating)	$\overline{\text{WR}}$ output ("H" level output)	
P51	Input port (floating)	$\overline{\text{BHE}}$ output (undefined)	
P52	Input port (floating)	$\overline{\text{RD}}$ output ("H" level output)	
P53	Input port (floating)	BCLK output	
P54	Input port (floating)	$\overline{\text{HLDA}}$ output (The output value depends on the input to the HOLD pin)	
P55	Input port (floating)	HOLD input (floating)	
P56	Input port (floating)	RAS output	
P57	Input port (floating)	$\overline{\text{RDY}}$ input (floating)	
P6 to P15 (Note)	Input port (floating)	Input port (floating)	

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

(1) Processor mode register 0	(Note 1) (0004 ₁₆)	80 ₁₆	(26) UART2 receive /ACK interrupt control register	(006B ₁₆)	XXXXXXXX?000
(2) Processor mode register 1	(0005 ₁₆)	X000000XX	(27) Timer A0 interrupt control register	(006C ₁₆)	XXXXXXXX?000
(3) System clock control register 0	(0006 ₁₆)	0000X000	(28) UART3 receive/ACK interrupt control register	(006D ₁₆)	XXXXXXXX?000
(4) System clock control register 1	(0007 ₁₆)	20 ₁₆	(29) Timer A2 interrupt control register	(006E ₁₆)	XXXXXXXX?000
(5) Wait control register	(0008 ₁₆)	FF ₁₆	(30) UART4 receive/ACK interrupt control register	(006F ₁₆)	XXXXXXXX?000
(6) Address match interrupt control register	(0009 ₁₆)	XXXXXXXX0000	(31) Timer A4 interrupt control register	(0070 ₁₆)	XXXXXXXX?000
(7) Protect register	(000A ₁₆)	XXXXXXXX0000	(32) UART0/UART3 bus collision detection interrupt control register	(0071 ₁₆)	XXXXXXXX?000
(8) External data bus width control register	(Note 2) (000B ₁₆)	XXXXXXXX0000	(33) UART0 receive/ACK interrupt control register	(0072 ₁₆)	XXXXXXXX?000
(9) Main clock divided register	(000C ₁₆)	XXXXXXXX01000	(34) A-D0 interrupt control register	(0073 ₁₆)	XXXXXXXX?000
(10) Oscillation stop detect register	(000D ₁₆)	00 ₁₆	(35) UART1 receive/ACK interrupt control register	(0074 ₁₆)	XXXXXXXX?000
(11) Watchdog timer start register	(000E ₁₆)	?? ₁₆	(36) Intelligent I/O interrupt control register 0	(0075 ₁₆)	XXXXXXXX?000
(12) Watchdog timer control register	(000F ₁₆)	0000????	(37) Timer B1 interrupt control register	(0076 ₁₆)	XXXXXXXX?000
(13) Address match interrupt register 0	(0010 ₁₆)	00 ₁₆	(38) Intelligent I/O interrupt control register 2	(0077 ₁₆)	XXXXXXXX?000
	(0011 ₁₆)	00 ₁₆	(39) Timer B3 interrupt control register	(0078 ₁₆)	XXXXXXXX?000
	(0012 ₁₆)	00 ₁₆	(40) Intelligent I/O interrupt control register 4	(0079 ₁₆)	XXXXXXXX?000
(14) Address match interrupt register 1	(0014 ₁₆)	00 ₁₆	(41) INT5 interrupt control register	(007A ₁₆)	XXXX00?000
	(0015 ₁₆)	00 ₁₆	(42) Intelligent I/O interrupt control register 6	(007B ₁₆)	XXXXXXXX?000
	(0016 ₁₆)	00 ₁₆	(43) INT3 interrupt control register	(007C ₁₆)	XXXX00?000
(15) VDC control register for PLL	(0017 ₁₆)	XXXXXXXXX01	(44) Intelligent I/O interrupt control register 8	(007D ₁₆)	XXXXXXXX?000
(16) Address match interrupt register 2	(0018 ₁₆)	00 ₁₆	(45) INT1 interrupt control register	(007E ₁₆)	XXXX00?000
	(0019 ₁₆)	00 ₁₆	(46) Intelligent I/O interrupt control register 10/ CAN interrupt 1 control register	(007F ₁₆)	XXXXXXXX?000
	(001A ₁₆)	00 ₁₆	(47) Intelligent I/O interrupt control register 11/ CAN interrupt 2 control register	(0081 ₁₆)	XXXXXXXX?000
(17) VDD control register 1	(001B ₁₆)	00 ₁₆	(48) A-D1 interrupt control register	(0086 ₁₆)	XXXXXXXX?000
(18) Address match interrupt register 3	(001C ₁₆)	00 ₁₆	(49) DMA1 interrupt control register	(0088 ₁₆)	XXXXXXXX?000
	(001D ₁₆)	00 ₁₆	(50) UART2 transmit /NACK interrupt control register	(0089 ₁₆)	XXXXXXXX?000
	(001E ₁₆)	00 ₁₆	(51) DMA3 interrupt control register	(008A ₁₆)	XXXXXXXX?000
(19) VDD control register 1	(001F ₁₆)	00 ₁₆	(52) UART3 transmit /NACK interrupt control register	(008B ₁₆)	XXXXXXXX?000
(20) DRAM control register	(0040 ₁₆)	?XXXXX???	(53) Timer A1 interrupt control register	(008C ₁₆)	XXXXXXXX?000
(21) DRAM refresh interval set register	(0041 ₁₆)	?? ₁₆	(54) UART4 transmit /NACK interrupt control register	(008D ₁₆)	XXXXXXXX?000
(22) Flash memory control register 0	(0057 ₁₆)	XX000001	(55) Timer A3 interrupt control register	(008E ₁₆)	XXXXXXXX?000
(23) DMA0 interrupt control register	(0068 ₁₆)	XXXXXXXX?000	(56) UART2 bus collision detection interrupt control register	(008F ₁₆)	XXXXXXXX?000
(24) Timer B5 interrupt control register	(0069 ₁₆)	XXXXXXXX?000	(57) UART0 transmit /NACK interrupt control register	(0090 ₁₆)	XXXXXXXX?000
(25) DMA2 interrupt control register	(006A ₁₆)	XXXXXXXX?000	(58) UART1/UART4 bus collision detection interrupt control register	(0091 ₁₆)	XXXXXXXX?000

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 Vcc level is applied to the CNVss pin, it is 03₁₆ at a reset.
Note 2: When the BYTE pin is "L", bit 3 is "1". When the BYTE pin is "H", bit 3 is "0".

Figure 1.4.3. Device's internal status after a reset is cleared (1/10)

(59) UART1 transmit/NACK interrupt control register (009216)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> ?0000	(92) Interrupt enable register 7 (00B716)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
(60) Key input interrupt control register (009316)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> ?0000	(93) Interrupt enable register 8 (00B816)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
(61) Timer B0 interrupt control register (009416)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> ?0000	(94) Interrupt enable register 9 (00B916)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
(62) Intelligent I/O interrupt control register 1 (009516)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> ?0000	(95) Interrupt enable register 10 (00BA16)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
(63) Timer B2 interrupt control register (009616)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> ?0000	(96) Interrupt enable register 11 (00BB16)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>
(64) Intelligent I/O interrupt control register 3 (009716)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> ?0000	(97) Group 0 time measurement/waveform generate register 0 (00C016)	<input type="text" value="??16"/>
(65) Timer B4 interrupt control register (009816)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> ?0000	(98) Group 0 time measurement/waveform generate register 1 (00C116)	<input type="text" value="??16"/>
(66) Intelligent I/O interrupt control register 5 (009916)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> ?0000	(99) Group 0 time measurement/waveform generate register 2 (00C216)	<input type="text" value="??16"/>
(67) INT4 interrupt control register (009A16)	<input type="checkbox"/> <input type="checkbox"/> 00?0000	(100) Group 0 time measurement/waveform generate register 3 (00C316)	<input type="text" value="??16"/>
(68) Intelligent I/O interrupt control register 7 (009B16)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> ?0000	(101) Group 0 time measurement/waveform generate register 4 (00C416)	<input type="text" value="??16"/>
(69) INT2 interrupt control register (009C16)	<input type="checkbox"/> <input type="checkbox"/> 00?0000	(102) Group 0 time measurement/waveform generate register 5 (00C516)	<input type="text" value="??16"/>
(70) Intelligent I/O interrupt control register 9/ CAN interrupt 0 control register (009D16)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> ?0000	(103) Group 0 time measurement/waveform generate register 6 (00C616)	<input type="text" value="??16"/>
(71) INT0 interrupt control register (009E16)	<input type="checkbox"/> <input type="checkbox"/> 00?0000	(104) Group 0 time measurement/waveform generate register 7 (00C716)	<input type="text" value="??16"/>
(72) Exit priority register (009F16)	<input type="checkbox"/> <input type="checkbox"/> 00X0000	(105) Group 0 time measurement/waveform generate register 8 (00C816)	<input type="text" value="??16"/>
(73) Interrupt request register 0 (00A016)	<input type="checkbox"/> <input type="checkbox"/> 00X000X	(106) Group 0 time measurement/waveform generate register 9 (00C916)	<input type="text" value="??16"/>
(74) Interrupt request register 1 (00A116)	<input type="checkbox"/> <input type="checkbox"/> 00X000X	(107) Group 0 time measurement/waveform generate register 10 (00CA16)	<input type="text" value="??16"/>
(75) Interrupt request register 2 (00A216)	<input type="checkbox"/> <input type="checkbox"/> 00X000X	(108) Group 0 time measurement/waveform generate register 11 (00CB16)	<input type="text" value="??16"/>
(76) Interrupt request register 3 (00A316)	<input type="checkbox"/> <input type="checkbox"/> 0000000X	(109) Group 0 time measurement/waveform generate register 12 (00CC16)	<input type="text" value="??16"/>
(77) Interrupt request register 4 (00A416)	<input type="checkbox"/> <input type="checkbox"/> 0000000X	(110) Group 0 time measurement/waveform generate register 13 (00CD16)	<input type="text" value="??16"/>
(78) Interrupt request register 5 (00A516)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 000000X	(111) Group 0 time measurement/waveform generate register 14 (00CE16)	<input type="text" value="??16"/>
(79) Interrupt request register 6 (00A616)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 000000X	(112) Group 0 time measurement/waveform generate register 15 (00CF16)	<input type="text" value="??16"/>
(80) Interrupt request register 7 (00A716)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 000000X	(113) Group 0 waveform generate control register 0 (00D016)	<input type="checkbox"/> <input type="checkbox"/> 00X0000
(81) Interrupt request register 8 (00A816)	<input type="checkbox"/> <input type="checkbox"/> 0000000X	(114) Group 0 waveform generate control register 1 (00D116)	<input type="checkbox"/> <input type="checkbox"/> 00X0000
(82) Interrupt request register 9 (00A916)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 000000X	(115) Group 0 waveform generate control register 4 (00D416)	<input type="checkbox"/> <input type="checkbox"/> 00X0000
(83) Interrupt request register 10 (00AA16)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 000000X	(116) Group 0 waveform generate control register 5 (00D516)	<input type="checkbox"/> <input type="checkbox"/> 00X0000
(84) Interrupt request register 11 (00AB16)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 000000X	(117) Group 0 time measurement control register 0 (00D816)	<input type="text" value="0016"/>
(85) Interrupt enable register 0 (00B016)	<input type="checkbox"/> <input type="checkbox"/> 00X0000	(118) Group 0 time measurement control register 1 (00D916)	<input type="text" value="0016"/>
(86) Interrupt enable register 1 (00B116)	<input type="checkbox"/> <input type="checkbox"/> 00X0000	(119) Group 0 time measurement control register 2 (00DA16)	<input type="text" value="0016"/>
(87) Interrupt enable register 2 (00B216)	<input type="checkbox"/> <input type="checkbox"/> 00X0000	(120) Group 0 time measurement control register 3 (00DB16)	<input type="text" value="0016"/>
(88) Interrupt enable register 3 (00B316)	<input type="checkbox"/> <input type="checkbox"/> 00000000	(121) Group 0 time measurement control register 4 (00DC16)	<input type="text" value="0016"/>
(89) Interrupt enable register 4 (00B416)	<input type="checkbox"/> <input type="checkbox"/> 00000000	(122) Group 0 time measurement control register 5 (00DD16)	<input type="text" value="0016"/>
(90) Interrupt enable register 5 (00B516)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 0000000	(123) Group 0 time measurement control register 6 (00DE16)	<input type="text" value="0016"/>
(91) Interrupt enable register 6 (00B616)	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> 0000000	(124) Group 0 time measurement control register 7 (00DF16)	<input type="text" value="0016"/>

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.

Figure 1.4.3. Device's internal status after a reset is cleared (2/10)

(117) Group 0 base timer register	(00E0 ₁₆)	?? ₁₆	(144) Group 1 time measurement/waveform generate register 2	(0104 ₁₆)	?? ₁₆
	(00E1 ₁₆)	?? ₁₆		(0105 ₁₆)	?? ₁₆
(118) Group 0 base timer control register 0	(00E2 ₁₆)	00 ₁₆	(145) Group 1 time measurement/waveform generate register 3	(0106 ₁₆)	?? ₁₆
(119) Group 0 base timer control register 1	(00E3 ₁₆)	00 ₁₆		(0107 ₁₆)	?? ₁₆
(120) Group 0 time measurement prescaler register 6	(00E4 ₁₆)	00 ₁₆	(146) Group 1 time measurement/waveform generate register 4	(0108 ₁₆)	?? ₁₆
(121) Group 0 time measurement prescaler register 7	(00E5 ₁₆)	00 ₁₆		(0109 ₁₆)	?? ₁₆
(122) Group 0 function enable register	(00E6 ₁₆)	00 ₁₆	(147) Group 1 time measurement/waveform generate register 5	(010A ₁₆)	?? ₁₆
(123) Group 0 function select register	(00E7 ₁₆)	00 ₁₆		(010B ₁₆)	?? ₁₆
(124) Group 0 SI/O receive buffer register	(00E8 ₁₆)	?? ₁₆	(148) Group 1 time measurement/waveform generate register 6	(010C ₁₆)	?? ₁₆
	(00E9 ₁₆)	X000XXXXXX		(010D ₁₆)	?? ₁₆
(125) Group 0 transmit buffer/receive data register	(00EA ₁₆)	?? ₁₆	(149) Group 1 time measurement/waveform generate register 7	(010E ₁₆)	?? ₁₆
(126) Group 0 receive input register	(00EC ₁₆)	?? ₁₆		(010F ₁₆)	?? ₁₆
(127) Group 0 SI/O communication mode register	(00ED ₁₆)	00 ₁₆	(150) Group 1 waveform generate control register 0	(0110 ₁₆)	0X00X000
(128) Group 0 transmit output register	(00EE ₁₆)	?? ₁₆	(151) Group 1 waveform generate control register 1	(0111 ₁₆)	0X00X000
(129) Group 0 SI/O communication control register	(00EF ₁₆)	0000X011	(152) Group 1 waveform generate control register 2	(0112 ₁₆)	0X00X000
(130) Group 0 data compare register 0	(00F0 ₁₆)	?? ₁₆	(153) Group 1 waveform generate control register 3	(0113 ₁₆)	0X00X000
(131) Group 0 data compare register 1	(00F1 ₁₆)	?? ₁₆	(154) Group 1 waveform generate control register 4	(0114 ₁₆)	0X00X000
(132) Group 0 data compare register 2	(00F2 ₁₆)	?? ₁₆	(155) Group 1 waveform generate control register 5	(0115 ₁₆)	0X00X000
(133) Group 0 data compare register 3	(00F3 ₁₆)	?? ₁₆	(156) Group 1 waveform generate control register 6	(0116 ₁₆)	0X00X000
(134) Group 0 data mask register 0	(00F4 ₁₆)	?? ₁₆	(157) Group 1 waveform generate control register 7	(0117 ₁₆)	0X00X000
(135) Group 0 data mask register 1	(00F5 ₁₆)	?? ₁₆	(158) Group 1 time measurement control register 1	(0119 ₁₆)	00 ₁₆
(136) Group 0 receive CRC code register	(00F8 ₁₆)	?? ₁₆	(159) Group 1 time measurement control register 2	(011A ₁₆)	00 ₁₆
	(00F9 ₁₆)	?? ₁₆	(160) Group 1 time measurement control register 6	(011E ₁₆)	00 ₁₆
(137) Group 0 transmit CRC code register	(00FA ₁₆)	00 ₁₆	(161) Group 1 time measurement control register 7	(011F ₁₆)	00 ₁₆
	(00FB ₁₆)	00 ₁₆	(162) Group 1 base timer register	(0120 ₁₆)	?? ₁₆
(138) Group 0 SI/O expansion mode register	(00FC ₁₆)	00 ₁₆		(0121 ₁₆)	?? ₁₆
(139) Group 0 SI/O expansion receive control register	(00FD ₁₆)	00 ₁₆	(163) Group 1 base timer control register 0	(0122 ₁₆)	00 ₁₆
(140) Group 0 SI/O special communication interrupt detect register	(00FE ₁₆)	000000XX	(164) Group 1 base timer control register 1	(0123 ₁₆)	00 ₁₆
(141) Group 0 SI/O expansion transmit control register	(00FF ₁₆)	000000XXXX			
(142) Group 1 time measurement/waveform generate register 0	(0100 ₁₆)	?? ₁₆			
	(0101 ₁₆)	?? ₁₆			
(143) Group 1 time measurement/waveform generate register 1	(0102 ₁₆)	?? ₁₆			
	(0103 ₁₆)	?? ₁₆			

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.

Figure 1.4.3. Device's internal status after a reset is cleared (3/10)

(165) Group 1 time measurement prescaler register 6 (0124 ₁₆)	00 ₁₆	(191) Group 2 waveform generate register 4	(0148 ₁₆)	?? ₁₆	
(166) Group 1 time measurement prescaler register 7 (0125 ₁₆)	00 ₁₆	(192) Group 2 waveform generate register 5	(014A ₁₆)	?? ₁₆	
(167) Group 1 function enable register	(0126 ₁₆)	00 ₁₆	(014B ₁₆)	?? ₁₆	
(168) Group 1 function select register	(0127 ₁₆)	00 ₁₆	(014C ₁₆)	?? ₁₆	
(169) Group 1 SI/O receive buffer register	(0128 ₁₆)	?? ₁₆	(014D ₁₆)	?? ₁₆	
	(0129 ₁₆)	X000XXXX			
(170) Group 1 transmit buffer/receive data register	(012A ₁₆)	?? ₁₆	(194) Group 2 waveform generate register 7	(014E ₁₆)	?? ₁₆
(171) Group 1 receive input register	(012C ₁₆)	?? ₁₆	(195) Group 2 waveform generate control register 0	(0150 ₁₆)	00 ₁₆
(172) Group 1 SI/O communication mode register	(012D ₁₆)	00 ₁₆	(196) Group 2 waveform generate control register 1	(0151 ₁₆)	00 ₁₆
(173) Group 1 transmit output register	(012E ₁₆)	?? ₁₆	(197) Group 2 waveform generate control register 2	(0152 ₁₆)	00 ₁₆
(174) Group 1 SI/O communication control register	(012F ₁₆)	0000X011	(198) Group 2 waveform generate control register 3	(0153 ₁₆)	00 ₁₆
(175) Group 1 data compare register 0	(0130 ₁₆)	?? ₁₆	(199) Group 2 waveform generate control register 4	(0154 ₁₆)	00 ₁₆
(176) Group 1 data compare register 1	(0131 ₁₆)	?? ₁₆	(200) Group 2 waveform generate control register 5	(0155 ₁₆)	00 ₁₆
(177) Group 1 data compare register 2	(0132 ₁₆)	?? ₁₆	(201) Group 2 waveform generate control register 6	(0156 ₁₆)	00 ₁₆
(178) Group 1 data compare register 3	(0133 ₁₆)	?? ₁₆	(202) Group 2 waveform generate control register 7	(0157 ₁₆)	00 ₁₆
(179) Group 1 data mask register 0	(0134 ₁₆)	?? ₁₆	(203) Group 2 base timer register	(0160 ₁₆)	?? ₁₆
(180) Group 1 data mask register 1	(0135 ₁₆)	?? ₁₆	(204) Group 2 base timer control register 0	(0162 ₁₆)	00 ₁₆
(181) Group 1 receive CRC code register	(0138 ₁₆)	?? ₁₆	(205) Group 2 base timer control register 1	(0163 ₁₆)	00 ₁₆
	(0139 ₁₆)	?? ₁₆	(206) Base timer start register	(0164 ₁₆)	XXXX0000
(182) Group 1 transmit CRC code register	(013A ₁₆)	00 ₁₆	(207) Group 2 function enable register	(0166 ₁₆)	00 ₁₆
	(013B ₁₆)	00 ₁₆	(208) Group 2 RTP output buffer register	(0167 ₁₆)	00 ₁₆
(183) Group 1 SI/O expansion mode register	(013C ₁₆)	00 ₁₆	(209) Group 2 SI/O communication mode register	(016A ₁₆)	00XXXX00
(184) Group 1 SI/O expansion receive control register	(013D ₁₆)	00 ₁₆	(210) Group 2 SI/O communication control register	(016B ₁₆)	0000X110
(185) Group 1 SI/O special communication interrupt detect register	(013E ₁₆)	0000000X	(211) Group 2 SI/O transmit buffer register	(016C ₁₆)	?? ₁₆
(186) Group 1 SI/O expansion transmit control register	(013F ₁₆)	000000XX		(016D ₁₆)	??XX??
(187) Group 2 waveform generate register 0	(0140 ₁₆)	?? ₁₆	(212) Group 2 SI/O receive buffer register	(016E ₁₆)	?? ₁₆
	(0141 ₁₆)	?? ₁₆		(016F ₁₆)	XXXXXX
(188) Group 2 waveform generate register 1	(0142 ₁₆)	?? ₁₆	(213) Group 2 IEBus address register	(0170 ₁₆)	?? ₁₆
	(0143 ₁₆)	?? ₁₆		(0171 ₁₆)	XXXX??
(189) Group 2 waveform generate register 2	(0144 ₁₆)	?? ₁₆	(214) Group 2 IEBus control register	(0172 ₁₆)	00XXXX00
	(0145 ₁₆)	?? ₁₆			
(190) Group 2 waveform generate register 3	(0146 ₁₆)	?? ₁₆			
	(0147 ₁₆)	?? ₁₆			

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.

Figure 1.4.3. Device's internal status after a reset is cleared (4/10)

(215) Group 2 IEBus transmit interrupt cause detect register	(0173 ₁₆)	<input type="text" value="x x x 0 0 0 0 0 0"/>	(238) Group 3 waveform generate mask register 4	(0198 ₁₆)	<input type="text" value="??16"/>
(216) Group 2 IEBus receive interrupt cause detect register	(0174 ₁₆)	<input type="text" value="x x x 0 0 0 0 0 0"/>		(0199 ₁₆)	<input type="text" value="??16"/>
(217) Input function select register	(0178 ₁₆)	<input type="text" value="0016"/>	(239) Group 3 waveform generate mask register 5	(019A ₁₆)	<input type="text" value="??16"/>
(218) Group 3 SI/O communication mode register	(017A ₁₆)	<input type="text" value="0 0 x x 0 0 0 0"/>		(019B ₁₆)	<input type="text" value="??16"/>
(219) Group 3 SI/O communication control register	(017B ₁₆)	<input type="text" value="0 0 ? 0 x ? ? 0"/>	(240) Group 3 waveform generate mask register 6	(019C ₁₆)	<input type="text" value="??16"/>
(220) Group 3 SI/O transmit buffer register	(017C ₁₆)	<input type="text" value="??16"/>		(019D ₁₆)	<input type="text" value="??16"/>
	(017D ₁₆)	<input type="text" value="??16"/>	(241) Group 3 waveform generate mask register 7	(019E ₁₆)	<input type="text" value="??16"/>
(221) Group 3 SI/O receive buffer register	(017E ₁₆)	<input type="text" value="??16"/>		(019F ₁₆)	<input type="text" value="??16"/>
	(017F ₁₆)	<input type="text" value="??16"/>	(242) Group 3 base timer register	(01A0 ₁₆)	<input type="text" value="??16"/>
(222) Group 3 waveform generate register 0	(0180 ₁₆)	<input type="text" value="??16"/>		(01A1 ₁₆)	<input type="text" value="??16"/>
	(0181 ₁₆)	<input type="text" value="??16"/>	(243) Group 3 base timer control register 0	(01A2 ₁₆)	<input type="text" value="0016"/>
(223) Group 3 waveform generate register 1	(0182 ₁₆)	<input type="text" value="??16"/>	(244) Group 3 base timer control register 1	(01A3 ₁₆)	<input type="text" value="0 x x 0 x 0 0 0 0"/>
	(0183 ₁₆)	<input type="text" value="??16"/>	(245) Group 3 function enable register	(01A6 ₁₆)	<input type="text" value="0016"/>
(224) Group 3 waveform generate register 2	(0184 ₁₆)	<input type="text" value="??16"/>	(246) Group 3 RTP output buffer register	(01A7 ₁₆)	<input type="text" value="0016"/>
	(0185 ₁₆)	<input type="text" value="??16"/>	(247) Group 3 high-speed HDLC communication control register 1	(01AB ₁₆)	<input type="text" value="0 0 x x x x x 0"/>
(225) Group 3 waveform generate register 3	(0186 ₁₆)	<input type="text" value="??16"/>	(248) Group 3 high-speed HDLC communication control register	(01AC ₁₆)	<input type="text" value="0016"/>
	(0187 ₁₆)	<input type="text" value="??16"/>	(249) Group 3 high-speed HDLC communication register	(01AD ₁₆)	<input type="text" value="??16"/>
(226) Group 3 waveform generate register 4	(0188 ₁₆)	<input type="text" value="??16"/>	(250) Group 3 high-speed HDLC transmit counter	(01AE ₁₆)	<input type="text" value="0016"/>
	(0189 ₁₆)	<input type="text" value="??16"/>		(01AF ₁₆)	<input type="text" value="0016"/>
(227) Group 3 waveform generate register 5	(018A ₁₆)	<input type="text" value="??16"/>	(251) Group 3 high-speed HDLC data compare register 0	(01B0 ₁₆)	<input type="text" value="0016"/>
	(018B ₁₆)	<input type="text" value="??16"/>		(01B1 ₁₆)	<input type="text" value="0016"/>
(228) Group 3 waveform generate register 6	(018C ₁₆)	<input type="text" value="??16"/>	(252) Group 3 high-speed HDLC data mask register 0	(01B2 ₁₆)	<input type="text" value="0016"/>
	(018D ₁₆)	<input type="text" value="??16"/>		(01B3 ₁₆)	<input type="text" value="0016"/>
(229) Group 3 waveform generate register 7	(018E ₁₆)	<input type="text" value="??16"/>	(253) Group 3 high-speed HDLC data compare register 1	(01B4 ₁₆)	<input type="text" value="0016"/>
	(018F ₁₆)	<input type="text" value="??16"/>		(01B5 ₁₆)	<input type="text" value="0016"/>
(230) Group 3 waveform generate control register 0	(0190 ₁₆)	<input type="text" value="0016"/>	(254) Group 3 high-speed HDLC data mask register 1	(01B6 ₁₆)	<input type="text" value="0016"/>
(231) Group 3 waveform generate control register 1	(0191 ₁₆)	<input type="text" value="0016"/>		(01B7 ₁₆)	<input type="text" value="0016"/>
(232) Group 3 waveform generate control register 2	(0192 ₁₆)	<input type="text" value="0016"/>	(255) Group 3 high-speed HDLC data compare register 2	(01B8 ₁₆)	<input type="text" value="0016"/>
(233) Group 3 waveform generate control register 3	(0193 ₁₆)	<input type="text" value="0016"/>		(01B9 ₁₆)	<input type="text" value="0016"/>
(234) Group 3 waveform generate control register 4	(0194 ₁₆)	<input type="text" value="0016"/>	(256) Group 3 high-speed HDLC data mask register 2	(01BA ₁₆)	<input type="text" value="0016"/>
(235) Group 3 waveform generate control register 5	(0195 ₁₆)	<input type="text" value="0016"/>		(01BB ₁₆)	<input type="text" value="0016"/>
(236) Group 3 waveform generate control register 6	(0196 ₁₆)	<input type="text" value="0016"/>	(257) Group 3 high-speed HDLC data compare register 3	(01BC ₁₆)	<input type="text" value="0016"/>
(237) Group 3 waveform generate control register 7	(0197 ₁₆)	<input type="text" value="0016"/>		(01BD ₁₆)	<input type="text" value="0016"/>

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.

Figure 1.4.3. Device's internal status after a reset is cleared (5/10)

(258) Group 3 high-speed HDLC data mask register 3	(01BE ₁₆) 00 ₁₆	(282) CAN0 message slot buffer 0 data 6	(01EC ₁₆) ?? ₁₆ (Note)
	(01BF ₁₆) 00 ₁₆	(283) CAN0 message slot buffer 0 data 7	(01ED ₁₆) ?? ₁₆ (Note)
(259) A-D1 register 0	(01C0 ₁₆) ?? ₁₆	(284) CAN0 message slot buffer 0 time stamp high	(01EE ₁₆) ?? ₁₆ (Note)
	(01C1 ₁₆) ?? ₁₆	(285) CAN0 message slot buffer 0 time stamp low	(01EF ₁₆) ?? ₁₆ (Note)
(260) A-D1 register 1	(01C2 ₁₆) ?? ₁₆	(286) CAN1 message slot buffer 0 standard ID 0	(01F0 ₁₆) XXXX?? ₁₆ (Note)
	(01C3 ₁₆) ?? ₁₆	(287) CAN1 message slot buffer 0 standard ID 1	(01F1 ₁₆) XX?? ₁₆ (Note)
(261) A-D1 register 2	(01C4 ₁₆) ?? ₁₆	(288) CAN1 message slot buffer 0 extended ID 0	(01F2 ₁₆) XXXX?? ₁₆ (Note)
	(01C5 ₁₆) ?? ₁₆	(289) CAN1 message slot buffer 0 extended ID 1	(01F3 ₁₆) ?? ₁₆
(262) A-D1 register 3	(01C6 ₁₆) ?? ₁₆	(290) CAN1 message slot buffer 0 extended ID 2	(01F4 ₁₆) XX?? ₁₆ (Note)
	(01C7 ₁₆) ?? ₁₆	(291) CAN1 message slot buffer 0 data length code	(01F5 ₁₆) XXXX?? ₁₆ (Note)
(263) A-D1 register 4	(01C8 ₁₆) ?? ₁₆	(292) CAN1 message slot buffer 0 data 0	(01F6 ₁₆) ?? ₁₆ (Note)
	(01C9 ₁₆) ?? ₁₆	(293) CAN1 message slot buffer 0 data 1	(01F7 ₁₆) ?? ₁₆ (Note)
(264) A-D1 register 5	(01CA ₁₆) ?? ₁₆	(294) CAN1 message slot buffer 0 data 2	(01F8 ₁₆) ?? ₁₆ (Note)
	(01CB ₁₆) ?? ₁₆	(295) CAN1 message slot buffer 0 data 3	(01F9 ₁₆) ?? ₁₆ (Note)
(265) A-D1 register 6	(01CC ₁₆) ?? ₁₆	(296) CAN1 message slot buffer 0 data 4	(01FA ₁₆) ?? ₁₆ (Note)
	(01CD ₁₆) ?? ₁₆	(297) CAN1 message slot buffer 0 data 5	(01FB ₁₆) ?? ₁₆ (Note)
(266) A-D1 register 7	(01CE ₁₆) ?? ₁₆	(298) CAN1 message slot buffer 0 data 6	(01FC ₁₆) ?? ₁₆ (Note)
	(01CF ₁₆) ?? ₁₆	(299) CAN1 message slot buffer 0 data 7	(01FD ₁₆) ?? ₁₆ (Note)
(267) A-D1 control register 2	(01D4 ₁₆) X00XX000	(300) CAN1 message slot buffer 0 time stamp high	(01FE ₁₆) ?? ₁₆ (Note)
(268) A-D1 control register 0	(01D6 ₁₆) 00 ₁₆	(301) CAN1 message slot buffer 0 time stamp low	(01FF ₁₆) ?? ₁₆ (Note)
(269) A-D1 control register 1	(01D7 ₁₆) XXX00000	(302) CAN0 control register 0	(0200 ₁₆) XX010X01 (Note)
(270) CAN0 message slot buffer 0 standard ID 0	(01E0 ₁₆) XX?? ₁₆ (Note)	(0201 ₁₆) XXXX0000	
(271) CAN0 message slot buffer 0 standard ID 1	(01E1 ₁₆) XX?? ₁₆ (Note)	(303) CAN0 status register	(0202 ₁₆) 00 ₁₆ (Note)
(272) CAN0 message slot buffer 0 extended ID 0	(01E2 ₁₆) XXXX?? ₁₆ (Note)	(0203 ₁₆) X0000X01	
(273) CAN0 message slot buffer 0 extended ID 1	(01E3 ₁₆) ?? ₁₆ (Note)	(304) CAN0 expansion ID register	(0204 ₁₆) 00 ₁₆ (Note)
(274) CAN0 message slot buffer 0 extended ID 2	(01E4 ₁₆) XX?? ₁₆ (Note)	(0205 ₁₆) 00 ₁₆	
(275) CAN0 message slot buffer 0 data length code	(01E5 ₁₆) XXXX?? ₁₆ (Note)	(305) CAN0 configuration register	(0206 ₁₆) 0000XXXX (Note)
(276) CAN0 message slot buffer 0 data 0	(01E6 ₁₆) ?? ₁₆ (Note)	(0207 ₁₆) 00 ₁₆	
(277) CAN0 message slot buffer 0 data 1	(01E7 ₁₆) ?? ₁₆ (Note)	(306) CAN0 time stamp register	(0208 ₁₆) 00 ₁₆ (Note)
(278) CAN0 message slot buffer 0 data 2	(01E8 ₁₆) ?? ₁₆ (Note)	(0209 ₁₆) 00 ₁₆	
(279) CAN0 message slot buffer 0 data 3	(01E9 ₁₆) ?? ₁₆ (Note)	(307) CAN0 transmit error count register	(020A ₁₆) 00 ₁₆ (Note)
(280) CAN0 message slot buffer 0 data 4	(01EA ₁₆) ?? ₁₆ (Note)	(308) CAN0 receive error count register	(020B ₁₆) 00 ₁₆ (Note)
(281) CAN0 message slot buffer 0 data 5	(01EB ₁₆) ?? ₁₆ (Note)		

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: This applies when the CAN module is supplied with a clock by setting the sleep mode control bit (bit 0 at address 0242₁₆) to 1 after reset.

Figure 1.4.3. Device's internal status after a reset is cleared (6/10)

(309) CAN0 slot interrupt status register	(020C ₁₆) <input type="text" value="00<sub>16</sub>"/> (Note) (020D ₁₆) <input type="text" value="00<sub>16</sub>"/>	(339) X0 register/Y0 register	(02C0 ₁₆) <input type="text" value="??<sub>16</sub>"/> (02C1 ₁₆) <input type="text" value="??<sub>16</sub>"/>
(310) CAN0 slot interrupt mask register	(0210 ₁₆) <input type="text" value="00<sub>16</sub>"/> (Note) (0211 ₁₆) <input type="text" value="00<sub>16</sub>"/>	(340) X1 register/Y1 register	(02C2 ₁₆) <input type="text" value="??<sub>16</sub>"/> (02C3 ₁₆) <input type="text" value="??<sub>16</sub>"/>
(311) CAN0 error interrupt mask register	(0214 ₁₆) <input type="text" value="xxxxxx0000"/> (Note)	(341) X2 register/Y2 register	(02C4 ₁₆) <input type="text" value="??<sub>16</sub>"/>
(312) CAN0 error interrupt status register	(0215 ₁₆) <input type="text" value="xxxxxx0000"/> (Note)		(02C5 ₁₆) <input type="text" value="??<sub>16</sub>"/>
(313) CAN0 baud rate prescaler	(0217 ₁₆) <input type="text" value="01<sub>16</sub>"/> (Note)	(342) X3 register/Y3 register	(02C6 ₁₆) <input type="text" value="??<sub>16</sub>"/>
(314) CAN0 global mask register standard ID0	(0228 ₁₆) <input type="text" value="xxx0000000"/> (Note)		(02C7 ₁₆) <input type="text" value="??<sub>16</sub>"/>
(315) CAN0 global mask register standard ID1	(0229 ₁₆) <input type="text" value="xx00000000"/> (Note)	(343) X4 register/Y4 register	(02C8 ₁₆) <input type="text" value="??<sub>16</sub>"/>
(316) CAN0 global mask register extended ID0	(022A ₁₆) <input type="text" value="??<sub>16</sub>"/> (Note)		(02C9 ₁₆) <input type="text" value="??<sub>16</sub>"/>
(317) CAN0 global mask register extended ID1	(022B ₁₆) <input type="text" value="??<sub>16</sub>"/> (Note)	(344) X5 register/Y5 register	(02CA ₁₆) <input type="text" value="??<sub>16</sub>"/>
(318) CAN0 global mask register extended ID2	(022C ₁₆) <input type="text" value="??<sub>16</sub>"/> (Note)		(02CB ₁₆) <input type="text" value="??<sub>16</sub>"/>
(319) CAN0 message slot 0 control register / CAN0 local mask register A standard ID0	(0230 ₁₆) <input type="text" value="xxxx000000"/> (Note)	(345) X6 register/Y6 register	(02CC ₁₆) <input type="text" value="??<sub>16</sub>"/>
(320) CAN0 message slot 1 control register / CAN0 local mask register A standard ID1	(0231 ₁₆) <input type="text" value="xx00000000"/> (Note)		(02CD ₁₆) <input type="text" value="??<sub>16</sub>"/>
(321) CAN0 message slot 2 control register / CAN0 local mask register A extended ID0	(0232 ₁₆) <input type="text" value="00<sub>16</sub>"/> (Note)	(346) X7 register/Y7 register	(02CE ₁₆) <input type="text" value="??<sub>16</sub>"/>
(322) CAN0 message slot 3 control register / CAN0 local mask register A extended ID1	(0233 ₁₆) <input type="text" value="00<sub>16</sub>"/> (Note)		(02CF ₁₆) <input type="text" value="??<sub>16</sub>"/>
(323) CAN0 message slot 4 control register / CAN0 local mask register A extended ID2	(0234 ₁₆) <input type="text" value="00<sub>16</sub>"/> (Note)	(347) X8 register/Y8 register	(02D0 ₁₆) <input type="text" value="??<sub>16</sub>"/>
(324) CAN0 message slot 5 control register	(0235 ₁₆) <input type="text" value="00<sub>16</sub>"/> (Note)		(02D1 ₁₆) <input type="text" value="??<sub>16</sub>"/>
(325) CAN0 message slot 6 control register	(0236 ₁₆) <input type="text" value="00<sub>16</sub>"/> (Note)	(348) X9 register/Y9 register	(02D2 ₁₆) <input type="text" value="??<sub>16</sub>"/>
(326) CAN0 message slot 7 control register	(0237 ₁₆) <input type="text" value="00<sub>16</sub>"/> (Note)		(02D3 ₁₆) <input type="text" value="??<sub>16</sub>"/>
(327) CAN0 message slot 8 control register / CAN0 local mask register B standard ID0	(0238 ₁₆) <input type="text" value="xxx0000000"/> (Note)	(349) X10 register/Y10 register	(02D4 ₁₆) <input type="text" value="??<sub>16</sub>"/>
(328) CAN0 message slot 9 control register / CAN0 local mask register B standard ID1	(0239 ₁₆) <input type="text" value="xx00000000"/> (Note)		(02D5 ₁₆) <input type="text" value="??<sub>16</sub>"/>
(329) CAN0 message slot 10 control register / CAN0 local mask register B extended ID0	(023A ₁₆) <input type="text" value="00<sub>16</sub>"/> (Note)	(350) X11 register/Y11 register	(02D6 ₁₆) <input type="text" value="??<sub>16</sub>"/>
(330) CAN0 message slot 11 control register / CAN0 local mask register B extended ID1	(023B ₁₆) <input type="text" value="00<sub>16</sub>"/> (Note)		(02D7 ₁₆) <input type="text" value="??<sub>16</sub>"/>
(331) CAN0 message slot 12 control register / CAN0 local mask register B extended ID2	(023C ₁₆) <input type="text" value="00<sub>16</sub>"/> (Note)	(351) X12 register/Y12 register	(02D8 ₁₆) <input type="text" value="??<sub>16</sub>"/>
(332) CAN0 message slot 13 control register	(023D ₁₆) <input type="text" value="00<sub>16</sub>"/> (Note)		(02D9 ₁₆) <input type="text" value="??<sub>16</sub>"/>
(333) CAN0 message slot 14 control register	(023E ₁₆) <input type="text" value="00<sub>16</sub>"/> (Note)	(352) X13 register/Y13 register	(02DA ₁₆) <input type="text" value="??<sub>16</sub>"/>
(334) CAN0 message slot 15 control register	(023F ₁₆) <input type="text" value="00<sub>16</sub>"/> (Note)		(02DB ₁₆) <input type="text" value="??<sub>16</sub>"/>
(335) CAN0 slot buffer select register	(0240 ₁₆) <input type="text" value="00<sub>16</sub>"/> (Note)	(353) X14 register/Y14 register	(02DC ₁₆) <input type="text" value="??<sub>16</sub>"/>
(336) CAN0 control register 1	(0241 ₁₆) <input type="text" value="xx00000xxx"/> (Note)		(02DD ₁₆) <input type="text" value="??<sub>16</sub>"/>
(337) CAN0 sleep control register	(0242 ₁₆) <input type="text" value="xxxxxx0000"/> (Note)	(354) X15 register/Y15 register	(02DE ₁₆) <input type="text" value="??<sub>16</sub>"/>
(338) CAN0 acceptance filter support register	(0244 ₁₆) <input type="text" value="00<sub>16</sub>"/> (Note) (0245 ₁₆) <input type="text" value="01<sub>16</sub>"/>	(355) XY control register	(02E0 ₁₆) <input type="text" value="xxxxxxxx00"/>

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: This applies when the CAN module is supplied with a clock by setting the sleep mode control bit (bit 0 at address 0242₁₆) to 1 after reset.

Figure 1.4.3. Device's internal status after a reset is cleared (7/10)

(356) UART1 special mode register 4	(02E4 ₁₆)	00 ₁₆	(382) Three-phase output buffer register 0	(030A ₁₆)	X ₁₆ X ₁₆ 0 ₁₆ 0 ₁₆ 0 ₁₆ 0 ₁₆
(357) UART1 special mode register 3	(02E5 ₁₆)	00 ₁₆	(383) Three-phase output buffer register 1	(030B ₁₆)	X ₁₆ X ₁₆ 0 ₁₆ 0 ₁₆ 0 ₁₆ 0 ₁₆
(358) UART1 special mode register 2	(02E6 ₁₆)	00 ₁₆	(384) Dead time timer	(030C ₁₆)	?? ₁₆
(359) UART1 special mode register	(02E7 ₁₆)	00 ₁₆	(385) Timer B2 interrupt occurrence frequency set counter	(030D ₁₆)	X ₁₆ X ₁₆ X ₁₆ X ₁₆ ? ₁₆ ? ₁₆ ? ₁₆
(360) UART1 transmit-receive mode register	(02E8 ₁₆)	00 ₁₆	(386) Timer B3 register	(0310 ₁₆)	?? ₁₆
(361) UART1 bit rate generator	(02E9 ₁₆)	?? ₁₆	(387) Timer B4 register	(0311 ₁₆)	?? ₁₆
(362) UART1 transmit buffer register	(02EA ₁₆)	?? ₁₆	(388) Timer B5 register	(0312 ₁₆)	?? ₁₆
	(02EB ₁₆)	X ₁₆ X ₁₆ X ₁₆ X ₁₆ X ₁₆ X ₁₆ ? ₁₆		(0313 ₁₆)	?? ₁₆
(363) UART1 transmit-receive control register 0	(02EC ₁₆)	08 ₁₆		(0314 ₁₆)	?? ₁₆
(364) UART1 transmit-receive control register 1	(02ED ₁₆)	02 ₁₆		(0315 ₁₆)	?? ₁₆
(365) UART1 receive buffer register	(02EE ₁₆)	?? ₁₆	(389) Timer B3 mode register	(031B ₁₆)	0 ₁₆ 0 ₁₆ ? ₁₆ X ₁₆ 0 ₁₆ 0 ₁₆ 0 ₁₆
	(02EF ₁₆)	? ₁₆ ? ₁₆ ? ₁₆ ? ₁₆ X ₁₆ X ₁₆ ? ₁₆	(390) Timer B4 mode register	(031C ₁₆)	0 ₁₆ 0 ₁₆ ? ₁₆ X ₁₆ 0 ₁₆ 0 ₁₆ 0 ₁₆
(366) UART4 special mode register 4	(02F4 ₁₆)	00 ₁₆	(391) Timer B5 mode register	(031D ₁₆)	0 ₁₆ 0 ₁₆ ? ₁₆ 0 ₁₆ 0 ₁₆ 0 ₁₆ 0 ₁₆
(367) UART4 special mode register 3	(02F5 ₁₆)	00 ₁₆	(392) External interrupt cause select register	(031F ₁₆)	00 ₁₆
(368) UART4 special mode register 2	(02F6 ₁₆)	00 ₁₆	(393) UART3 special mode register 4	(0324 ₁₆)	00 ₁₆
(369) UART4 special mode register	(02F7 ₁₆)	00 ₁₆	(394) UART3 special mode register 3	(0325 ₁₆)	00 ₁₆
(370) UART4 transmit-receive mode register	(02F8 ₁₆)	00 ₁₆	(395) UART3 special mode register 2	(0326 ₁₆)	00 ₁₆
(371) UART4 bit rate generator	(02F9 ₁₆)	?? ₁₆	(396) UART3 special mode register	(0327 ₁₆)	00 ₁₆
(372) UART4 transmit buffer register	(02FA ₁₆)	?? ₁₆	(397) UART3 transmit-receive mode register	(0328 ₁₆)	00 ₁₆
	(02FB ₁₆)	X ₁₆ X ₁₆ X ₁₆ X ₁₆ X ₁₆ X ₁₆ ? ₁₆	(398) UART3 bit rate generator	(0329 ₁₆)	?? ₁₆
(373) UART4 transmit-receive control register 0	(02FC ₁₆)	08 ₁₆	(399) UART3 transmit buffer register	(032A ₁₆)	?? ₁₆
(374) UART4 transmit-receive control register 1	(02FD ₁₆)	02 ₁₆		(032B ₁₆)	X ₁₆ X ₁₆ X ₁₆ X ₁₆ X ₁₆ X ₁₆ ? ₁₆
(375) UART4 receive buffer register	(02FE ₁₆)	?? ₁₆	(400) UART3 transmit-receive control register 0	(032C ₁₆)	08 ₁₆
	(02FF ₁₆)	? ₁₆ ? ₁₆ ? ₁₆ ? ₁₆ X ₁₆ X ₁₆ ? ₁₆	(401) UART3 transmit-receive control register 1	(032D ₁₆)	02 ₁₆
(376) Timer B3,B4,B5 count start flag	(0300 ₁₆)	0 ₁₆ 0 ₁₆ 0 ₁₆ X ₁₆ X ₁₆ X ₁₆ X ₁₆	(402) UART3 receive buffer register	(032E ₁₆)	?? ₁₆
(377) Timer A1-1 register	(0302 ₁₆)	?? ₁₆		(032F ₁₆)	? ₁₆ ? ₁₆ ? ₁₆ ? ₁₆ X ₁₆ X ₁₆ ? ₁₆
	(0303 ₁₆)	?? ₁₆	(403) UART2 special mode register 4	(0334 ₁₆)	00 ₁₆
(378) Timer A2-1 register	(0304 ₁₆)	?? ₁₆	(404) UART2 special mode register 3	(0335 ₁₆)	00 ₁₆
	(0305 ₁₆)	?? ₁₆	(405) UART2 special mode register 2	(0336 ₁₆)	00 ₁₆
(379) Timer A4-1 register	(0306 ₁₆)	?? ₁₆	(406) UART2 special mode register	(0337 ₁₆)	00 ₁₆
	(0307 ₁₆)	?? ₁₆	(407) UART2 transmit-receive mode register	(0338 ₁₆)	00 ₁₆
(380) Three-phase PWM control register 0	(0308 ₁₆)	00 ₁₆	(408) UART2 bit rate generator	(0339 ₁₆)	?? ₁₆
(381) Three-phase PWM control register 1	(0309 ₁₆)	00 ₁₆			

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.

Figure 1.4.3. Device's internal status after a reset is cleared (8/10)

(409) UART2 transmit buffer register	(033A ₁₆)	?? ₁₆	(432) Timer B1 mode register	(035C ₁₆)	00?X0000
	(033B ₁₆)	XXXXXX?X?	(433) Timer B2 mode register	(035D ₁₆)	00?X0000
(410) UART2 transmit/receive control register 0	(033C ₁₆)	08 ₁₆	(434) Timer B2 special mode register	(035E ₁₆)	XXXXXXXXXX
(411) UART2 transmit/receive control register 1	(033D ₁₆)	02 ₁₆	(435) Count source prescaler register	(035F ₁₆)	0XXXX0000
(412) UART2 receive buffer register	(033E ₁₆)	?? ₁₆	(436) UART0 pecial mode register 4	(0364 ₁₆)	00 ₁₆
	(033F ₁₆)	??/?/?X?X?	(437) UART0 special mode register 3	(0365 ₁₆)	00 ₁₆
(413) Count start flag	(0340 ₁₆)	00 ₁₆	(438) UART0 special mode register 2	(0366 ₁₆)	00 ₁₆
(414) Clock prescaler reset flag	(0341 ₁₆)	0XXXXXX	(439) UART0 special mode register	(0367 ₁₆)	00 ₁₆
(415) One-shot start flag	(0342 ₁₆)	00 ₁₆	(440) UART0 transmit/receive mode register	(0368 ₁₆)	00 ₁₆
(416) Trigger select register	(0343 ₁₆)	00 ₁₆	(441) UART0 bit rate generator	(0369 ₁₆)	?? ₁₆
(417) Up-down flag	(0344 ₁₆)	00 ₁₆	(442) UART0 transmit buffer register	(036A ₁₆)	?? ₁₆
(418) Timer A0	(0346 ₁₆)	?? ₁₆		(036B ₁₆)	XXXXXXXXX?
	(0347 ₁₆)	?? ₁₆	(443) UART0 transmit/receive control register 0	(036C ₁₆)	08 ₁₆
(419) Timer A1	(0348 ₁₆)	?? ₁₆	(444) UART0 transmit/receive control register 1	(036D ₁₆)	02 ₁₆
	(0349 ₁₆)	?? ₁₆	(445) UART0 receive buffer register	(036E ₁₆)	?? ₁₆
(420) Timer A2	(034A ₁₆)	?? ₁₆		(036F ₁₆)	?/?/?/?X?X?
	(034B ₁₆)	?? ₁₆	(446) PLL control register 0	(0376 ₁₆)	00110100
(421) Timer A3	(034C ₁₆)	?? ₁₆	(447) DMA0 cause select register	(0378 ₁₆)	0X000000
	(034D ₁₆)	?? ₁₆	(448) DMA1 cause select register	(0379 ₁₆)	0X000000
(422) Timer A4	(034E ₁₆)	?? ₁₆	(449) DMA2 cause select register	(037A ₁₆)	0X000000
	(034F ₁₆)	?? ₁₆	(450) DMA3 cause select register	(037B ₁₆)	0X000000
(423) Timer B0	(0350 ₁₆)	?? ₁₆	(451) CRC data register	(037C ₁₆)	?? ₁₆
	(0351 ₁₆)	?? ₁₆		(037D ₁₆)	?? ₁₆
(424) Timer B1	(0352 ₁₆)	?? ₁₆	(452) CRC input register	(037E ₁₆)	?? ₁₆
	(0353 ₁₆)	?? ₁₆	(453) A-D0 register 0	(0380 ₁₆)	?? ₁₆
(425) Timer B2	(0354 ₁₆)	?? ₁₆		(0381 ₁₆)	?? ₁₆
	(0355 ₁₆)	?? ₁₆	(454) A-D0 register 1	(0382 ₁₆)	?? ₁₆
(426) Timer A0 mode register	(0356 ₁₆)	00000X00		(0383 ₁₆)	?? ₁₆
(427) Timer A1 mode register	(0357 ₁₆)	00000X00	(455) A-D0 register 2	(0384 ₁₆)	?? ₁₆
(428) Timer A2 mode register	(0358 ₁₆)	00000X00		(0385 ₁₆)	?? ₁₆
(429) Timer A3 mode register	(0359 ₁₆)	00000X00	(456) A-D0 register 3	(0386 ₁₆)	?? ₁₆
(430) Timer A4 mode register	(035A ₁₆)	00000X00		(0387 ₁₆)	?? ₁₆
(431) Timer B0 mode register	(035B ₁₆)	00?00000			

x : Nothing is mapped to this bit
? : Undefined

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

Figure 1.4.3. Device's internal status after a reset is cleared (9/10)

(457) A-D0 register 4	(0388 ₁₆)	?? ₁₆	(486) Port P9	(03C5 ₁₆)	?? ₁₆
	(0389 ₁₆)	?? ₁₆	(487) Port P8 direction register	(03C6 ₁₆)	00X00000
(458) A-D0 register 5	(038A ₁₆)	?? ₁₆	(488) Port P9 direction register	(03C7 ₁₆)	00 ₁₆
	(038B ₁₆)	?? ₁₆	(489) Port P10	(03C8 ₁₆)	?? ₁₆
(459) A-D0 register 6	(038C ₁₆)	?? ₁₆	(490) Port P11	(Note) (03C9 ₁₆)	XXXX???
	(038D ₁₆)	?? ₁₆	(491) Port P10 direction register	(Note) (03CA ₁₆)	00 ₁₆
(460) A-D0 register 7	(038E ₁₆)	?? ₁₆	(492) Port P11 direction register	(Note) (03CB ₁₆)	XXXX0000
	(038F ₁₆)	?? ₁₆	(493) Port P12	(Note) (03CC ₁₆)	?? ₁₆
(461) A-D0 control register 2	(0394 ₁₆)	X0000000	(494) Port P13	(Note) (03CD ₁₆)	?? ₁₆
(462) A-D0 control register 0	(0396 ₁₆)	00 ₁₆	(495) Port P12 direction register	(Note) (03CE ₁₆)	00 ₁₆
(463) A-D0 control register 1	(0397 ₁₆)	00 ₁₆	(496) Port P13 direction register	(Note) (03CF ₁₆)	00 ₁₆
(464) D-A register 0	(0398 ₁₆)	?? ₁₆	(497) Port P14	(Note) (03D0 ₁₆)	X???
(465) D-A register 1	(039A ₁₆)	?? ₁₆	(498) Port P15	(Note) (03D1 ₁₆)	?? ₁₆
(466) D-A control register	(039C ₁₆)	XXXXXX00	(499) Port P14 direction register	(Note) (03D2 ₁₆)	X0000000
(467) Function select register A8	(Note) (03A0 ₁₆)	XXXXX000	(500) Port P15 direction register	(Note) (03D3 ₁₆)	00 ₁₆
(468) Function select register A9	(Note) (03A1 ₁₆)	00 ₁₆	(501) Pull-up control register 2	(03DA ₁₆)	00 ₁₆
(469) Function select register C	(03AF ₁₆)	00X00000	(502) Pull-up control register 3	(03DB ₁₆)	00 ₁₆
(470) Function select register A0	(03B0 ₁₆)	00 ₁₆	(503) Pull-up control register 4	(Note) (03DC ₁₆)	XXXX0000
(471) Function select register A1	(03B1 ₁₆)	00 ₁₆	(504) Port P0	(03E0 ₁₆)	?? ₁₆
(472) Function select register B0	(03B2 ₁₆)	00 ₁₆	(505) Port P1	(03E1 ₁₆)	?? ₁₆
(473) Function select register B1	(03B3 ₁₆)	00 ₁₆	(506) Port P0 direction register	(03E2 ₁₆)	00 ₁₆
(474) Function select register A2	(03B4 ₁₆)	XXXXX000	(507) Port P1 direction register	(03E3 ₁₆)	00 ₁₆
(475) Function select register A3	(03B5 ₁₆)	00 ₁₆	(508) Port P2	(03E4 ₁₆)	?? ₁₆
(476) Function select register B2	(03B6 ₁₆)	XXXXXX00	(509) Port P3	(03E5 ₁₆)	?? ₁₆
(477) Function select register B3	(03B7 ₁₆)	00 ₁₆	(510) Port P2 direction register	(03E6 ₁₆)	00 ₁₆
(478) Function select register A5	(Note) (03B9 ₁₆)	XXXXX000	(511) Port P3 direction register	(03E7 ₁₆)	00 ₁₆
(479) Function select register A6	(Note) (03BC ₁₆)	00 ₁₆	(512) Port P4	(03E8 ₁₆)	?? ₁₆
(480) Function select register A7	(Note) (03BD ₁₆)	00 ₁₆	(513) Port P5	(03E9 ₁₆)	?? ₁₆
(481) Port P6	(03C0 ₁₆)	?? ₁₆	(514) Port P4 direction register	(03EA ₁₆)	00 ₁₆
(482) Port P7	(03C1 ₁₆)	?? ₁₆	(515) Port P5 direction register	(03EB ₁₆)	00 ₁₆
(483) Port P6 direction register	(03C2 ₁₆)	00 ₁₆	(516) Pull-up control register 0	(03F0 ₁₆)	00 ₁₆
(484) Port P7 direction register	(03C3 ₁₆)	00 ₁₆	(517) Pull-up control register 1	(03F1 ₁₆)	XXXX0000
(485) Port P8	(03C4 ₁₆)	?? ₁₆	(518) Port control register	(03FF ₁₆)	XXXXXXXXX0

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 : This register exists in 144-pin version.

Figure 1.4.3. Device's internal status after a reset is cleared (10/10)

SFR

SFR

Address	Register	
0000 ₁₆		
0001 ₁₆		
0002 ₁₆		
0003 ₁₆		
0004 ₁₆	Processor mode register 0	PM0
0005 ₁₆	Processor mode register 1	PM1
0006 ₁₆	System clock control register 0	CM0
0007 ₁₆	System clock control register 1	CM1
0008 ₁₆	Wait control register	WCR
0009 ₁₆	Address match interrupt control register	AIER
000A ₁₆	Protect register	PRCR
000B ₁₆	External data bus width control register	DS
000C ₁₆	Main clock divided register	MCD
000D ₁₆	Oscillation stop detect register	CM2
000E ₁₆	Watchdog timer start register	WDTS
000F ₁₆	Watchdog timer control register	WDC
0010 ₁₆		
0011 ₁₆	Address match interrupt register 0	RMAD0
0012 ₁₆		
0013 ₁₆		
0014 ₁₆		
0015 ₁₆	Address match interrupt register 1	RAMD1
0016 ₁₆		
0017 ₁₆	VDC control register for PLL	PLV
0018 ₁₆		
0019 ₁₆	Address match interrupt register 2	RAMD2
001A ₁₆		
001B ₁₆	VDC control register 1	VDC1 *
001C ₁₆		
001D ₁₆	Address match interrupt register 3	RAMD3
001E ₁₆		
001F ₁₆	VDC control register 0	VDC0 *
0020 ₁₆		
0021 ₁₆	Emulator interrupt vector table register	EIAD0 *
0022 ₁₆		
0023 ₁₆	Emulator interrupt detect register	EITD *
0024 ₁₆	Emulator protect register	EPRR *
0025 ₁₆		
0026 ₁₆		
0027 ₁₆		
0028 ₁₆		
0029 ₁₆		
002A ₁₆		
002B ₁₆		
002C ₁₆		
002D ₁₆		
002E ₁₆		
002F ₁₆		

Address	Register	
0030 ₁₆	ROM area set register	ROA *
0031 ₁₆	Debug moritor area set register	DBA *
0032 ₁₆	Expansion area set register 0	EXA0 *
0033 ₁₆	Expansion area set register 1	EXA1 *
0034 ₁₆	Expansion area set register 2	EXA2 *
0035 ₁₆	Expansion area set register 3	EXA3 *
0036 ₁₆		
0037 ₁₆		
0038 ₁₆		
0039 ₁₆		
003A ₁₆		
003B ₁₆		
003C ₁₆		
003D ₁₆		
003E ₁₆		
003F ₁₆		
0040 ₁₆	DRAM control register	DRAMCONT
0041 ₁₆	DRAM refresh interval set register	REFCNT
0042 ₁₆		
0043 ₁₆		
0044 ₁₆		
0045 ₁₆		
0046 ₁₆		
0047 ₁₆		
0048 ₁₆		
0049 ₁₆		
004A ₁₆		
004B ₁₆		
004C ₁₆		
004D ₁₆		
004E ₁₆		
004F ₁₆		
0050 ₁₆		
0051 ₁₆		
0052 ₁₆		
0053 ₁₆		
0054 ₁₆		
0055 ₁₆	Flash memory control register 2	FMR2 *
0056 ₁₆	Flash memory control register 1	FMR1 *
0057 ₁₆	Flash memory control register 0	FMR0
0058 ₁₆		
0059 ₁₆		
005A ₁₆		
005B ₁₆		
005C ₁₆		
005D ₁₆		
005E ₁₆		
005F ₁₆		

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

*: User cannot use this. Do not access to the register.

Address	Register
0060 ₁₆	
0061 ₁₆	
0062 ₁₆	
0063 ₁₆	
0064 ₁₆	
0065 ₁₆	
0066 ₁₆	
0067 ₁₆	
0068 ₁₆	DMA0 interrupt control register DM0IC
0069 ₁₆	Timer B5 interrupt control register TB5IC
006A ₁₆	DMA2 interrupt control register DM2IC
006B ₁₆	UART2 receive /ACK interrupt control register S2RIC
006C ₁₆	Timer A0 interrupt control register TA0IC
006D ₁₆	UART3 receive /ACK interrupt control register S3RIC
006E ₁₆	Timer A2 interrupt control register TA2IC
006F ₁₆	UART4 receive /ACK interrupt control register S4RIC
0070 ₁₆	Timer A4 interrupt control register TA4IC
0071 ₁₆	UART0/UART3 bus collision detection interrupt control register BCN0IC
0072 ₁₆	UART0 receive/ACK interrupt control register S0RIC
0073 ₁₆	A-D0 interrupt control register AD0IC
0074 ₁₆	UART1 receive/ACK interrupt control register S1RIC
0075 ₁₆	Intelligent I/O interrupt control register 0 IIO0IC
0076 ₁₆	Timer B1 interrupt control register TB1IC
0077 ₁₆	Intelligent I/O interrupt control register 2 IIO2IC
0078 ₁₆	Timer B3 interrupt control register TB3IC
0079 ₁₆	Intelligent I/O interrupt control register 4 IIO4IC
007A ₁₆	INT5 interrupt control register INT5IC
007B ₁₆	Intelligent I/O interrupt control register 6 IIO6IC
007C ₁₆	INT3 interrupt control register INT3IC
007D ₁₆	Intelligent I/O interrupt control register 8 IIO8IC
007E ₁₆	INT1 interrupt control register INT1IC
007F ₁₆	Intelligent I/O interrupt control register 10/ CAN interrupt 1 control register IIO10IC CAN1IC
0080 ₁₆	
0081 ₁₆	Intelligent I/O interrupt control register 11/ CAN interrupt 2 control register IIO11IC CAN2IC
0082 ₁₆	
0083 ₁₆	
0084 ₁₆	
0085 ₁₆	
0086 ₁₆	A-D1 interrupt control register AD1IC
0087 ₁₆	
0088 ₁₆	DMA1 interrupt control register DM1IC
0089 ₁₆	UART2 transmit /NACK interrupt control register S2TIC
008A ₁₆	DMA3 interrupt control register DM3IC
008B ₁₆	UART3 transmit /NACK interrupt control register S3TIC
008C ₁₆	Timer A1 interrupt control register TA1IC
008D ₁₆	UART4 transmit /NACK interrupt control register S4TIC
008E ₁₆	Timer A3 interrupt control register TA3IC
008F ₁₆	UART2 bus collision detection interrupt control register BCN2IC

Address	Register
0090 ₁₆	UART0 transmit /NACK interrupt control register S0TIC
0091 ₁₆	UART1/UART4 bus collision detection interrupt control register BCN1IC
0092 ₁₆	UART1 transmit/NACK interrupt control register S1TIC
0093 ₁₆	Key input interrupt control register KUPIC
0094 ₁₆	Timer B0 interrupt control register TBOIC
0095 ₁₆	Intelligent I/O interrupt control register 1 IIO1IC
0096 ₁₆	Timer B2 interrupt control register TB2IC
0097 ₁₆	Intelligent I/O interrupt control register 3 IIO3IC
0098 ₁₆	Timer B4 interrupt control register TB4IC
0099 ₁₆	Intelligent I/O interrupt control register 5 IIO5IC
009A ₁₆	INT4 interrupt control register INT4IC
009B ₁₆	Intelligent I/O interrupt control register 7 IIO7IC
009C ₁₆	INT2 interrupt control register INT2IC
009D ₁₆	Intelligent I/O interrupt control register 9/ CAN interrupt 0 control register IIO9IC CAN0IC
009E ₁₆	INT0 interrupt control register INTOIC
009F ₁₆	Exit priority register RLVL
00A0 ₁₆	Interrupt request register 0 IIO0IR
00A1 ₁₆	Interrupt request register 1 IIO1IR
00A2 ₁₆	Interrupt request register 2 IIO2IR
00A3 ₁₆	Interrupt request register 3 IIO3IR
00A4 ₁₆	Interrupt request register 4 IIO4IR
00A5 ₁₆	Interrupt request register 5 IIO5IR
00A6 ₁₆	Interrupt request register 6 IIO6IR
00A7 ₁₆	Interrupt request register 7 IIO7IR
00A8 ₁₆	Interrupt request register 8 IIO8IR
00A9 ₁₆	Interrupt request register 9 IIO9IR
00AA ₁₆	Interrupt request register 10 IIO10IR
00AB ₁₆	Interrupt request register 11 IIO11IR
00AC ₁₆	
00AD ₁₆	
00AE ₁₆	
00AF ₁₆	
00B0 ₁₆	Interrupt enable register 0 IIO0IE
00B1 ₁₆	Interrupt enable register 1 IIO1IE
00B2 ₁₆	Interrupt enable register 2 IIO2IE
00B3 ₁₆	Interrupt enable register 3 IIO3IE
00B4 ₁₆	Interrupt enable register 4 IIO4IE
00B5 ₁₆	Interrupt enable register 5 IIO5IE
00B6 ₁₆	Interrupt enable register 6 IIO6IE
00B7 ₁₆	Interrupt enable register 7 IIO7IE
00B8 ₁₆	Interrupt enable register 8 IIO8IE
00B9 ₁₆	Interrupt enable register 9 IIO9IE
00BA ₁₆	Interrupt enable register 10 IIO10IE
00BB ₁₆	Interrupt enable register 11 IIO11IE
00BC ₁₆	
00BD ₁₆	
00BE ₁₆	
00BF ₁₆	

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

Address	Register	
00C0 ₁₆	Group 0 TM /WG register 0	G0TM0/G0PO0
00C1 ₁₆		
00C2 ₁₆	Group 0 TM /WG register 1	G0TM1/G0PO1
00C3 ₁₆		
00C4 ₁₆	Group 0 TM /WG register 2	G0TM2/G0PO2
00C5 ₁₆		
00C6 ₁₆	Group 0 TM /WG register 3	G0TM3/G0PO3
00C7 ₁₆		
00C8 ₁₆	Group 0 TM /WG register 4	G0TM4/G0PO4
00C9 ₁₆		
00CA ₁₆	Group 0 TM /WG register 5	G0TM5/G0PO5
00CB ₁₆		
00CC ₁₆	Group 0 TM /WG register 6	G0TM6/G0PO6
00CD ₁₆		
00CE ₁₆	Group 0 TM /WG register 7	G0TM7/G0PO7
00CF ₁₆		
00D0 ₁₆	Group 0 waveform generate control register 0	G0POCR0
00D1 ₁₆	Group 0 waveform generate control register 1	G0POCR1
00D2 ₁₆		
00D3 ₁₆		
00D4 ₁₆	Group 0 waveform generate control register 4	G0POCR4
00D5 ₁₆	Group 0 waveform generate control register 5	G0POCR5
00D6 ₁₆		
00D7 ₁₆		
00D8 ₁₆	Group 0 time measurement control register 0	G0TMCR0
00D9 ₁₆	Group 0 time measurement control register 1	G0TMCR1
00DA ₁₆	Group 0 time measurement control register 2	G0TMCR2
00DB ₁₆	Group 0 time measurement control register 3	G0TMCR3
00DC ₁₆	Group 0 time measurement control register 4	G0TMCR4
00DD ₁₆	Group 0 time measurement control register 5	G0TMCR5
00DE ₁₆	Group 0 time measurement control register 6	G0TMCR6
00DF ₁₆	Group 0 time measurement control register 7	G0TMCR7
00E0 ₁₆	Group 0 base timer register	G0BT
00E1 ₁₆		
00E2 ₁₆	Group 0 base timer control register 0	G0BCR0
00E3 ₁₆	Group 0 base timer control register 1	G0BCR1
00E4 ₁₆	Group 0 time measurement prescaler register 6	G0TPR6
00E5 ₁₆	Group 0 time measurement prescaler register 7	G0TPR7
00E6 ₁₆	Group 0 function enable register	G0FE
00E7 ₁₆	Group 0 function select register	G0FS
00E8 ₁₆	Group 0 SI/O receive buffer register	G0BF
00E9 ₁₆		
00EA ₁₆	Group 0 transmit buffer/receive data register	G0DR
00EB ₁₆		
00EC ₁₆	Group 0 receive input register	G0RI
00ED ₁₆	Group 0 SI/O communication mode register	G0MR
00EE ₁₆	Group 0 transmit output register	G0TO
00EF ₁₆	Group 0 SI/O communication control register	G0CR

Address	Register	
00F0 ₁₆	Group 0 data compare register 0	G0CMP0
00F1 ₁₆	Group 0 data compare register 1	G0CMP1
00F2 ₁₆	Group 0 data compare register 2	G0CMP2
00F3 ₁₆	Group 0 data compare register 3	G0CMP3
00F4 ₁₆	Group 0 data mask register 0	G0MSK0
00F5 ₁₆	Group 0 data mask register 1	G0MSK1
00F6 ₁₆		
00F7 ₁₆		
00F8 ₁₆	Group 0 receive CRC code register	G0RCRC
00F9 ₁₆		
00FA ₁₆	Group 0 transmit CRC code register	G0TCRC
00FB ₁₆		
00FC ₁₆	Group 0 SI/O expansion mode register	G0EMR
00FD ₁₆	Group 0 SI/O expansion receive control register	G0ERC
00FE ₁₆	Group 0 SI/O special communication interrupt detect register	G0IRF
00FF ₁₆	Group 0 SI/O expansion transmit control register	G0ETC
0100 ₁₆		
0101 ₁₆	Group 1 TM /WG register 0	G1TM0/G1PO0
0102 ₁₆	Group 1 TM /WG register 1	G1TM1/G1PO1
0103 ₁₆		
0104 ₁₆	Group 1 TM /WG register 2	G1TM2/G1PO2
0105 ₁₆		
0106 ₁₆	Group 1 TM /WG register 3	G1TM3/G1PO3
0107 ₁₆		
0108 ₁₆	Group 1 TM /WG register 4	G1TM4/G1PO4
0109 ₁₆		
010A ₁₆	Group 1 TM /WG register 5	G1TM5/G1PO5
010B ₁₆		
010C ₁₆	Group 1 TM /WG register 6	G1TM6/G1PO6
010D ₁₆		
010E ₁₆	Group 1 TM /WG register 7	G1TM7/G1PO7
010F ₁₆		
0110 ₁₆	Group 1 waveform generate control register 0	G1POCR0
0111 ₁₆	Group 1 waveform generate control register 1	G1POCR1
0112 ₁₆	Group 1 waveform generate control register 2	G1POCR2
0113 ₁₆	Group 1 waveform generate control register 3	G1POCR3
0114 ₁₆	Group 1 waveform generate control register 4	G1POCR4
0115 ₁₆	Group 1 waveform generate control register 5	G1POCR5
0116 ₁₆	Group 1 waveform generate control register 6	G1POCR6
0117 ₁₆	Group 1 waveform generate control register 7	G1POCR7
0118 ₁₆		
0119 ₁₆	Group 1 time measurement control register 1	G1TMCR1
011A ₁₆	Group 1 time measurement control register 2	G1TMCR2
011B ₁₆		
011C ₁₆		
011D ₁₆		
011E ₁₆	Group 1 time measurement control register 6	G1TMCR6
011F ₁₆	Group 1 time measurement control register 7	G1TMCR7

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

Address	Register	Address	Register
0120 ₁₆	Group 1 base timer register	0150 ₁₆	Group 2 waveform generate control register 0
0121 ₁₆		G1BT	Group 2 waveform generate control register 1
0122 ₁₆	Group 1 base timer control register 0	0152 ₁₆	Group 2 waveform generate control register 2
0123 ₁₆	Group 1 base timer control register 1	0153 ₁₆	Group 2 waveform generate control register 3
0124 ₁₆	Group 1 time measurement prescaler register 6	0154 ₁₆	Group 2 waveform generate control register 4
0125 ₁₆	Group 1 time measurement prescaler register 7	0155 ₁₆	Group 2 waveform generate control register 5
0126 ₁₆	Group 1 function enable register	0156 ₁₆	Group 2 waveform generate control register 6
0127 ₁₆	Group 1 function select register	0157 ₁₆	Group 2 waveform generate control register 7
0128 ₁₆	Group 1 SI/O receive buffer register	0158 ₁₆	
0129 ₁₆		G1BF	0159 ₁₆
012A ₁₆	Group 1 transmit buffer/receive data register	015A ₁₆	
012B ₁₆		015B ₁₆	
012C ₁₆	Group 1 receive input register	015C ₁₆	
012D ₁₆	Group 1 SI/O communication mode register	015D ₁₆	
012E ₁₆	Group 1 transmit output register	015E ₁₆	
012F ₁₆	Group 1 SI/O communication control register	015F ₁₆	
0130 ₁₆	Group 1 data compare register 0	0160 ₁₆	Group 2 base timer register
0131 ₁₆	Group 1 data compare register 1	0161 ₁₆	
0132 ₁₆	Group 1 data compare register 2	0162 ₁₆	Group 2 base timer control register 0
0133 ₁₆	Group 1 data compare register 3	0163 ₁₆	Group 2 base timer control register 1
0134 ₁₆	Group 1 data mask register 0	0164 ₁₆	Base timer start register
0135 ₁₆	Group 1 data mask register 1	0165 ₁₆	
0136 ₁₆		0166 ₁₆	Group 2 function enable register
0137 ₁₆		0167 ₁₆	Group 2 RTP output buffer register
0138 ₁₆	Group 1 receive CRC code register	0168 ₁₆	
0139 ₁₆		G1RCRC	0169 ₁₆
013A ₁₆	Group 1 transmit CRC code register	016A ₁₆	Group 2 SI/O communication mode register
013B ₁₆		G1TCRC	016B ₁₆
013C ₁₆	Group 1 SI/O expansion mode register	016C ₁₆	Group 2 SI/O transmit buffer register
013D ₁₆	Group 1 SI/O expansion receive control register	016D ₁₆	
013E ₁₆	Group 1 SI/O special communication interrupt detect register	016E ₁₆	Group 2 SI/O receive buffer register
013F ₁₆	Group 1 SI/O expansion transmit control register	016F ₁₆	
0140 ₁₆	Group 2 waveform generate register 0	0170 ₁₆	Group 2 IEBus address register
0141 ₁₆		G2PO0	
0142 ₁₆	Group 2 waveform generate register 1	0172 ₁₆	Group 2 IEBus control register
0143 ₁₆		G2PO1	0173 ₁₆
0144 ₁₆	Group 2 waveform generate register 2	0174 ₁₆	Group 2 IEBus receive interrupt cause detect register
0145 ₁₆		G2PO2	0175 ₁₆
0146 ₁₆	Group 2 waveform generate register 3	0176 ₁₆	
0147 ₁₆		G2PO3	0177 ₁₆
0148 ₁₆	Group 2 waveform generate register 4	0178 ₁₆	Input function select register
0149 ₁₆		G2PO4	0179 ₁₆
014A ₁₆	Group 2 waveform generate register 5	017A ₁₆	Group 3 SI/O communication mode register
014B ₁₆		G2PO5	017B ₁₆
014C ₁₆	Group 2 waveform generate register 6	017C ₁₆	Group 3 SI/O transmit buffer register
014D ₁₆		G2PO6	
014E ₁₆	Group 2 waveform generate register 7	017E ₁₆	Group 3 SI/O receive buffer register
014F ₁₆		G2PO7	

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

Address	Register	Address	Register
0180 ₁₆ 0181 ₁₆	Group 3 waveform generate register 0	G3PO0	
0182 ₁₆ 0183 ₁₆	Group 3 waveform generate register 1	G3PO1	
0184 ₁₆ 0185 ₁₆	Group 3 waveform generate register 2	G3PO2	
0186 ₁₆ 0187 ₁₆	Group 3 waveform generate register 3	G3PO3	
0188 ₁₆ 0189 ₁₆	Group 3 waveform generate register 4	G3PO4	
018A ₁₆ 018B ₁₆	Group 3 waveform generate register 5	G3PO5	
018C ₁₆ 018D ₁₆	Group 3 waveform generate register 6	G3PO6	
018E ₁₆ 018F ₁₆	Group 3 waveform generate register 7	G3PO7	
0190 ₁₆	Group 3 waveform generate control register 0	G3POCR0	
0191 ₁₆	Group 3 waveform generate control register 1	G3POCR1	
0192 ₁₆	Group 3 waveform generate control register 2	G3POCR2	
0193 ₁₆	Group 3 waveform generate control register 3	G3POCR3	
0194 ₁₆	Group 3 waveform generate control register 4	G3POCR4	
0195 ₁₆	Group 3 waveform generate control register 5	G3POCR5	
0196 ₁₆	Group 3 waveform generate control register 6	G3POCR6	
0197 ₁₆	Group 3 waveform generate control register 7	G3POCR7	
0198 ₁₆ 0199 ₁₆	Group 3 waveform generate mask register 4	G3MK4	
019A ₁₆ 019B ₁₆	Group 3 waveform generate mask register 5	G3MK5	
019C ₁₆ 019D ₁₆	Group 3 waveform generate mask register 6	G3MK6	
019E ₁₆ 019F ₁₆	Group 3 waveform generate mask register 7	G3MK7	
01A0 ₁₆ 01A1 ₁₆	Group 3 base timer register	G3BT	
01A2 ₁₆	Group 3 base timer control register 0	G3BCR0	
01A3 ₁₆	Group 3 base timer control register 1	G3BCR1	
01A4 ₁₆			
01A5 ₁₆			
01A6 ₁₆	Group 3 function enable register	G3FE	
01A7 ₁₆	Group 3 RTP output buffer register	G3RTP	
01A8 ₁₆			
01A9 ₁₆			
01AA ₁₆			
01AB ₁₆	Group 3 high-speed HDLC communication control register 1	HDLC1	
01AC ₁₆	Group 3 high-speed HDLC communication control register	HDLC	
01AD ₁₆	Group 3 high-speed HDLC communication register	HDLCF	
01AE ₁₆ 01AF ₁₆	Group 3 high-speed HDLC transmit counter	HDLCC	
01B0 ₁₆ 01B1 ₁₆	Group 3 high-speed HDLC data compare register 0	HDLC _{CP} 0	
01B2 ₁₆ 01B3 ₁₆	Group 3 high-speed HDLC data mask register 0	HDLC _{CM} 0	
01B4 ₁₆ 01B5 ₁₆	Group 3 high-speed HDLC data compare register1	HDLC _{CP} 1	
01B6 ₁₆ 01B7 ₁₆	Group 3 high-speed HDLC data mask register 1	HDLC _{CM} 1	
01B8 ₁₆ 01B9 ₁₆	Group 3 high-speed HDLC data compare register 2	HDLC _{CP} 2	
01BA ₁₆ 01BB ₁₆	Group 3 high-speed HDLC data mask register 2	HDLC _{CM} 2	
01BC ₁₆ 01BD ₁₆	Group 3 high-speed HDLC data compare register 3	HDLC _{CP} 3	
01BE ₁₆ 01BF ₁₆	Group 3 high-speed HDLC data mask register 3	HDLC _{CM} 3	
01C0 ₁₆	A-D1 register 0		AD10
01C1 ₁₆			
01C2 ₁₆	A-D1 register 1		AD11
01C3 ₁₆			
01C4 ₁₆	A-D1 register 2		AD12
01C5 ₁₆			
01C6 ₁₆	A-D1 register 3		AD13
01C7 ₁₆			
01C8 ₁₆	A-D1 register 4		AD14
01C9 ₁₆			
01CA ₁₆	A-D1 register 5		AD15
01CB ₁₆			
01CC ₁₆	A-D1 register 6		AD16
01CD ₁₆			
01CE ₁₆	A-D1 register 7		AD17
01CF ₁₆			
01D0 ₁₆			
01D1 ₁₆			
01D2 ₁₆			
01D3 ₁₆			
01D4 ₁₆	A-D1 control register 2		AD1 _{CON} 2
01D5 ₁₆			
01D6 ₁₆	A-D1 control register 0		AD1 _{CON} 0
01D7 ₁₆	A-D1 control register 1		AD1 _{CON} 1
01D8 ₁₆			
01D9 ₁₆			
01DA ₁₆			
01DB ₁₆			
01DC ₁₆			
01DD ₁₆			
01DE ₁₆			
01DF ₁₆			

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

Address	Register	
01E0 ₁₆	CAN0 message slot buffer 0 standard ID0	C0SLOT0_0
01E1 ₁₆	CAN0 message slot buffer 0 standard ID1	C0SLOT0_1
01E2 ₁₆	CAN0 message slot buffer 0 extend ID0	C0SLOT0_2
01E3 ₁₆	CAN0 message slot buffer 0 extend ID1	C0SLOT0_3
01E4 ₁₆	CAN0 message slot buffer 0 extend ID2	C0SLOT0_4
01E5 ₁₆	CAN0 message slot buffer 0 data length code	C0SLOT0_5
01E6 ₁₆	CAN0 message slot buffer 0 data 0	C0SLOT0_6
01E7 ₁₆	CAN0 message slot buffer 0 data 1	C0SLOT0_7
01E8 ₁₆	CAN0 message slot buffer 0 data 2	C0SLOT0_8
01E9 ₁₆	CAN0 message slot buffer 0 data 3	C0SLOT0_9
01EA ₁₆	CAN0 message slot buffer 0 data 4	C0SLOT0_10
01EB ₁₆	CAN0 message slot buffer 0 data 5	C0SLOT0_11
01EC ₁₆	CAN0 message slot buffer 0 data 6	C0SLOT0_12
01ED ₁₆	CAN0 message slot buffer 0 data 7	C0SLOT0_13
01EE ₁₆	CAN0 message slot buffer 0 time stamp high	C0SLOT0_14
01EF ₁₆	CAN0 message slot buffer 0 time stamp low	C0SLOT0_15
01F0 ₁₆	CAN0 message slot buffer 1 standard ID0	C0SLOT1_0
01F1 ₁₆	CAN0 message slot buffer 1 standard ID1	C0SLOT1_1
01F2 ₁₆	CAN0 message slot buffer 1 extend ID0	C0SLOT1_2
01F3 ₁₆	CAN0 message slot buffer 1 extend ID1	C0SLOT1_3
01F4 ₁₆	CAN0 message slot buffer 1 extend ID2	C0SLOT1_4
01F5 ₁₆	CAN0 message slot buffer 1 data length code	C0SLOT1_5
01F6 ₁₆	CAN0 message slot buffer 1 data 0	C0SLOT1_6
01F7 ₁₆	CAN0 message slot buffer 1 data 1	C0SLOT1_7
01F8 ₁₆	CAN0 message slot buffer 1 data 2	C0SLOT1_8
01F9 ₁₆	CAN0 message slot buffer 1 data 3	C0SLOT1_9
01FA ₁₆	CAN0 message slot buffer 1 data 4	C0SLOT1_10
01FB ₁₆	CAN0 message slot buffer 1 data 5	C0SLOT1_11
01FC ₁₆	CAN0 message slot buffer 1 data 6	C0SLOT1_12
01FD ₁₆	CAN0 message slot buffer 1 data 7	C0SLOT1_13
01FE ₁₆	CAN0 message slot buffer 1 time stamp high	C0SLOT1_14
01FF ₁₆	CAN0 message slot buffer 1 time stamp low	C0SLOT1_15
0200 ₁₆		
0201 ₁₆	CAN0 control register 0	C0CTRL0
0202 ₁₆		
0203 ₁₆	CAN0 status register	C0STR
0204 ₁₆		
0205 ₁₆	CAN0 expansion ID register	C0IDR
0206 ₁₆		
0207 ₁₆	CAN0 configuration register	C0CONR
0208 ₁₆		
0209 ₁₆	CAN0 time stamp register	C0TSR
020A ₁₆	CAN0 transmit error count register	C0TEC
020B ₁₆	CAN0 receive error count register	C0REC
020C ₁₆		
020D ₁₆	CAN0 slot interrupt status register	C0SISTR
020E ₁₆		
020F ₁₆		

Address	Register	
0210 ₁₆		
0211 ₁₆	CAN0 slot interrupt mask register	C0SIMKR
0212 ₁₆		
0213 ₁₆		
0214 ₁₆	CAN0 error interrupt mask register	C0EIMKR
0215 ₁₆	CAN0 error interrupt status register	C0EISTR
0216 ₁₆		
0217 ₁₆	CAN0 baud rate prescaler	C0BPR
0218 ₁₆		
0219 ₁₆		
021A ₁₆		
021B ₁₆		
021C ₁₆		
021D ₁₆		
021E ₁₆		
021F ₁₆		
0220 ₁₆		
0221 ₁₆		
0222 ₁₆		
0223 ₁₆		
0224 ₁₆		
0225 ₁₆		
0226 ₁₆		
0227 ₁₆		
0228 ₁₆	CAN0 global mask register standard ID0	C0GMR0
0229 ₁₆	CAN0 global mask register standard ID1	C0GMR1
022A ₁₆	CAN0 global mask register extend ID0	C0GMR2
022B ₁₆	CAN0 global mask register extend ID1	C0GMR3
022C ₁₆	CAN0 global mask register extend ID2	C0GMR4
022D ₁₆		
022E ₁₆		
022F ₁₆		
0230 ₁₆	CAN0 message slot 0 control register / CAN0 local mask register A standard ID0	C0MCTL0/ C0LMAR0
0231 ₁₆	CAN0 message slot 1 control register / CAN0 local mask register A standard ID1	C0MCTL1/ C0LMAR1
0232 ₁₆	CAN0 message slot 2 control register / CAN0 local mask register A extend ID0	C0MCTL2/ C0LMAR2
0233 ₁₆	CAN0 message slot 3 control register / CAN0 local mask register A extend ID1	C0MCTL3/ C0LMAR3
0234 ₁₆	CAN0 message slot 4 control register / CAN0 local mask register A extend ID2	C0MCTL4/ C0LMAR4
0235 ₁₆	CAN0 message slot 5 control register	C0MCTL5
0236 ₁₆	CAN0 message slot 6 control register	C0MCTL6
0237 ₁₆	CAN0 message slot 7 control register	C0MCTL7
0238 ₁₆	CAN0 message slot 8 control register / CAN0 local mask register B standard ID0	C0MCTL8/ C0LMBR0

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

Note 1: CAN0 message slot i control registers (i=0 to 15) are allocated to addresses 0230₁₆ to 023F₁₆ by switching banks.

Address	Register	Address	Register
0239 ₁₆	CAN0 message slot 9 control register / CAN0 local mask register B standard ID1	02C0 ₁₆	X0 register/Y0 register
		02C1 ₁₆	X0R/Y0R
023A ₁₆	CAN0 message slot 10 control register / CAN0 local mask register B extend ID0	02C2 ₁₆	X1 register/Y1 register
		02C3 ₁₆	X1R/Y1R
023B ₁₆	CAN0 message slot 11 control register / CAN0 local mask register B extend ID1	02C4 ₁₆	X2 register/Y2 register
		02C5 ₁₆	X2R/Y2R
023C ₁₆	CAN0 message slot 12 control register / CAN0 local mask register B extend ID2	02C6 ₁₆	X3 register/Y3 register
		02C7 ₁₆	X3R/Y3R
023D ₁₆	CAN0 message slot 13 control register	02C8 ₁₆	X4 register/Y4 register
023E ₁₆	CAN0 message slot 14 control register	02C9 ₁₆	X4R/Y4R
023F ₁₆	CAN0 message slot 15 control register	02CA ₁₆	X5 register/Y5 register
0240 ₁₆	CAN0 slot buffer select register	02CB ₁₆	X5R/Y5R
0241 ₁₆	CAN0 control register 1	02CC ₁₆	X6 register/Y6 register
0242 ₁₆	CAN0 sleep control register	02CD ₁₆	X6R/Y6R
0243 ₁₆		02CE ₁₆	X7 register/Y7 register
0244 ₁₆		02CF ₁₆	X7R/Y7R
0245 ₁₆	CAN0 acceptance filter support register	02D0 ₁₆	X8 register/Y8 register
		02D1 ₁₆	X8R/Y8R
		02D2 ₁₆	X9 register/Y9 register
		02D3 ₁₆	X9R/Y9R
		02D4 ₁₆	X10 register/Y10 register
		02D5 ₁₆	X10R/Y10R
		02D6 ₁₆	X11 register/Y11 register
		02D7 ₁₆	X11R/Y11R
		02D8 ₁₆	X12 register/Y12 register
		02D9 ₁₆	X12R/Y12R
		02DA ₁₆	X13 register/Y13 register
		02DB ₁₆	X13R/Y13R
		02DC ₁₆	X14 register/Y14 register
		02DD ₁₆	X14R/Y14R
		02DE ₁₆	X15 register/Y15 register
		02DF ₁₆	X15R/Y15R
		02E0 ₁₆	XY control register
			XYC
		02E1 ₁₆	
		02E2 ₁₆	
		02E3 ₁₆	
		02E4 ₁₆	UART1 special mode register 4
			U1SMR4
		02E5 ₁₆	UART1 special mode register 3
			U1SMR3
		02E6 ₁₆	UART1 special mode register 2
			U1SMR2
		02E7 ₁₆	UART1 special mode register
			U1SMR
		02E8 ₁₆	UART1 transmit-receive mode register
			U1MR
		02E9 ₁₆	UART1 bit rate generator
			U1BRG
		02EA ₁₆	UART1 transmit buffer register
		02EB ₁₆	UART1 transmit buffer register
			U1TB
		02EC ₁₆	UART1 transmit-receive control register 0
			U1C0
		02ED ₁₆	UART1 transmit-receive control register 1
			U1C1
		02EE ₁₆	
		02EF ₁₆	UART1 receive buffer register
			U1RB

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

Address	Register	
02F0 ₁₆		
02F1 ₁₆		
02F2 ₁₆		
02F3 ₁₆		
02F4 ₁₆	UART4 special mode register 4	U4SMR4
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		
0300 ₁₆	Timer B3,B4,B5 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	Three-phase output buffer register 0	IDB0
030B ₁₆	Three-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	External interrupt cause select register	IFSR

Address	Register	
0320 ₁₆		
0321 ₁₆		
0322 ₁₆		
0323 ₁₆		
0324 ₁₆	UART3 special mode register 4	U3SMR4
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	UART2 special mode register 4	U2SMR4
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		
0340 ₁₆	Count start flag	TABS
0341 ₁₆	Clock prescaler reset flag	CPSRF
0342 ₁₆	One-shot start flag	ONSF
0343 ₁₆	Trigger select register	TRGSR
0344 ₁₆	Up-down flag	UDF
0345 ₁₆		
0346 ₁₆	Timer A0 register	TA0
0347 ₁₆		
0348 ₁₆	Timer A1 register	TA1
0349 ₁₆		
034A ₁₆	Timer A2 register	TA2
034B ₁₆		
034C ₁₆	Timer A3 register	TA3
034D ₁₆		
034E ₁₆	Timer A4 register	TA4
034F ₁₆		

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

Address	Register	
0350 ¹⁶	Timer B0 register	TB0
0351 ¹⁶		
0352 ¹⁶	Timer B1 register	TA1
0353 ¹⁶		
0354 ¹⁶	Timer B2 register	TA2
0355 ¹⁶		
0356 ¹⁶	Timer A0 mode register	TA0MR
0357 ¹⁶	Timer A1 mode register	TA1MR
0358 ¹⁶	Timer A2 mode register	TA2MR
0359 ¹⁶	Timer A3 mode register	TA3MR
035A ¹⁶	Timer A4 mode register	TA4MR
035B ¹⁶	Timer B0 mode register	TB0MR
035C ¹⁶	Timer B1 mode register	TB1MR
035D ¹⁶	Timer B2 mode register	TB2MR
035E ¹⁶	Timer B2 special mode register	TB2SC
035F ¹⁶	Count source prescaler register	TCSPR
0360 ¹⁶		
0361 ¹⁶		
0362 ¹⁶		
0363 ¹⁶		
0364 ¹⁶	UART0 special mode register 4	U0SMR4
0365 ¹⁶	UART0 special mode register 3	U0SMR3
0366 ¹⁶	UART0 special mode register 2	U0SMR2
0367 ¹⁶	UART0 special mode register	U0SMR
0368 ¹⁶	UART0 transmit/receive mode register	U0MR
0369 ¹⁶	UART0 bit rate generator	U0BRG
036A ¹⁶	UART0 transmit buffer register	U0TB
036B ¹⁶		
036C ¹⁶	UART0 transmit/receive control register 0	U0C0
036D ¹⁶	UART0 transmit/receive control register 1	U0C1
036E ¹⁶	UART0 receive buffer register	U0RB
036F ¹⁶		
0370 ¹⁶		
0371 ¹⁶		
0372 ¹⁶		
0373 ¹⁶		
0374 ¹⁶		
0375 ¹⁶		
0376 ¹⁶	PLL control register 0	PLC0
0377 ¹⁶		
0378 ¹⁶	DMA0 cause select register	DM0SL
0379 ¹⁶	DMA1 cause select register	DM1SL
037A ¹⁶	DMA2 cause select register	DM2SL
037B ¹⁶	DMA3 cause select register	DM3SL
037C ¹⁶	CRC data register	CRCD
037D ¹⁶		
037E ¹⁶	CRC input register	CRCIN
037F ¹⁶		

Address	Register	
0380 ¹⁶	A-D0 register 0	AD00
0381 ¹⁶		
0382 ¹⁶	A-D0 register 1	AD01
0383 ¹⁶		
0384 ¹⁶	A-D0 register 2	AD02
0385 ¹⁶		
0386 ¹⁶	A-D0 register 3	AD03
0387 ¹⁶		
0388 ¹⁶	A-D0 register 4	AD04
0389 ¹⁶		
038A ¹⁶	A-D0 register 5	AD05
038B ¹⁶		
038C ¹⁶	A-D0 register 6	AD06
038D ¹⁶		
038E ¹⁶	A-D0 register 7	AD07
038F ¹⁶		
0390 ¹⁶		
0391 ¹⁶		
0392 ¹⁶		
0393 ¹⁶		
0394 ¹⁶	A-D0 control register 2	AD0CON2
0395 ¹⁶		
0396 ¹⁶	A-D0 control register 0	AD0CON0
0397 ¹⁶	A-D0 control register 1	AD0CON1
0398 ¹⁶	D-A register 0	DA0
0399 ¹⁶		
039A ¹⁶	D-A register 1	DA1
039B ¹⁶		
039C ¹⁶	D-A control register	DACON
039D ¹⁶		
039E ¹⁶		
039F ¹⁶		

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

<144-pin version>

Address	Register		Address	Register	
03A0 ₁₆	Function select register A8	PS8	03D0 ₁₆	Port P14 register	P14
03A1 ₁₆	Function select register A9	PS9	03D1 ₁₆	Port P15 register	P15
03A2 ₁₆			03D2 ₁₆	Port P14 direction register	PD14
03A3 ₁₆			03D3 ₁₆	Port P15 direction register	PD15
03A4 ₁₆			03D4 ₁₆		
03A5 ₁₆			03D5 ₁₆		
03A6 ₁₆			03D6 ₁₆		
03A7 ₁₆			03D7 ₁₆		
03A8 ₁₆			03D8 ₁₆		
03A9 ₁₆			03D9 ₁₆		
03AA ₁₆			03DA ₁₆	Pull-up control register 2	PUR2
03AB ₁₆			03DB ₁₆	Pull-up control register 3	PUR3
03AC ₁₆			03DC ₁₆	Pull-up control register 4	PUR4
03AD ₁₆			03DD ₁₆		
03AE ₁₆			03DE ₁₆		
03AF ₁₆	Function select register C	PSC	03DF ₁₆		
03B0 ₁₆	Function select register A0	PS0	03E0 ₁₆	Port P0 register	P0
03B1 ₁₆	Function select register A1	PS1	03E1 ₁₆	Port P1 register	P1
03B2 ₁₆	Function select register B0	PSL0	03E2 ₁₆	Port P0 direction register	PD0
03B3 ₁₆	Function select register B1	PSL1	03E3 ₁₆	Port P1 direction register	PD1
03B4 ₁₆	Function select register A2	PS2	03E4 ₁₆	Port P2 register	P2
03B5 ₁₆	Function select register A3	PS2	03E5 ₁₆	Port P3 register	P3
03B6 ₁₆	Function select register B2	PSL2	03E6 ₁₆	Port P2 direction register	PD2
03B7 ₁₆	Function select register B3	PSL3	03E7 ₁₆	Port P3 direction register	PD3
03B8 ₁₆			03E8 ₁₆	Port P4 register	P4
03B9 ₁₆	Function select register A5	PS5	03E9 ₁₆	Port P5 register	P5
03BA ₁₆			03EA ₁₆	Port P4 direction register	PD4
03BB ₁₆			03EB ₁₆	Port P5 direction register	PD5
03BC ₁₆	Function select register A6	PS6	03EC ₁₆		
03BD ₁₆	Function select register A7	PS7	03ED ₁₆		
03BE ₁₆			03EE ₁₆		
03BF ₁₆			03EF ₁₆		
03C0 ₁₆	Port P6 register	P6	03F0 ₁₆	Pull-up control register 0	PUR0
03C1 ₁₆	Port P7 register	P7	03F1 ₁₆	Pull-up control register 1	PUR1
03C2 ₁₆	Port P6 direction register	PD6	03F2 ₁₆		
03C3 ₁₆	Port P7 direction register	PD7	03F3 ₁₆		
03C4 ₁₆	Port P8 register	P8	03F4 ₁₆		
03C5 ₁₆	Port P9 register	P9	03F5 ₁₆		
03C6 ₁₆	Port P8 direction register	PD8	03F6 ₁₆		
03C7 ₁₆	Port P9 direction register	PD9	03F7 ₁₆		
03C8 ₁₆	Port P10 register	P10	03F8 ₁₆		
03C9 ₁₆	Port P11 register	P11	03F9 ₁₆		
03CA ₁₆	Port P10 direction register	PD10	03FA ₁₆		
03CB ₁₆	Port P11 direction register	PD11	03FB ₁₆		
03CC ₁₆	Port P12 register	P12	03FC ₁₆		
03CD ₁₆	Port P13 register	P13	03FD ₁₆		
03CE ₁₆	Port P12 direction register	PD12	03FE ₁₆		
03CF ₁₆	Port P13 direction register	PD13	03FF ₁₆	Port control register	PCR


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

<100-pin version>


Address	Register
03A0 ₁₆	
03A1 ₁₆	
03A2 ₁₆	
03A3 ₁₆	
03A4 ₁₆	
03A5 ₁₆	
03A6 ₁₆	
03A7 ₁₆	
03A8 ₁₆	
03A9 ₁₆	
03AA ₁₆	
03AB ₁₆	
03AC ₁₆	
03AD ₁₆	
03AE ₁₆	
03AF ₁₆	Function select register C PSC
03B0 ₁₆	Function select register A0 PS0
03B1 ₁₆	Function select register A1 PS1
03B2 ₁₆	Function select register B0 PSL0
03B3 ₁₆	Function select register B1 PSL1
03B4 ₁₆	Function select register A2 PS2
03B5 ₁₆	Function select register A3 PS3
03B6 ₁₆	Function select register B2 PSL2
03B7 ₁₆	Function select register B3 PSL3
03B8 ₁₆	
03B9 ₁₆	
03BA ₁₆	
03BB ₁₆	
03BC ₁₆	
03BD ₁₆	
03BE ₁₆	
03BF ₁₆	
03C0 ₁₆	Port P6 register P6
03C1 ₁₆	Port P7 register P7
03C2 ₁₆	Port P6 direction register PD6
03C3 ₁₆	Port P7 direction register PD7
03C4 ₁₆	Port P8 register P8
03C5 ₁₆	Port P9 register P9
03C6 ₁₆	Port P8 direction register PD8
03C7 ₁₆	Port P9 direction register PD9
03C8 ₁₆	Port P10 register P10
03C9 ₁₆	
03CA ₁₆	Port P10 direction register PD10
03CB ₁₆	
03CC ₁₆	
03CD ₁₆	
03CE ₁₆	
03CF ₁₆	

Address	Register
03D0 ₁₆	
03D1 ₁₆	
03D2 ₁₆	
03D3 ₁₆	
03D4 ₁₆	
03D5 ₁₆	
03D6 ₁₆	
03D7 ₁₆	
03D8 ₁₆	
03D9 ₁₆	
03DA ₁₆	Pull-up control register 2 PUR2
03DB ₁₆	Pull-up control register 3 PUR3
03DC ₁₆	
03DD ₁₆	
03DE ₁₆	
03DF ₁₆	
03E0 ₁₆	Port P0 register P0
03E1 ₁₆	Port P1 register P1
03E2 ₁₆	Port P0 direction register PD0
03E3 ₁₆	Port P1 direction register PD1
03E4 ₁₆	Port P2 register P2
03E5 ₁₆	Port P3 register P3
03E6 ₁₆	Port P2 direction register PD2
03E7 ₁₆	Port P3 direction register PD3
03E8 ₁₆	Port P4 register P4
03E9 ₁₆	Port P5 register P5
03EA ₁₆	Port P4 direction register PD4
03EB ₁₆	Port P5 direction register PD5
03EC ₁₆	
03ED ₁₆	
03EE ₁₆	
03EF ₁₆	
03F0 ₁₆	Pull-up control register 0 PUR0
03F1 ₁₆	Pull-up control register 1 PUR1
03F2 ₁₆	
03F3 ₁₆	
03F4 ₁₆	
03F5 ₁₆	
03F6 ₁₆	
03F7 ₁₆	
03F8 ₁₆	
03F9 ₁₆	
03FA ₁₆	
03FB ₁₆	
03FC ₁₆	
03FD ₁₆	
03FE ₁₆	
03FF ₁₆	Port control register PCR

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

Note 1:  Addresses 03CB₁₆, 03CE₁₆, 03CF₁₆, 03D2₁₆, 03D3₁₆ does not exist in 100-pin version. Must set "FF₁₆" to the addresses at initial setting.

Note 2:  Addresses 03DC₁₆ area does not exist in 100-pin version. Must set "00₁₆" to addresses 03DC₁₆ at initial setting.

Note 3:  Addresses 03A0₁₆, 03A1₁₆, 03B9₁₆, 03BC₁₆, 03BD₁₆, 03C9₁₆, 03CC₁₆, 03CD₁₆, 03D30₁₆, 03D11₆ does not exist in 100-pin version.

Software Reset

Writing “1” to bit 3 of the processor mode register 0 (address 000416) 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 micro-processor mode. The functions of some pins, memory map, and 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 P15 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 an address bus, a 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 CNVss pin and the processor mode bits (bits 1 and 0 at address 000416). Do not set the processor mode bits to “102”.

Regardless of the level of the CNVss 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 Vss to CNVss pin**

The microcomputer begins operation in single-chip mode after being reset. Memory expansion mode is selected by writing “012” to the processor mode is selected bits.

- **Applying Vcc to CNVss 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.

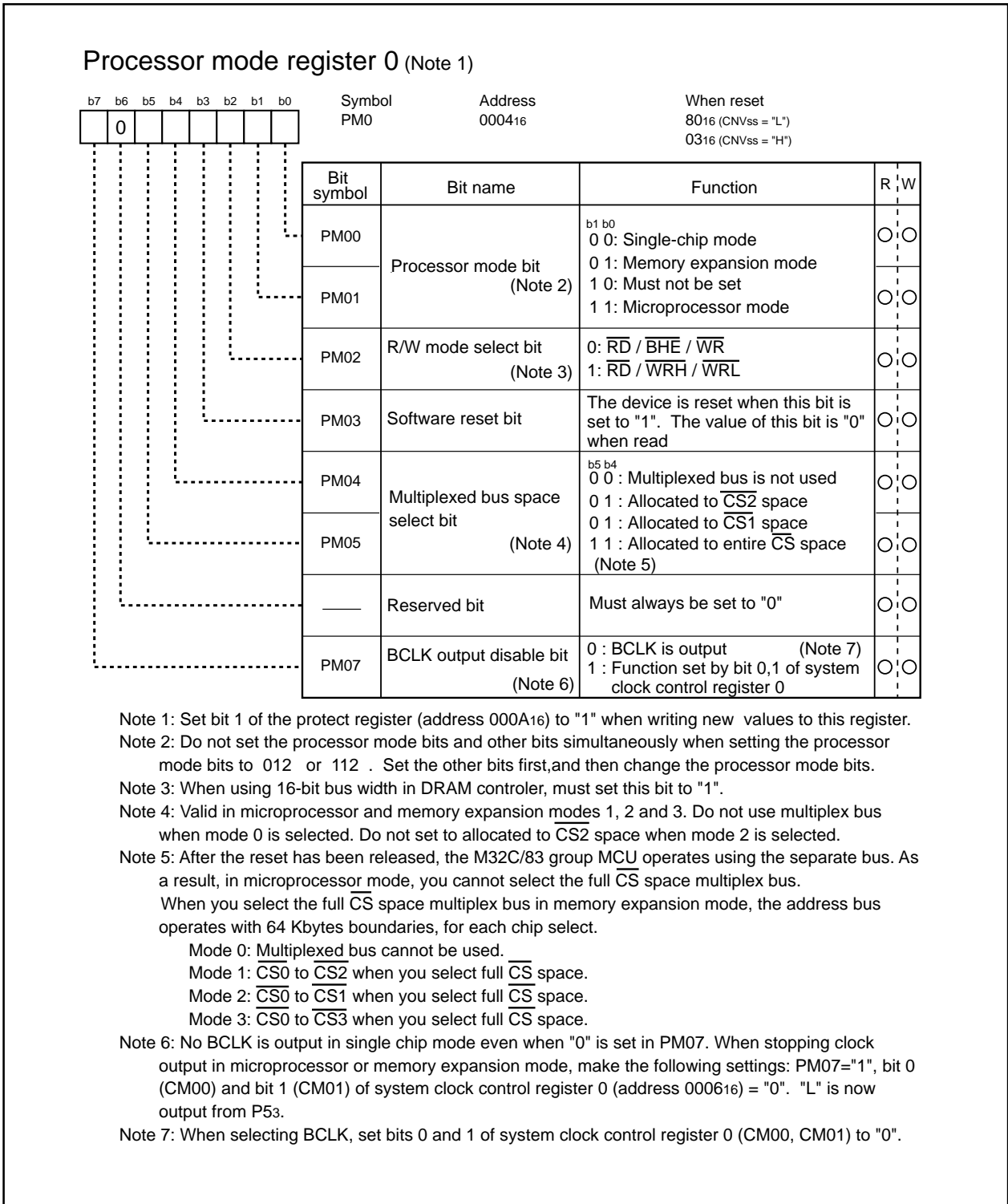


Figure 1.6.1. Processor mode register 0

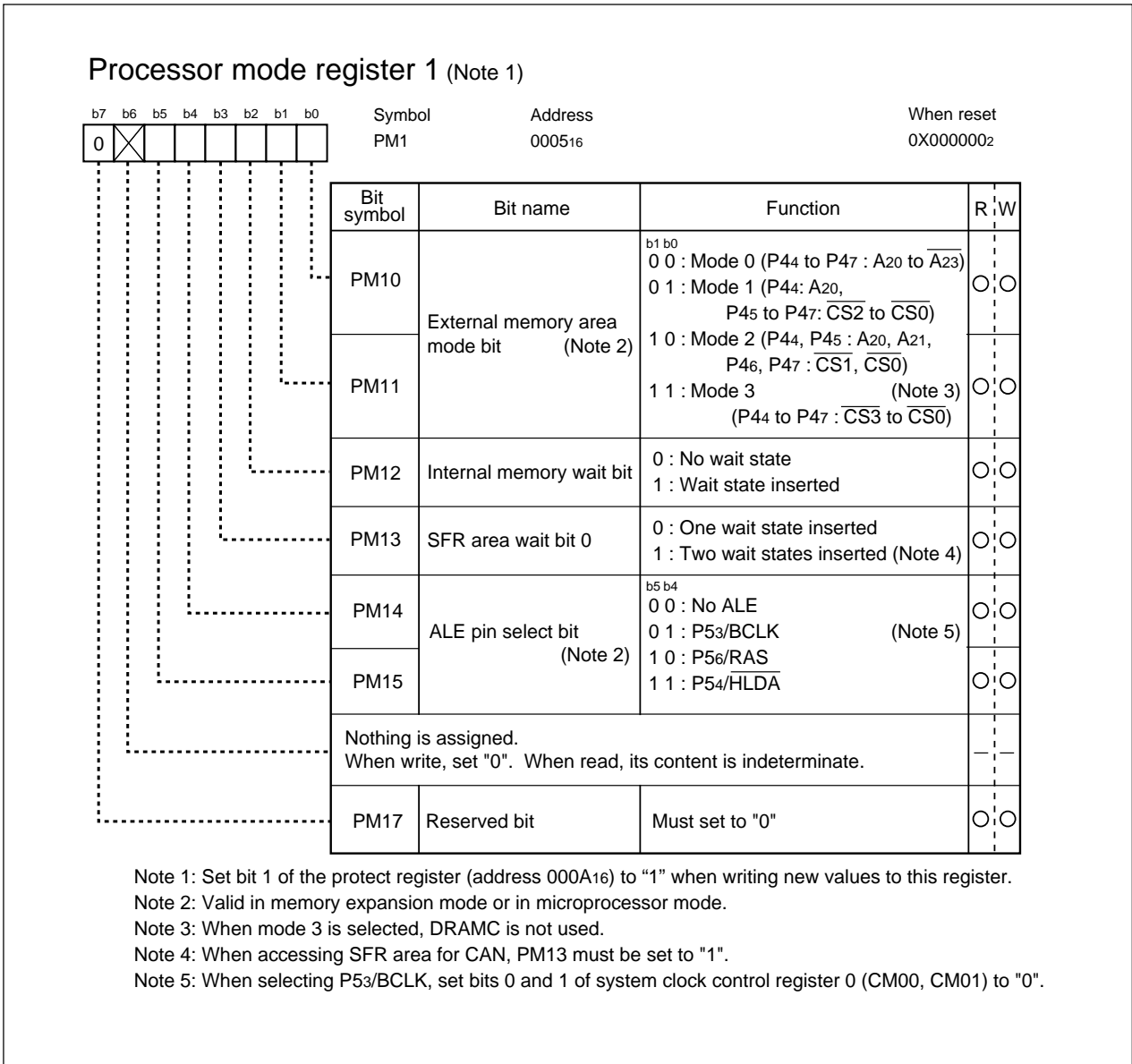


Figure 1.6.2. Processor mode register 1

	Memory expanded mode				Microprocessor mode			
	Mode 0	Mode 1	Mode 2	Mode 3	Mode 0	Mode 1	Mode 2	Mode 3
00000016	SFR area	SFR area	SFR area	SFR area	SFR area	SFR area	SFR area	SFR area
00040016	Internal RAM area	Internal RAM area	Internal RAM area	Internal RAM area	Internal RAM area	Internal RAM area	Internal RAM area	Internal RAM area
00080016	Internal reserved area	Internal reserved area	Internal reserved area	Internal reserved area	Internal reserved area	Internal reserved area	Internal reserved area	Internal reserved area
20000016	External area 0	CS1 2Mbytes (Note1) External area 0	CS1 4Mbytes (Note2) External area 0	No use CS1, 1Mbytes External area 0	External area 0	CS1 2Mbytes (Note1) External area 0	CS1 4Mbytes (Note2) External area 0	No use CS1, 1Mbytes External area 0
40000016	External area 1	CS2 2Mbytes External area 1	CS2 2Mbytes External area 1	No use CS2, 1Mbytes External area 1	External area 1	CS2 2Mbytes External area 1	CS2 2Mbytes External area 1	No use CS2, 1Mbytes External area 1
60000016	Connect with DRAM 0, 0.5 to 8MB (When not connect with DRAM, use as external area.) (External area 2)	Connect with DRAM 0, 0.5 to 8MB (When open area is under 8MB, cannot use the rest of this area.) (External area 2)	Connect with DRAM 0, 0.5 to 8MB (When open area is under 8MB, cannot use the rest of this area.) (External area 2)	No use (Cannot use as DRAM area or external area.)	Connect with DRAM 0, 0.5 to 8MB (When not connect with DRAM, use as external area.) (External area 2)	Connect with DRAM 0, 0.5 to 8MB (When open area is under 8MB, cannot use the rest of this area.) (External area 2)	Connect with DRAM 0, 0.5 to 8MB (When open area is under 8MB, cannot use the rest of this area.) (External area 2)	No use (Cannot use as DRAM area or external area.)
C0000016	External area 3	CS0 2Mbytes External area 3	CS0 3Mbytes External area 3	CS3, 1Mbytes External area 2	External area 3	No use	CS0 4Mbytes External area 3	CS3, 1Mbytes External area 2
E0000016	Internal reserved area	No use	External area 3	No use	Internal reserved area	Internal reserved area	Internal reserved area	No use
F0000016	Internal ROM area	Internal reserved area	Internal reserved area	CS0, 1Mbytes External area 3	Internal ROM area	Internal reserved area	Internal ROM area	CS0, 1Mbytes External area 3
FFFFF16	Internal ROM area	Internal ROM area	Internal ROM area	Internal ROM area	Internal ROM area	Internal ROM area	Internal ROM area	CS0, 1Mbytes External area 3

Note 1: 20000016~00800016=2016 Kbytes. 32 K less than 2 MB.
Note 2: 40000016~00800016=4064 Kbytes. 32 K less than 4 MB.

Each CS0 to CS3 can set 0 to 3 WAIT.

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 000B₁₆), bits 4 and 5 of the processor mode register 0 (address 0004₁₆) and bit 0 and 1 of the processor mode register 1 (address 0005₁₆) 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
Selecting external area	Bits 0 and 1 of processor mode register 1

(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 A₀ to A₁₅ 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 buses.

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 D₀ to D₇ are multiplexed with the 8 bits A₀ to A₇. For areas for which 16-bit has been selected using the external data bus width control register, the 16 bits D₀ to D₁₅ are multiplexed with the 16 bits A₀ to A₁₅. 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 a 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

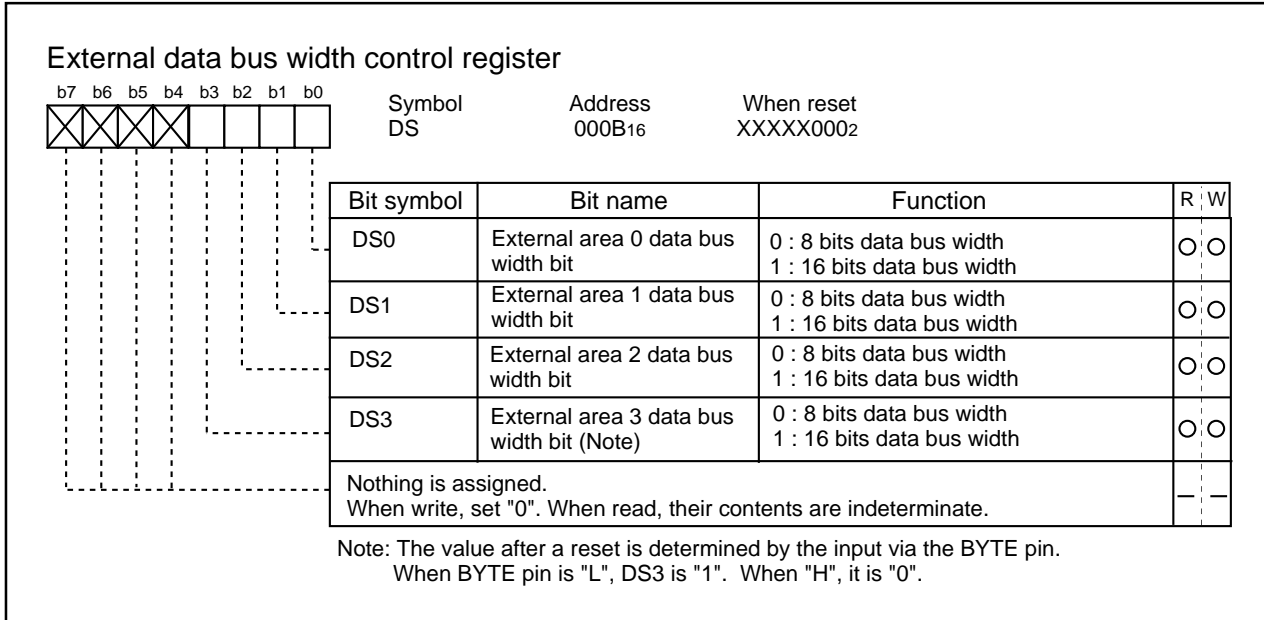


Figure 1.7.1. External data bus width control register

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 1FFFFFF ₁₆	<CS1 area> 008000 ₁₆ to 1FFFFFF ₁₆	<CS1 area> 008000 ₁₆ to 1FFFFFF ₁₆	<CS1 area> 100000 ₁₆ to 1FFFFFF ₁₆
External area 1	Memory expansion mode, Microprocessor mode	200000 ₁₆ to 3FFFFFF ₁₆	<CS2 area> 200000 ₁₆ to 3FFFFFF ₁₆	No area is selected.	<CS2 area> 200000 ₁₆ to 2FFFFFF ₁₆
External area 2	Memory expansion mode, Microprocessor mode	400000 ₁₆ to BFFFFFF ₁₆ (Note 1)	<DRAMC area> 400000 ₁₆ to BFFFFFF ₁₆	<DRAMC area> 400000 ₁₆ to BFFFFFF ₁₆	<CS3 area> C00000 ₁₆ to CFFFFFF ₁₆
External area 3	Memory expansion mode	C00000 ₁₆ to EFFFFFF ₁₆	<CS0 area> C00000 ₁₆ to EFFFFFF ₁₆	<CS0 area> C00000 ₁₆ to EFFFFFF ₁₆	<CS0 area> E00000 ₁₆ to EFFFFFF ₁₆
	Microprocessor mode	C00000 ₁₆ to FFFFFFF ₁₆	<CS0 area> E00000 ₁₆ to FFFFFFF ₁₆	<CS0 area> C00000 ₁₆ to FFFFFFF ₁₆	<CS0 area> F00000 ₁₆ to FFFFFFF 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 0005₁₆).

Table 1.7.3. Each processor mode and port function

Processor mode	Single-chip mode	Memory expansion mode/microprocessor modes				Memory expansion mode	
		"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
P00 to P07	I/O port	Data bus	Data bus	Data bus	Data bus	I/O port	I/O port
P10 to P17	I/O port	I/O port	I/O port	Data bus	I/O port	I/O port	I/O port
P20 to P27	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
P30 to P37	I/O port	Address bus	Address bus /data bus (Note 2)	Address bus	Address bus	Address bus	Address bus /data bus
P40 to P43	I/O port	Address bus	Address bus	Address bus	Address bus	I/O port	I/O port
P44 to P46	I/O port	CS (chip select) or address bus (A23) (For details, refer to "Bus control") (Note 5)					
P47	I/O port	\overline{CS} (chip select) or address bus (A23) (For details, refer to "Bus control") (Note 5)					
P50 to P53	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)					
P54	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)
P55	I/O port	\overline{HOLD}	\overline{HOLD}	\overline{HOLD}	\overline{HOLD}	\overline{HOLD}	\overline{HOLD}
P56	I/O port	RAS (Note 3)	RAS (Note 3)	RAS (Note 3)	RAS (Note 3)	RAS (Note 3)	RAS (Note 3)
P57	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 CS space multiplex bus" cannot be selected in microprocessor mode. When you select "Full 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 the DRAM controller and access the DRAM area, these are outputs \overline{CASL} , \overline{CASH} , \overline{DW} , and BCLK.

Note 5: The 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, A0 to A22 and $\overline{A23}$ for the address bus for accessing the 16 Mbytes address space. $\overline{A23}$ 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 000B₁₆) selects the 8-bit data bus, D0 to D7 for each external area, or the 16-bit data bus, D0 to D15. After a reset, there is by default an 8-bit data bus for the external area 3 when the BYTE pin is High, or a 16-bit data bus when the BYTE pin is Low.

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 A8 to A20.

(2) Chip select signals

The chip select signals share A0 to A22 and $\overline{A23}$. You can use bits 0 and 1 of the processor mode register 1 (address 0005₁₆) 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 Blocks or Areas 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

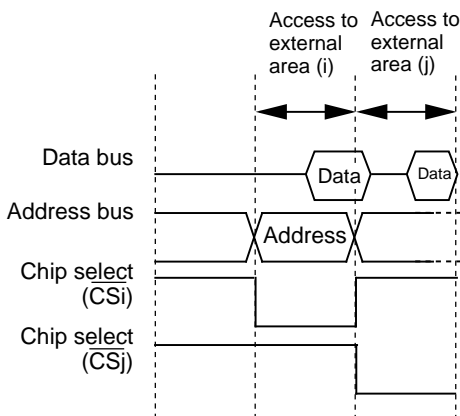
Memory space expansion mode	Processor mode	Chip select signal				
		$\overline{CS0}$	$\overline{CS1}$	$\overline{CS2}$	$\overline{CS3}$	
Specified address range	Mode 0	$\overline{(A23)}$	$\overline{(A22)}$	$\overline{(A21)}$	$\overline{(A20)}$	
	Mode 1	Memory expansion mode	C0000 ₁₆ to DFFFFFF ₁₆ (2 Mbytes)	008000 ₁₆ to 1FFFFFF ₁₆ (2016 Kbytes)	200000 ₁₆ to 3FFFFFF ₁₆ (2 Mbytes)	$\overline{(A20)}$
		Microprocessor mode	E00000 ₁₆ to FFFFFFF ₁₆ (2 Mbytes)			
	Mode 2	Memory expansion mode	C00000 ₁₆ to EFFFFFF ₁₆ (3 Mbytes)	008000 ₁₆ to 3FFFFFF ₁₆ (4064 Kbytes)	$\overline{(A21)}$	$\overline{(A20)}$
		Microprocessor mode	C00000 ₁₆ to FFFFFFF ₁₆ (4 Mbytes)			
	Mode 3	Memory expansion mode	E00000 ₁₆ to EFFFFFF ₁₆ (1 Mbytes)	100000 ₁₆ to 1FFFFFF ₁₆ (1 Mbytes)	200000 ₁₆ to 2FFFFFF ₁₆ (1 Mbytes)	C00000 ₁₆ to CFFFFFF ₁₆ (1 Mbytes)
		Microprocessor mode	F00000 ₁₆ to FFFFFFF ₁₆ (1 Mbytes)			

Bus Control

The chip select signal turns Low (active) in synchronize with the address bus. However, its turning High 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.

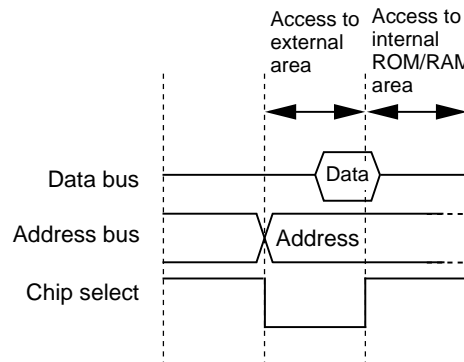
Example 1: After accessing the external area, the address bus and chip select signal both are changed in the next cycle.

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.



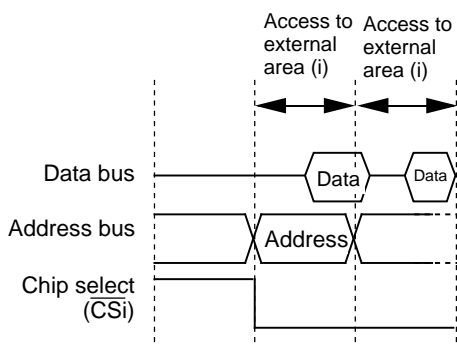
Example 2: After accessing the external area, only the chip select signal is changed in the next cycle. (The address bus does not change.)

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.



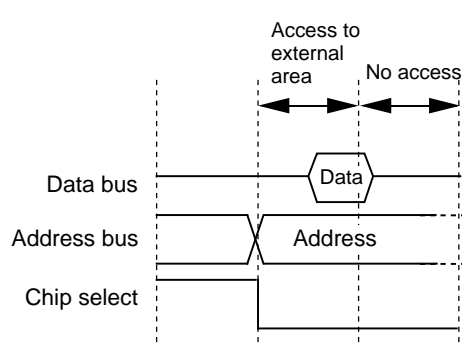
Example 3: After accessing the external area, only the address bus is changed in the next cycle. (The chip select signal does not change.)

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 0004₁₆) selects the combinations of \overline{RD} , \overline{BHE} , and \overline{WR} signals or \overline{RD} , \overline{WRL} , and \overline{WRH} signals. With a 8-bit full space data bus, use the combination of \overline{RD} , \overline{WR} , and \overline{BHE} signals as read/write signals. (Set "0" to bit 2 of the processor mode register 0 (address 0004₁₆.) When using both 8-bit and 16-bit data bus widths to access a 8-bit data bus area, the \overline{RD} , \overline{WR} and \overline{BHE} signals combination is selected regardless of the value of bit 2 of the processor mode register 0 (address 0004₁₆).

Tables 1.7.5 and 1.7.6 show the operation of these signals.

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

When switching to the \overline{RD} , \overline{WRL} , and \overline{WRH} combination, do not write to external memory until bit 2 of the processor mode register 0 (address 0004₁₆) 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 000A₁₆) to "1".

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

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

Data bus width	\overline{RD}	\overline{WRL}	\overline{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{WR} signal.

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

Data bus width	\overline{RD}	\overline{WR}	\overline{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 000516).

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

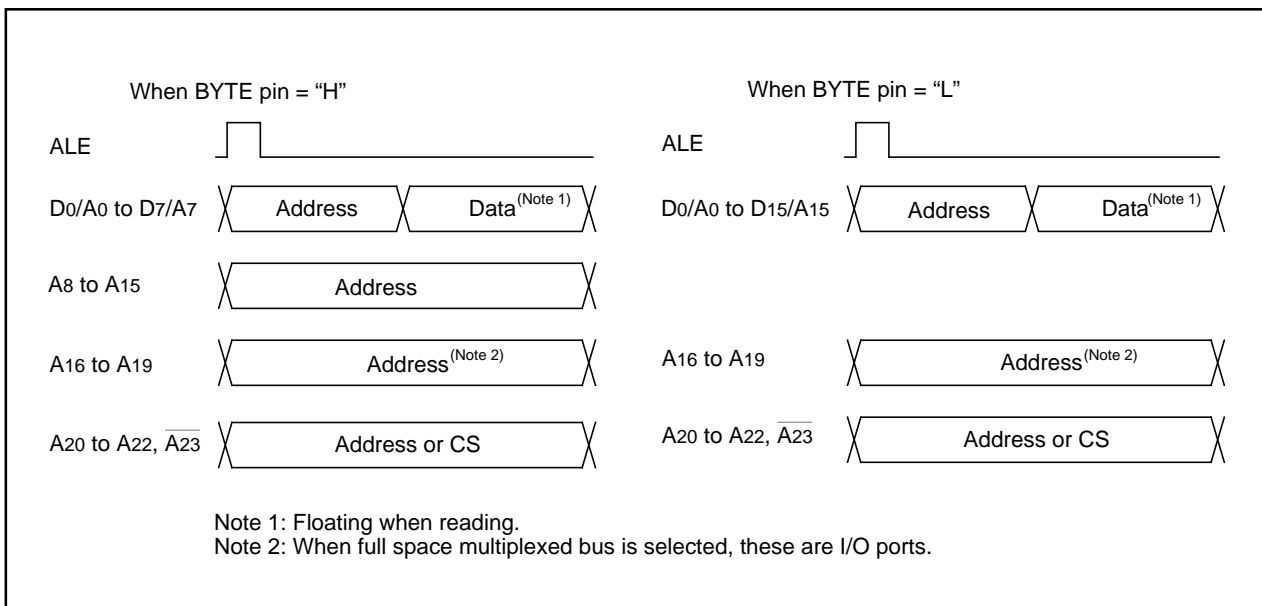


Figure 1.7.3. ALE signal and address/data bus

(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{RDY} pin at the falling edge of BCLK causes the microcomputer to enter the ready state. Inputting "H" to the \overline{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{RD} signal being extended using the \overline{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{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
$\overline{RD}/\overline{WR}$ signal, address bus, data bus, \overline{CS} ALE signal, \overline{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.

Bus Control

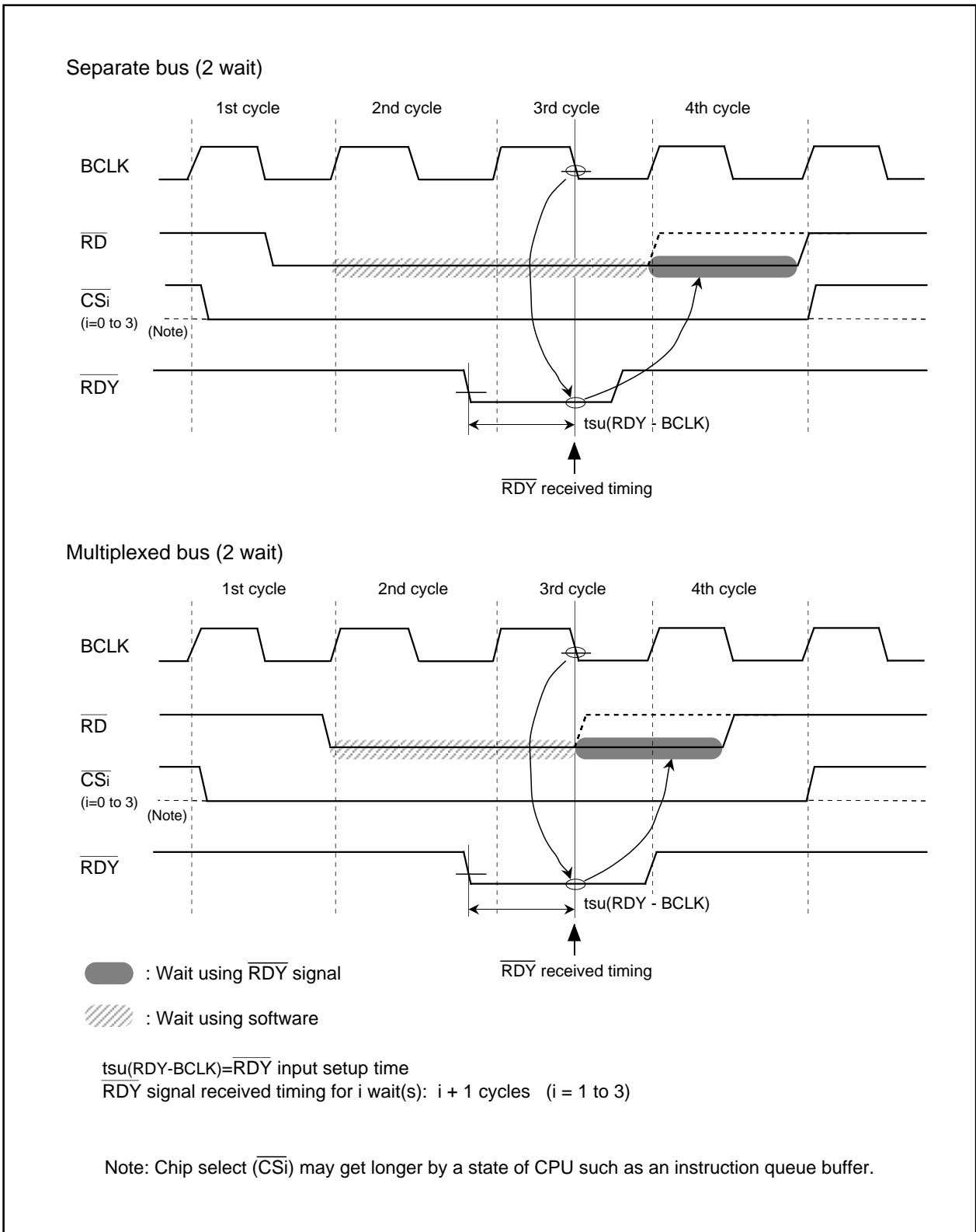


Figure 1.7.4. Example of RD signal extended by RDY signal

(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.

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

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
RD/WR signal, address bus, data bus, $\overline{\text{CS}}$, BHE	Floating
Programmable I/O ports: P0 to P15	Maintains status when hold signal is received
HLDA	Output “L”
Internal peripheral circuits	ON (but watchdog timer stops)
ALE signal	Output “L”

(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
RD, WR, WRL, WRH	Output “H”	
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 0004₁₆:PM07) and bit 1 and bit 0 of the system clock select register 0 (address 0006₁₆:CM01, CM00). Setting PM07 to “0” and CM01 and CM00 to “00” outputs the BCLK signal from P53. However, in single chip mode, BCLK signal is inactive. When setting PM07 to “1”, the function is 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 000416) select the DRAM space and enable the DRAM controller. The DRAM controller signals are 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	H	H	Read 1 byte of data from even address
	L	L	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 000816). 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 000516). Setting the internal memory wait bit = "0" sets "No wait". Setting the internal memory wait bit = "1" specifies a wait.

SFR area is accessed with either "1 wait" (BCLK 2-cycle) or "2 waits" (BCLK 3-cycle) by setting the SFR wait bit (bit 3) of the processor mode register 1 (address 000516). SFR area of CAN must be accessed with "2 waits".

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

Bus Control

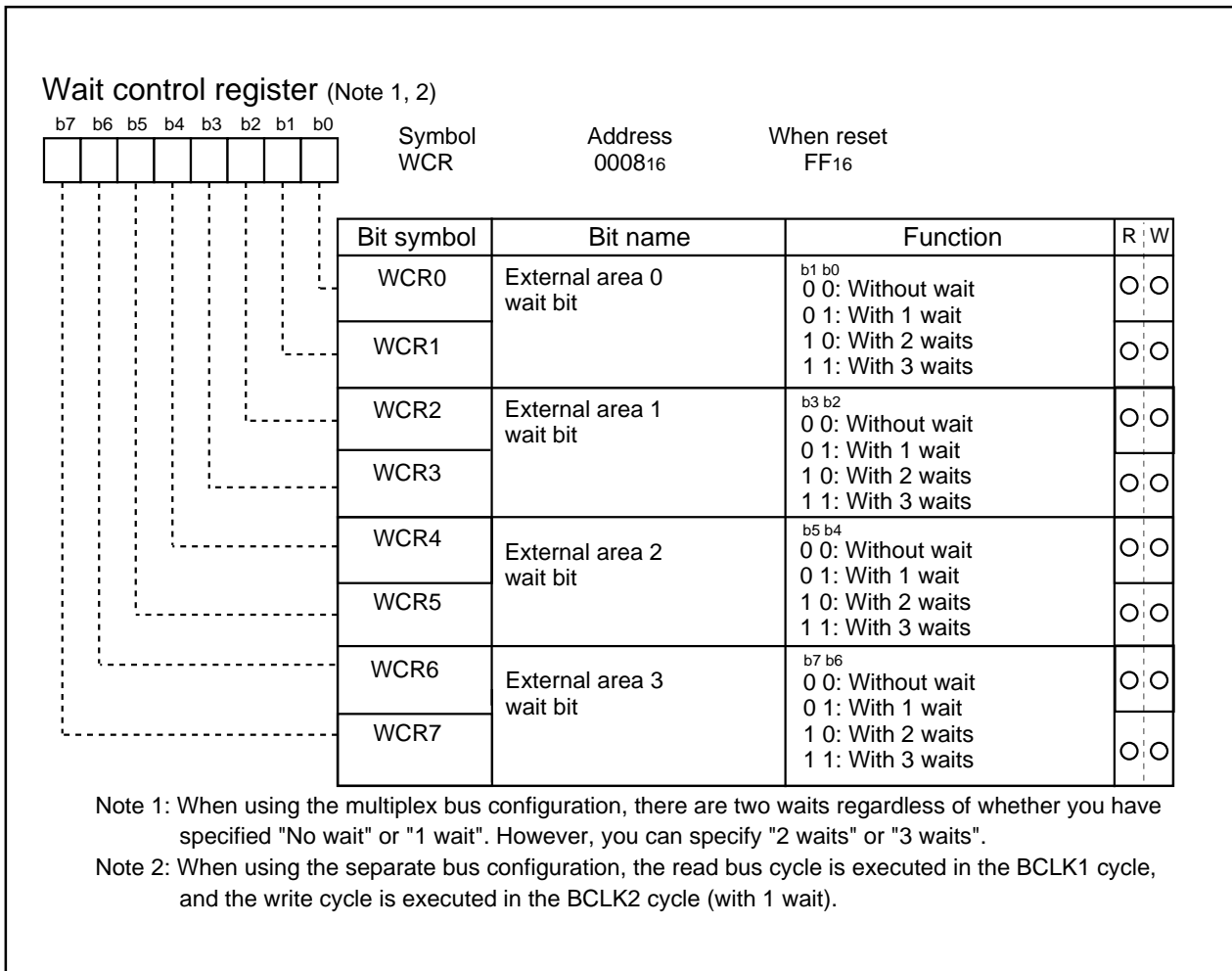


Figure 1.7.6. Wait control register

Table 1.7.11. Software waits and bus cycles

Area	Bus status	SFR area wait bit	Internal memory wait bit	External memory area i wait bit	Bus cycle	
SFR	_____	0	_____	_____	2 BCLK cycles	
	_____	1	_____	_____	3 BCLK cycles	
Internal ROM/RAM	_____	_____	0	_____	1 BCLK cycle	
	_____	_____	1	_____	2 BCLK cycles	
External memory area	Separate bus	_____	_____	002	Read :1 BCLK cycle Write : 2 BCLK cycles	
				012	2 BCLK cycles	
				102	3 BCLK cycles	
				112	4 BCLK cycles	
	Multiplex bus	_____	_____	_____	002	3 BCLK cycle
					012	3 BCLK cycles
					102	3 BCLK cycles
					112	4 BCLK cycles

Bus Control

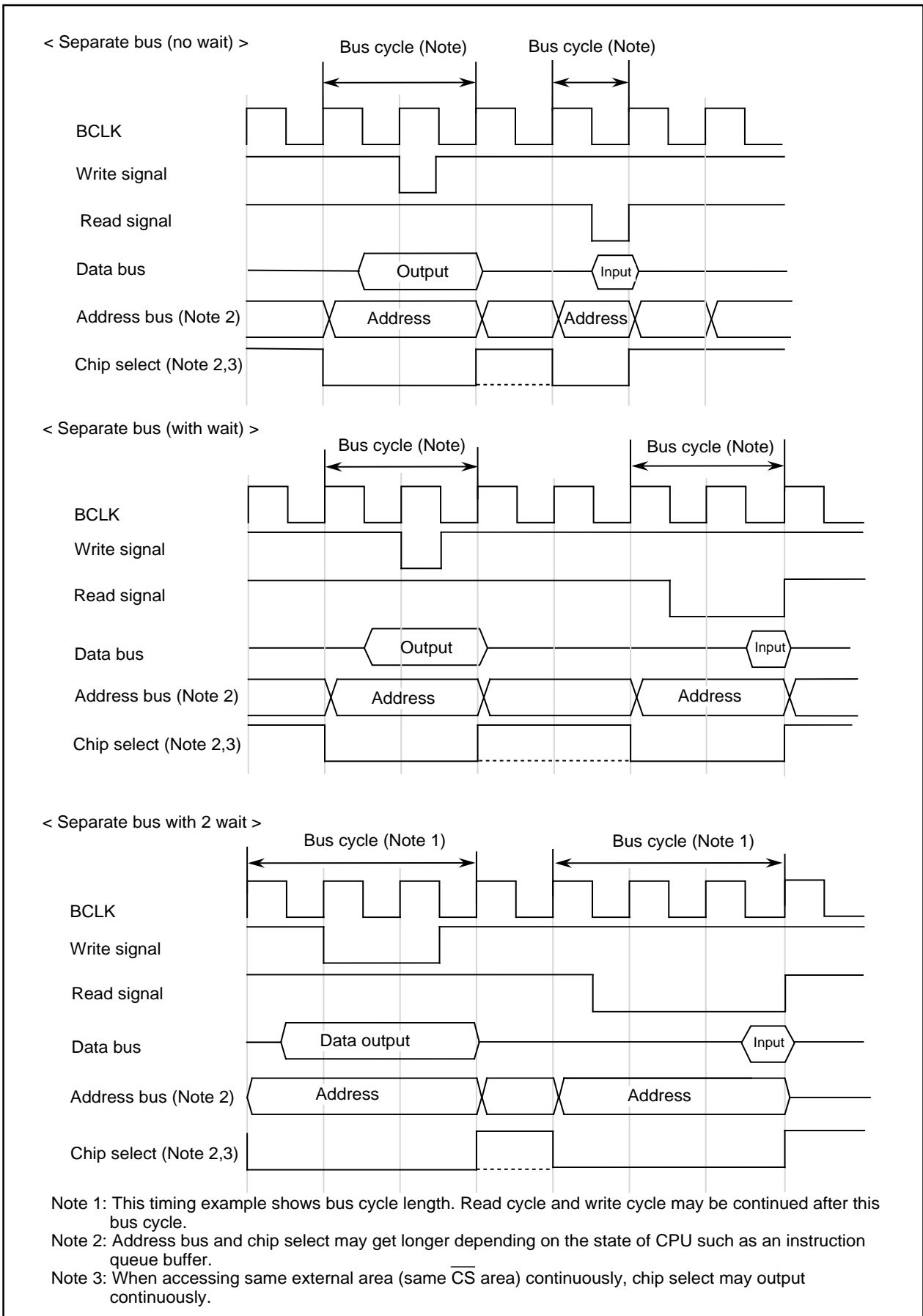


Figure 1.7.7. Typical bus timings using software wait

Bus Control

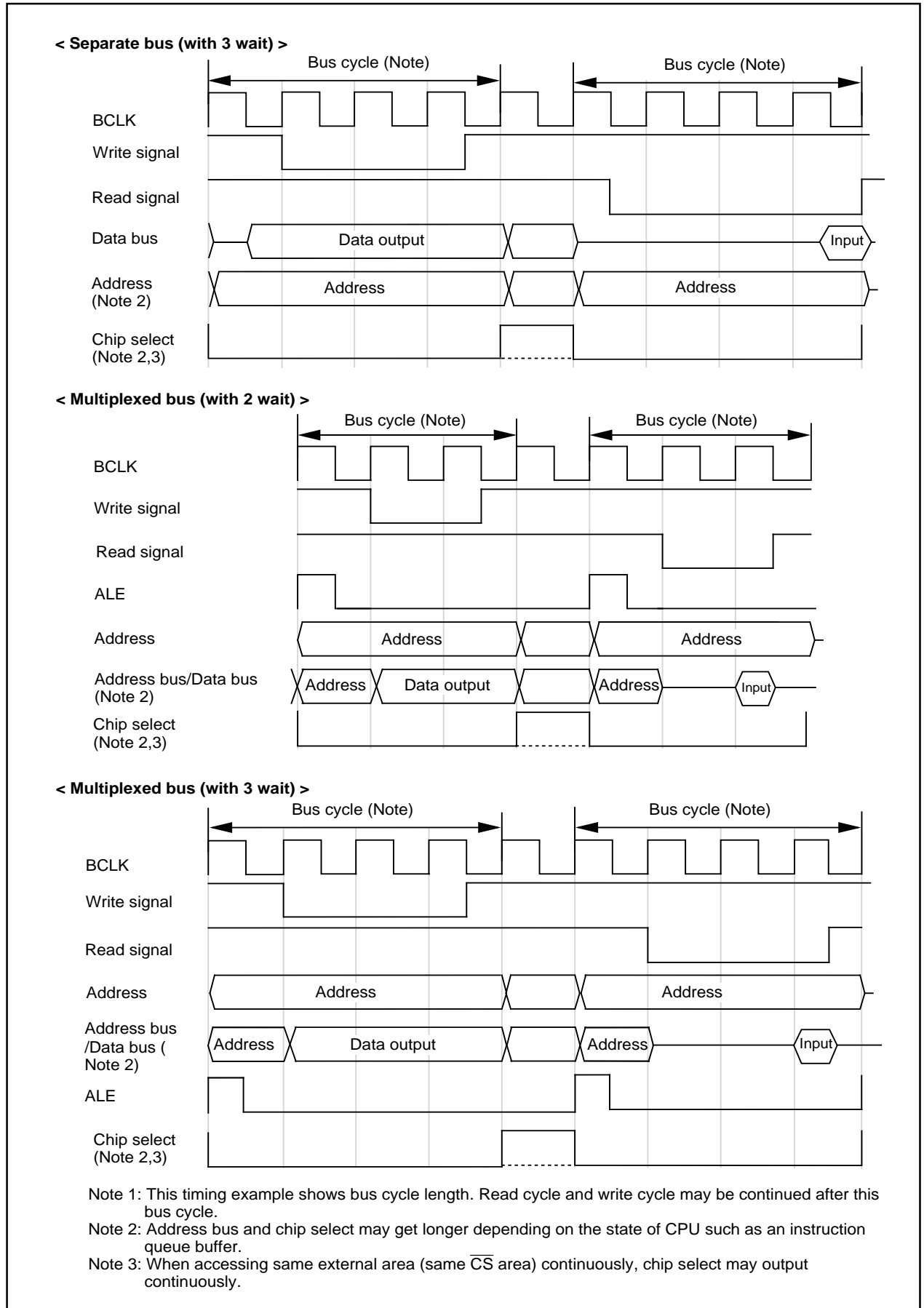


Figure 1.7.8. Typical bus timings using software wait

Clock Generating Circuit

System Clock

Clock Generating Circuit

The clock generating circuit contains three oscillator circuits as follows:

- (1) Main clock generating circuit
- (2) Sub clock generating circuit
- (3) Ring oscillator (oscillation stop detect function)

Table 1.8.1 lists the clock generating circuit specifications and Table 1.8.2 lists registers controlling each clock generating circuit. Figure 1.8.1 shows block diagram of the system clock generating circuit. Figure 1.8.2 to 1.8.5 show clock control related registers.

Table 1.8.1. The clock oscillation circuit specifications

Item	Main clock generating circuit	Sub clock generating circuit	Ring oscillator
Use of clock	<ul style="list-style-type: none"> • CPU's operating clock source • Internal peripheral unit's operating clock source 	<ul style="list-style-type: none"> • CPU's operating clock source • Timer A/B's count clock source 	<ul style="list-style-type: none"> • CPU's operating clock source when main clock frequency stops
Clock frequency	0 to 30 MHz	32.768 kHz	About 1 MHz
Usable oscillator	<ul style="list-style-type: none"> • Ceramic oscillator • Crystal oscillator 	<ul style="list-style-type: none"> • Crystal oscillator 	_____
Pins to connect oscillator	XIN, XOUT	XCIN, XCOUT	_____
Oscillation stop/restart function	Presence	Presence	Presence
Oscillator status after reset	Oscillating	Stopped	Stopped
Other	Externally derived clock can be input		_____

Table 1.8.2. Control registers for each clock generating circuits

Clock generating circuit	Control register
Main clock	System clock control register 0 (address 0006 ₁₆) :CM0 System clock control register 1 (address 0007 ₁₆) :CM1 Main clock divide register (address 000C ₁₆) : MCD
Sub clock	System clock control register 0 (address 0006 ₁₆) : CM0 System clock control register 1 (address 0007 ₁₆) :CM1
Oscillation stop detect function	Oscillation stop detect register (address 000D ₁₆) : CM2

Note : CM0, CM1, CM2 and MCD registers are protected from a false write by program runaway. When you want to rewrite these registers, set "1" to bit 0 of protect register (address 000A₁₆) to release protect, then rewrite the register.

Clock Generating Circuit

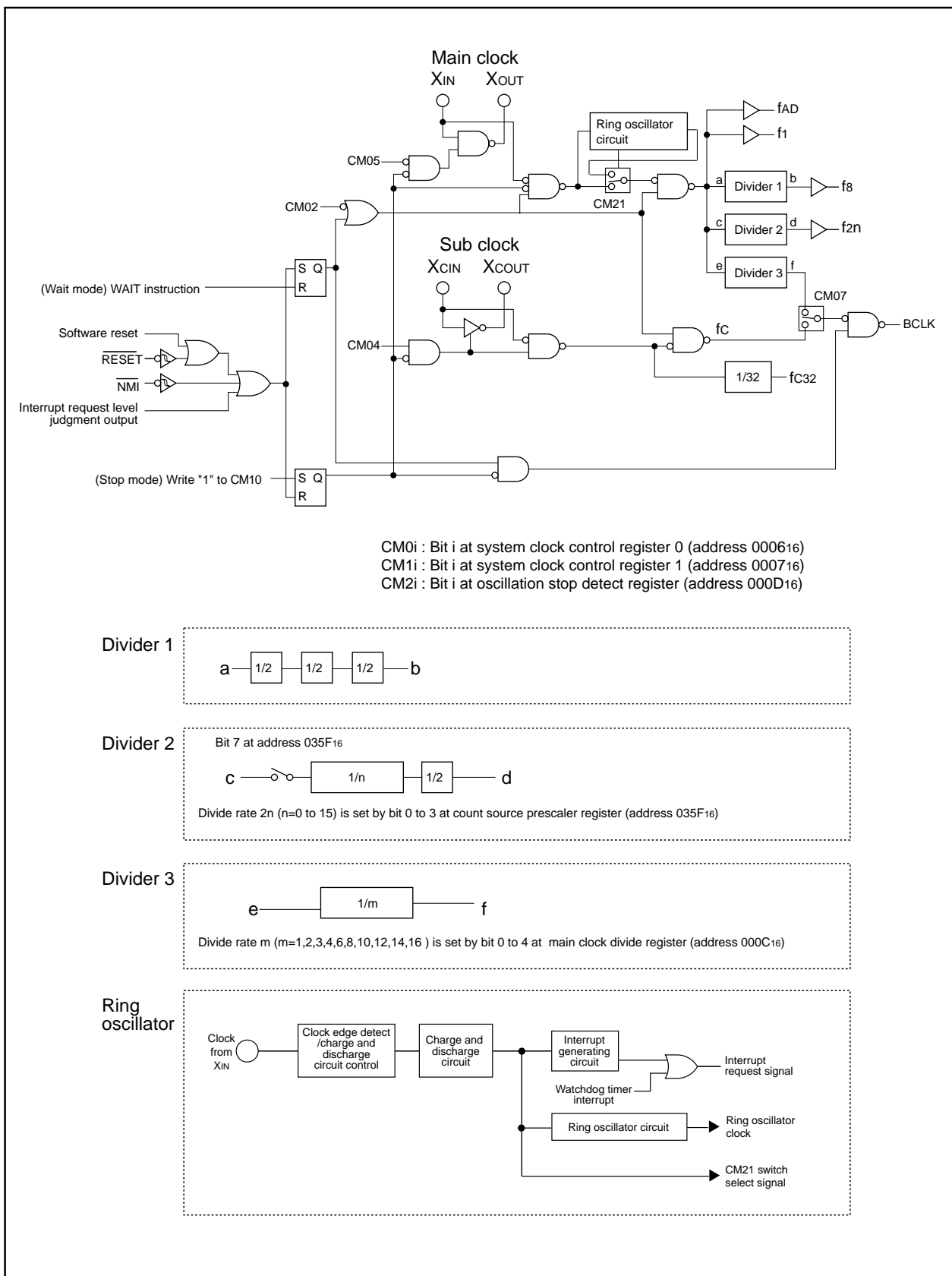


Figure 1.8.1. Clock generating circuit

Clock Generating Circuit

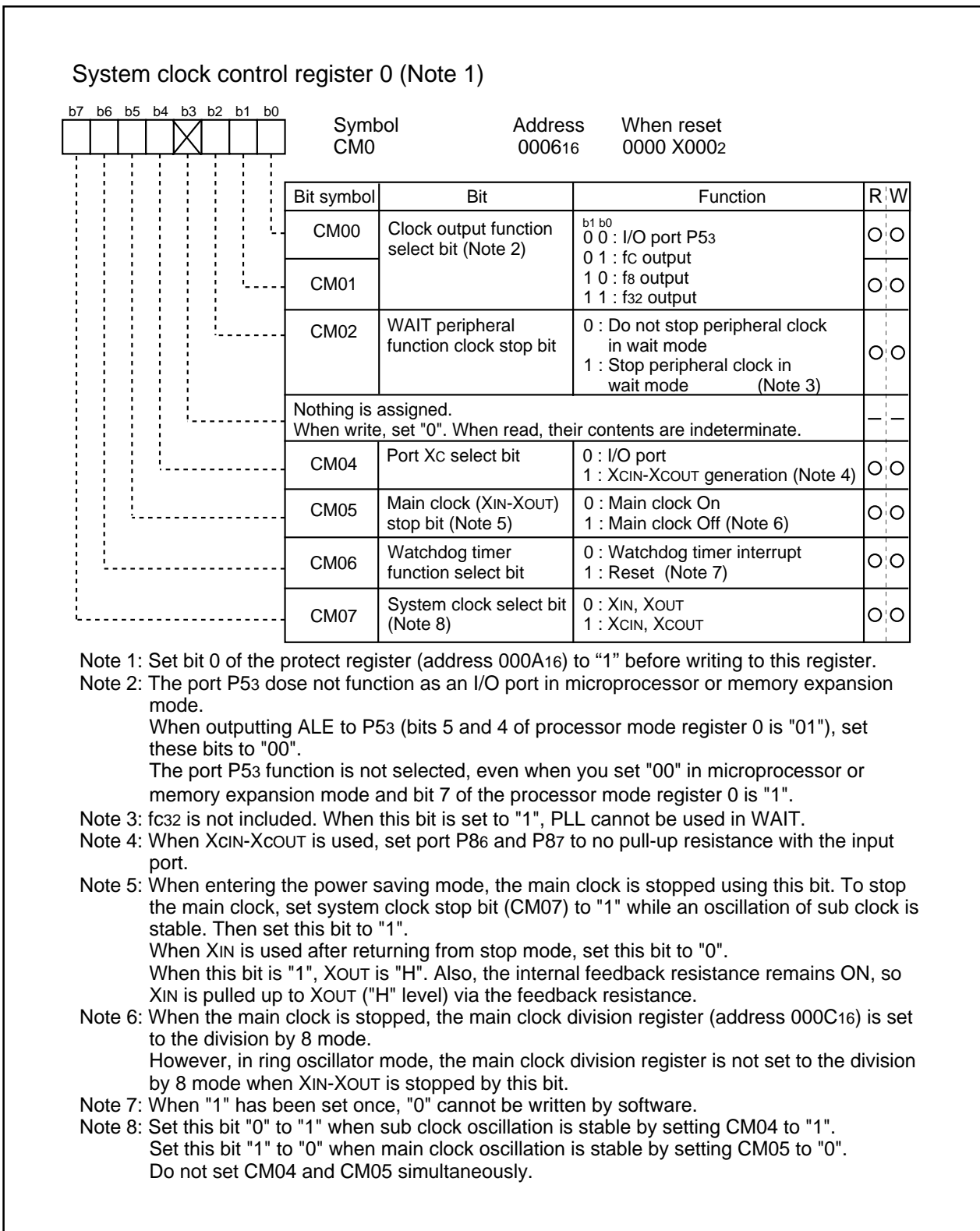


Figure 1.8.2. Clock control related register (1)

Clock Generating Circuit

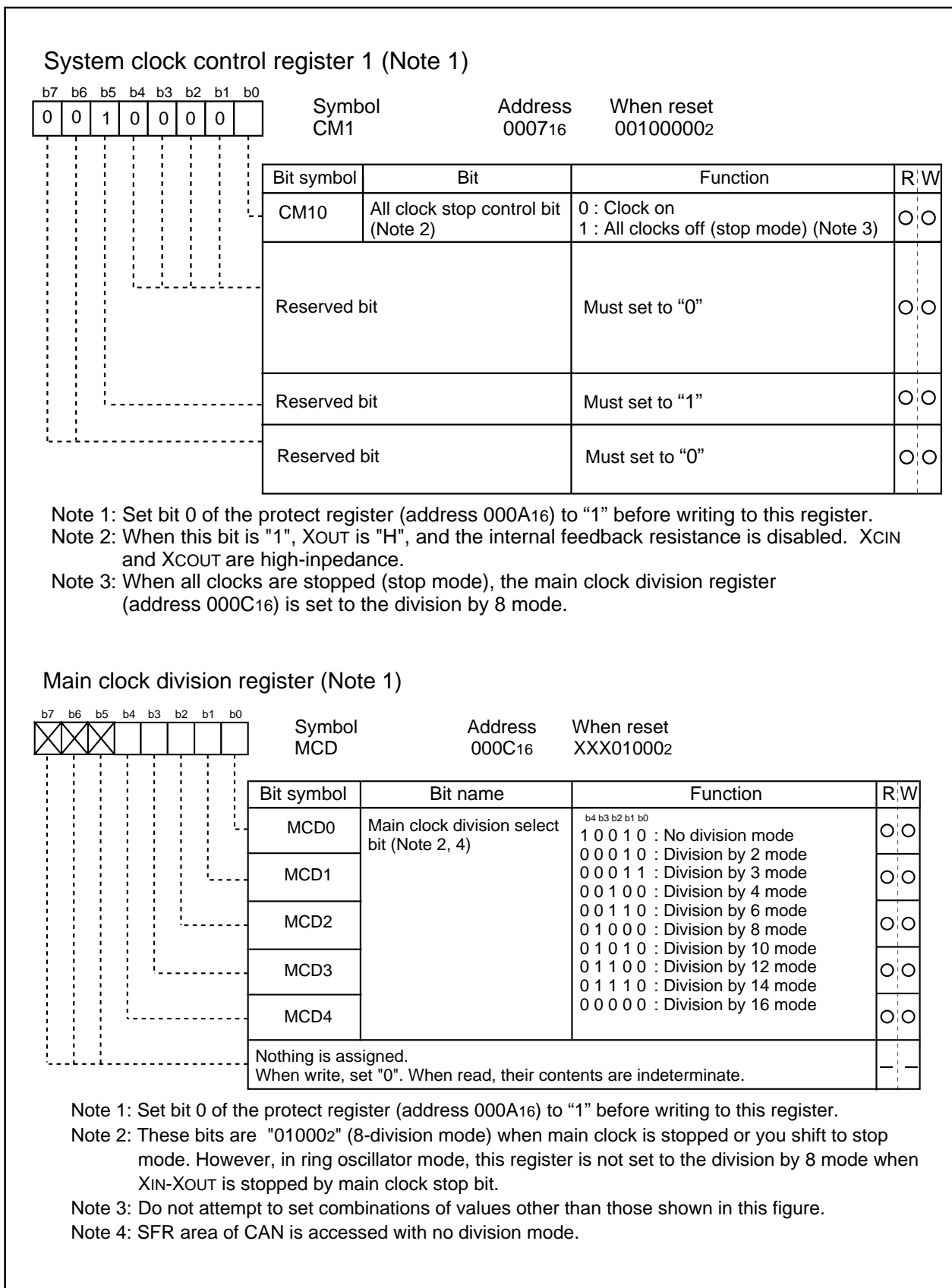


Figure 1.8.3. Clock control related registers (2)

Clock Generating Circuit

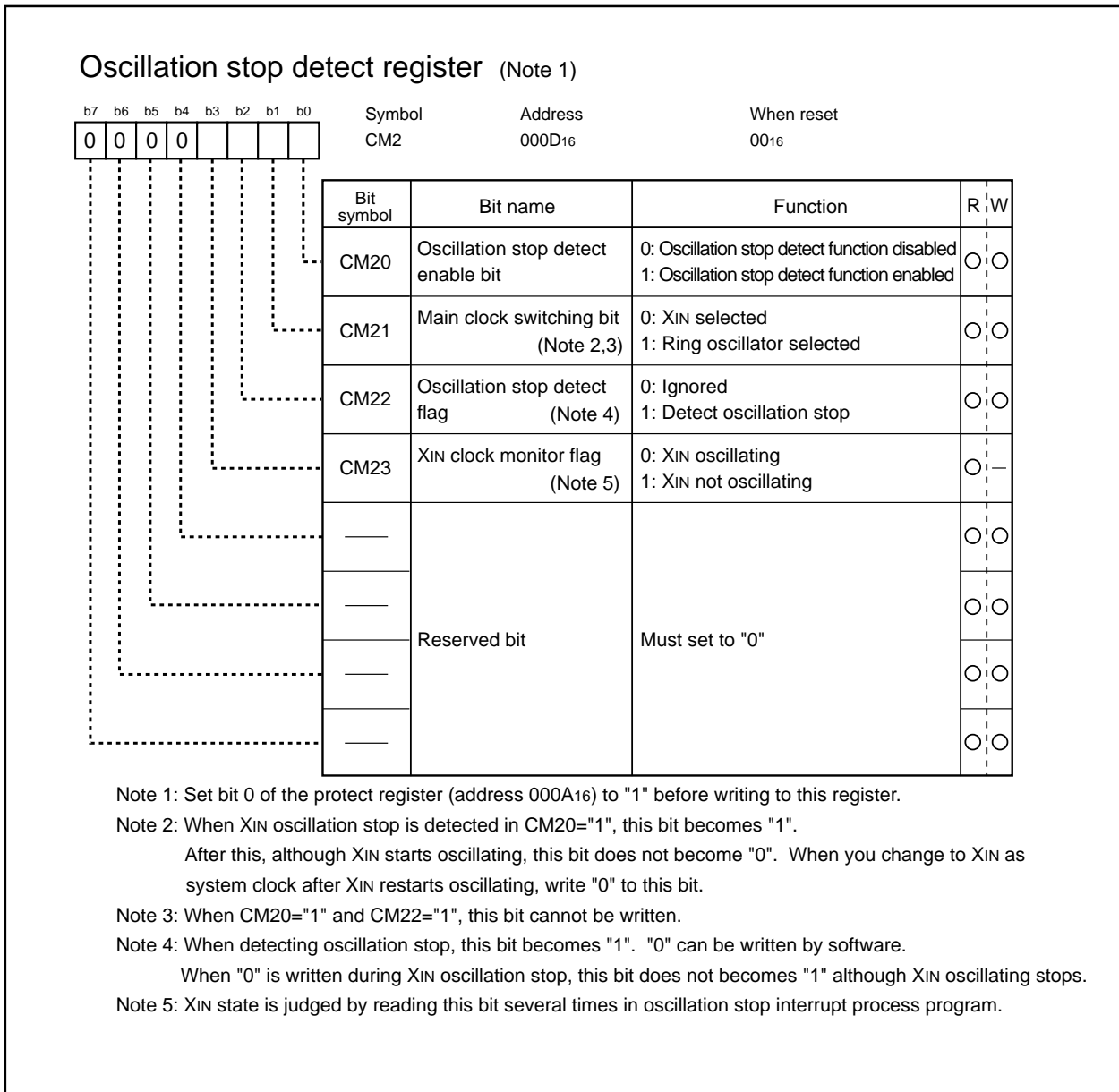
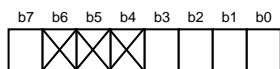


Figure 1.8.4. Clock control related register (3)

Clock Generating Circuit

Count source prescale register

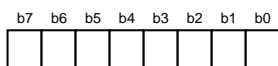


Symbol: TCSPR
Address: 035F₁₆
When reset: 0XX 000₂

Bit symbol	Bit name	Function	R ₁ W
CNT0	Division rate select bit	$b_3b_2b_1b_0$ 0 0 0 0: No-division 0 0 0 1: Division by 2 0 0 1 0: Division by 4 0 0 1 1: Division by 6 ⋮ 1 1 0 1: Division by 26 1 1 1 0: Division by 28 1 1 1 1: Division by 30	○ ○
CNT1			○ ○
CNT2			○ ○
CNT3			○ ○
—			○ ○
—	Nothing is assigned. When write, set to "0". When read, their contents are indeterminate.	— —	
—	Nothing is assigned. When write, set to "0". When read, their contents are indeterminate.	— —	
CST	Operation enable bit	0: Divider stops 1: Divider starts	○ ○

Note : Write to these bits during the count stop.

VDC control register for PLL (Note 1)



Symbol: PLV
Address: 0017₁₆
When reset: XXXXXX0₁₂

Bit symbol	Bit name	Function	R ₁ W
PLV00	PLL VDC enable bit (Note 2)	0 : Cut off power to PLL 1 : Power to PLL	○ ○
—	Reserved bit	Must set to "0"	○ ○
—	Nothing is assigned. When write, set to "0". When read, their contents are indeterminate.	— —	— —

Note 1: When rewriting this register, set bit 3 of protect register (address 000A₁₆) to "1".

Note 2: Set this bit to "0" before shifting to stop mode.

Figure 1.8.5. Clock control related register (4)

Clock Generating Circuit

(1) Main clock

The main clock is a clock source for CPU operation and peripheral I/O. Figure 1.8.6 shows example of a main clock. When a reset, the clock oscillates and after a reset, the clock is divided by 8 to the BCLK (CPU operating clock).

(a) Main clock On/Off function

- Main clock (XIN-XOUT) stop bit of system control register 0 (bit 5 at address 000616)

0: Main clock On

1: Main clock Off

Also, the clock is stopped by shifting to the stop mode.

- All clock stop control bit of system control register 1 (bit 0 at address 000716)

0: Clock on

1: All clocks off (stop mode)

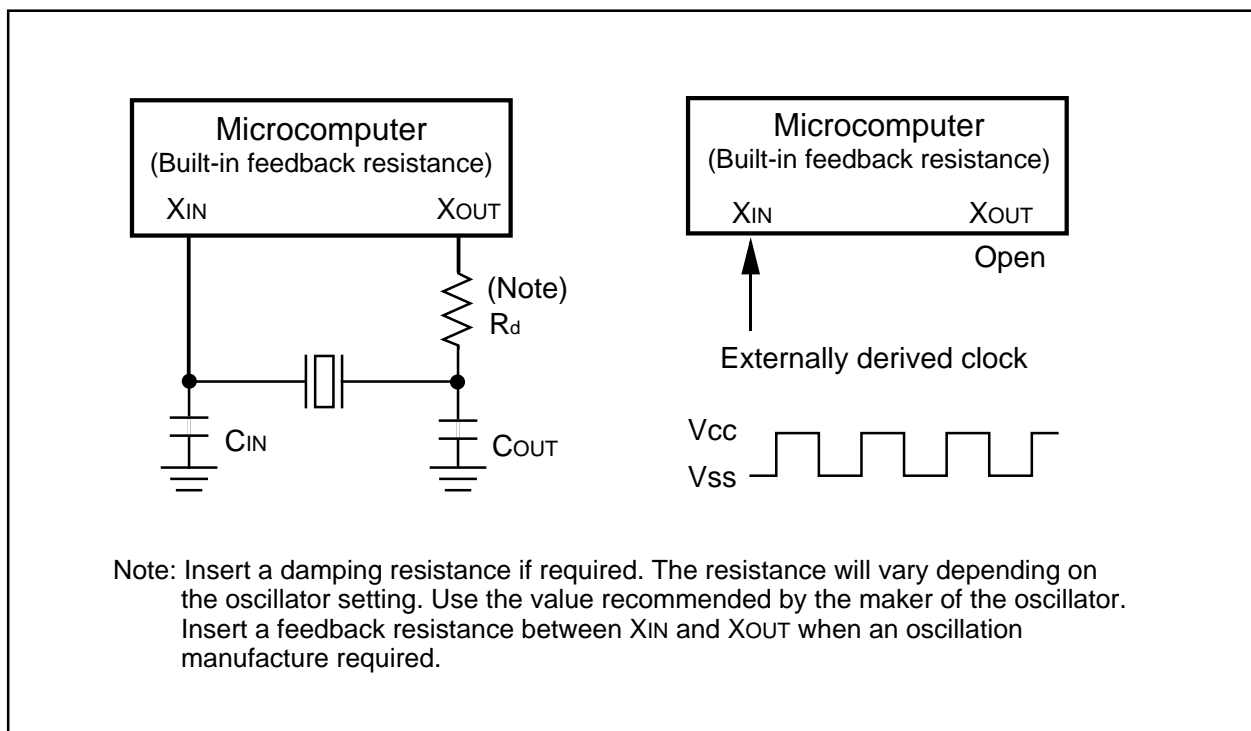


Figure 1.8.6. Examples of main clock

(2) Sub clock

The sub clock is a clock source for CPU operation and count source for timer A and B. Figure 1.8.7 shows example of sub clock. When the sub clock is used, set ports P86 and P87 to no pull-up resistance with the input port. No sub clock is generated during and after a reset.

(a) Sub clock On/Off function

When you want to use sub clock, set the following bit and sub clock enabled.

- Port Xc select bit of system control register 0 (bit 4 at address 000616)
 - 0: I/O port (sub clock off)
 - 1: XIN-XOUT generation (sub-clock on)

Also, shifting to the stop mode stops the clock.

- All clock stop control bit of system control register 1 (bit 0 at address 000716)
 - 0: Clock On
 - 1: All clock stop (stop mode)

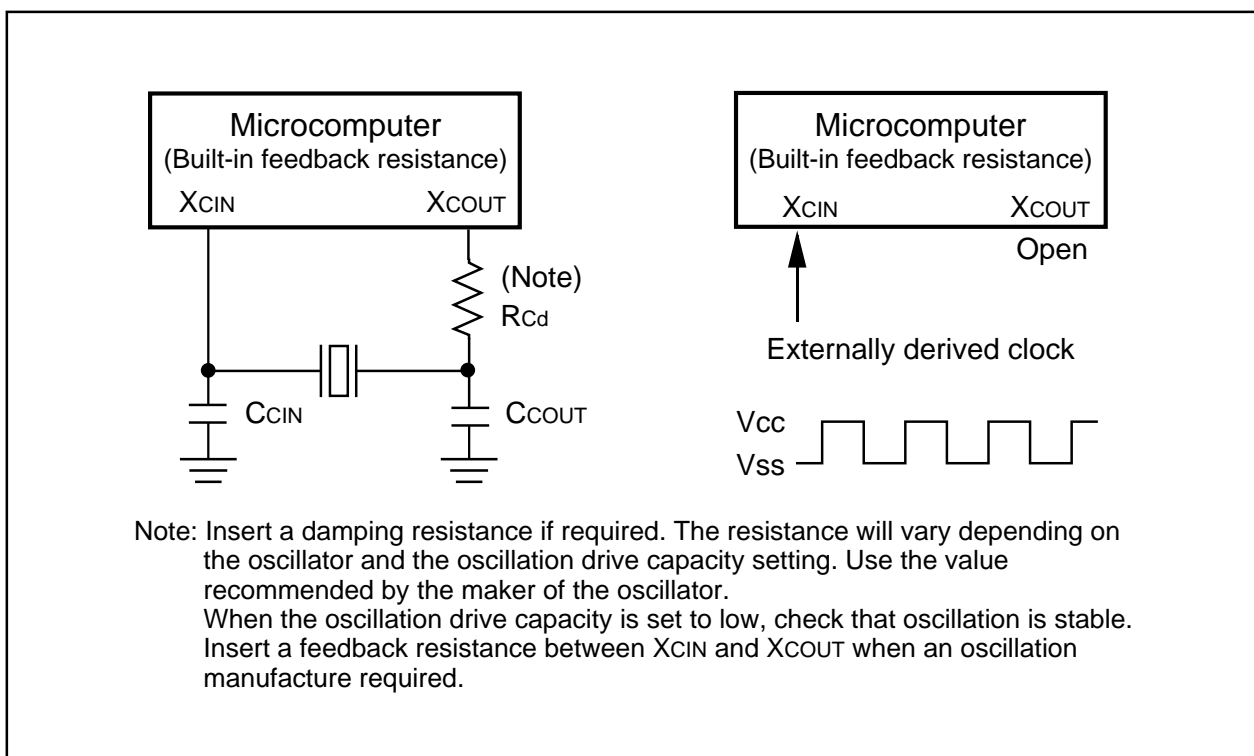


Figure 1.8.7. Examples of sub clock

Clock Generating Circuit

(3) Oscillation stop detect function (OSD function)

This function monitor the main clock (XIN pin). When main clock is stopped, internal ring oscillator starts oscillation and replace the main clock. Then oscillation stop detect interrupt process is operated.

When frequency of main clock is less or equal than 2MHz, this function does not work.

(a) OSD function enable/disable

- OSD enable bit of oscillation stop detect register (bit 0 at address 000D16)

0: OSD function disabled

1: OSD function enabled

Set OSD enable bit (bit 0) of oscillation stop detect register to "0" to disable OSD function before setting stop mode. Stop mode is canceled before setting this bit to "1".

(b) Operation when oscillation stop detects

- 1) When XIN oscillation stops, a built in ring oscillation starts as a main clock automatically.
- 2) OSD interrupt request is generated, jump to an address FFFFF0₁₆ to FFFFF3₁₆ allocated fixed vector table (watchdog timer interrupt vector) and execute program of jump address.
- 3) OSD interrupt shares vector table with watchdog timer interrupt. When using both OSD and watchdog timer interrupts, read and judge OSD flag in interrupt process routine.

OSD flag of oscillation stop detect register (bit 2 at address 000D16)

1: Oscillation stop detects

- 4) XIN does not become main clock although XIN On after oscillation stop detects. When you want XIN to be main clock, execute a process shown in Figure 1.8.8.

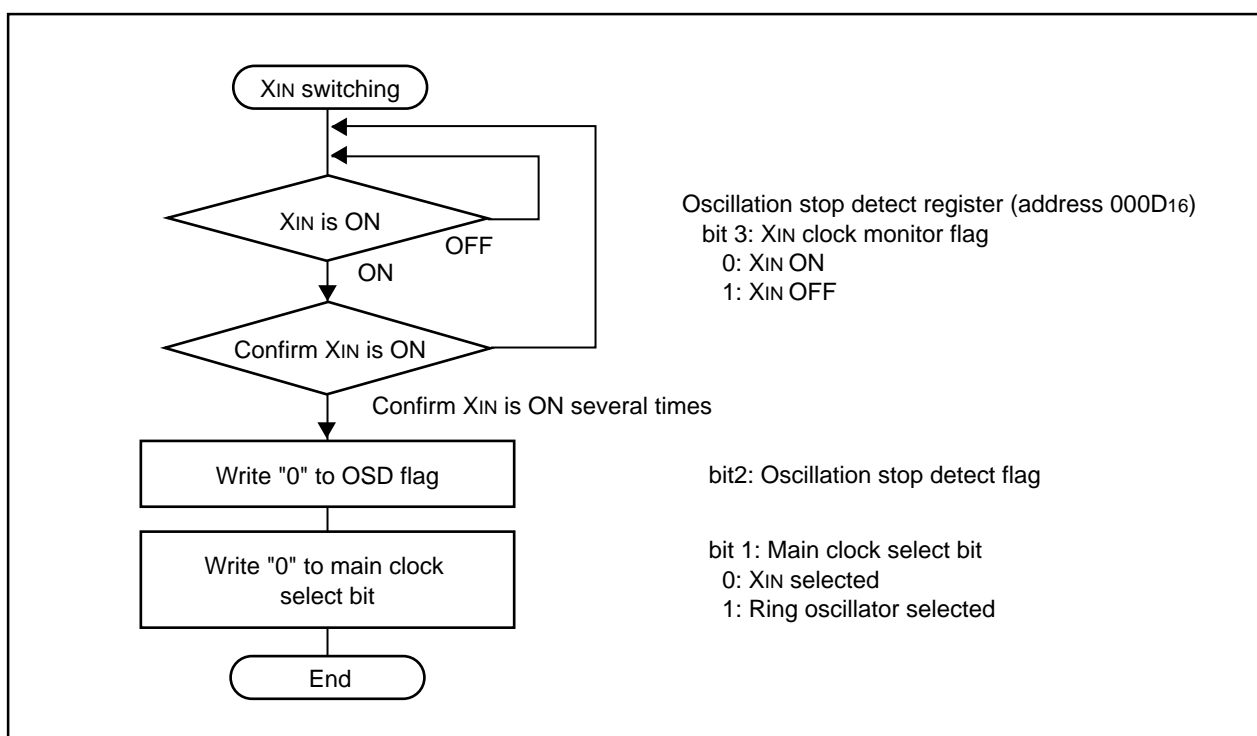


Figure 1.8.8. Main clock switching sequence

CPU clock (BCLK)

Main clock, sub clock or clock from ring oscillator can be selected as clock source for BCLK.

System clock select bit of system clock control register (bit 7 at address 0006₁₆)

0: Main clock is selected (XIN-XOUT)

1: Sub clock is selected (XCIN-XCOUT)

Main clock select bit of oscillation stop detect register (bit 1 at address 000D₁₆)

0: Main clock is selected (XIN-XOUT)

1: Clock from ring oscillator is selected

Table 1.8.3. BCLK source and setting bit

BCLK source	System clock select bit (Bit 7 of address 0006 ₁₆)	Main clock select bit (Bit 1 of address 000D ₁₆)
Main clock (XIN-XOUT)	0	0
Sub clock (XCIN-XCOUT)	1	0
Ring oscillator	0	1

When main clock or ring oscillator clock is selected as clock source for BCLK, the BCLK is the clock derived by dividing the main clock or ring oscillator clock by 1, 2, 3, 4, 6, 8, 10, 12, 14 or 16.

Main clock divide rate select bit of main clock division register (bit 0 to 4 at address 000C₁₆)

The BCLK is derived by dividing the main clock (XIN-XOUT) by 8 after a reset. (Main clock division register = "XXX01000₂")

When main clock is stopped under changing to stop mode or selecting XIN-XOUT (main clock select bit = "0"), the main clock division register is set to the division by 8 ("XXX01000₂").

When ring oscillator clock is selected as clock source for BCLK, although main clock is stopped, the contents of main clock division register is maintained.

Peripheral function clock

Main clock, sub clock, PLL clock or ring oscillator clock can be selected as clock source for peripheral function.

(1) f₁, f₈, f_{2n}

The clock is derived from the main clock or by dividing it by 1, 8 or 2ⁿ (n=1 to 15). It is used for the timer A and timer B counts and serial I/O and UART operation clock.

The f_{2n} division rate is set by the count source prescaler register. Figure 1.8.5 shows the count source prescaler register.

(2) f_{AD}

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

(3) f_{C32}

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

(4) f_{PLL}

This clock is 80 MHz generated by PLL synthesizer. It is used for the intelligent I/O group 3.

Clock Generating Circuit

Clock Output

You can output clock from the P53 pin.

- BCLK output function select bit of processor mode register 0 (bit 7 at address 0004₁₆)
- ALE select bits of processor mode register 1 (bit 4 and 5 at address 0005₁₆)
- Clock output function select bits of system clock select register (bits 1 and 0 at address 0006₁₆)

Table 1.8.4 shows clock output setting (single chip mode) and Table 1.8.5 shows clock output setting (memory expansion/microprocessor mode).

Table 1.8.4. Clock output setting (single chip mode)

BCLK output function select bit	Clock output function select bit		ALE pin select bit		P53/BCLK/ALE/CLKOUT pin function
	PM07	CM01	CM00	PM15	
Ignored	0	0	Ignored	Ignored	P53 I/O port
1	0	1	Ignored	Ignored	fc output (Note)
1	1	0	Ignored	Ignored	f8 output (Note)
1	1	1	Ignored	Ignored	f32 output (Note)

Note :Must use P57 as input port.

Table 1.8.5. Clock output setting (memory expansion/microprocessor mode)

BCLK output function select bit	Clock output function select bit		ALE pin select bit		P53/BCLK/ALE/CLKOUT pin function
	PM07	CM01	CM00	PM15	
0	0	0	"0, 0" "1, 0" "1, 1"		BCLK output
1	0	0			"L" output (not P53)
1	0	1			fc output
1	1	0			f8 output
1	1	1			f32 output
Ignored	0	0	0	1	ALE output

Note: The processor mode register 0 and 1 are protected from false write by program run away.

Set bit 1 to "1" at protect register (address 000A₁₆) and release protect before rewriting processor mode register 0 and 1.

Power Saving

There are three power save modes. Figure 1.8.9 shows the clock transition between each of the three modes, (1), (2), and (3).

- Normal operating mode
CPU and peripheral function operate when supplying clock. Power dissipation is reduced by making BCLK slow.
- Wait mode
BCLK is stopped. Peripheral function clock is stopped as desired. Main clock and sub clock isn't stopped. Power dissipation is reduced than normal operating mode.
- Stop mode (Note 1)
Main clock, sub clock and PLL synthesizer are stopped. CPU and peripheral function clock are stopped. Power dissipation is the most few in this mode.
Note :When using stop mode, oscillation stop detect function must be canceled.

(1) Normal operating mode

High-speed mode

Main clock one cycle forms CPU operating clock.

Medium-speed mode

The main clock divided into 2, 3, 4, 6, 8, 10, 12, 14, or 16 forms CPU operating clock.

Low-speed mode

Subclock (fc) forms CPU operating clock.

Low power-dissipation mode

This mode is selected when the main clock is stopped from low-speed mode. Only the peripheral functions for which the subclock was selected as the count source continue to run.

Ring oscillator mode

The ring oscillator clock divided into 2, 3, 4, 6, 8, 10, 12, 14, or 16 forms CPU operating clock.

Ring oscillator low power-dissipation mode

This mode is selected when the main clock is stopped from low-speed mode.

When switching BCLK from ring oscillator to main clock, switch clock after main clock oscillates fully stable. After setting divided by 8 (main clock division register =0816) in ring oscillator mode, switching to the middle mode (divided by 8) is recommended.

(2) Wait mode

In wait mode, BCLK is stopped and CPU and watchdog timer operated by BCLK are halted. The main clock, subclock and ring oscillator clock continue to run.

(a) Shifting to wait mode

Execute WAIT instruction.

(b) Peripheral function clock stop function

The f1, f8 and f2n being supplied to the internal peripheral functions stops. The internal peripheral functions operated by the clock stop.

WAIT peripheral function clock stop bit of system clock control register 0 (bit 2 at address 000616)

0: Do not stop f_1 , f_8 and f_{2n} in wait mode and do not stop supplying clock to PLL circuit

1: Stop f_1 , f_8 and f_{2n} in wait mode and stop supplying clock to PLL circuit

(c) The status of the ports in wait mode

Table 1.8.6 shows the status of the ports in wait mode.

(d) Exit from wait mode

Wait mode is cancelled by a hardware reset or interrupt. If a peripheral function interrupt is used to cancel wait mode, set the following registers.

Interrupt priority set bits for exiting a stop/wait state of exit priority register (bits 0 to 2 at address 009F16) :RLVL0 to RLVL2

Set the same level as the flag register (FLG) processor interrupt level (IPL).

Interrupt priority set bits of interrupt control register (bits 0 to 2)

Set to a priority level above the level set by RLVL0 to RLVL2 bits

Interrupt enable flag of FLG register

I = 1

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 1.8.6. Port status during wait mode

Pin		Memory expansion mode Microprocessor mode	Single-chip mode
Address bus, data bus, \overline{CS}_0 to \overline{CS}_3 , BHE		Retains status before wait mode	/
RD, WR, WRL, WRH, DW, CASL, CASH		"H" (Note)	
RAS		"H" (Note)	
\overline{HLDA} , BCLK		"H"	
ALE		"L"	
Port		Retains status before wait mode	
CLKOUT	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".

(3) Stop mode

All oscillation, main clock, subclock, and PLL synthesizer stop in this mode. Because the oscillation of BCLK and peripheral clock stops in stop mode, peripheral functions such as the A-D converter, timer A and B, serial I/O, intelligent I/O and watchdog timer do not function.

The content of the internal RAM is retained provided that VCC remains above 2.5V.

When changing to stop mode, the main clock division register (000C16) is set to "XXX010002" (division by 8 mode).

(a) Changing to stop mode

All clock stop control bit of system clock control register 1 (bit 0 at address 000716)

0: Clock ON

1: All clocks off (stop mode)

Before changing to stop mode, set bit 7 of PLL control register 0 (address 037616) to "0" to stop PLL.

Also, set bit 0 of VDC control register for PLL (address 001716) to "1" to turn PLL circuit power off.

(b) The status of the ports in stop mode

Table 1.8.7 shows the status of the ports in stop mode.

(c) Exit from stop mode

Stop mode is cancelled by a hardware reset or interrupt. If a peripheral function interrupt is used to cancel stop mode, set the following registers.

- Interrupt priority set bits for exiting a stop/wait state of exit priority register (bits 0 to 2 at address 009F16) :RLVL0 to RLVL2
Set the same level as the flag register (FLG) processor interrupt level (IPL).

- Interrupt priority set bits of interrupt control register (bits 0 to 2)

Set to a priority level above the level set by RLVL0 to RLVL2 bits

- Interrupt enable flag of FLG register

I = 1

When exiting from stop mode using peripheral interrupt request, CPU operates the following BCLK and the relevant interrupt routine is executed.

- When subclock was set as BCLK before changing to stop mode, subclock is set to BCLK after cancelled stop mode
- When main clock was set as BCLK before changing to stop mode, the main clock division by 8 is set to BCLK after cancelled stop mode.

Table 1.8.7. Port status during stop mode

Pin		Memory expansion mode Microprocessor mode	Single-chip mode
Address bus, data bus, CS ₀ to CS ₃ , BHE		Retains status before stop mode	/
RD, WR, WRL, WRH, DW, CASL, CASH		"H" (Note)	
RAS		"H" (Note)	
HLDA, BCLK		"H"	
ALE		"H"	
Port		Retains status before stop mode	
CLKOUT	When fc selected	"H"	
	When f8, f32 selected	Retains status before stop mode	

Note :When self-refresh is done in operating DRAM control, CAS and RAS becomes "L".

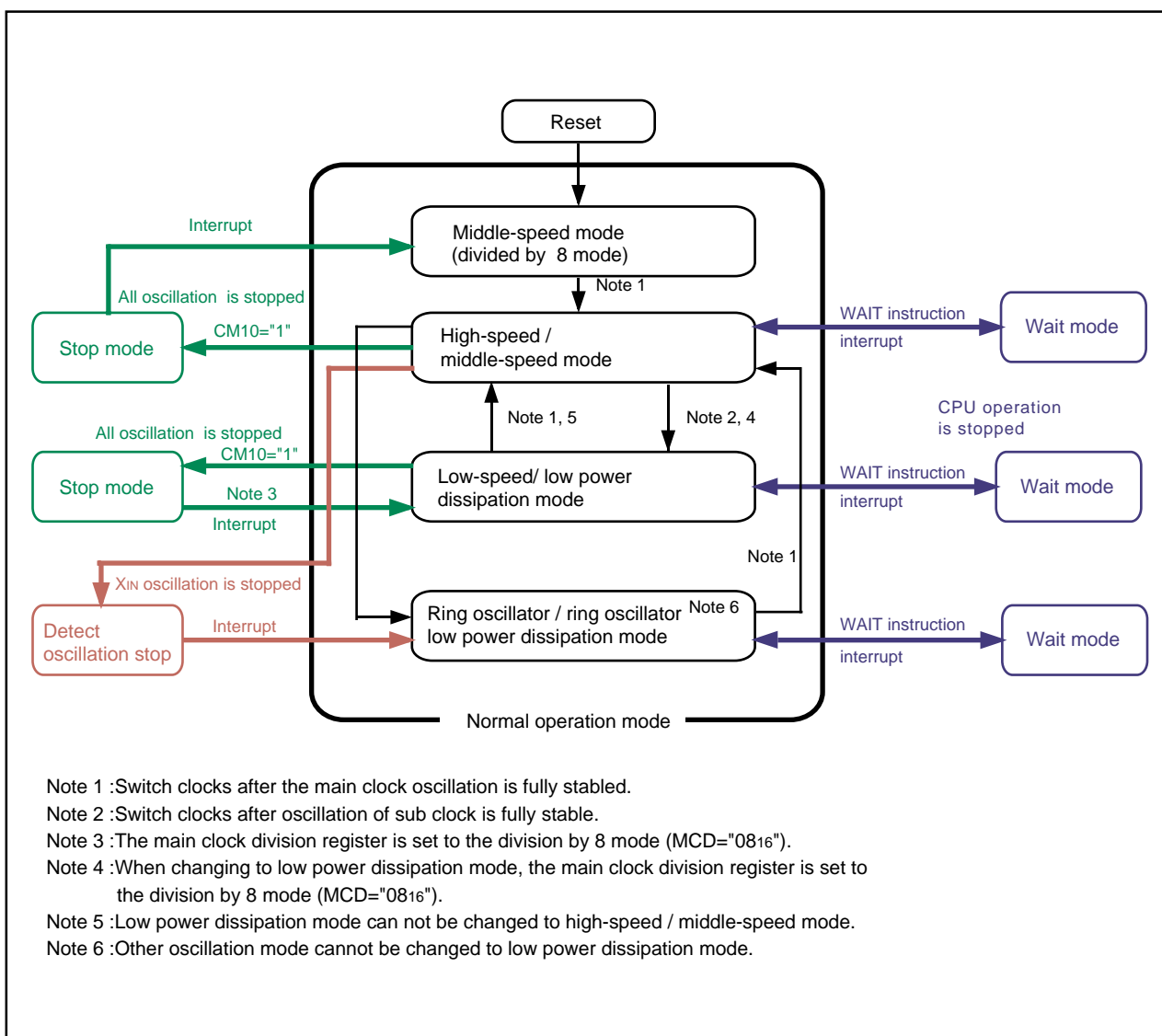


Figure 1.8.9. Clock transition

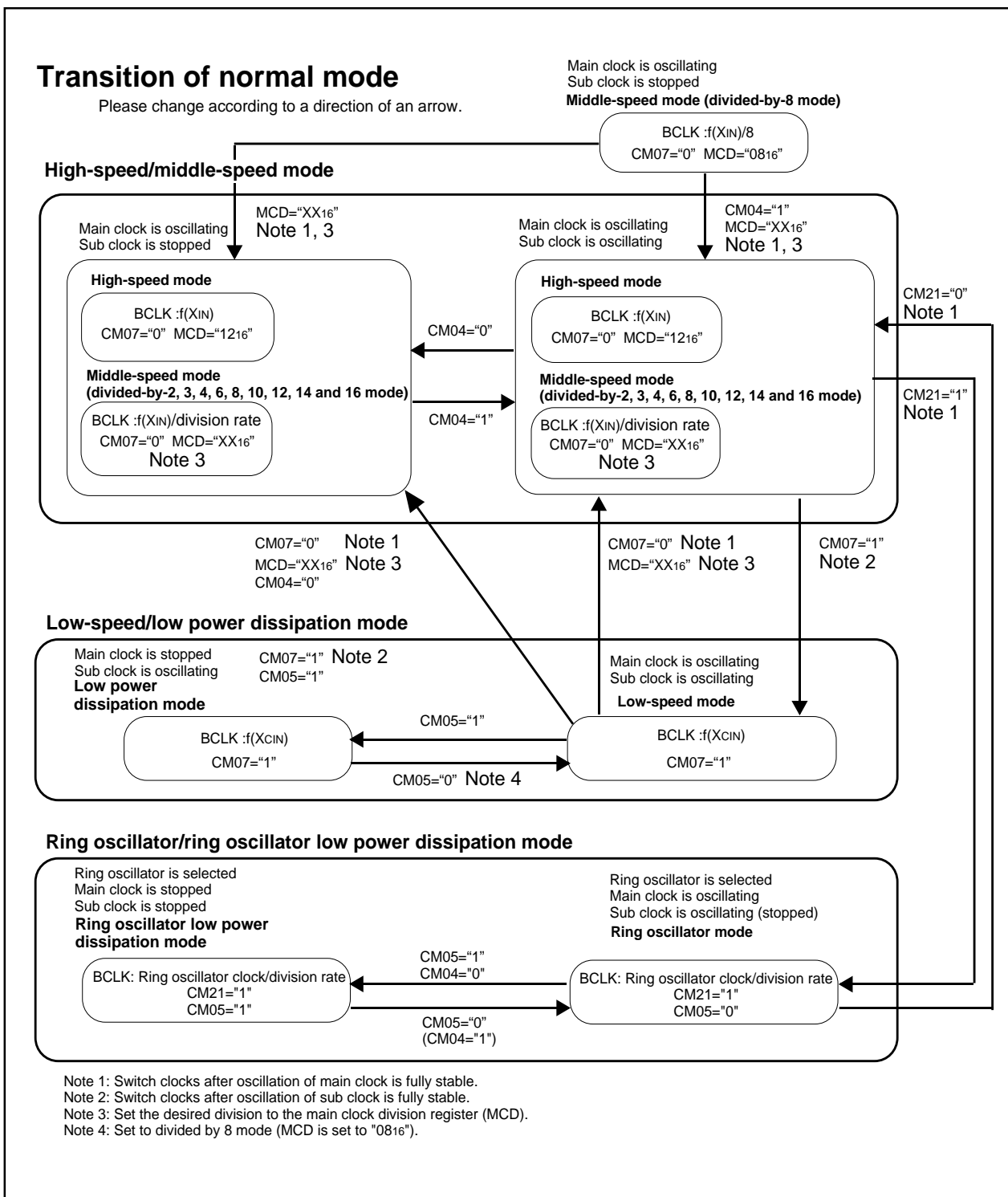


Figure 1.8.10. Clock transition

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 1.8.11 shows the protect register. The following registers are protected by the protect register.

(1) Registers protected by PRC0 (bit 0)

- System clock control registers 0 and 1 (addresses 0006₁₆ and 0007₁₆)
- Main clock division register (address 000C₁₆)
- Oscillation stop detect register (address 000D₁₆)
- PLL control register 0 (address 0376₁₆)

(2) Registers protected by PRC1 (bit 1)

- Processor mode registers 0 and 1 (addresses 0004₁₆ and 0005₁₆)
- Three-phase PWM control registers 0 and 1 (addresses 0308₁₆ and 0309₁₆)

(3) Registers protected by PRC2 (bit 2)

- Port P9 direction register (address 03C7₁₆)
- Function select register A3 (address 03B5₁₆)

(4) Registers protected by PRC3 (bit 3)

- VDC control register for PLL (address 0017₁₆)
- VDC control register 0 (address 001F₁₆)

If, after "1" (write-enabled) has been written to the PRC2, 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, PRC1 and PRC3 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".

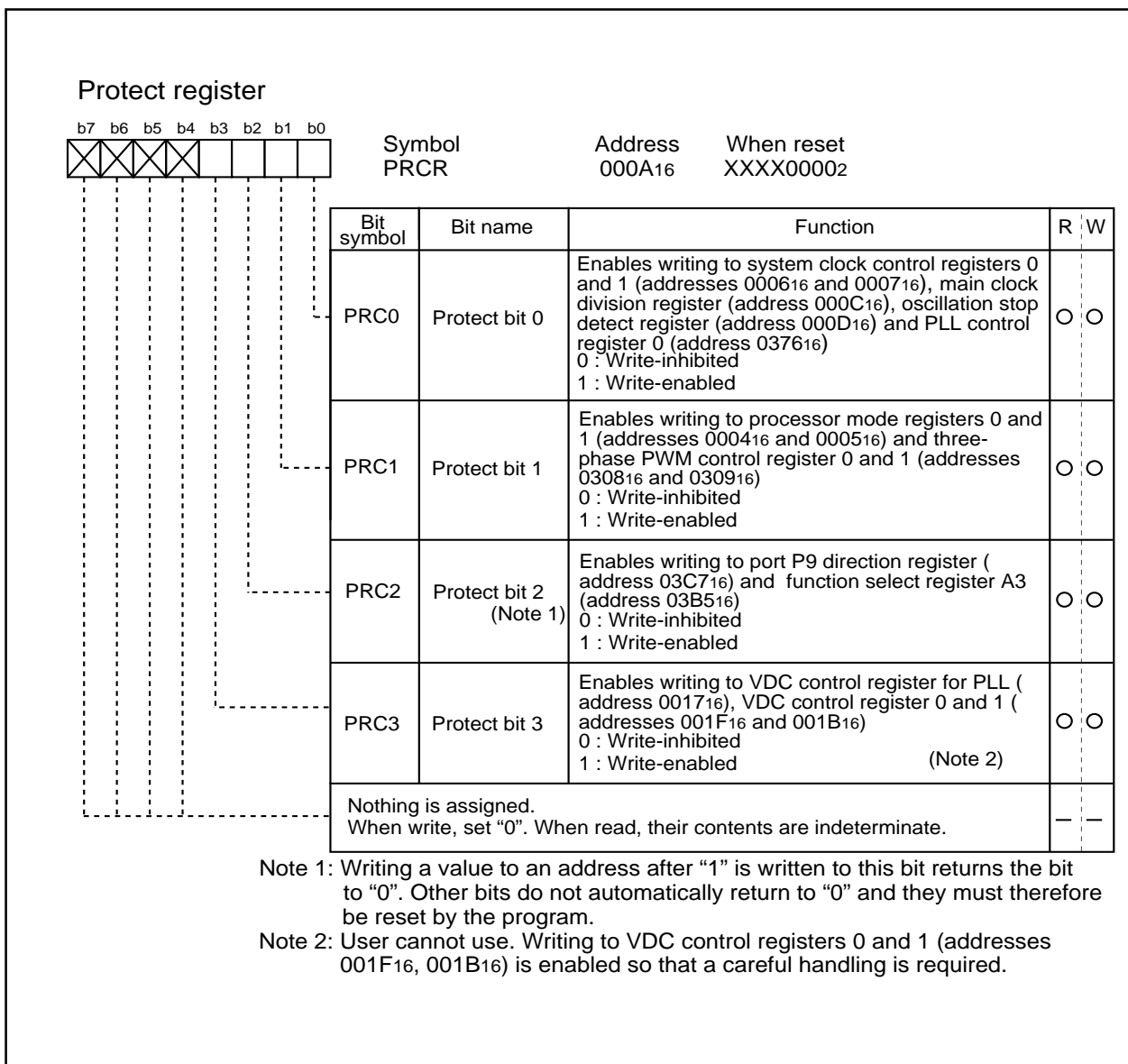


Figure 1.8.11. Protect register

Interrupt Outline

Types of Interrupts

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

Figure 1.9.1 lists the types of interrupts.

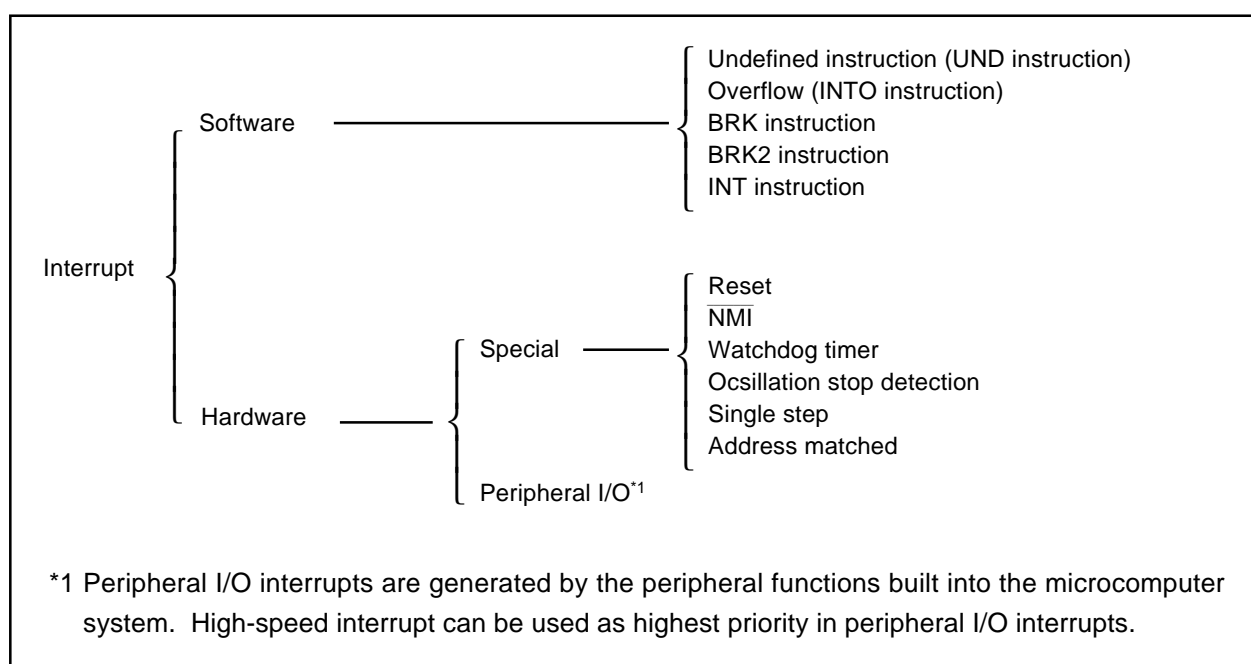


Figure 1.9.1. Classification of interrupts

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 7 to 54 and 57 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 54 and 57.

Hardware Interrupts

There are Two types of 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.

- **Oscillation stop detect interrupt**

This interrupt is caused by the oscillation stop detect function.

It occurs when detecting the X_{IN} oscillation is stopped.

- **Single-step interrupt**

This interrupt is used exclusively for debugger purposes. These interrupts normally do not need to use this interrupt. A single-step interrupt occurs when the D flag is set (= 1); in this case, an interrupt is generated each time an instruction is executed.

- **Address-match interrupt**

This interrupt occurs when the program's execution address matches the contents of the address match register while the address match interrupt enable bit is set (= 1).

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

(2) Peripheral I/O interrupts

A peripheral I/O interrupt is generated by one of the 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 7 through 54 and 57 the INT instruction uses. Peripheral I/O interrupts are maskable interrupts.

- **UART related interrupt (UART0 to 4)**

- UART transmission/NACK interrupt
- UART reception/ACK interrupt
- Bus collision detection, start/stop condition detection interrupts

This is an interrupt that the serial I/O bus collision detection generates. When I²C mode is selected, start, stop condition interrupt is selected.

- **DMA0 through DMA3 interrupts**

- **Key-input interrupt**

A key-input interrupt occurs if an "L" is input to the \overline{KI} pin.

- **A-D conversion interrupt (AD0, 1)**

- **Timer A interrupt (TA0 to 4)**

- **Timer B interrupt (TB0 to 5)**

- **\overline{INT} interrupt ($\overline{INT0}$ to $\overline{INT5}$)**

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

- **Intelligent I/O interrupt**

- **CAN interrupt**

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 an 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 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. When using high speed interrupt, DMA II cannot be used.

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 relocatable vector table, in which addresses can be varied by the setting.

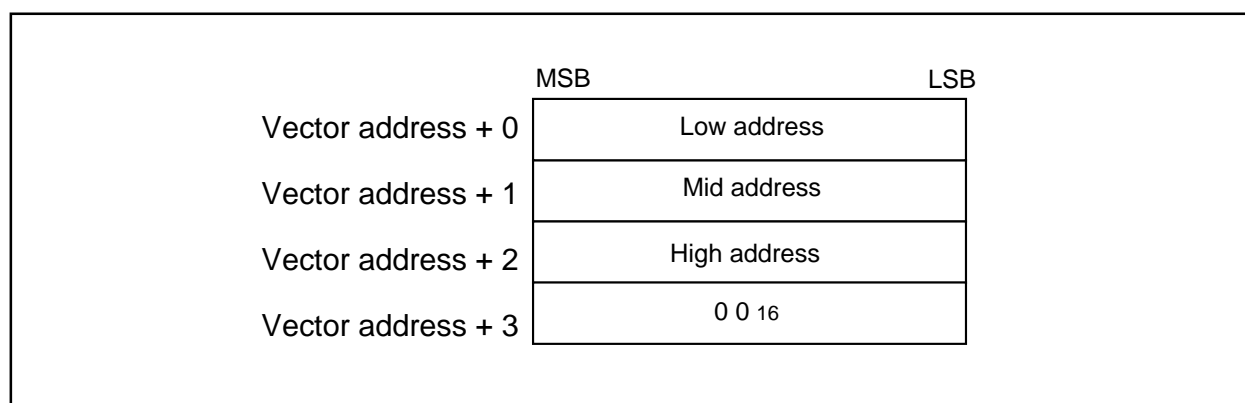


Figure 1.9.2. Format for specifying interrupt vector addresses

• 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 FFFFDC₁₆ to FFFFFFF₁₆. Each vector 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 FFFF E3 ₁₆	Interrupt on INTO instruction
BRK instruction	FFFFE4 ₁₆ to FFFF E7 ₁₆	If contents of FFFF E7 ₁₆ is filled with FF ₁₆ , program execution starts from the address shown by the vector in the relocatable vector table
Address match	FFFFE8 ₁₆ to FFFF EB ₁₆	There is an address-matching interrupt enable bit
Watchdog timer interrupt	FFFFF0 ₁₆ to FFFF F3 ₁₆	Share it with watchdog timer and oscillation stop detect
NMI	FFFFF8 ₁₆ to FFFF FB ₁₆	External interrupt by input to $\overline{\text{NMI}}$ pin
Reset	FFFFFC ₁₆ to FFFF F ₁₆	

- **Vector table dedicated for emulator**

Table 1.9.2 shows interrupt vector address, which is vector table register dedicated for emulator (address 000020₁₆ to 000022₁₆). 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 the interrupt vector table register dedicated for emulator (address 000020₁₆ to 000022₁₆).

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	Interrupt for debugger
Single step	000020 ₁₆ to 000022 ₁₆	

- **Relocatable vector tables**

The addresses in the relocatable 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 relocatable 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 relocatable 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) (1/2)

Softwear interrupt number	Vector table address Address(L)to address(H) (Note 1)	Interrupt source
Softwear interrupt number 0 (Note 2)	+0 to +3 (0000 ₁₆ to 0003 ₁₆)	BRK instruction
Softwear interrupt number 7	+28 to +31 (001C ₁₆ to 001F ₁₆)	A-D channel 1
Softwear interrupt number 8	+32 to +35 (0020 ₁₆ to 0023 ₁₆)	DMA0
Softwear interrupt number 9	+36 to +39 (0024 ₁₆ to 0027 ₁₆)	DMA1
Softwear interrupt number 10	+40 to +43 (0028 ₁₆ to 002B ₁₆)	DMA2
Softwear interrupt number 11	+44 to +47 (002C ₁₆ to 002F ₁₆)	DMA3
Softwear interrupt number 12	+48 to +51 (0030 ₁₆ to 0033 ₁₆)	Timer A0
Softwear interrupt number 13	+52 to +55 (0034 ₁₆ to 0037 ₁₆)	Timer A1
Softwear interrupt number 14	+56 to +59 (0038 ₁₆ to 003B ₁₆)	Timer A2
Softwear interrupt number 15	+60 to +63 (003C ₁₆ to 003F ₁₆)	Timer A3
Softwear interrupt number 16	+64 to +67 (0040 ₁₆ to 0043 ₁₆)	Timer A4
Softwear interrupt number 17	+68 to +71 (0044 ₁₆ to 0047 ₁₆)	UART0 transmit/NACK (Note 3)
Softwear interrupt number 18	+72 to +75 (0048 ₁₆ to 004B ₁₆)	UART0 receive/ACK (Note 3)
Softwear interrupt number 19	+76 to +79 (004C ₁₆ to 004F ₁₆)	UART1 transmit/NACK (Note 3)
Softwear interrupt number 20	+80 to +83 (0050 ₁₆ to 0053 ₁₆)	UART1 receive/ACK (Note 3)
Softwear interrupt number 21	+84 to +87 (0054 ₁₆ to 0057 ₁₆)	Timer B0
Softwear interrupt number 22	+88 to +91 (0058 ₁₆ to 005B ₁₆)	Timer B1
Softwear interrupt number 23	+92 to +95 (005C ₁₆ to 005F ₁₆)	Timer B2
Softwear interrupt number 24	+96 to +99 (0060 ₁₆ to 0063 ₁₆)	Timer B3
Softwear interrupt number 25	+100 to +103 (0064 ₁₆ to 0067 ₁₆)	Timer B4
Softwear interrupt number 26	+104 to +107 (0068 ₁₆ to 006B ₁₆)	INT5
Softwear interrupt number 27	+108 to +111 (006C ₁₆ to 006F ₁₆)	INT4
Softwear interrupt number 28	+112 to +115 (0070 ₁₆ to 0073 ₁₆)	INT3
Softwear interrupt number 29	+116 to +119 (0074 ₁₆ to 0077 ₁₆)	INT2
Softwear interrupt number 30	+120 to +123 (0078 ₁₆ to 007B ₁₆)	INT1
Softwear interrupt number 31	+124 to +127 (007C ₁₆ to 007F ₁₆)	INT0
Softwear interrupt number 32	+128 to +131 (0080 ₁₆ to 0083 ₁₆)	Timer B5
Softwear interrupt number 33	+132 to +135 (0084 ₁₆ to 0087 ₁₆)	UART2 transmit/NACK (Note 3)
Softwear interrupt number 34	+136 to +139 (0088 ₁₆ to 008B ₁₆)	UART2 receive/ACK (Note 3)
Softwear interrupt number 35	+140 to +143 (008C ₁₆ to 008F ₁₆)	UART3 transmit/NACK (Note 3)
Softwear interrupt number 36	+144 to +147 (0090 ₁₆ to 0093 ₁₆)	UART3 receive/ACK (Note 3)
Softwear interrupt number 37	+148 to +151 (0094 ₁₆ to 0097 ₁₆)	UART4 transmit/NACK (Note 3)
Softwear interrupt number 38	+152 to +155 (0098 ₁₆ to 009B ₁₆)	UART4 receive/ACK (Note 3)
Softwear interrupt number 39	+156 to +159 (009C ₁₆ to 009F ₁₆)	Bus collision detection, start/stop condition detection (UART2)(Note 3)
Softwear interrupt number 40	+160 to +163 (00A0 ₁₆ to 00A3 ₁₆)	Bus collision detection, start/stop condition detection (UART3/UART0)(Note 3)
Softwear interrupt number 41	+164 to +167 (00A4 ₁₆ to 00A7 ₁₆)	Bus collision detection, start/stop condition detection (UART4/UART1)(Note 3)
Softwear interrupt number 42	+168 to +171 (00A8 ₁₆ to 00AB ₁₆)	A-D channel 0
Softwear interrupt number 43	+172 to +175 (00AC ₁₆ to 00AF ₁₆)	Key input interrupt
Softwear interrupt number 44	+176 to +179 (00B0 ₁₆ to 00B3 ₁₆)	Intelligent I/O interrupt 0
Softwear interrupt number 45	+180 to +183 (00B4 ₁₆ to 00B7 ₁₆)	Intelligent I/O interrupt 1
Softwear interrupt number 46	+184 to +187 (00B8 ₁₆ to 00BB ₁₆)	Intelligent I/O interrupt 2
Softwear interrupt number 47	+188 to +191 (00BC ₁₆ to 00BF ₁₆)	Intelligent I/O interrupt 3
Softwear interrupt number 48	+192 to +195 (00C0 ₁₆ to 00C3 ₁₆)	Intelligent I/O interrupt 4
Softwear interrupt number 49	+196 to +199 (00C4 ₁₆ to 00C7 ₁₆)	Intelligent I/O interrupt 5
Softwear interrupt number 50	+200 to +203 (00C8 ₁₆ to 00CB ₁₆)	Intelligent I/O interrupt 6
Softwear interrupt number 51	+204 to +207 (00CC ₁₆ to 00CF ₁₆)	Intelligent I/O interrupt 7
Softwear interrupt number 52	+208 to +211 (00D0 ₁₆ to 00D3 ₁₆)	Intelligent I/O interrupt 8
Softwear interrupt number 53	+212 to +215 (00D4 ₁₆ to 00D7 ₁₆)	Intelligent I/O interrupt 9/CAN interrupt 0
Softwear interrupt number 54	+216 to +219 (00D8 ₁₆ to 00DB ₁₆)	Intelligent I/O interrupt 10/CAN interrupt 1

Table 1.9.3. Interrupt causes (variable interrupt vector addresses) (2/2)

Softwear interrupt number	Vector table address Address(L)to address(H) (Note 1)	Interrutp source
Softwear interrupt number 55	+220 to +223 (00DC ₁₆ to 00DF ₁₆)	Softwea interrupt
Softwear interrupt number 56	+224 to +227 (00E0 ₁₆ to 00E3 ₁₆)	Softwea interrupt
Softwear interrupt number 57	+228 to +231 (00E4 ₁₆ to 00E7 ₁₆)	Intelligent I/O interrupt 11/CAN interrupt 2
Softwear interrupt number 58 (Note 2) to Softwear interrupt number 63	+232 to +235 (00E8 ₁₆ to 00EB ₁₆) to +252 to +255 (00FC ₁₆ to 00FF ₁₆)	Softwea interrupt

Note 1: Address relative to address in interrupt table register (INTB).

Note 2: Cannot be masked by I flag.

Note 3: When IIC mode is selected, NACK/ACK, start/stop condition detection interrupts are selected. The fault error interrupt is selected when SS pin is selected.

Interrupt request reception

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), the processor interrupt priority level (IPL), interrupt request bit and interrupt priority level select bit are all independent of each other, so they do not affect any other bit. There are I flag and IPL in flag register (FLG). This flag and bit are described below.

Interrupt Enable Flag (I Flag) and processor Interrupt Priority Level (IPL)

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.

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.

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

Table 1.9.4. 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.

Interrupt control registers and Exit priority register

Peripheral I/O interrupts have their own interrupt control registers. Figure 1.9.3 and 1.9.4 show the interrupt control registers and figure 1.9.5 shows exit priority register.

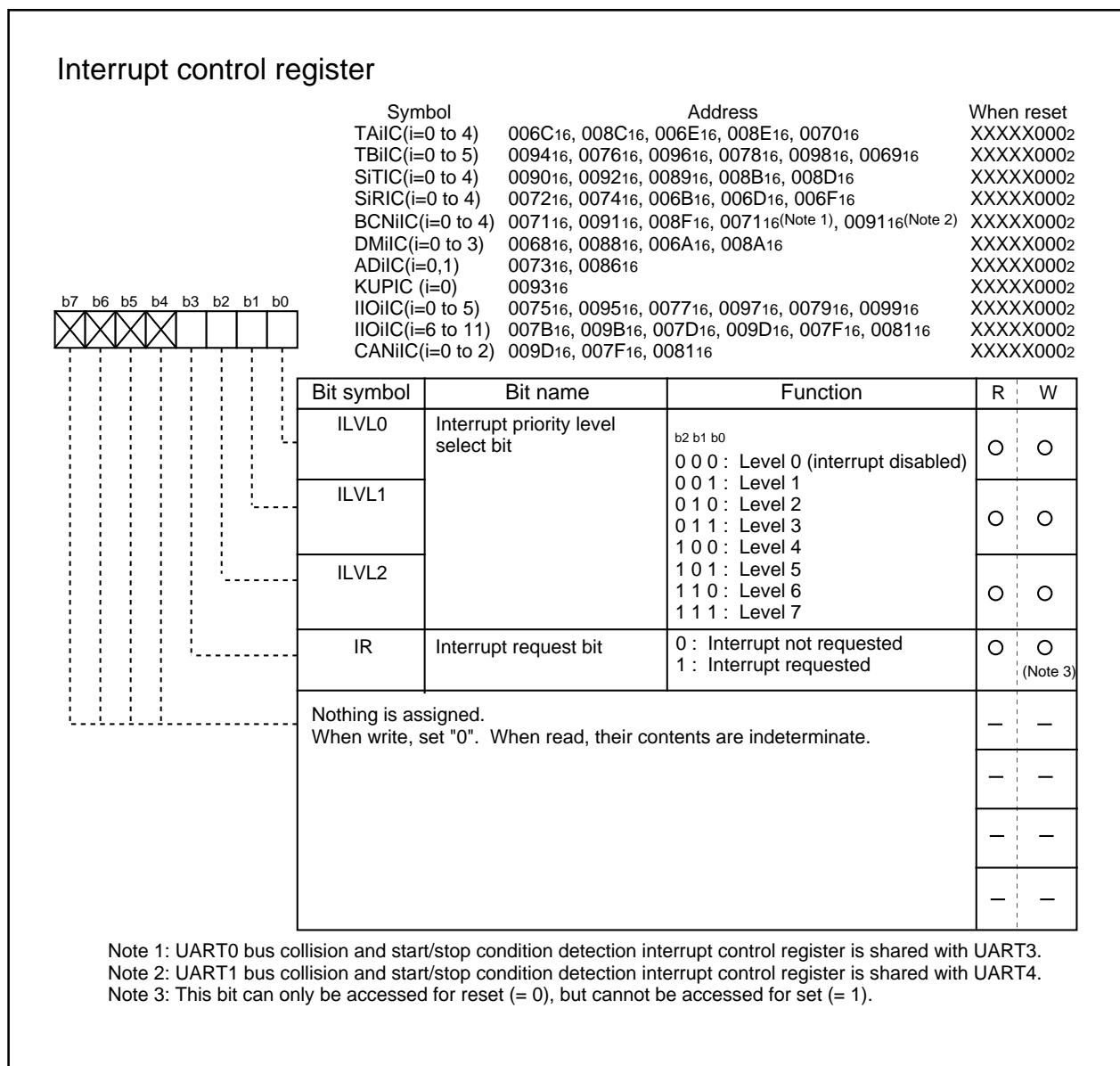


Figure 1.9.3. Interrupt control register (1)

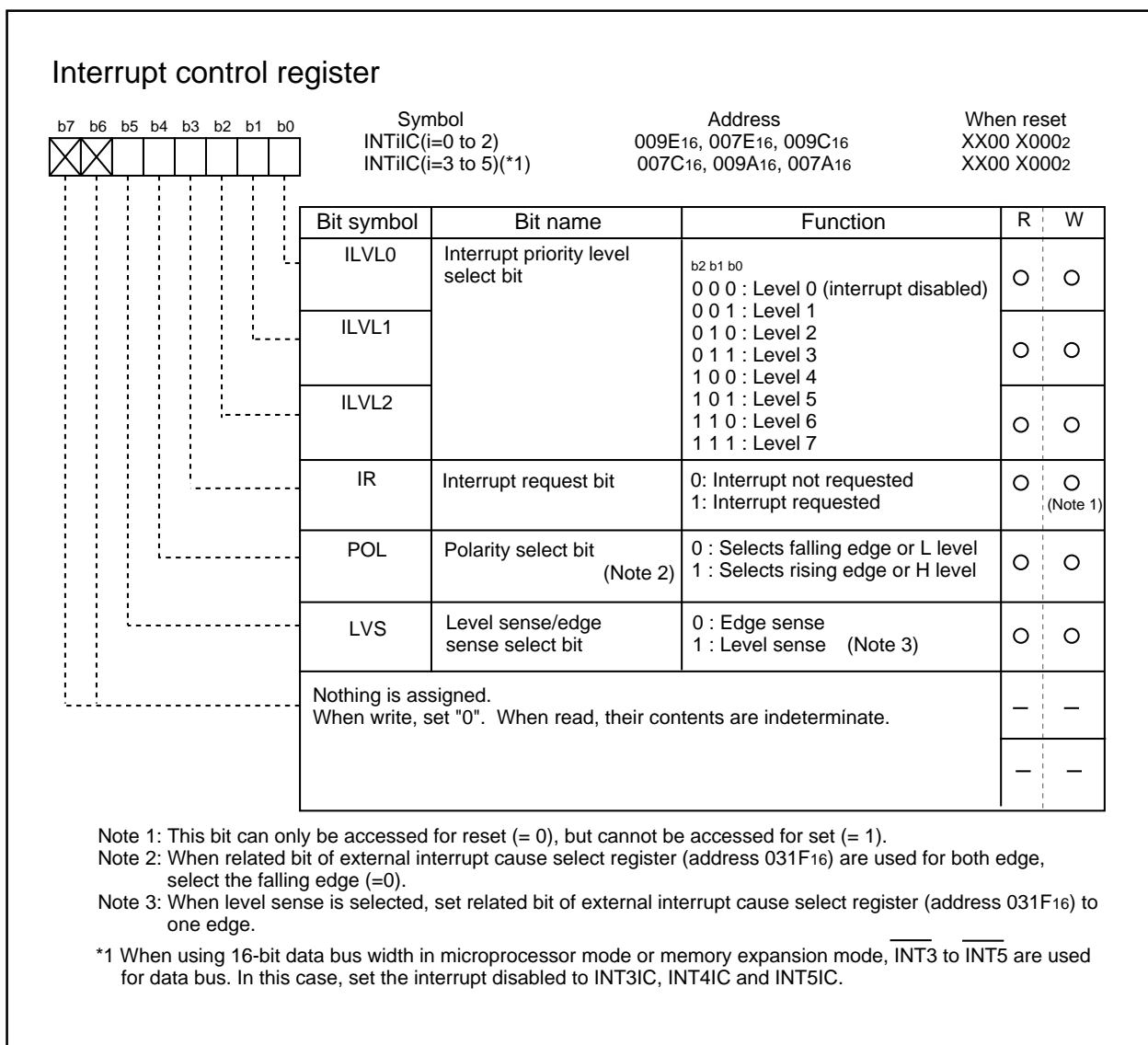


Figure 1.9.4. Interrupt control register (2)

Bit 0 to 2: Interrupt Priority Level Select Bits (ILVL0 to ILVL2)

Interrupt priority levels are set by ILVL0 to ILVL2 bits. When an interrupt request is generated, the interrupt priority level of this interrupt is compared with IPL. This interrupt is enabled only when its interrupt priority level is greater than IPL. This means that you can disable any particular interrupt by setting its interrupt priority level to 0.

Bit 3: Interrupt Request Bit (IR)

This bit is set (= 1) by hardware when an interrupt request is generated. The bit is cleared (= 0) by hardware when the interrupt request is acknowledged and jump to the interrupt vector.

This bit can be cleared (= 0) (but never be set to 1) in software.

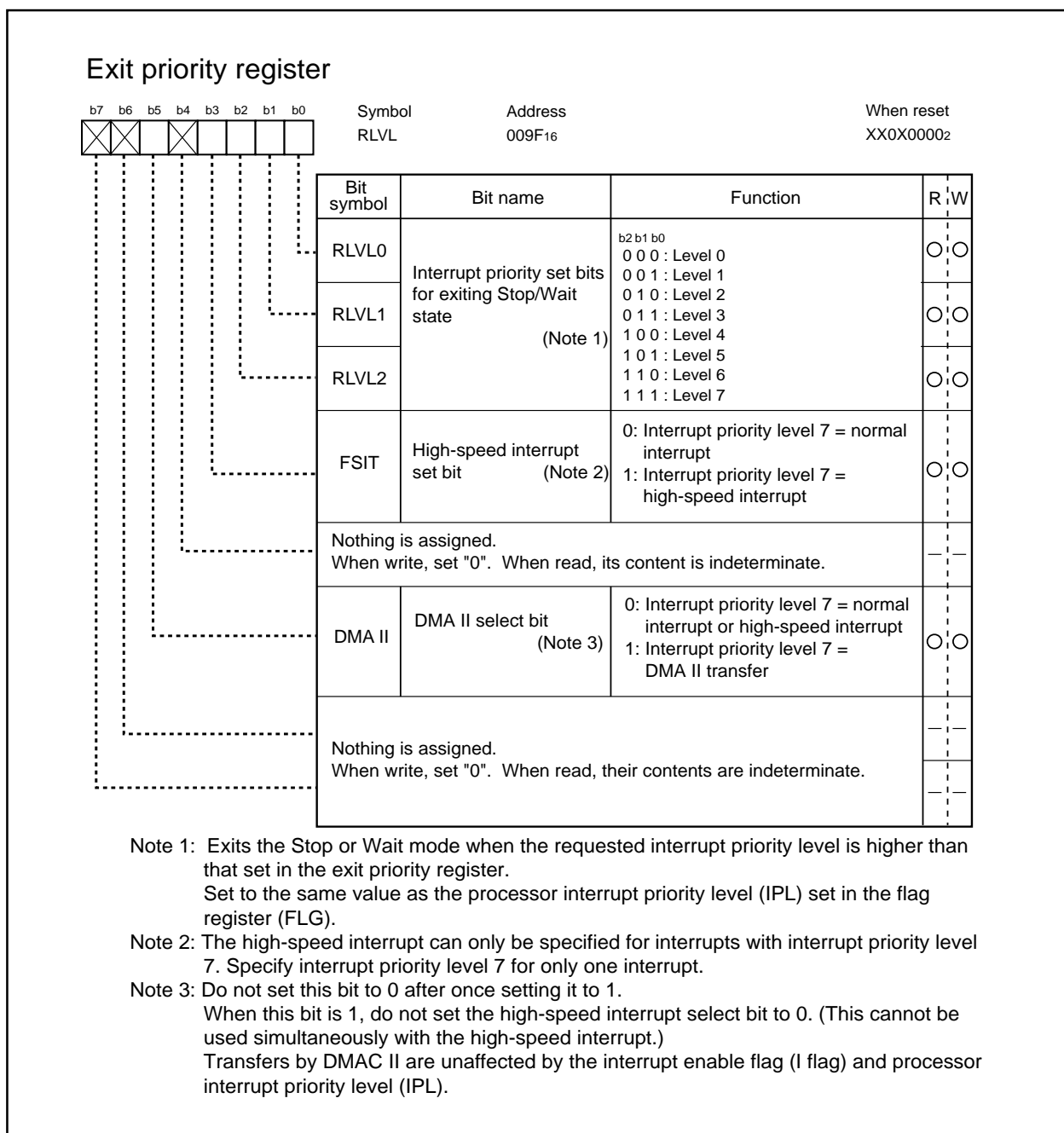


Figure 1.9.5. Exit priority register

Bit 0 to 2: Interrupt priority set bits for exiting Stop/Wait state (RLVL0 to RLVL2)

When using an interrupt to exit Stop mode or Wait mode, the relevant interrupt must be enabled and set to a priority level above the level set by the RLVL0 to RLVL2 bits. Set the RLVL0 to RLVL2 bits to the same level as the flag register (FLG) IPL.

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 contents of the flag register (FLG) 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 contents of the temporary register (Note) 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 1.9.6 shows the interrupt response time.

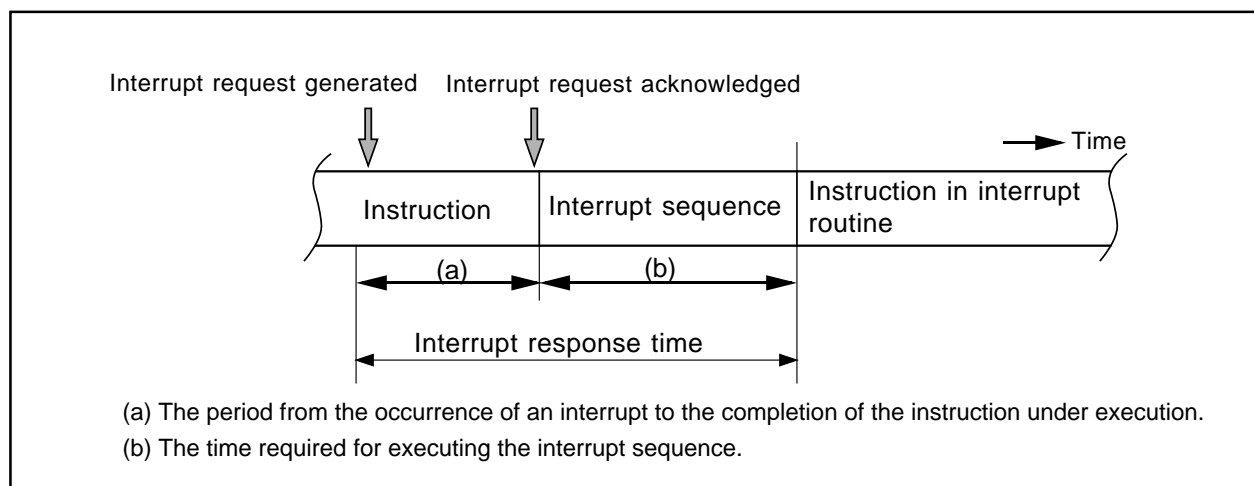


Figure 1.9.6. Interrupt response time

Interrupts

Time (a) varies with each instruction being executed. The DIVX instruction requires a maximum time of 29* cycles.

Time (b) is shown in table 1.9.5.

* 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 bit bus area, double the value of X and Y.

Table 1.9.5 Interrupt Sequence Execution Time

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

Note 1: Allocate interrupt vector addresses in even addresses as much as 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 1.9.6 is set to the IPL.

Table 1.9.6 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 are 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 1.9.7 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) are saved to the flag save register (SVF) and program counter (PC) are 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.

In high speed interrupt, switch register bank, then register bank 1 is used as high speed interrupt register. In this case, switch register bank mode for high-speed interrupt routine.

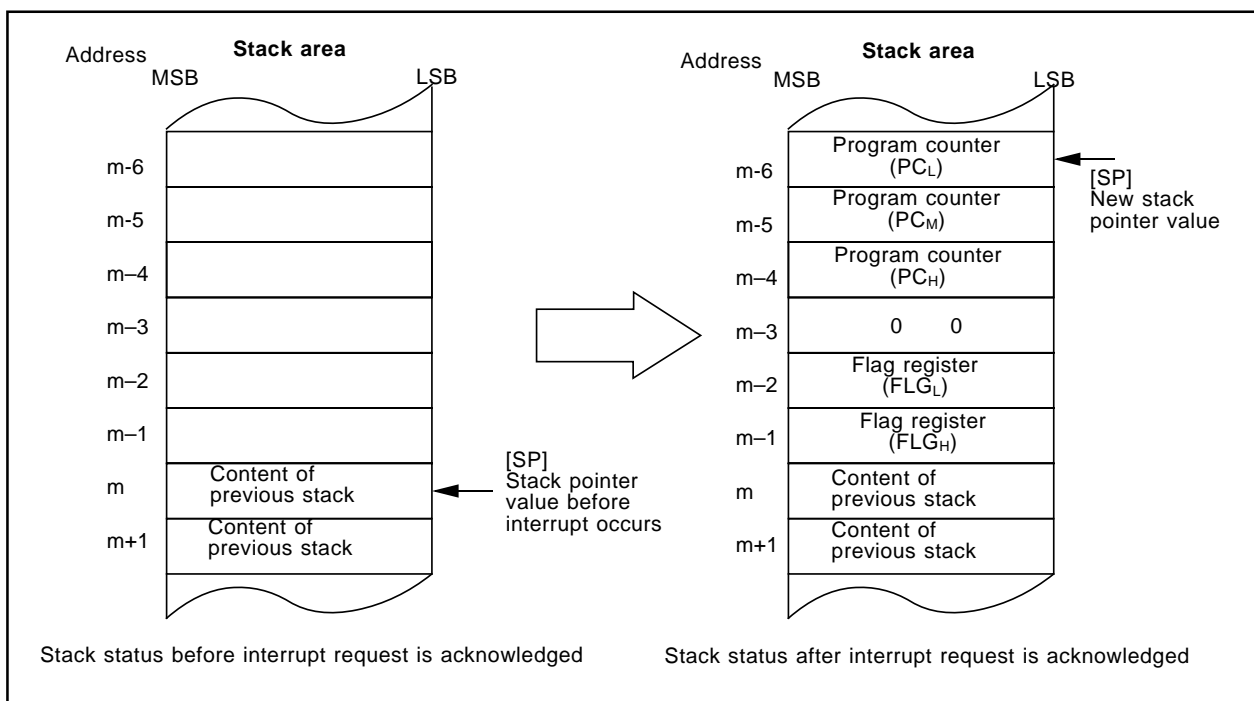


Figure 1.9.7. Stack status before and after an interrupt request is acknowledged

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, the interrupt with the highest priority will be acknowledged.

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 are 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 1.9.8 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 1.9.9 shows the interrupt resolution circuit.

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

Figure 1.9.8. Interrupt priority that is set in hardware

Interrupts

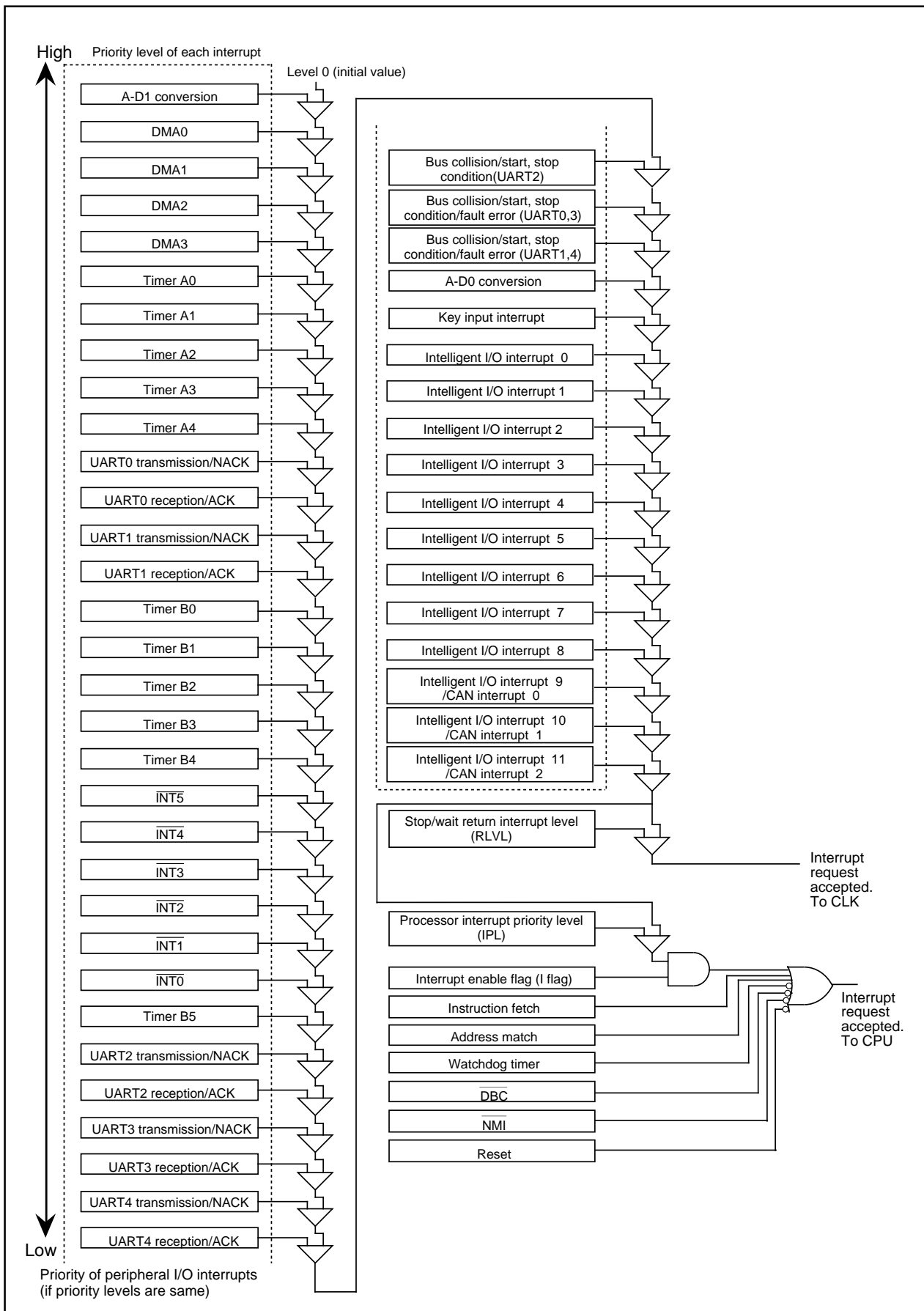


Figure 1.9.9. Interrupt resolution circuit

Interrupts

INT Interrupts

INT0 to 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 031F16) 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, set the INTi interrupt polarity switch bit of the interrupt request select register (address 031F16) to "0".

Figure 1.9.10 shows the interrupt request select register.

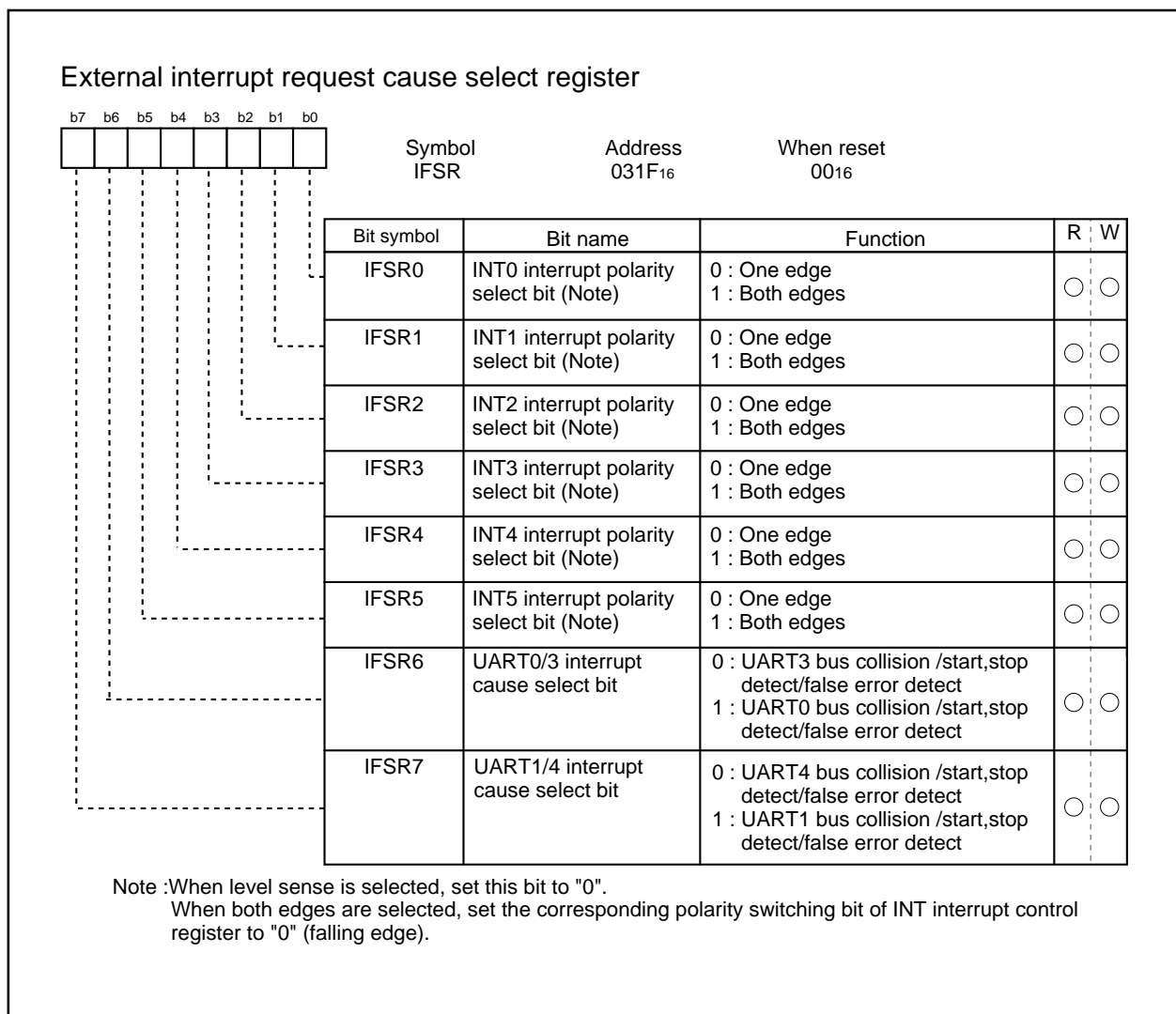


Figure 1.9.10. External interrupt request cause select register

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 P85 register (bit 5 at address 03C416).

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 VCC (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 P104 to P107 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 P104 to P107 as A-D input ports. Figure 1.9.11 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 03AF16) 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.

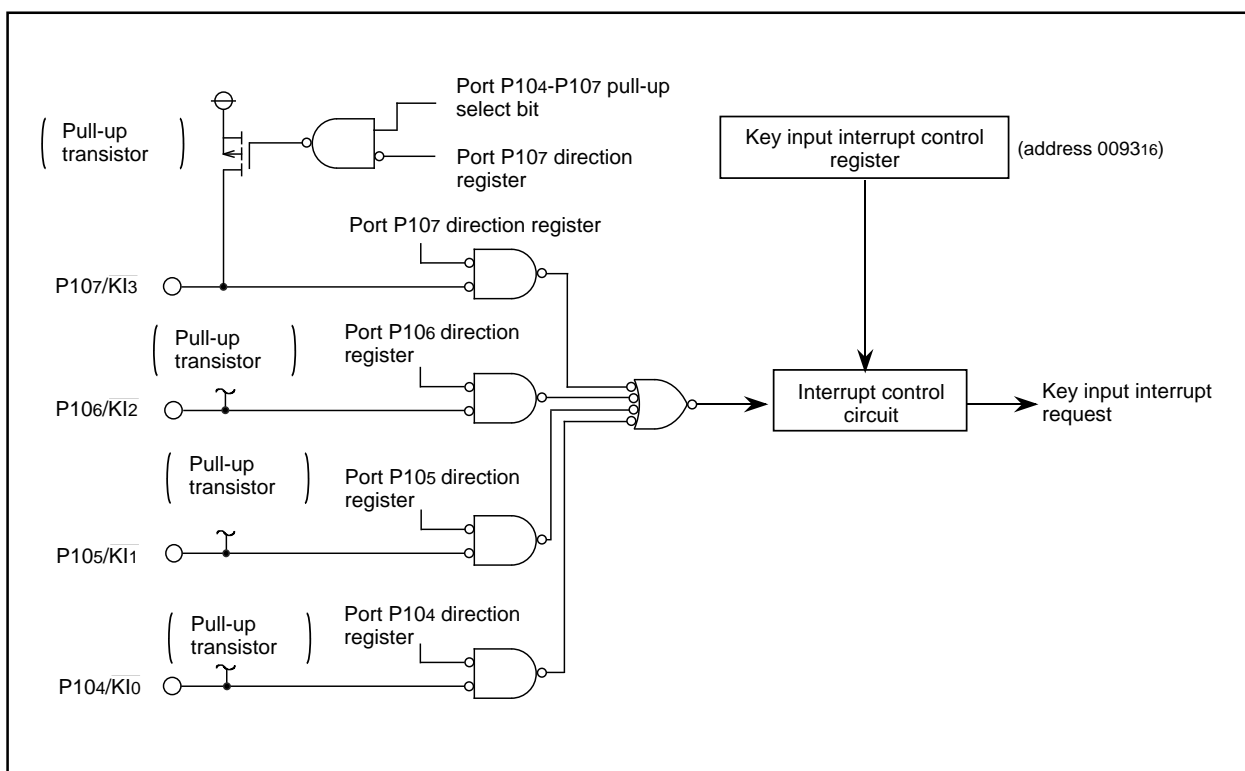


Figure 1.9.11. Block diagram of key input interrupt

Interrupts

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 1.9.12 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.

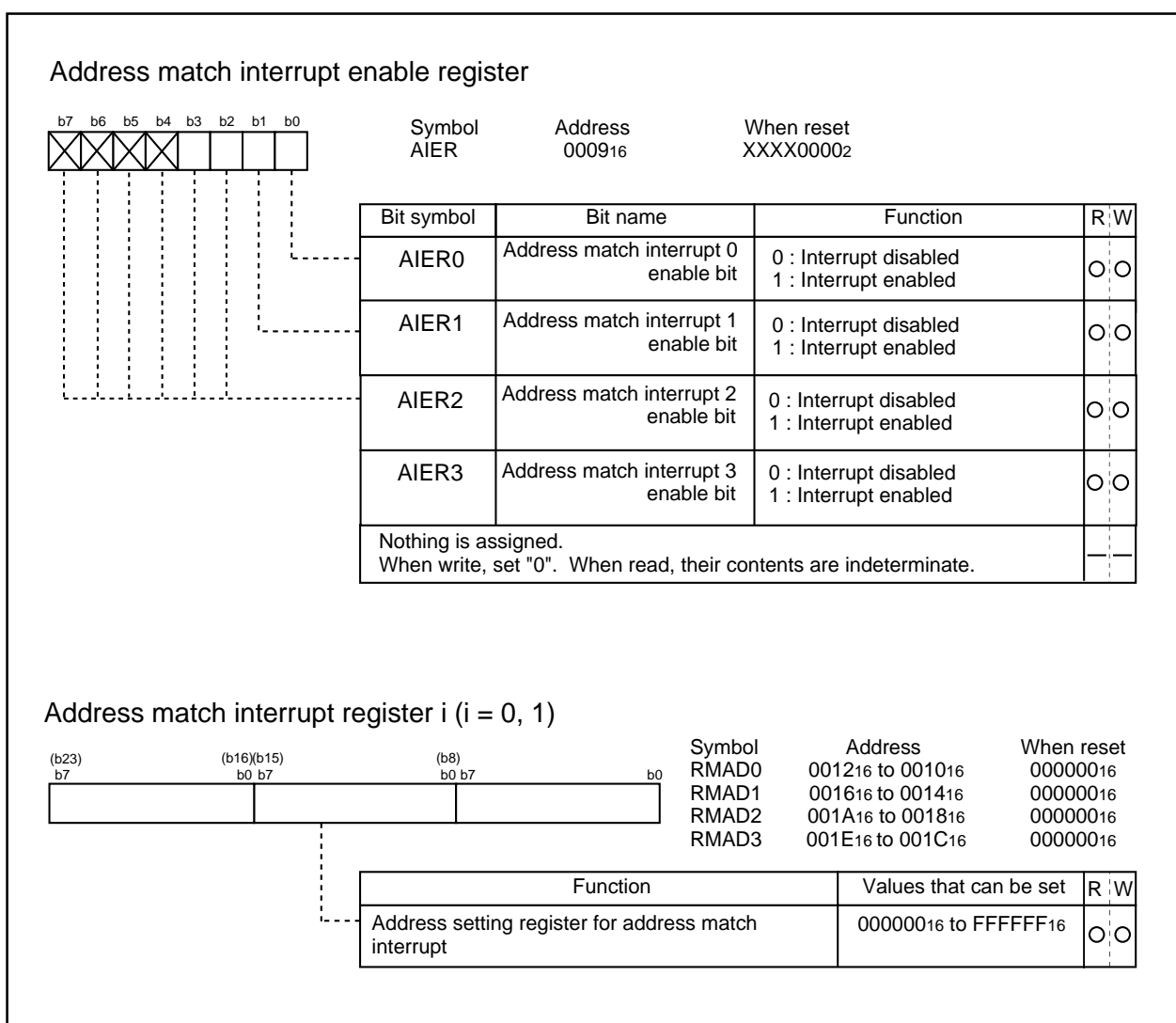


Figure 1.9.12. Address match interrupt-related registers

Intelligent I/O and CAN Interrupt

Group 0 to 3 intelligent I/O interrupts and CAN interrupt are assigned to software interrupt numbers 44 to 54 and 57.

As intelligent I/O interrupt request, there are base timer interrupt request signals, time measurement interrupt request signals, waveform generation interrupt request signals and interrupt request signals from various communication circuits.

Figure 1.9.13 shows the intelligent I/O interrupts and CAN interrupt block diagram, figure 1.9.14 shows the interrupt request register and figure 1.9.15 shows interrupt enable register.

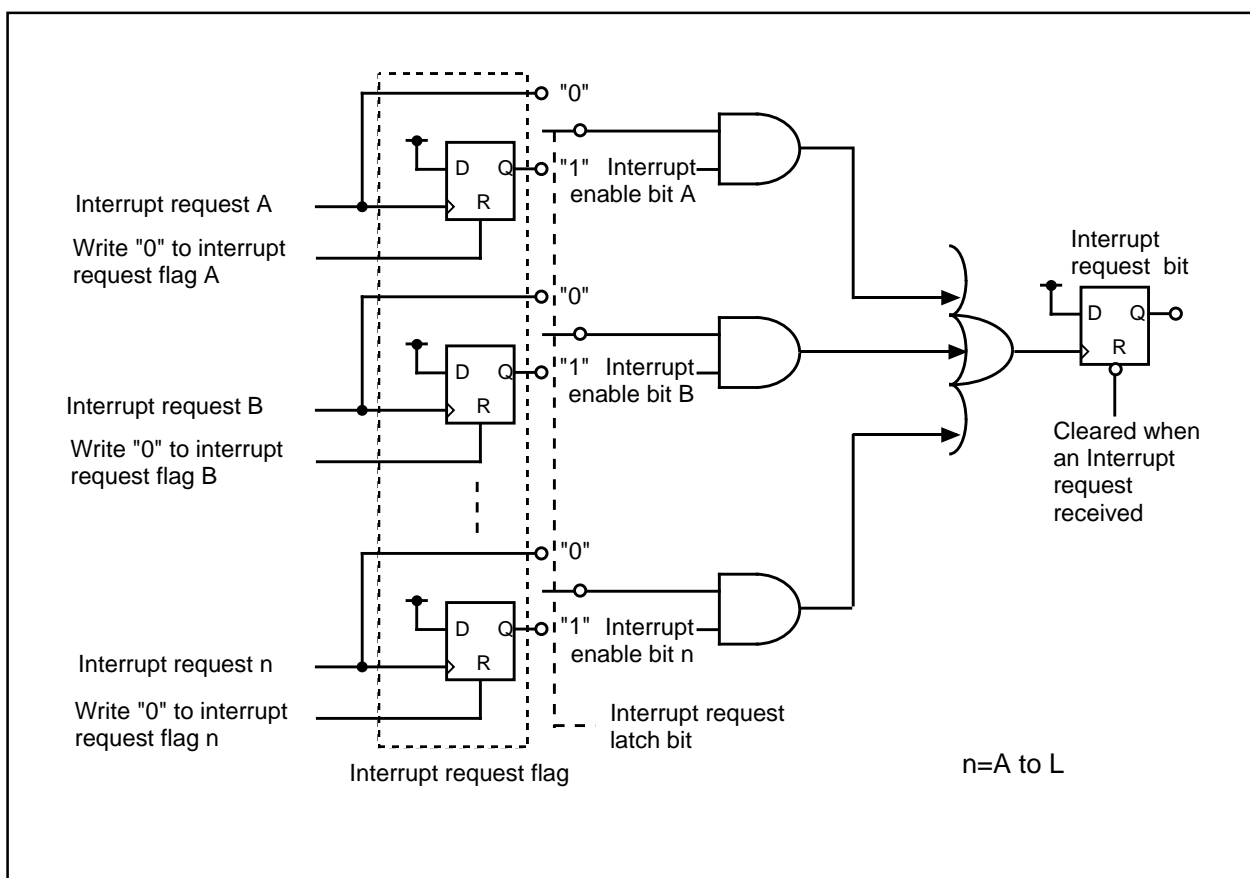


Figure 1.9.13. Intelligent I/O and CAN interrupt block diagram

When using the intelligent I/O or CAN interrupt as a starting factor for DMA II, the interrupt latch bit must be set to "0" in order to enable only the interrupt request factor used by the interrupt enable register.

Interrupts

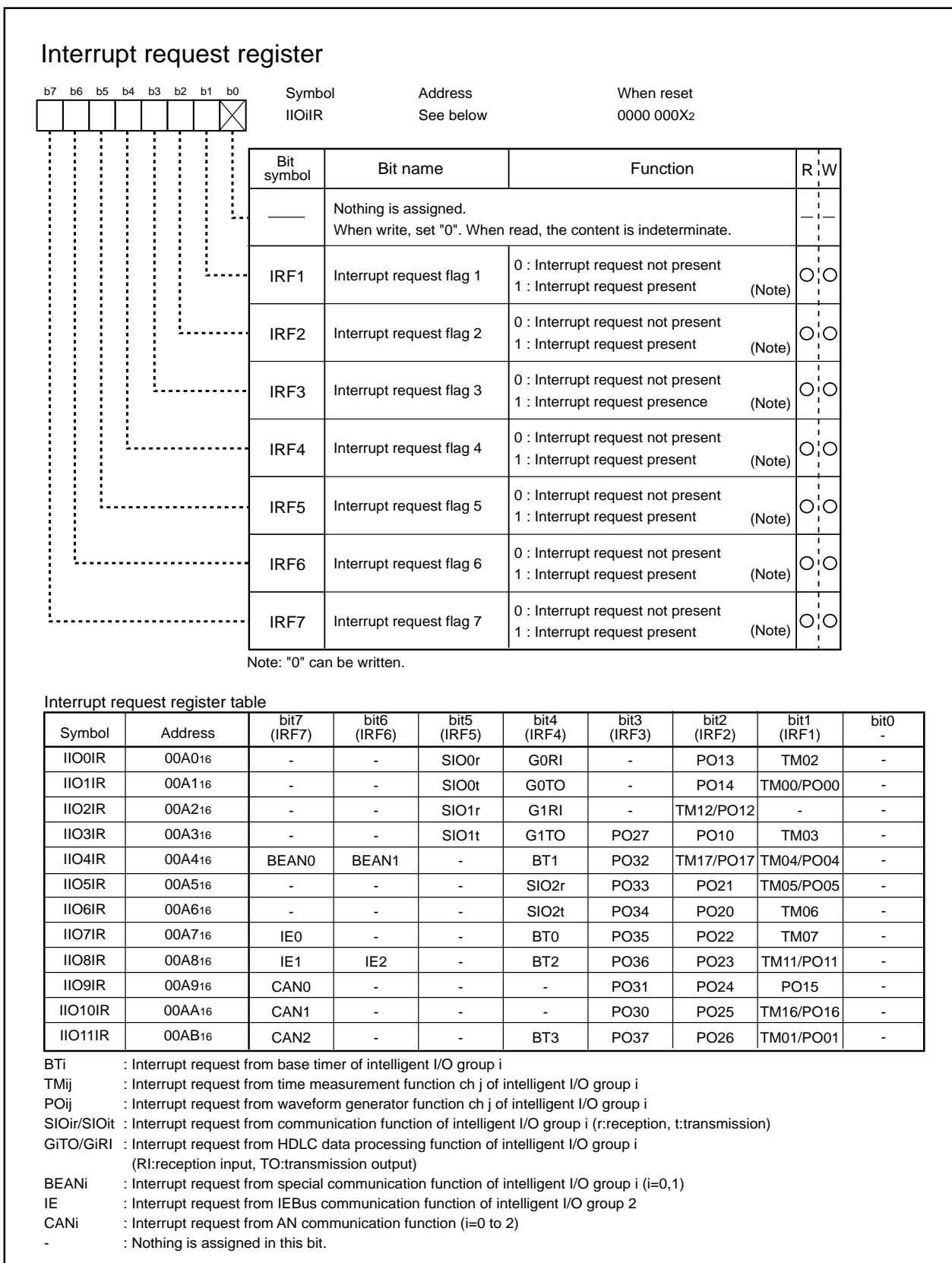


Figure 1.9.14. Interrupt request registers

Bit 1 to bit 7: Interrupt request flag (IRF1 to IRF7)

To retain respective interrupt requests and judge interrupt kind occurred in the interrupt process routine.

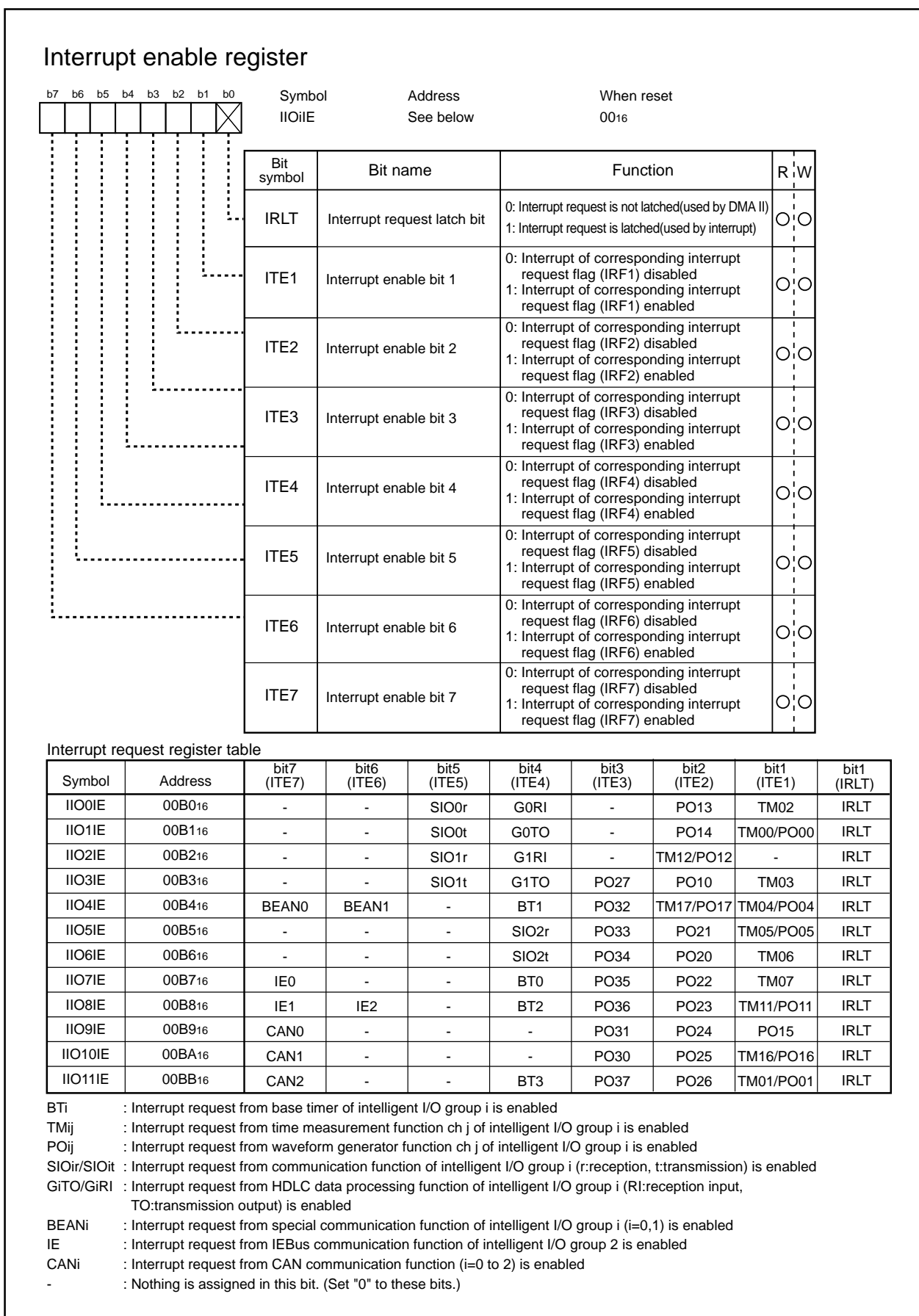


Figure 1.9.15. Interrupt enable registers

Bit 0: Interrupt request latch bit (IRLT)

An interrupt signal or latched signal of the interrupt signal is selected as an interrupt request signal. When the latched signal of an interrupt signal is used, this flag must be set to "0" after interrupt request flag is read in interrupt process routine. If this flag is not set to "0" and interrupt process is completed, although interrupt request occurs again, interrupt will not occur.

Bit 1 to bit 7: Interrupt enable bit (ITE 1 to ITE 7)

To enable/disable respective interrupts.

Precautions for Interrupts**(1) Reading addresses 000000₁₆ and 000002₁₆**

- When maskable interrupt occurs, CPU reads the interrupt information (the interrupt number and interrupt request level) in the interrupt sequence from address 000000₁₆. When a high-speed interrupt occurs, CPU reads from address 000002₁₆.

The interrupt request bit of the certain interrupt will then be set to "0".

However, reading addresses 000000₁₆ and 000002₁₆ 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 000000₁₆. 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 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 number to the stack pointer. Set an even address to the stack pointer so that operating efficiency is increased.

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

- As for the $\overline{\text{NMI}}$ interrupt pin, this interrupt cannot be disabled. Connect it to the Vcc pin via a pull-up resistor if unused.
- The $\overline{\text{NMI}}$ pin also serves as P85, 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.
- A low level signal with more than 1 clock cycle (BCLK) is necessary for $\overline{\text{NMI}}$ pin.

(4) External interrupt

- Edge sense
Either a low level or a high level for at least 250 ns is necessary for the signal input to pins INT0 to INT5 regardless of the CPU operation clock.
- Level sense
Either a low level or a high level of 1 cycle of BCLK + at least 200 ns width is necessary for the signal input to pins INT0 to INT5 regardless of the CPU operation clock. (When $X_{IN}=20\text{MHz}$ and no division mode, at least 250 ns width is necessary.)

- When the polarity of the INT₀ to INT₅ 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.

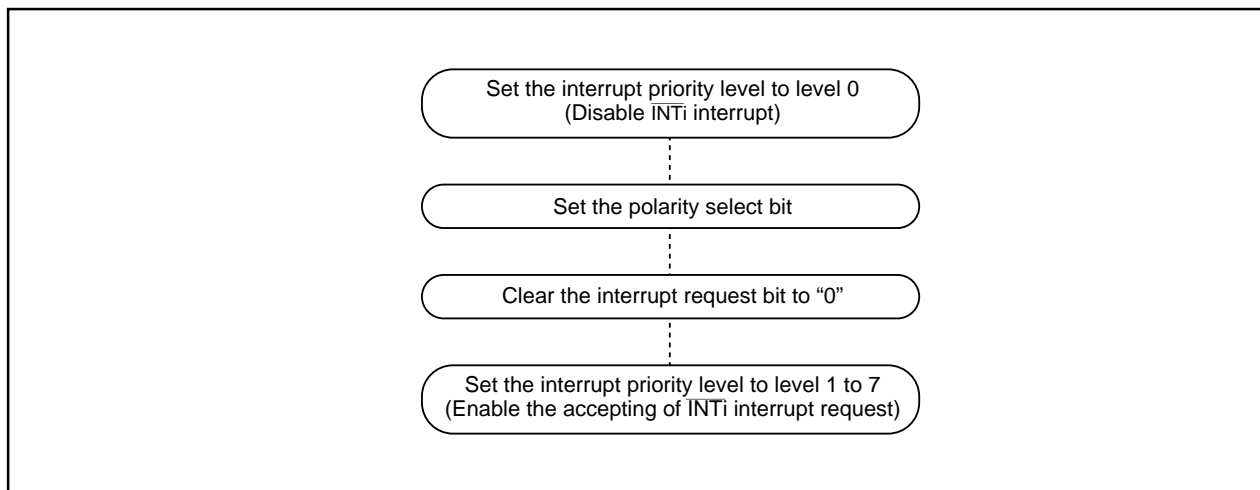


Figure 1.9.16. Switching condition of $\overline{\text{INT}}$ interrupt request

(5) Rewrite the interrupt control register

- When an 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

(6) Rewrite interrupt request register

- When writing to "0" to this register, the following instructions must be used.

Instructions : AND, BCLR

Watchdog Timer

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 (CM06) of the system control register 0 (address 000816) is "0" and reset is selected when CM06 is "1". No value other than "1" can be written in CM06. Once reset is selected (CM06="1"), watchdog timer interrupt cannot be selected by software.

When XIN is selected for the BCLK, bit 7 (WDC7) of the watchdog timer control register (address 000F16) selects the prescaler division ratio (by 16 or by 128). When XCIN is selected as the BCLK, the prescaler is set for division by 2 regardless of WDC7. Therefore, the watchdog timer cycle can be calculated as follows. However, errors can arise in the watchdog timer cycle due to the prescaler.

When XIN is selected in BCLK

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

When XCIN is selected in BCLK

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

For example, when BCLK is 20MHz and the prescaler division ratio is set to 16, the monitor timer cycle is approximately 26.2 ms, and approximately 17.5 ms when BCLK is 30MHz.

The watchdog timer is initialized by writing to the watchdog timer start register (address 000E16) 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 000E16). 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 previous value.

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 and 1.10.3 show the watchdog timer-related registers.

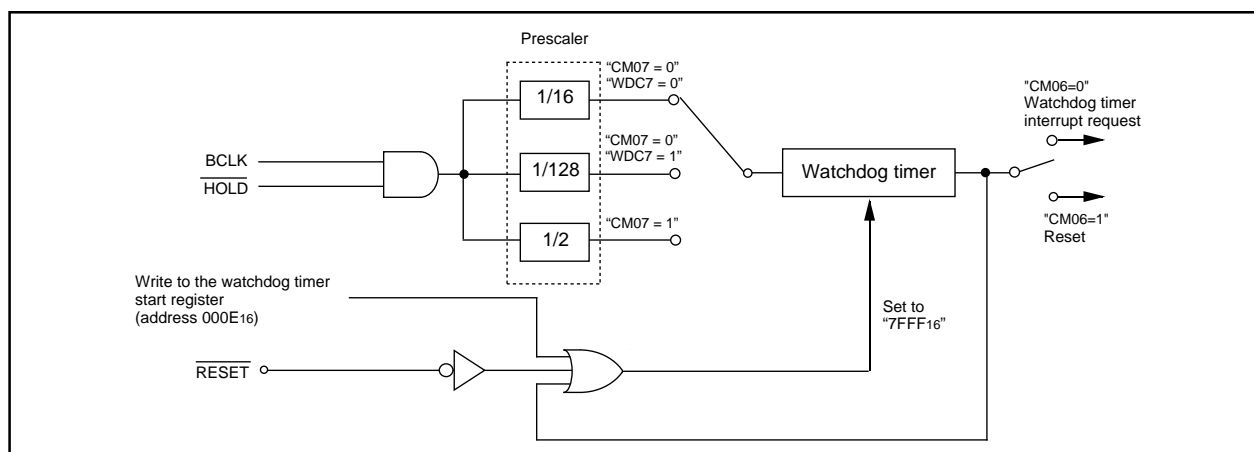


Figure 1.10.1. Block diagram of watchdog timer

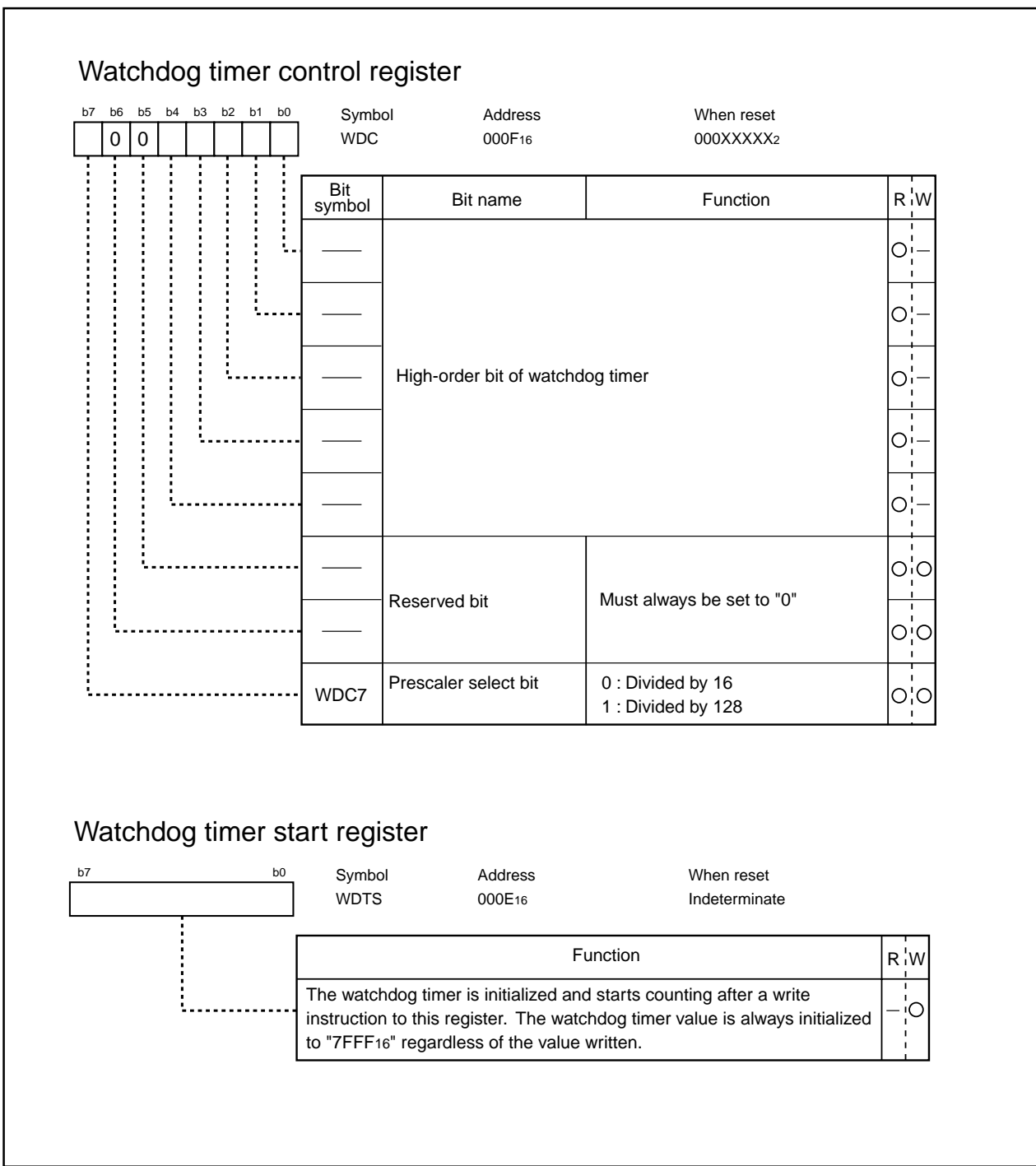


Figure 1.10.2. Watchdog timer control and start registers

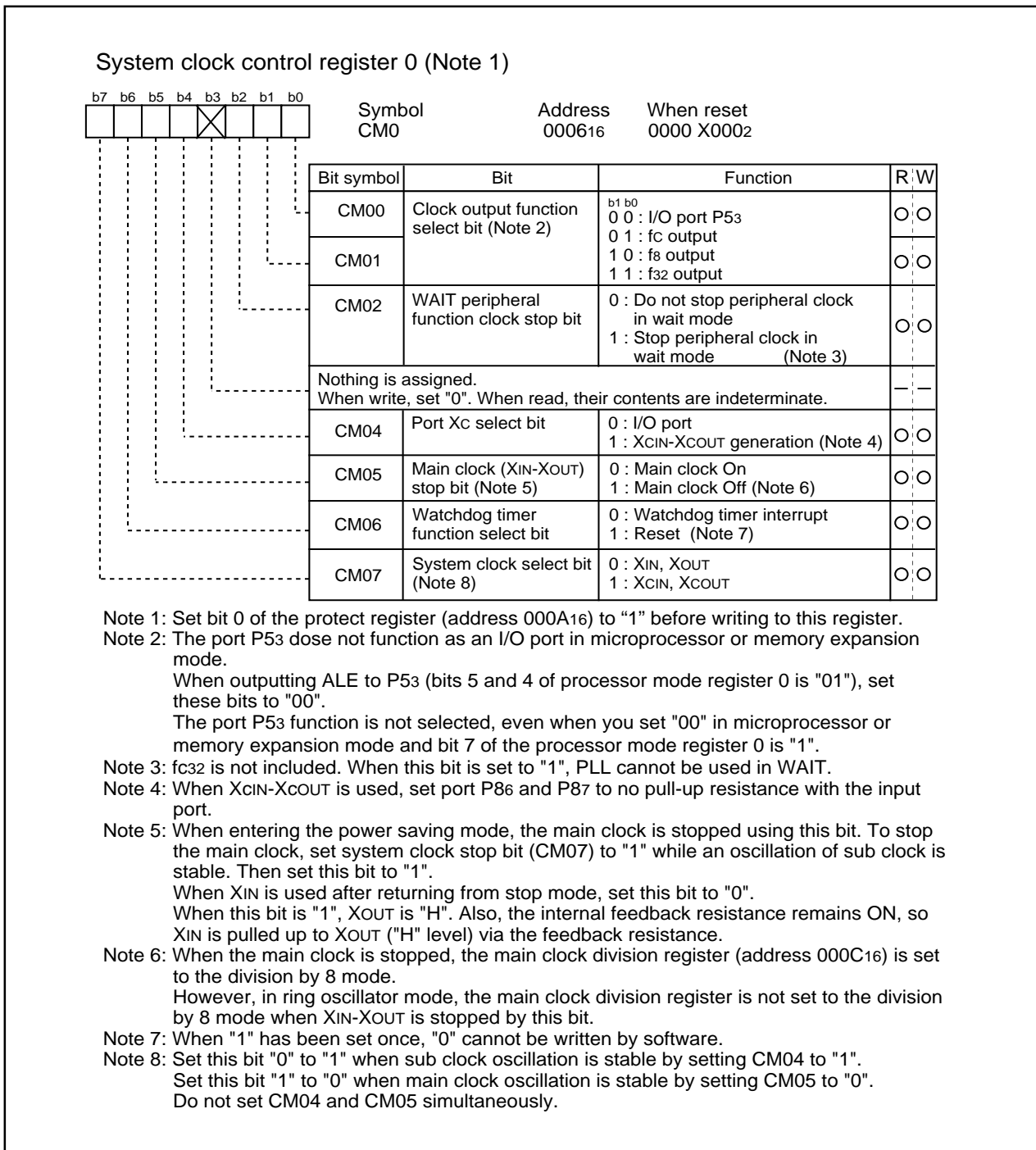


Figure 1.10.3. System clock control register 0

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 transmit delete 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 and high-speed interrupt register are used as DMAC registers. If you are using three or more DMAC channels, you cannot 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-stealing, 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 are allocated in the 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.

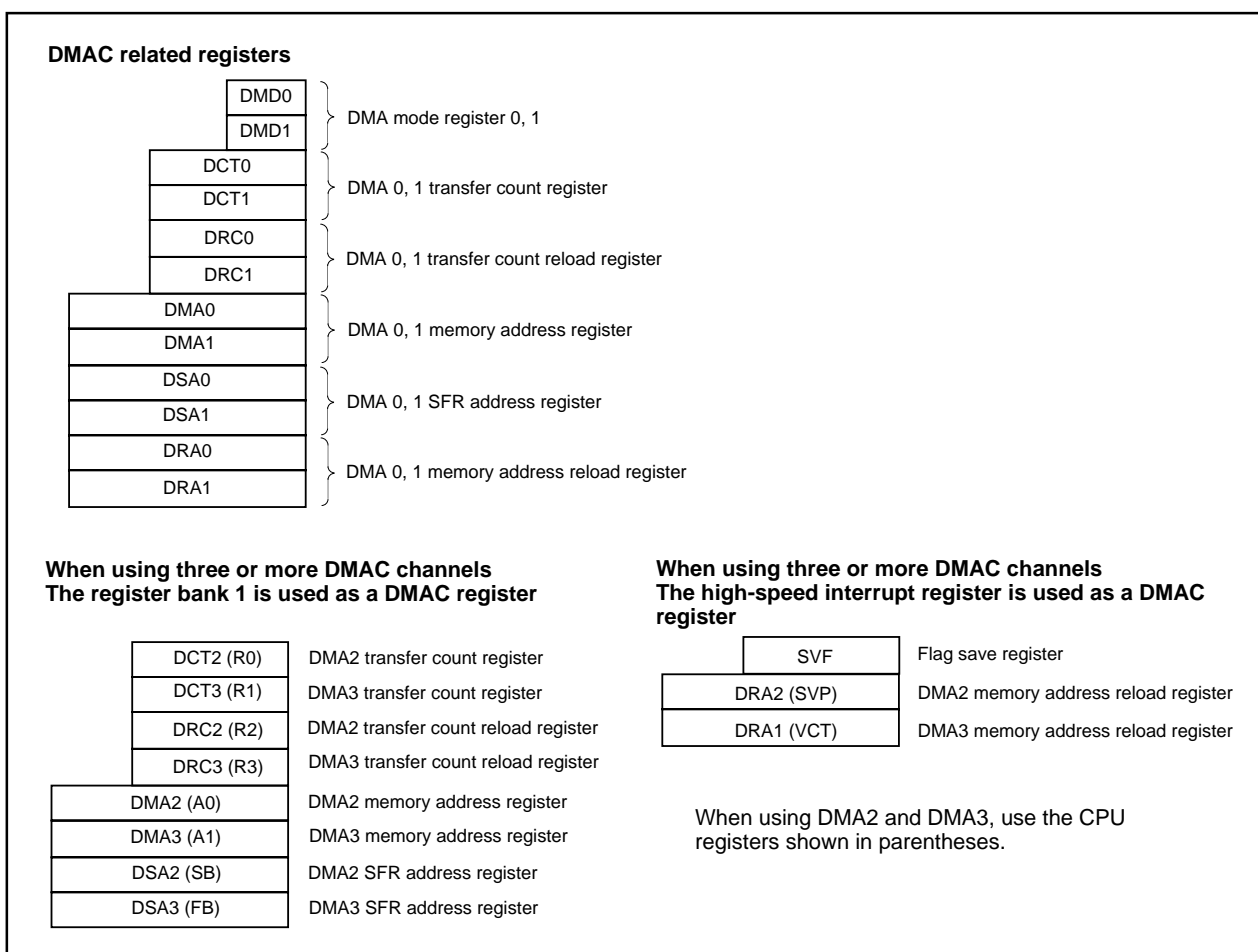


Figure 1.11.1. Register map using DMAC

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	<ul style="list-style-type: none"> 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 Mbytes 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 $\overline{INT0}$ to $\overline{INT3}$ 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 Intelligent I/O interrupt 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	<ul style="list-style-type: none"> Single transfer Transfer ends when the transfer count register is "0000₁₆". Repeat transfer When the transfer counter is "0000₁₆", 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	<ul style="list-style-type: none"> Single transfer Transfer starts when DMA transfer count register is more than "0001₁₆" and the DMA is requested after "012" is written to the channel i transfer mode select bits Repeat transfer Transfer starts when the DMA is requested after "112" is written to the channel i transfer mode select bits
DMA shutdown	<ul style="list-style-type: none"> Single transfer When "002" is written to the channel i transfer mode select bits and DMA transfer count register becomes "0000₁₆" by DMA transfer or write Repeat transfer When "002" 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 does not affect any interrupt.

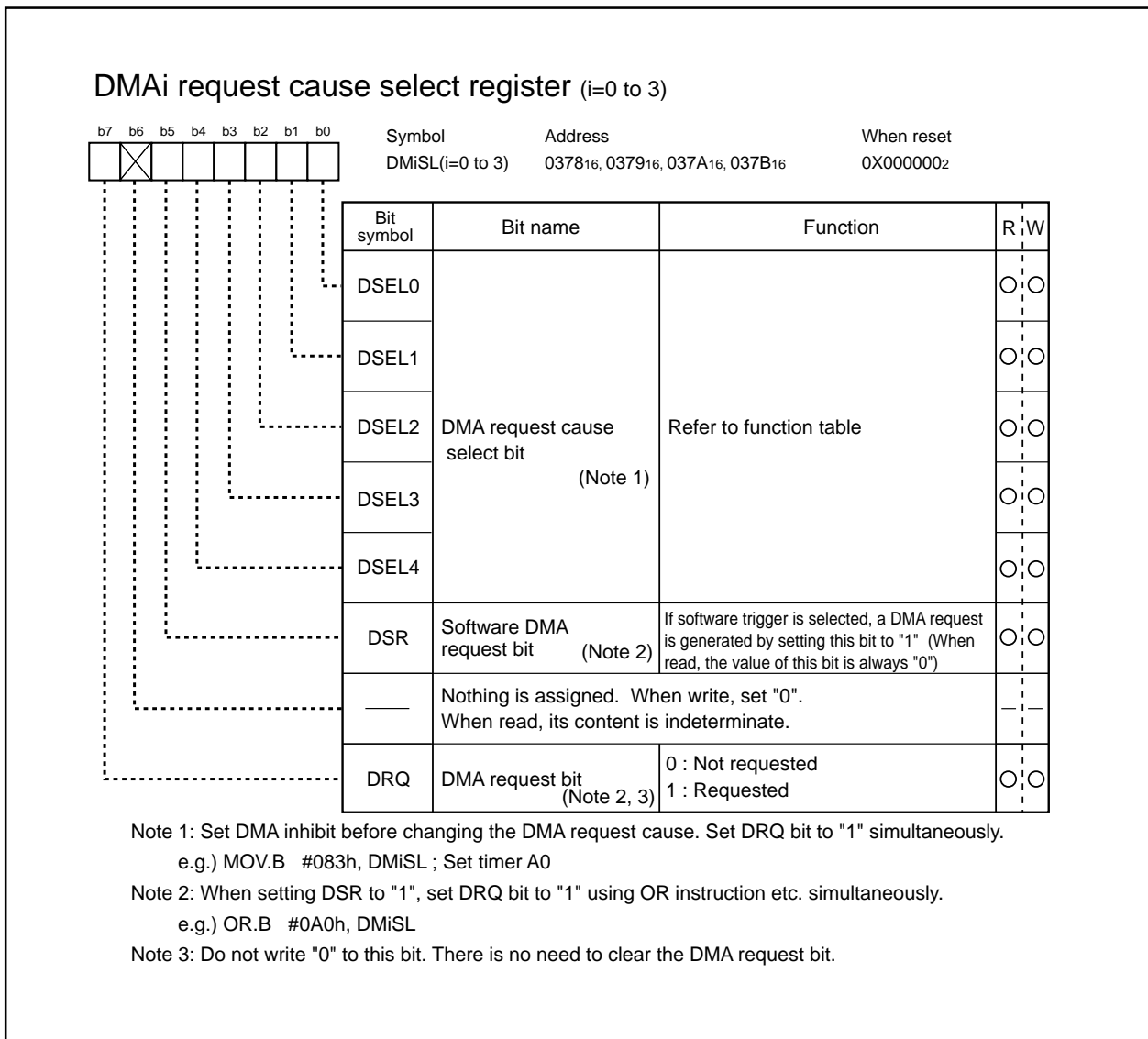


Figure 1.11.2. DMAC register (1)

Table 1.11.2. DMAi request cause select register function

Setting value	DMA request cause				
b4 b3 b2 b1 b0	DMA0	DMA1	DMA2	DMA3	
0 0 0 0 0	Software trigger				
0 0 0 0 1	Falling edge of INT0 pin	Falling edge of INT1 pin	Falling edge of INT2 pin	Falling edge of INT3 pin	(Note 1)
0 0 0 1 0	Both edges of INT0	Both edges of INT1	Both edges of INT2	Both edges of INT3	(Note 1)
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 /ACK				(Note 2)
1 0 0 0 0	UART1 transmit				
1 0 0 0 1	UART1 receive /ACK				(Note 2)
1 0 0 1 0	UART2 transmit				
1 0 0 1 1	UART2 receive /ACK				(Note 2)
1 0 1 0 0	UART3 transmit				
1 0 1 0 1	UART3 receive /ACK				(Note 2)
1 0 1 1 0	UART4 transmit				
1 0 1 1 1	UART4 receive /ACK				(Note 2)
1 1 0 0 0	A-D0	A-D1	A-D0	A-D1	
1 1 0 0 1	Intelligent I/O interrupt control register 0	Intelligent I/O interrupt control register 7	Intelligent I/O interrupt control register 2	Intelligent I/O interrupt control register 9	
1 1 0 1 0	Intelligent I/O interrupt control register 1	Intelligent I/O interrupt control register 8	Intelligent I/O interrupt control register 3	Intelligent I/O interrupt control register 10	
1 1 0 1 1	Intelligent I/O interrupt control register 2	Intelligent I/O interrupt control register 9	Intelligent I/O interrupt control register 4	Intelligent I/O interrupt control register 11	
1 1 1 0 0	Intelligent I/O interrupt control register 3	Intelligent I/O interrupt control register 10	Intelligent I/O interrupt control register 5	Intelligent I/O interrupt control register 0	
1 1 1 0 1	Intelligent I/O interrupt control register 4	Intelligent I/O interrupt control register 11	Intelligent I/O interrupt control register 6	Intelligent I/O interrupt control register 1	
1 1 1 1 0	Intelligent I/O interrupt control register 5	Intelligent I/O interrupt control register 0	Intelligent I/O interrupt control register 7	Intelligent I/O interrupt control register 2	
1 1 1 1 1	Intelligent I/O interrupt control register 6	Intelligent I/O interrupt control register 1	Intelligent I/O interrupt control register 8	Intelligent I/O interrupt control register 3	

Note 1: When $\overline{\text{INT3}}$ pin is data bus in microprocessor mode, INT3 edge cannot be used as DMA3 request cause.

Note 2: UARTi receive /ACK switched by setting of UARTi special mode register and UARTi special mode register 2 (i=0 to 3)

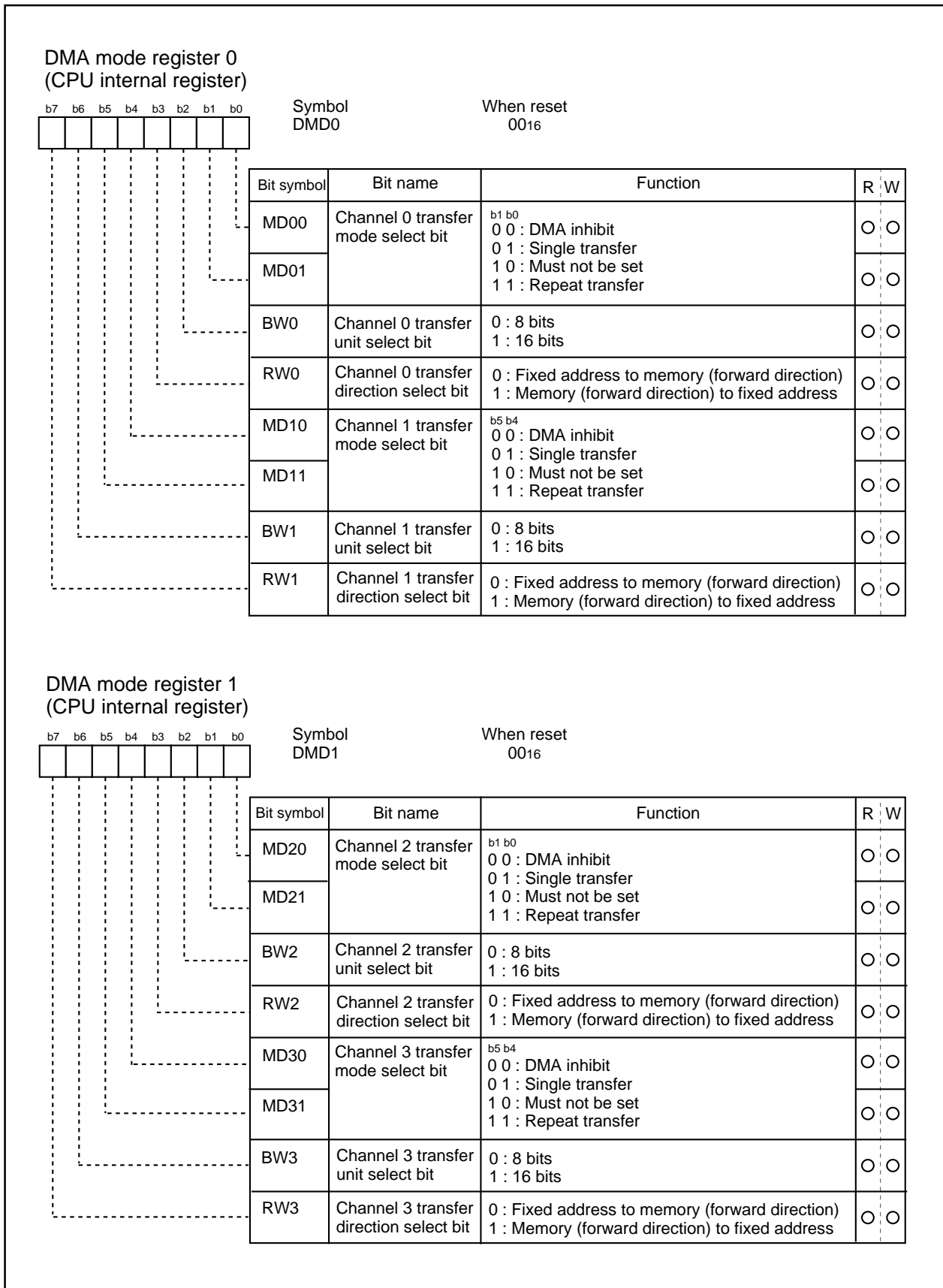


Figure 1.11.3. DMAC register (2)

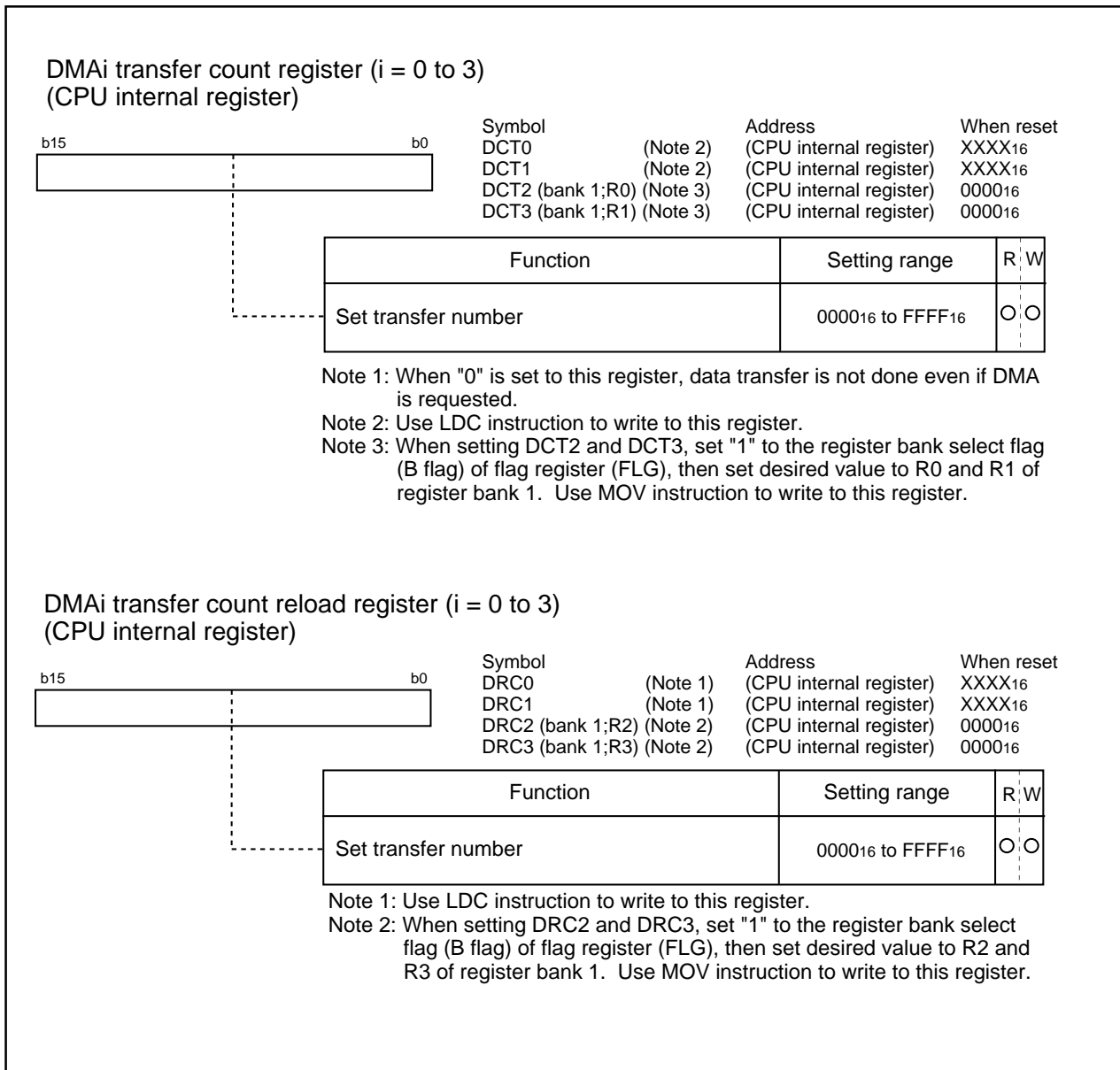


Figure 1.11.4. DMAC register (3)

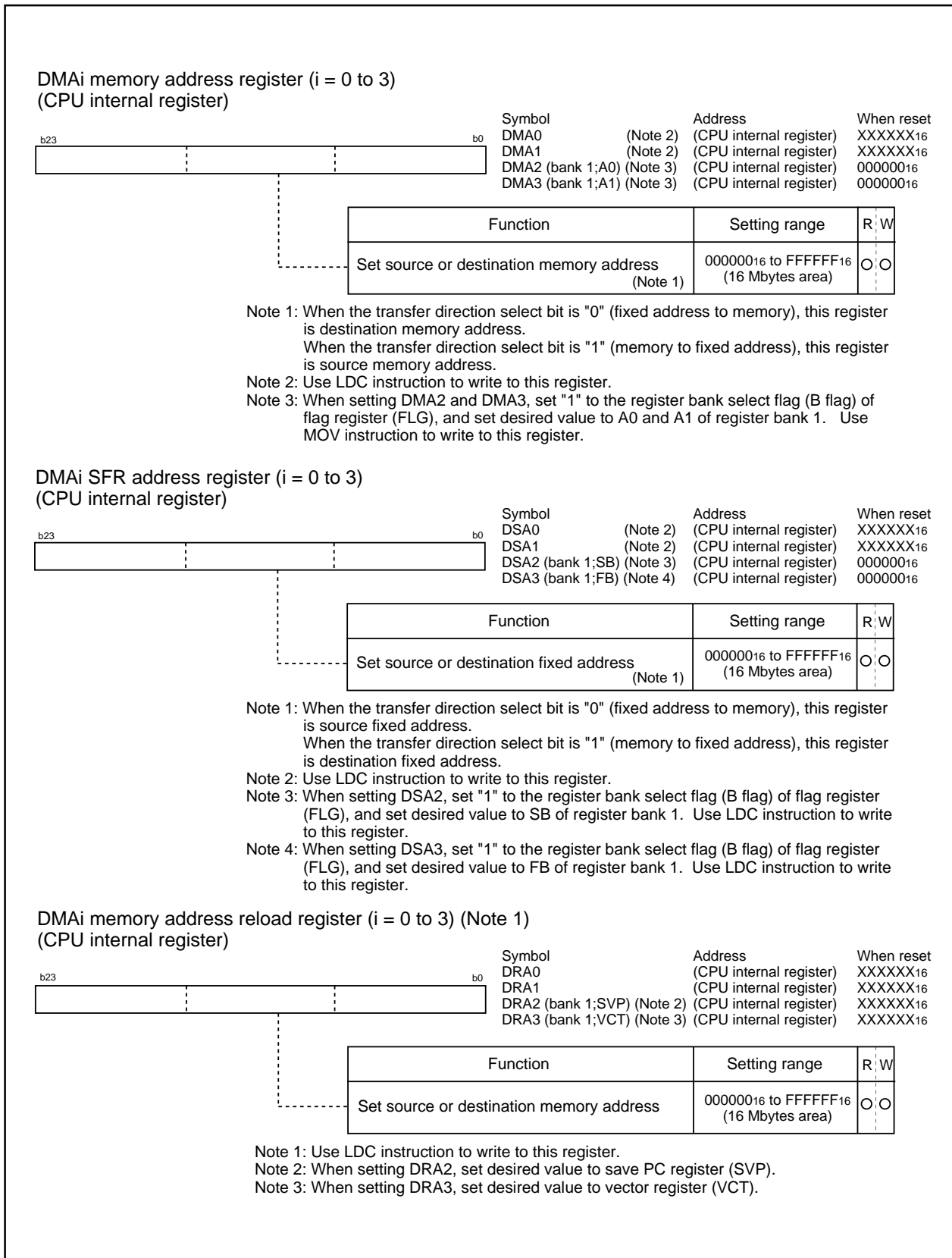


Figure 1.11.5. DMAC register (4)

(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 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 software 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.

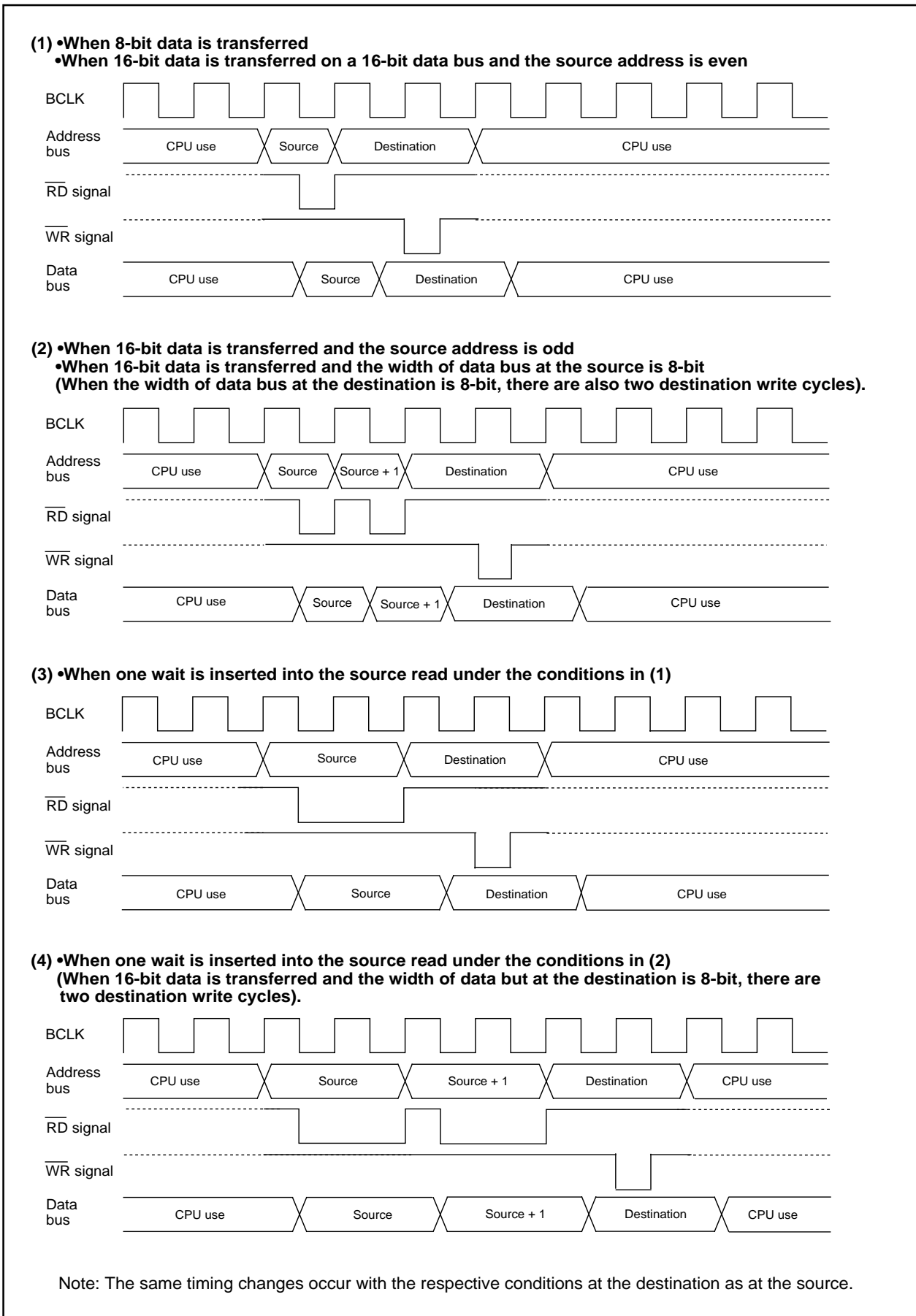


Figure 1.11.6. Example of the transfer cycles for a source read

(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:

$$\text{No. of transfer cycles per transfer unit} = \text{No. of read cycles} \times j + \text{No. of write cycles} \times k$$

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

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

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 DMA_i 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 DMA_i request bit is cleared to "0" when the DMA transfer starts or the DMA transfer is disabled (channel *i* transfer mode select bits are "00" and the DMA_i transfer count register is "0").

(2) External factors

These are DMA request factors that are generated by the input edge from the \overline{INT}_i pin (where *i* indicates the DMAC channel). When the \overline{INT}_i pin is selected by the DMA_i 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 DMA_i 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{INT}_i pins, or both edges.

When an external factor is selected, the DMA_i request bit is cleared, in the same way as the DMA_i 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 DMA_i request bits, and DMA transfer timing

When the request inputs to DMA_i occur in the same sampling cycle (between the falling edge of BCLK and the next falling edge), the DMA_i request bits are set simultaneously, but if the DMA_i 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.

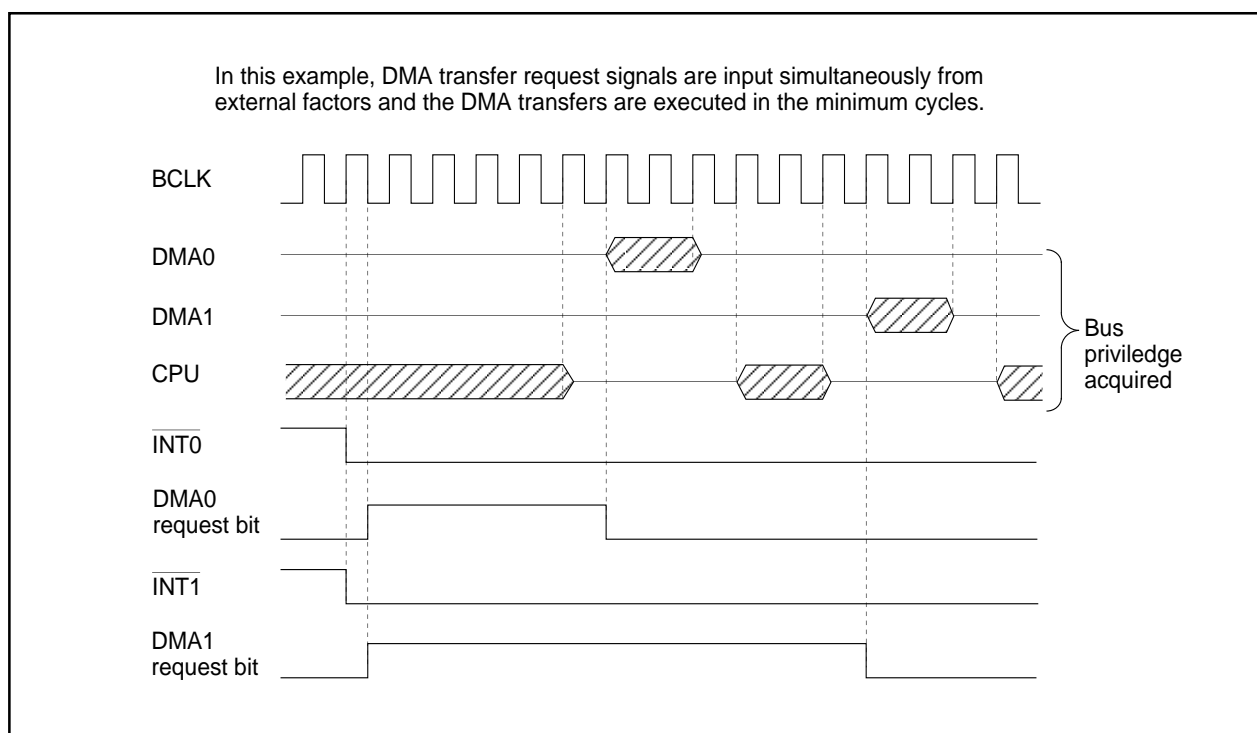


Figure 1.11.7. DMA transfer example by external factors

Precautions for DMAC

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

In M32C/83, 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, the corresponding DMA channel is set to disabled. At least $8 + 6 \times N$ (N: enabled channel number) clock cycles are needed from the instruction to write to the DMAi request cause select bit to enable DMA.

e.g.) 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)
nop
: } At least 8 + 6 x N cycles
; (N: enabled channel number)
ldc R0, DMD0 ; DMA0 enabled
pop.w R0 ; Restore R0 register

```


DMAC II

DMAC II

When requested by an interrupt from any peripheral I/O, the DMAC performs a memory-to-memory transfer, an immediate data transfer, or an arithmetic transfer (to transfer the sum of two data added).

Specifications of DMAC II are shown in Table 1.12.1.

Table 1.12.1 Specifications of DMAC II

Item	Specification
Causes to activate DMAC II	Interrupt request from any peripheral I/O whose interrupt priority is set to "level 7" by the Interrupt Control Register
Transfer data	(1) Memory -> memory (memory-to-memory transfer) (2) Immediate data -> memory (immediate data transfer) (3) Memory (or immediate data) + memory -> memory (arithmetic transfer)
Unit of transfer	Transferred in 8 or 16 bits
Transfer space	64-Kbyte space at address up to 0FFFF ₁₆ (Note)
Direction of transfer	Fixed or forward address Can be selected individually for the source and the destination of transfer.
Transfer mode	(1) Single transfer (2) Burst transfer
Chained transfer function	Parameters (transfer count, transfer address, and other information) are switched over when the transfer counter reaches zero.
Interrupt at end of transfer	Interrupt is generated when the transfer counter reaches zero.
Multiple transfer function	Multiple data transfers can be performed by one DMA II transfer request generated.

Note : When transfer unit is 16 bits and destination address is 0FFFF₁₆, data is transferred to addresses 0FFFF₁₆ and 10000₁₆. When source address is 0FFFF₁₆, data is transferred as in the previous.

Settings of DMAC II

DMAC II can be enabled for use by setting up the following registers and tables.

- Exit Priority Register (address 009F₁₆)
- DMAC II Index
- Interrupt Control Register for the peripheral I/O that requests a transfer by DMAC II
- Relocatable Vector Table for the peripheral I/O that requests a transfer by DMAC II
- When using an intelligent I/O or CAN interrupt, Interrupt Enable Register's interrupt request latch bit (bit 0)

(1) Exit priority register (address 009F₁₆)

If this register's DMAC II select bit (bit 5) and fast interrupt select bit (bit 3) respectively are set to 1 and 0, DMAC II is activated by an interrupt request from any peripheral I/O whose interrupt priority is set to "level 7" by the interrupt priority level select bit.

The configuration of the exit priority register is shown in Figure 1.12.1.

DMAC II

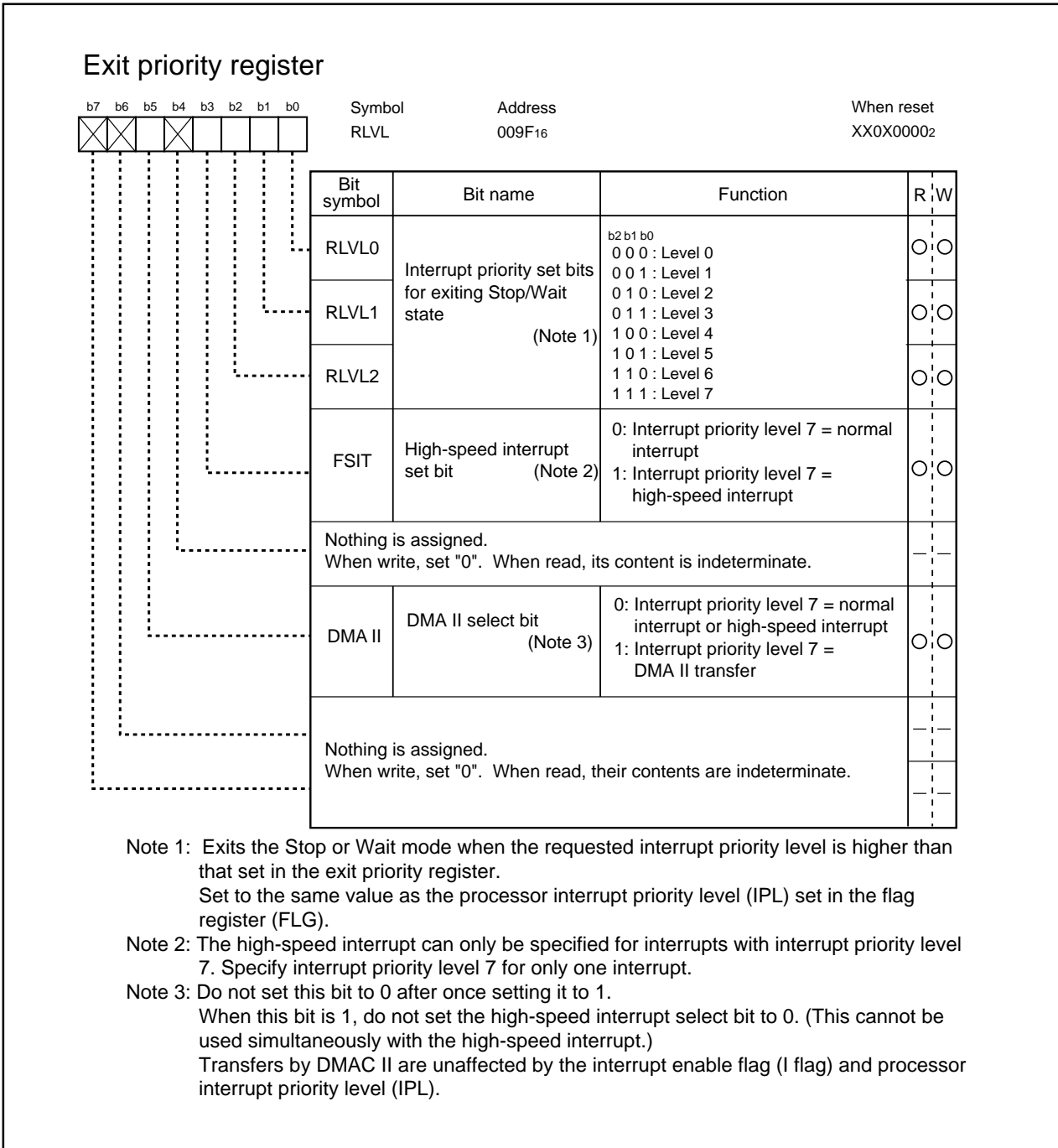


Figure 1.12.1. Exit priority register

DMAC II

(2) DMAC II Index

The DMAC II Index is a data table, comprised of 8 to 18 bytes (max. 32 kbytes when multiple transfer function is selected), which contains such parameters as transfer mode, transfer counter, transfer source address (or immediate data), operation address, transfer destination address, chained transfer address, and end-of-transfer interrupt address.

This DMAC II Index is located in the RAM area.

Configuration of the DMAC II Index is shown in Figure 1.12.2. The configuration of the DMAC II Index by transfer mode is shown in Table 1.12.2.

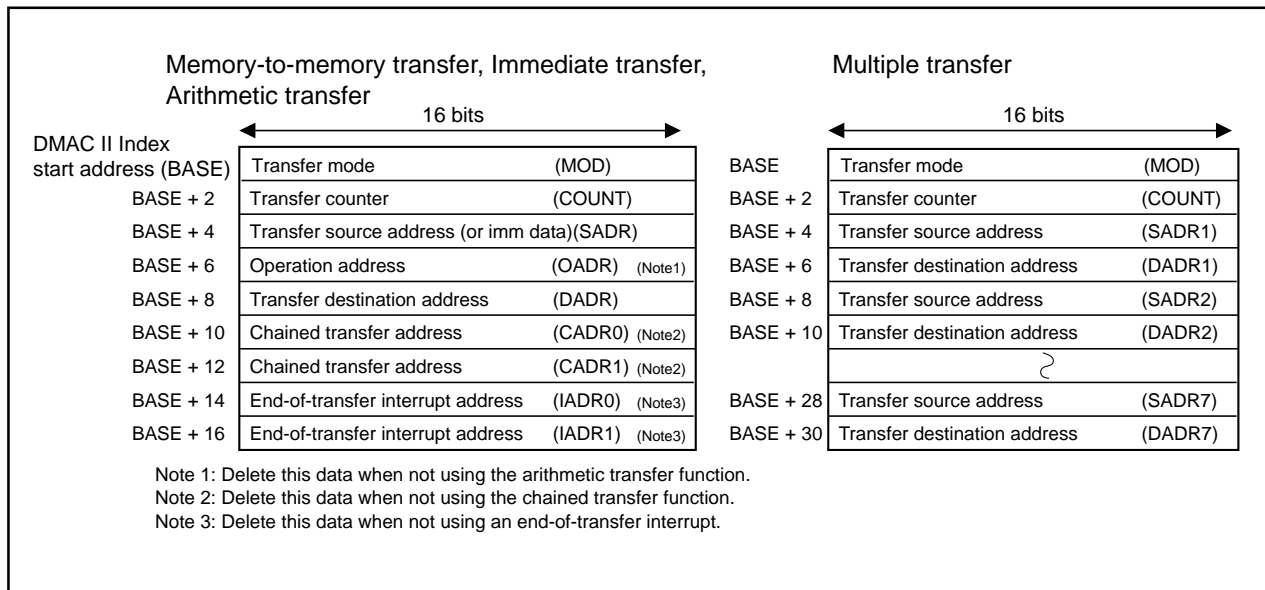


Figure 1.12.2. DMAC II index

- Transfer mode (MOD)

This two-byte data sets DMAC II transfer mode. Configuration of transfer modes is shown in Figure 1.12.3.

- Transfer counter (COUNT)

This two-byte data sets the number of times transfer is performed.

- Transfer source address (SADR)

This two-byte data sets the memory address from which data is transferred or immediate data.

- Operation address (OADR)

This two-byte data sets the memory address to be operated on for calculation. This data is added to the table only when using the arithmetic transfer function.

- Transfer destination address (DADR)

This two-byte data sets the memory address to which data is transferred.

- Chained transfer address (CADR)

This four-byte data sets the DMAC II Index start address for the next DMAC II transfer to be performed. This data is added to the table only when using the chained transfer function.

- End-of-transfer interrupt address (IADR)

This four-byte data sets the jump address for end-of-transfer interrupt processing. This data is added to the table only when using an end-of-transfer interrupt.

DMAC II

Table 1.12.2. The configuration of the DMAC II Index by transfer mode

Transmit data	Memory-to-memory transfer /immediate data transfer				Arithmetic transfer				Multiple transfer																																																			
Chained transfer	Not use	Use	Not use	Use	Not use	Use	Not use	Use	Cannot use																																																			
Interrupt at end of transfer	Not use	Not use	Use	Use	Not use	Not use	Use	Use	Cannot use																																																			
DMAC II index	<table border="1"> <tr><td>MOD</td></tr> <tr><td>COUNT</td></tr> <tr><td>SADR</td></tr> <tr><td>DADR</td></tr> </table> <p>8 bytes</p>	MOD	COUNT	SADR	DADR	<table border="1"> <tr><td>MOD</td></tr> <tr><td>COUNT</td></tr> <tr><td>SADR</td></tr> <tr><td>DADR</td></tr> <tr><td>CADR0</td></tr> <tr><td>CADR1</td></tr> </table> <p>12 bytes</p>	MOD	COUNT	SADR	DADR	CADR0	CADR1	<table border="1"> <tr><td>MOD</td></tr> <tr><td>COUNT</td></tr> <tr><td>SADR</td></tr> <tr><td>DADR</td></tr> <tr><td>IADR0</td></tr> <tr><td>IADR1</td></tr> </table> <p>12 bytes</p>	MOD	COUNT	SADR	DADR	IADR0	IADR1	<table border="1"> <tr><td>MOD</td></tr> <tr><td>COUNT</td></tr> <tr><td>SADR</td></tr> <tr><td>DADR</td></tr> <tr><td>CADR0</td></tr> <tr><td>CADR1</td></tr> <tr><td>IADR0</td></tr> <tr><td>IADR1</td></tr> </table> <p>16 bytes</p>	MOD	COUNT	SADR	DADR	CADR0	CADR1	IADR0	IADR1	<table border="1"> <tr><td>MOD</td></tr> <tr><td>COUNT</td></tr> <tr><td>SADR</td></tr> <tr><td>OADR</td></tr> <tr><td>DADR</td></tr> </table> <p>10 bytes</p>	MOD	COUNT	SADR	OADR	DADR	<table border="1"> <tr><td>MOD</td></tr> <tr><td>COUNT</td></tr> <tr><td>SADR</td></tr> <tr><td>OADR</td></tr> <tr><td>DADR</td></tr> <tr><td>CADR0</td></tr> <tr><td>CADR1</td></tr> </table> <p>14 bytes</p>	MOD	COUNT	SADR	OADR	DADR	CADR0	CADR1	<table border="1"> <tr><td>MOD</td></tr> <tr><td>COUNT</td></tr> <tr><td>SADR</td></tr> <tr><td>OADR</td></tr> <tr><td>DADR</td></tr> <tr><td>IADR0</td></tr> <tr><td>IADR1</td></tr> </table> <p>14 bytes</p>	MOD	COUNT	SADR	OADR	DADR	IADR0	IADR1	<table border="1"> <tr><td>MOD</td></tr> <tr><td>COUNT</td></tr> <tr><td>SADR1</td></tr> <tr><td>DADR1</td></tr> <tr><td> </td></tr> <tr><td> </td></tr> <tr><td> </td></tr> <tr><td>SADRi</td></tr> <tr><td>DADRi</td></tr> </table> <p>i=1 to 7 Max. 32 bytes (when i=7)</p>	MOD	COUNT	SADR1	DADR1				SADRi	DADRi
	MOD																																																											
COUNT																																																												
SADR																																																												
DADR																																																												
MOD																																																												
COUNT																																																												
SADR																																																												
DADR																																																												
CADR0																																																												
CADR1																																																												
MOD																																																												
COUNT																																																												
SADR																																																												
DADR																																																												
IADR0																																																												
IADR1																																																												
MOD																																																												
COUNT																																																												
SADR																																																												
DADR																																																												
CADR0																																																												
CADR1																																																												
IADR0																																																												
IADR1																																																												
MOD																																																												
COUNT																																																												
SADR																																																												
OADR																																																												
DADR																																																												
MOD																																																												
COUNT																																																												
SADR																																																												
OADR																																																												
DADR																																																												
CADR0																																																												
CADR1																																																												
MOD																																																												
COUNT																																																												
SADR																																																												
OADR																																																												
DADR																																																												
IADR0																																																												
IADR1																																																												
MOD																																																												
COUNT																																																												
SADR1																																																												
DADR1																																																												
SADRi																																																												
DADRi																																																												

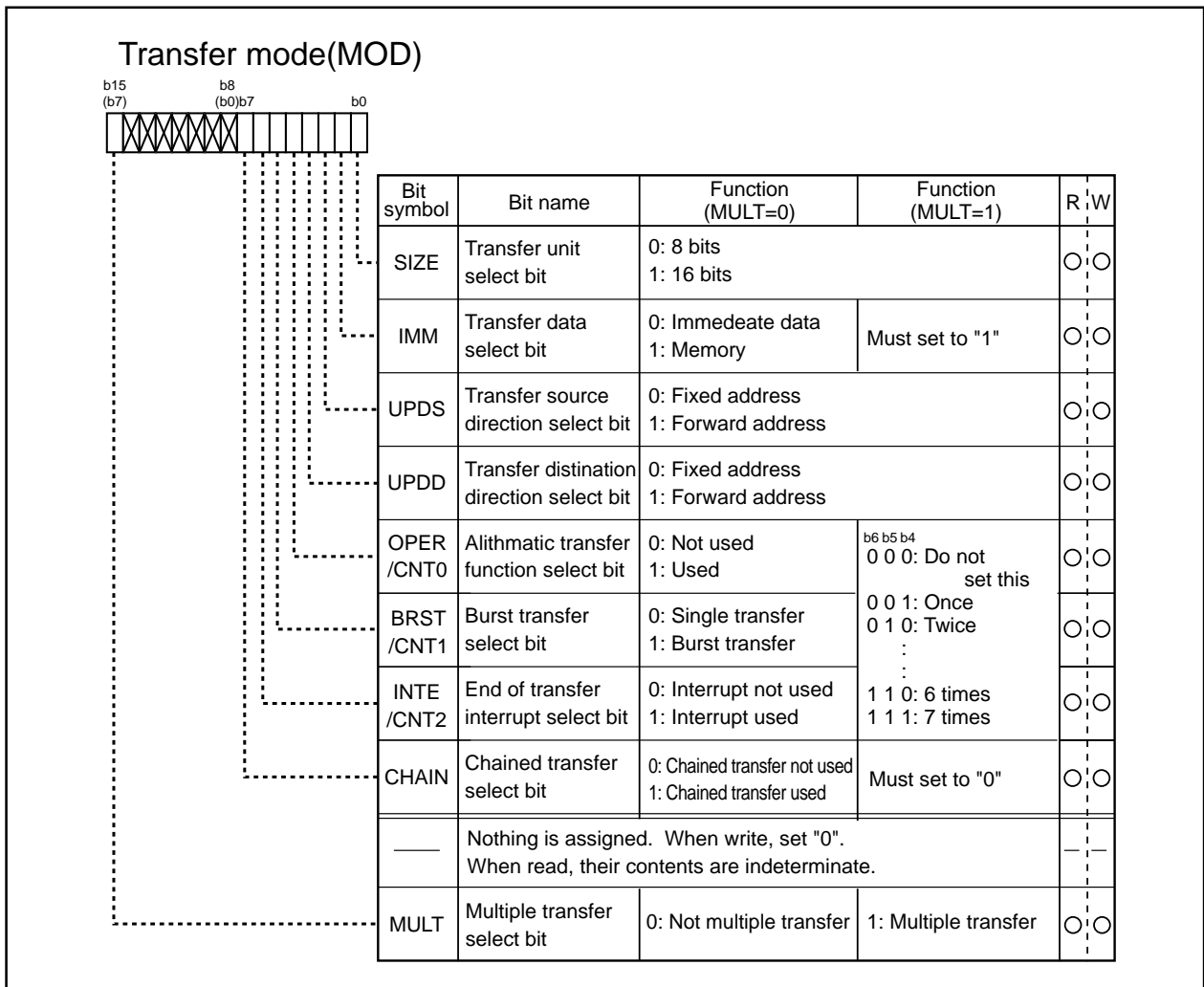


Figure 1.12.3. Transfer mode

DMAC II

(3) Interrupt Control Register for Peripheral I/O

For peripheral I/O interrupts used to request a transfer by DMAC II, set the Interrupt Control Register for each peripheral I/O to select "level 7" for their interrupt priority.

(4) Relocatable Vector Table for Peripheral I/O

In the relocatable vector table for each peripheral I/O that requests a transfer by DMAC II, set the DMAC II Index start address. (When using chained transfers, the relocatable vector table must be located in the RAM.)

(5) Interrupt Enable Register's interrupt request latch bit (bit 0)

When using an intelligent I/O or CAN interrupt to activate DMAC II, set to 0 the Interrupt Enable Register's interrupt request latch bit (bit 0) for the intelligent I/O or CAN interrupt that requests a transfer by DMAC II.

Operation of DMAC II

The DMAC II function is selected by setting the DMAC II select bit (bit 5 at address 009F₁₆) to 1. All peripheral I/O interrupt requests which have had their interrupt priorities set to "level 7" by the Interrupt Control Register comprise DMAC II interrupt requests. These interrupt requests (priority level = 7) do not generate an interrupt, however.

When an interrupt request is generated by any peripheral I/O whose interrupt priority is set to "level 7," DMAC II is activated no matter which state the I flag and processor interrupt priority level (IPL) is in. If an interrupt request with higher priority than that (e.g., $\overline{\text{NMI}}$ or watchdog timer) occurs, this higher priority interrupt has precedence over and is accepted before DMAC II transfers. The pending DMAC II transfer is started after the interrupt processing sequence for that interrupt finishes.

Transfer data

DMAC II transfers data in units of 8 or 16 bits as described below.

- Memory-to-memory transfer: Data is transferred from any memory location in the 64-Kbyte space to any memory location in the same space.
- Immediate data transfer: Data is transferred as immediate data to any memory location in the 64-Kbyte space.
- Arithmetic transfer: Two 8 or 16 bits of data are added together and the result is transferred to any memory location in the 64-Kbyte space.

When transfer unit is 16 bits and destination address is 0FFFF₁₆, data is transferred to addresses 0FFFF₁₆ and 10000₁₆. When source address is 0FFFF₁₆, data is transferred as in the previous.

DMAC II

(1) Memory-to-memory transfer

Data can be transferred from any memory location in the 64-Kbyte space to any memory location in the same space in one of the following four ways:

- Transfer from a fixed address to another fixed address
- Transfer from a fixed address to a variable address
- Transfer from a variable address to a fixed address
- Transfer from a variable address to another variable address

If variable address mode is selected, the transfer address is incremented for the next DMA II transfer to be performed. When transferred in units of 8 bits, the transfer address is incremented by one; when transferred in units of 16 bits, the transfer address is incremented by two. If the transfer source or destination address exceeds $0FFFF_{16}$ as a result of address incrementation, the transfer source or destination address recycles back to 00000_{16} .

(2) Immediate data transfer

Data is transferred as immediate data to any memory location in the 64-Kbyte space. A fixed or variable address can be selected for the transfer destination address. Store the immediate data in the DMAC II Index's transfer source address. When transferring 8-bit immediate data, set the data in the lower byte position of the transfer source address. (The upper byte is ignored.)

(3) Arithmetic transfer

Data in two memory locations of the 64-Kbyte space or immediate data and data in any memory location of the 64-Kbyte space are added together and the result is transferred to any memory location in the 64-Kbyte space. Set the memory location to be operated on or immediate data in the DMAC II Index's transfer source address field and the other memory location to be operated on in the DMAC II Index's operation address field. When performing this mode of transfer on two memory locations, a fixed or variable address can be selected for the transfer source and transfer destination addresses. If the transfer source address is chosen to be variable, the operation address also becomes variable. When performing this mode of transfer on immediate data and any memory location, a fixed or variable address can be selected for the transfer destination address.

Transfer modes

DMAC II supports single and burst transfers. Use the burst transfer select bit (bit 5) for transfer mode setup in the DMAC II index to choose single or burst transfer mode. Use the DMAC II index transfer counter to set the number of times a transfer is performed. Neither single transfer nor burst transfer is performed if the value " 0000_{16} " is set in the transfer counter.

(1) Single transfer

For a DMAC II transfer request, 8 or 16 bits of data (one transfer unit) is transferred once. If the transfer source or transfer destination address is chosen to be variable, the next DMA II transfer is performed on an incremented memory address.

The transfer counter is decremented by each DMA II transfer performed. When using the end-of-transfer interrupt facility, an end-of-transfer interrupt is generated at the time the transfer counter reaches zero.

DMAC II

(2) Burst transfer

For a DMAC II transfer request, data transfers are performed in succession a number of times as set by the DMAC II Index transfer counter. When using the end-of-transfer interrupt facility, an end-of-transfer interrupt is generated at the time a burst transfer finishes (i.e., when the transfer counter reaches zero after being decremented for each data transfer performed).

(3) Multiple transfers

For multiple transfers, use the multiple transfer select bit (bit 15) for transfer mode setup in the DMAC II Index. Setting this bit to 1 selects the multiple transfer function. For the multiple transfer function, memory to memory transfer can be performed.

Multiple transfers are performed for one DMAC II transfer request received. Use DMAC II Index transfer mode bits 4–6 to set the number of transfers to be performed. (Setting these bits to 001 performs one transfer; setting these bits to 111 performs 7 transfers. Setting these bits to 000 is inhibited.)

The transfer source and transfer destination addresses are alternately incremented beginning with the DMAC II Index BASE address + 4 (as many times as the number of transfers performed).

When using multiple transfer function, arithmetic transfer, burst transfer, end-of-transfer interrupt and chained transfer cannot be used.

(4) Chained transfer

For chained transfers, use the chained transfer select bit (bit 7) for transfer mode setup in the DMAC II Index. Setting this bit to 1 selects the chained transfer function. The following describes how a chained transfer is performed.

- 1) When a DMA II transfer request (interrupt request from any peripheral I/O) is received, a DMAC II Index transfer is performed corresponding to the received request.
- 2) When the DMAC II Index transfer counter reaches zero, the chained transfer address in the DMAC II Index (i.e., the start address of the DMAC II Index that contains a description of the next DMAC II transfer to be performed) is written to the relocatable vector table for the peripheral I/O.
- 3) From the next DMA II transfer request on, transfers are performed based on the DMAC II Index indicated by the rewritten relocatable vector table of the peripheral I/O.

Before the chained transfer function can be used, the relocatable vector table must be located in the RAM area.

(5) End-of-transfer interrupt

For end-of-transfer interrupts, use the end-of-transfer interrupt select bit (bit 6) for transfer mode setup in the DMAC II Index. Setting this bit to 1 selects the end-of-transfer interrupt function. Set the jump address for end-of-transfer interrupt processing in the DMAC II Index's end-of-transfer interrupt address field. An end-of-transfer interrupt is generated when the DMAC II Index transfer counter reaches zero.

DMAC II

Execution time

The number of DMAC II execution cycles is calculated by the equation below.

For other than multiple transfers, $t = 6 + (26 + A + B + C + D) \times m + (4 + E) \times n$ (cycles)

For multiple transfers, $t = 21 + (11 + B + C) \times k$ (cycles)

where

A: If the source of transfer is immediate data, $A = 0$; if it is memory, $A = -1$

B: If the source address of transfer is a variable address, $B = 0$; if it is a fixed address, $B = 1$

C: If the destination address of transfer is a variable address, $C = 0$; if it is a fixed address, $C = 1$

D: If the arithmetic function is not selected, $D = 0$; if the arithmetic function is selected and the source of transfer is immediate data or fixed address memory, $D = 7$; if the arithmetic function is selected and the source of transfer is variable address memory, $D = 8$

E: If the chained transfer function is not selected, $E = 0$; if the chained transfer function is selected, $E = 4$

m: For single transfer, $m = 1$; for burst transfer, $m =$ the value set by the transfer counter

n: If the transfer count is one, $n = 0$; if the transfer count is two or greater, $n = 1$

k: Number of transfers set by transfer mode bits 4–7

The above equation applies only when all of the following conditions are met, however.

- No bus wait states are inserted.
- The DMAC II Index is set to an even address.
- During word transfer, the transfer source address, transfer destination address, and operation address all are set to an even address.

Note that the first instruction in end-of-transfer interrupt processing is executed 7 cycles after DMAC II transfers are completed.

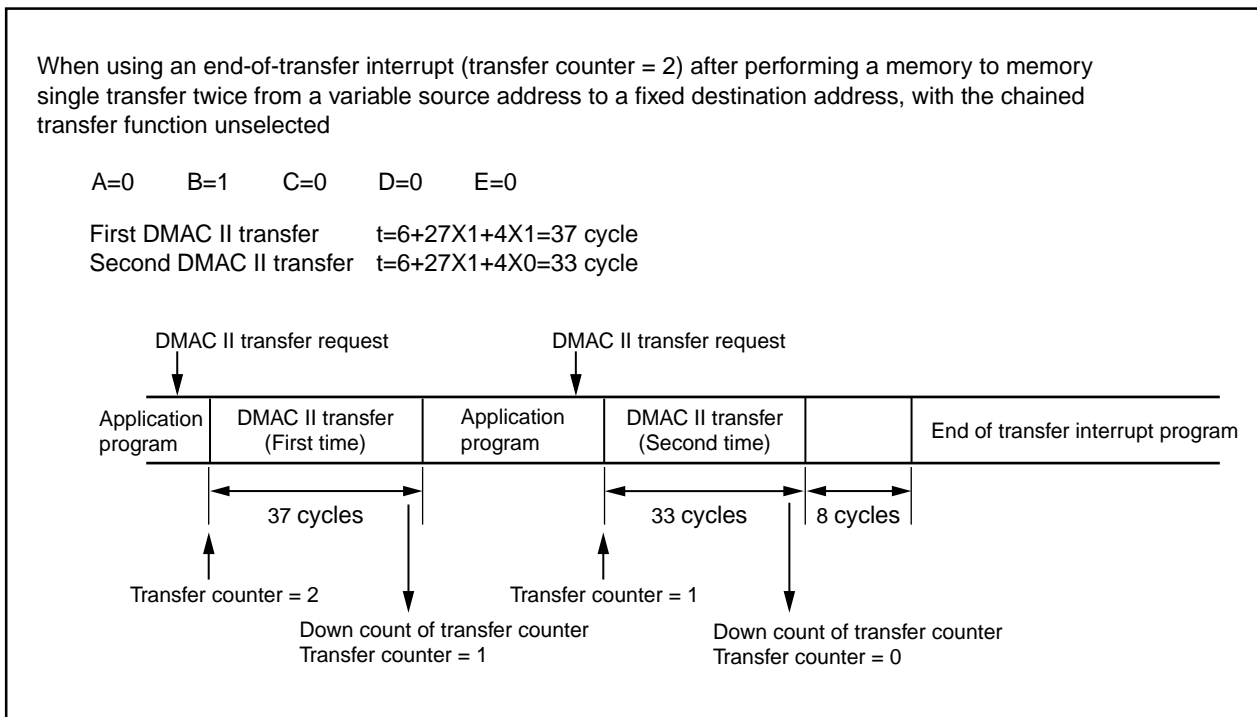


Figure 1.12.4. Transfer Time

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 1.13.1 and 1.13.2 show the block diagram of timers.

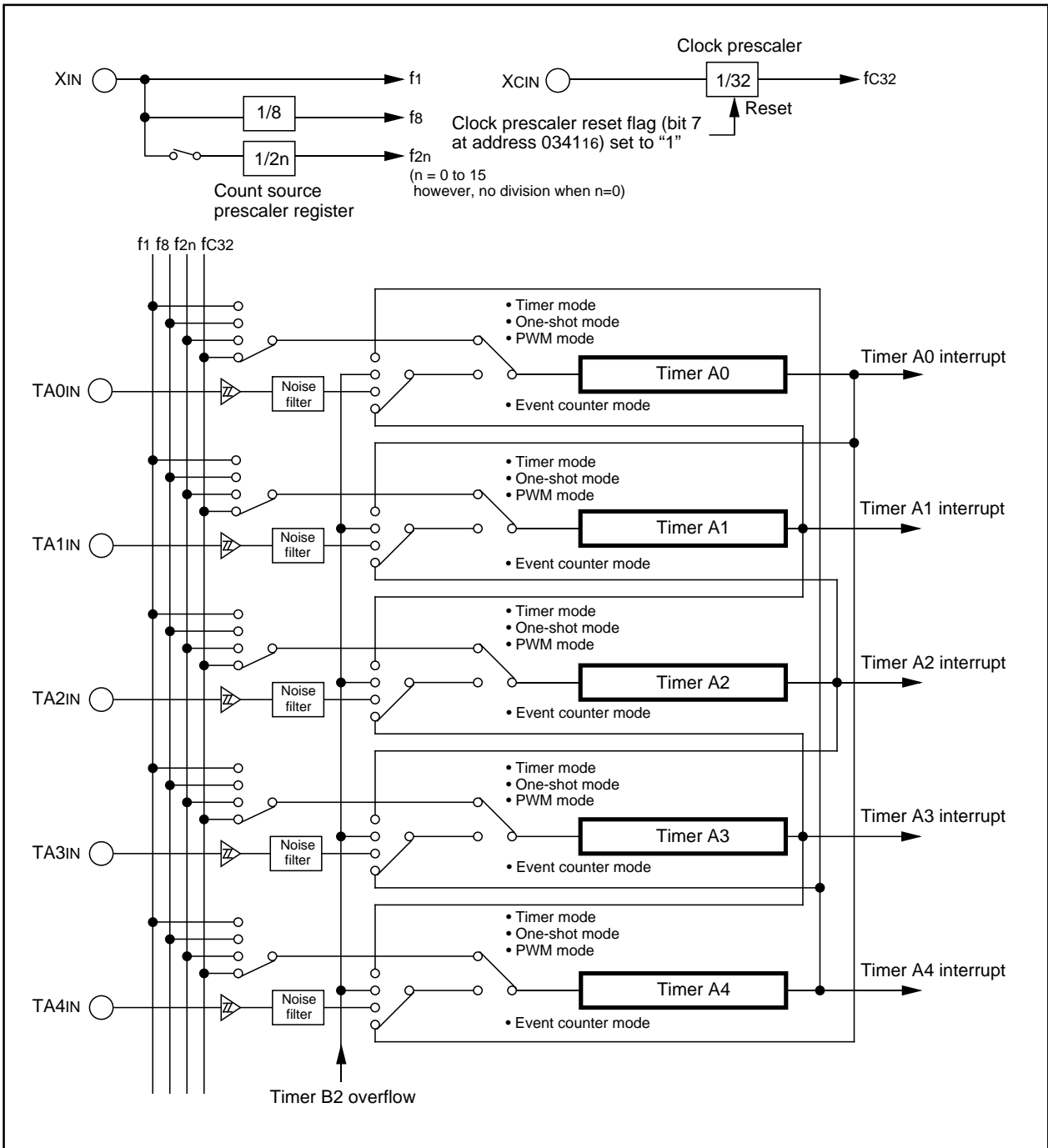


Figure 1.13.1. Timer A block diagram

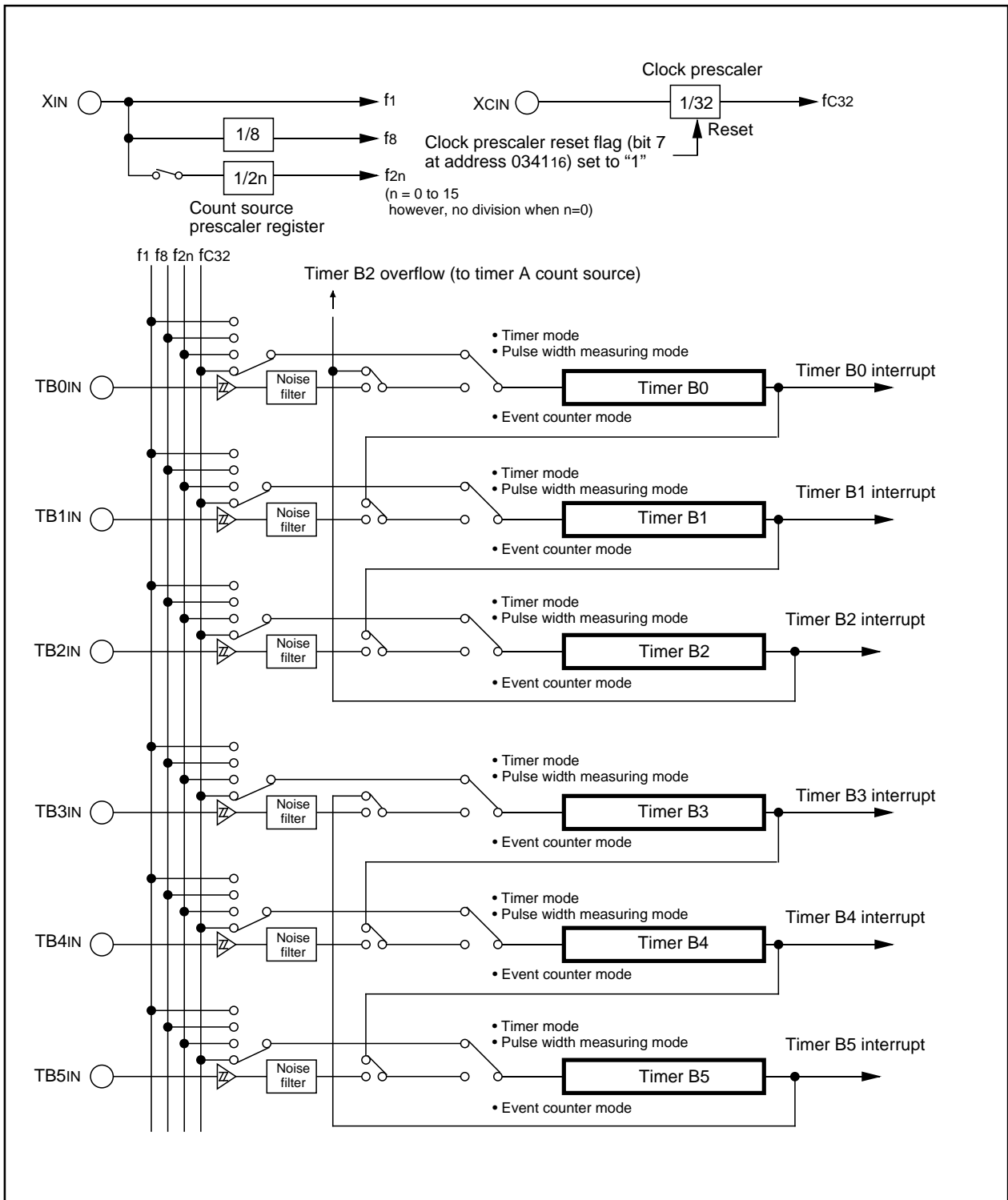


Figure 1.13.2. Timer B block diagram

Timer A

Timer A

Figure 1.14.1 shows the block diagram of timer A. Figures 1.14.2 to 1.14.6 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 outputs one effective pulse until the count reaches "0000₁₆".
- Pulse width modulation (PWM) mode: The timer outputs pulses of a given width.

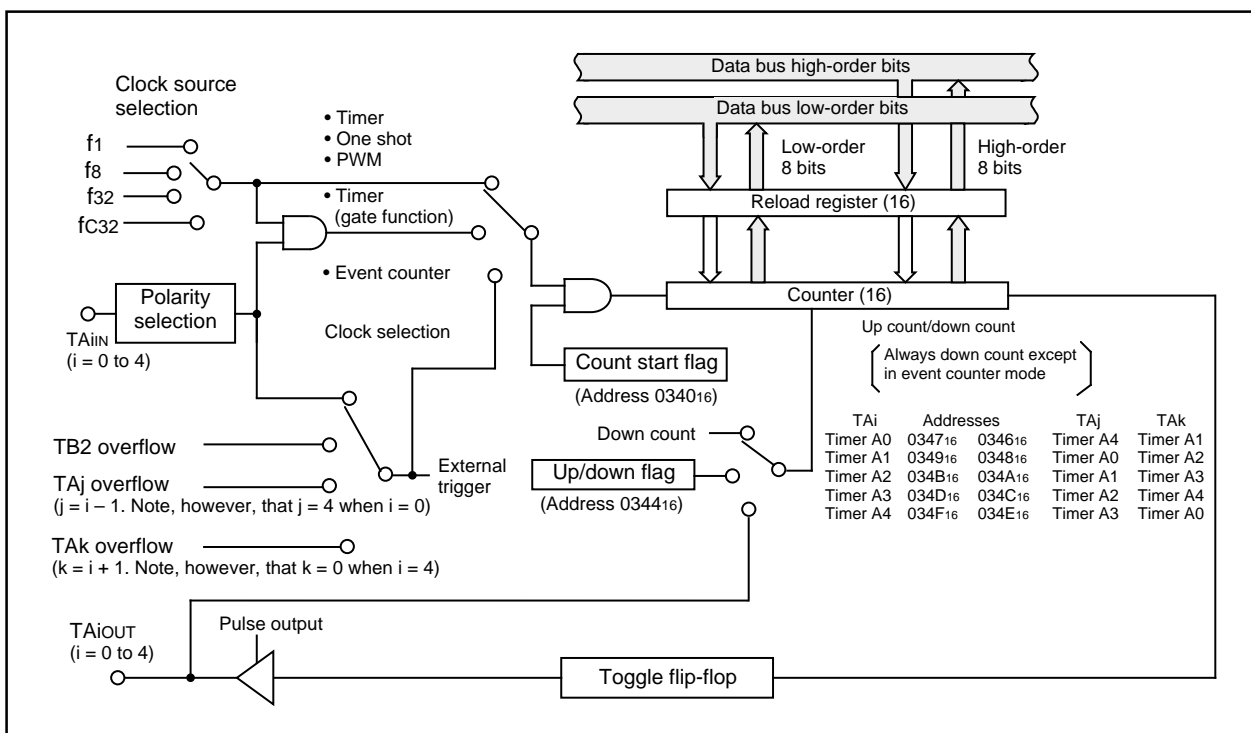
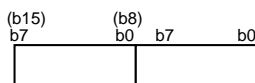


Figure 1.14.1. Block diagram of timer A

Timer A

Timer Ai register (i = 0 to 4) (Note 1)



Symbol	Address	When reset
TAi (i = 0 to 2)	0347 ₁₆ , 0346 ₁₆ , 0349 ₁₆ , 0348 ₁₆ , 034B ₁₆ , 034A ₁₆	Indeterminate
TAi (i = 3, 4)	034D ₁₆ , 034C ₁₆ , 034F ₁₆ , 034E ₁₆	Indeterminate

	Function	Values that can be set	R	W
Timer mode	16-bit counter (set to dividing ratio)	0000 ₁₆ to FFFF ₁₆	○	○
Event counter mode	16-bit counter (set to dividing ratio) (Note 2)	0000 ₁₆ to FFFF ₁₆	○	○
One-shot timer mode	16-bit counter (set to one shot width) (Note 6)	0000 ₁₆ to FFFF ₁₆ (Note 3)	—	○
Pulse width modulation mode (16-bit PWM)	16-bit pulse width modulator (set to PWM pulse "H" width) (Note 4, 7)	0000 ₁₆ to FFFE ₁₆ (Note 3)	—	○
Pulse width modulation mode (8-bit PWM)	Low-order 8 bits : 8-bit prescaler (set to PWM period) (Note 5, 7) High-order 8 bits : 8-bit pulse width modulator (set to PWM pulse "H" width)	00 ₁₆ to FE ₁₆ (High-order address) 00 ₁₆ to FF ₁₆ (Low-order address) (Note 3)	—	○

- Note 1: Read and write data in 16-bit units.
- Note 2: Counts pulses from an external source or timer overflow.
- Note 3: Use MOV instruction to write to this register.
- Note 4: When setting value is n, PWM period and "H" width of PWM pulse are as follows:
 PWM period : $(2^{16} - 1) / f_i$
 PWM pulse "H" width : n / f_i
- Note 5: When setting value of high-order address is n and setting value of low-order address is m, PWM period and "H" width of PWM pulse are as follows:
 PWM period : $(2^8 - 1) \times (m + 1) / f_i$
 PWM pulse "H" width : $(m + 1)n / f_i$
- Note 6: When the timer Ai register is set to "0000₁₆", 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 7: When the timer Ai register is set to "0000₁₆", 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 "00₁₆".

Figure 1.14.2. Timer A-related registers (1)

Timer A

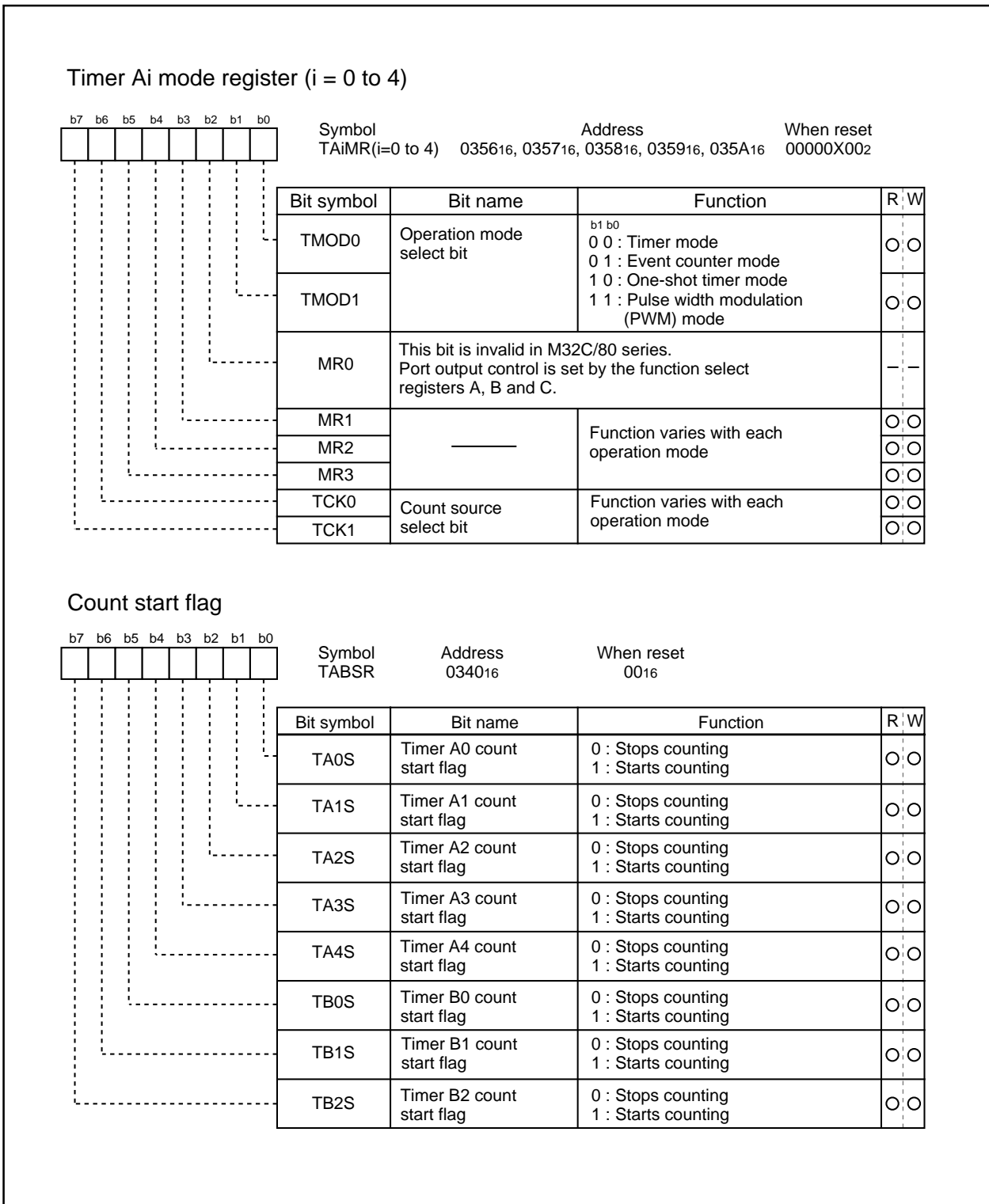


Figure 1.14.3. Timer A-related registers (2)

Timer A

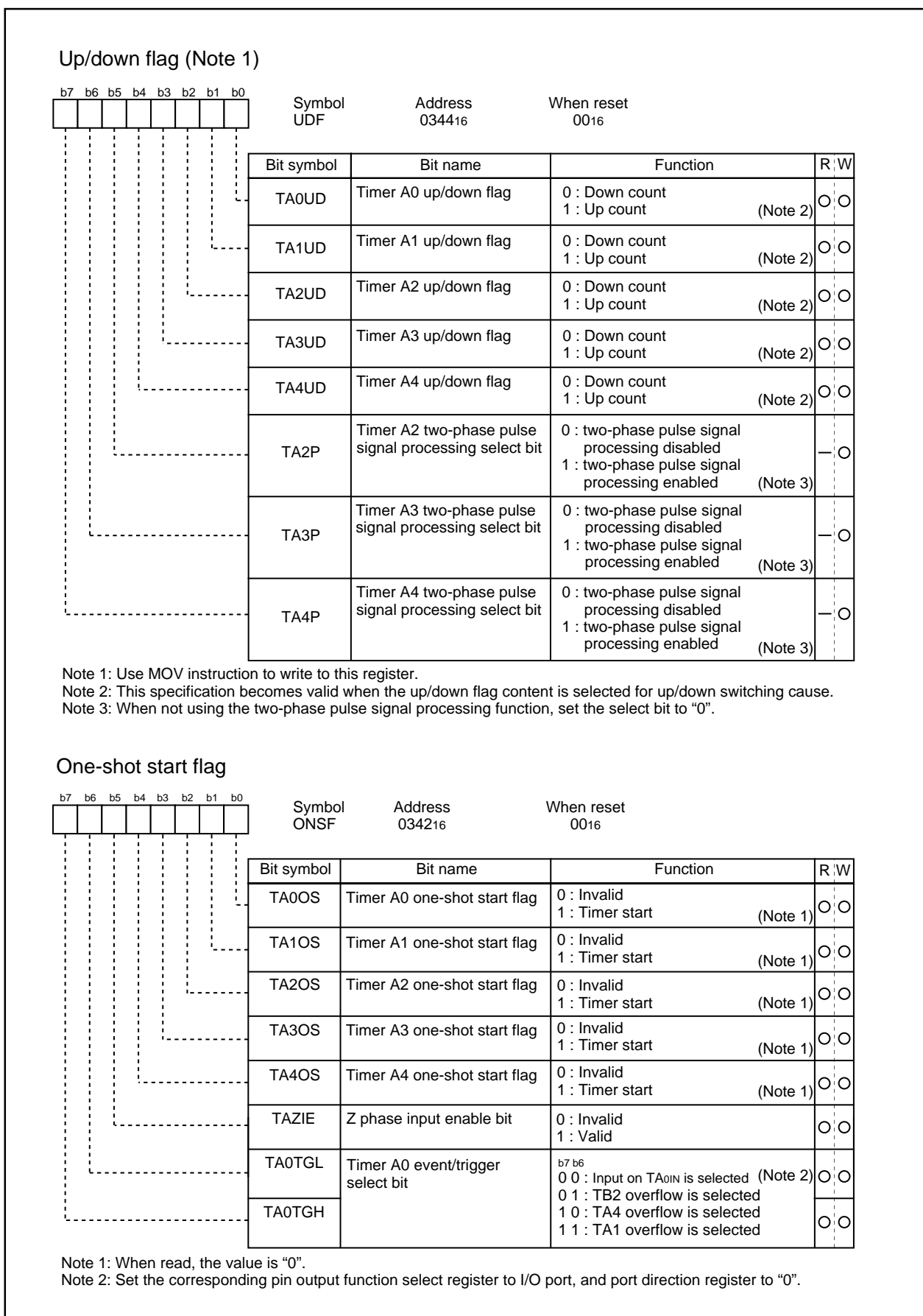


Figure 1.14.4. Timer A-related registers (3)

Timer A

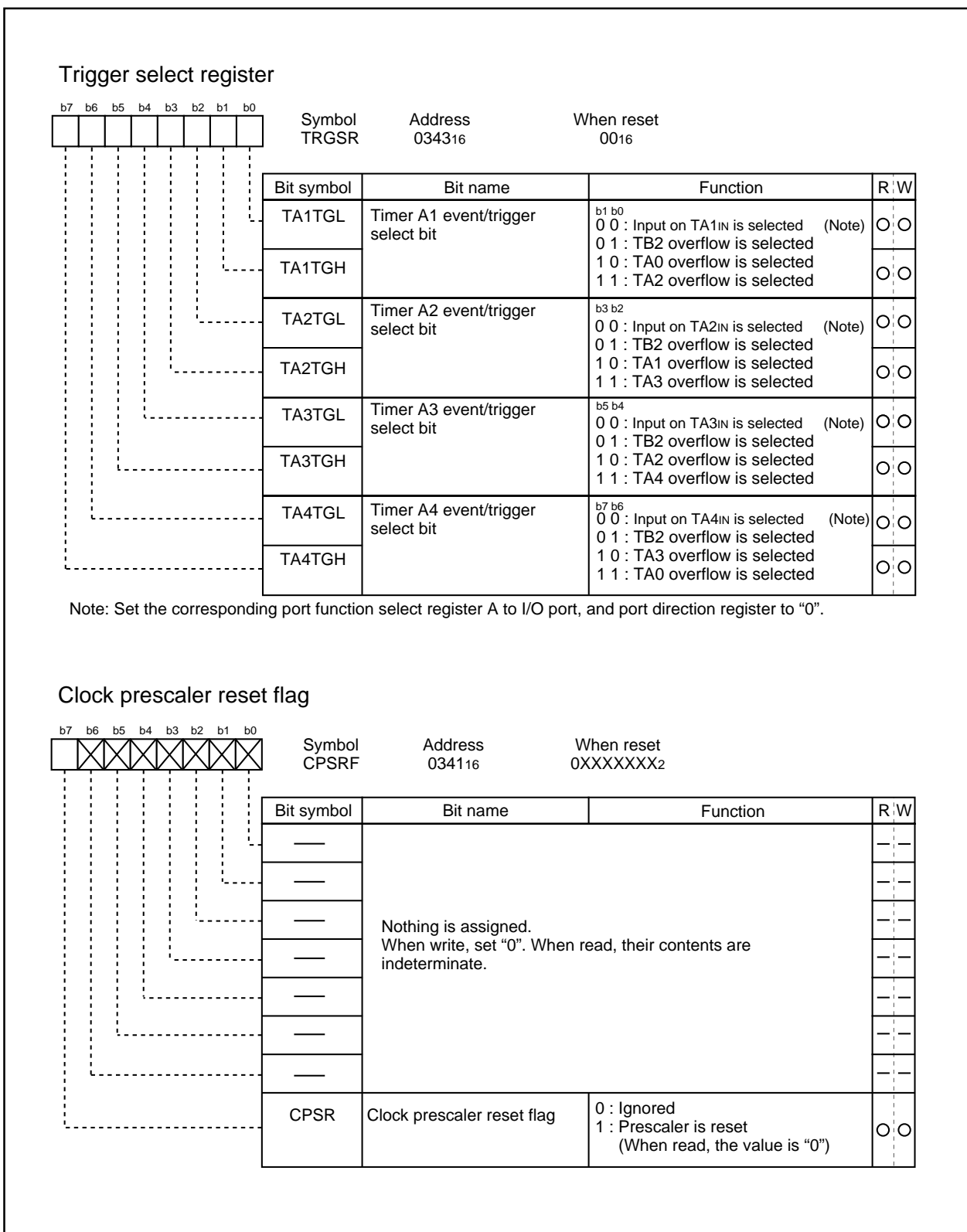


Figure 1.14.5. Timer A-related registers (4)

Timer A

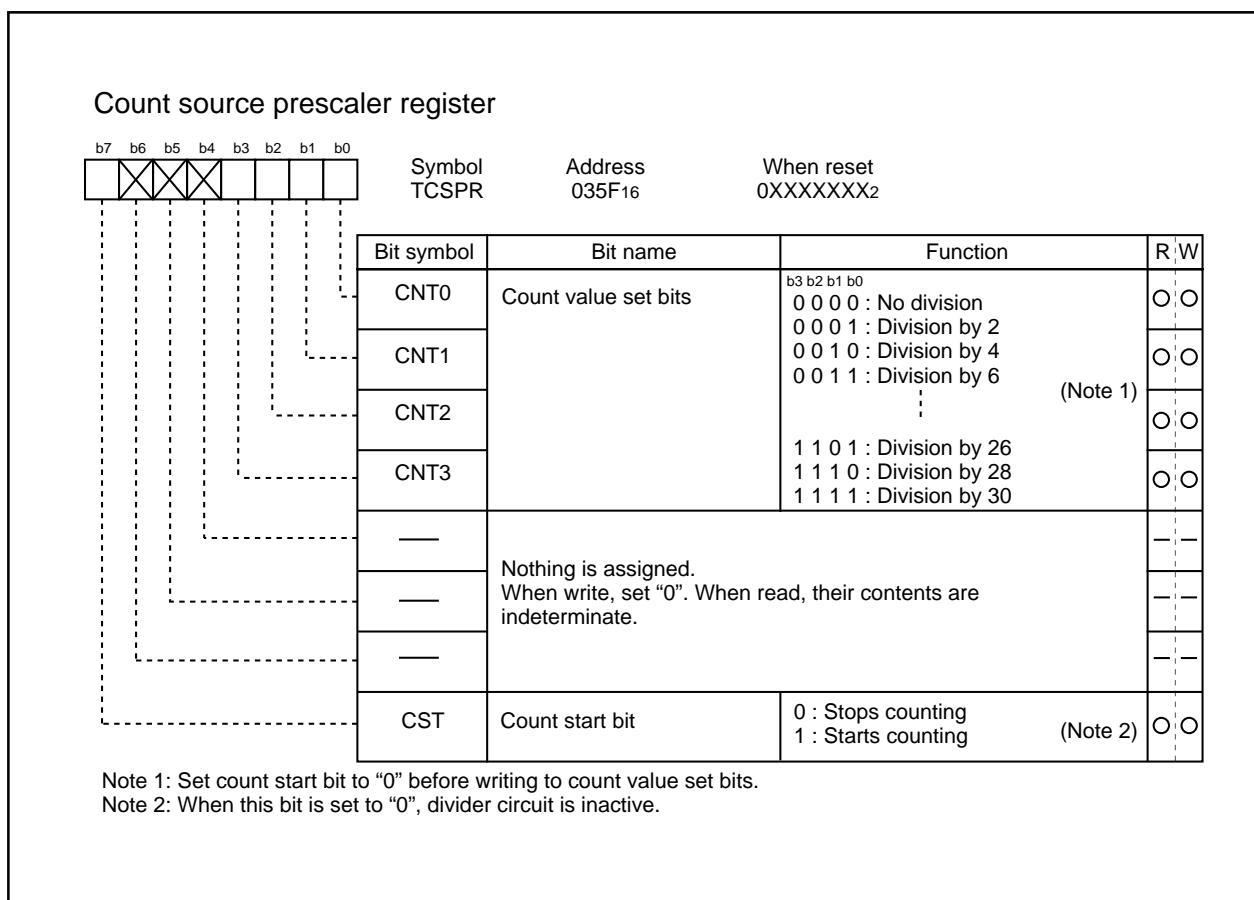


Figure 1.14.6. Timer A-related registers (5)

Timer A

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 1.14.1.) Figure 1.14.7 shows the timer Ai mode register in timer mode.

Table 1.14.1. Specifications of timer mode

Item	Specification
Count source	f1, f8, f2n, fc32
Count operation	<ul style="list-style-type: none"> • Down count • When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(m+1)m : 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 corresponding function select registers A, B and C)
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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

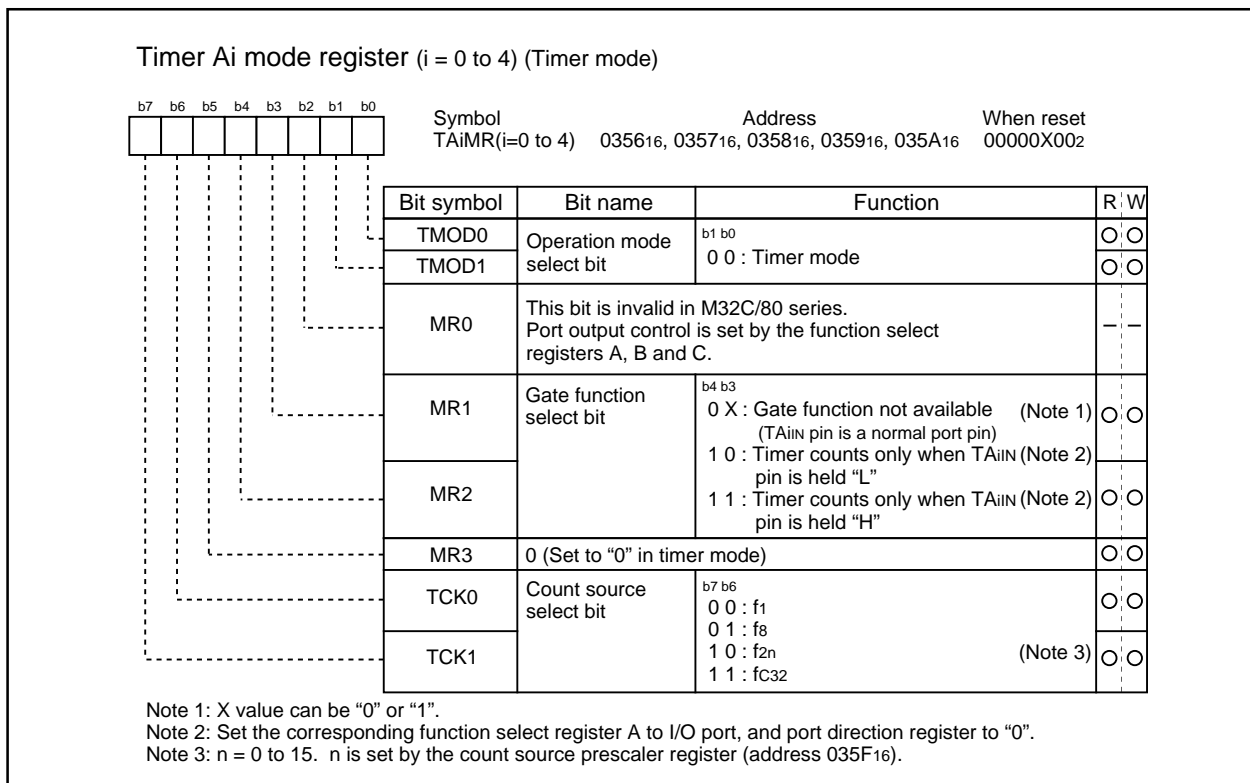


Figure 1.14.7. Timer Ai mode register in timer mode

(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 1.14.2 lists timer specifications when counting a single-phase external signal. Table 1.14.3 lists timer specifications when counting a two-phase external signal. Figure 1.14.8 shows the timer Ai mode register in event counter mode.

Table 1.14.2. Timer specifications in event counter mode (when not processing two-phase pulse signal)

Item	Specification
Count source	<ul style="list-style-type: none"> External signals input to TAIiN pin (effective edge can be selected by software) TB2 overflows or underflows, TAJ overflows or underflows
Count operation	<ul style="list-style-type: none"> Up count or down count can be selected by external signal or software When the timer overflows or underflows, it reloads the reload register contents before continuing counting (Note)
Divide ratio	<ul style="list-style-type: none"> $1 / (FFFF_{16} - n + 1)$ for up count $1 / (n + 1)$ for down count <p style="text-align: right;">n : Set value</p>
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
TAiIN pin function	Programmable I/O port or count source input
TAiOUT pin function	Programmable I/O port, pulse output, or up/down count select input (Setting by corresponding function select registers A, B and C)
Read from timer	Count value can be read out by reading timer Ai register
Write to timer	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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 TAIOUT pin's polarity is reversed

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

Timer A

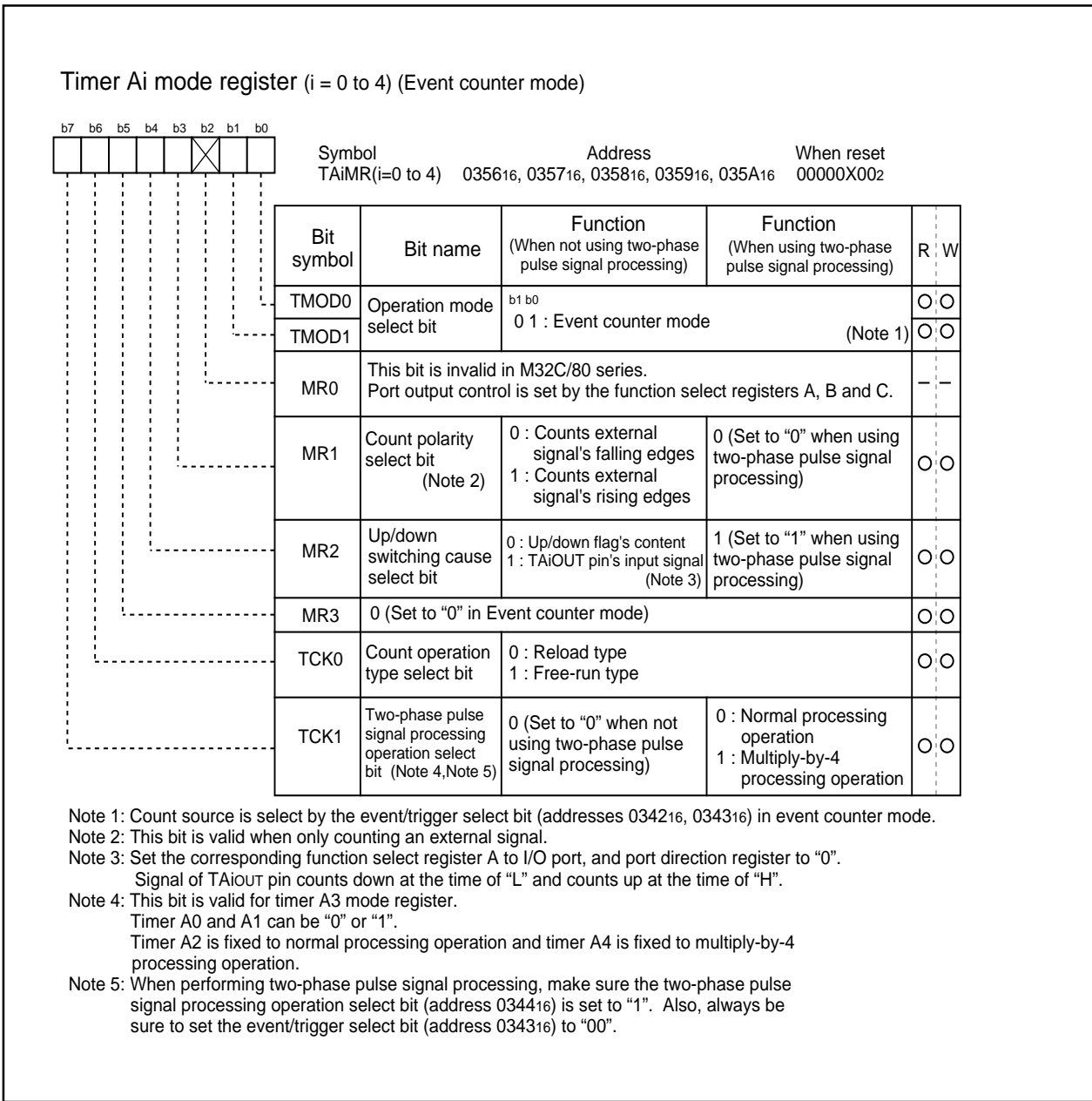
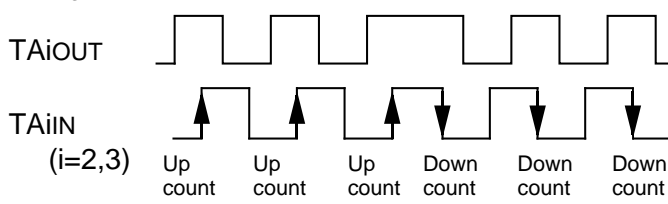
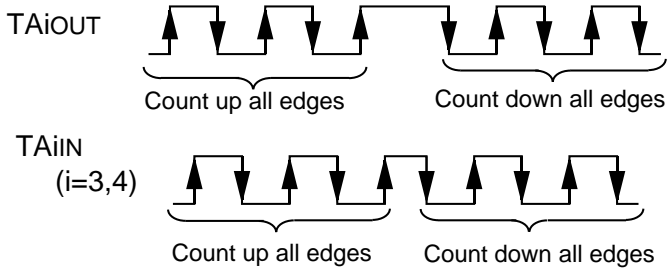


Figure 1.14.8. Timer Ai mode register in event counter mode

Timer A

Table 1.14.3. Timer specifications in event counter mode

Item	Specification
Count source	Two-phase pulse signals input to TAIIN or TAIOUT pin
Count operation	<ul style="list-style-type: none"> 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 1)
Divide ratio	<ul style="list-style-type: none"> 1/ (FFFF₁₆ - 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 corresponding function select register A for I/O port)
Read from timer	Count value can be read out by reading timer A2, A3, or A4 register
Write to timer	<ul style="list-style-type: none"> 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)	<ul style="list-style-type: none"> 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"  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. 

(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 A

• 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 034216) 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 “000016” to the timer A3 register (addresses 034D16 and 034C16).

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 009C16). The Z-phase must have a pulse width greater than 1 cycle of the timer A3 count source. Figure 1.14.9 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 1.14.10 shows the timing at which the counter is reset to “0”.

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

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

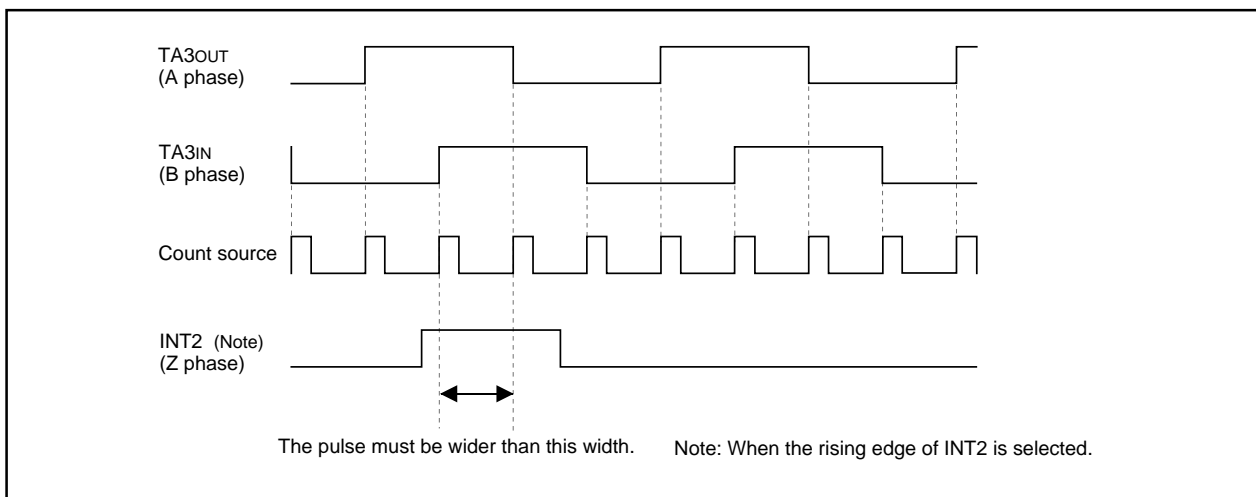


Figure 1.14.9. The relationship between the two-phase pulse (A phase and B phase) and the Z phase

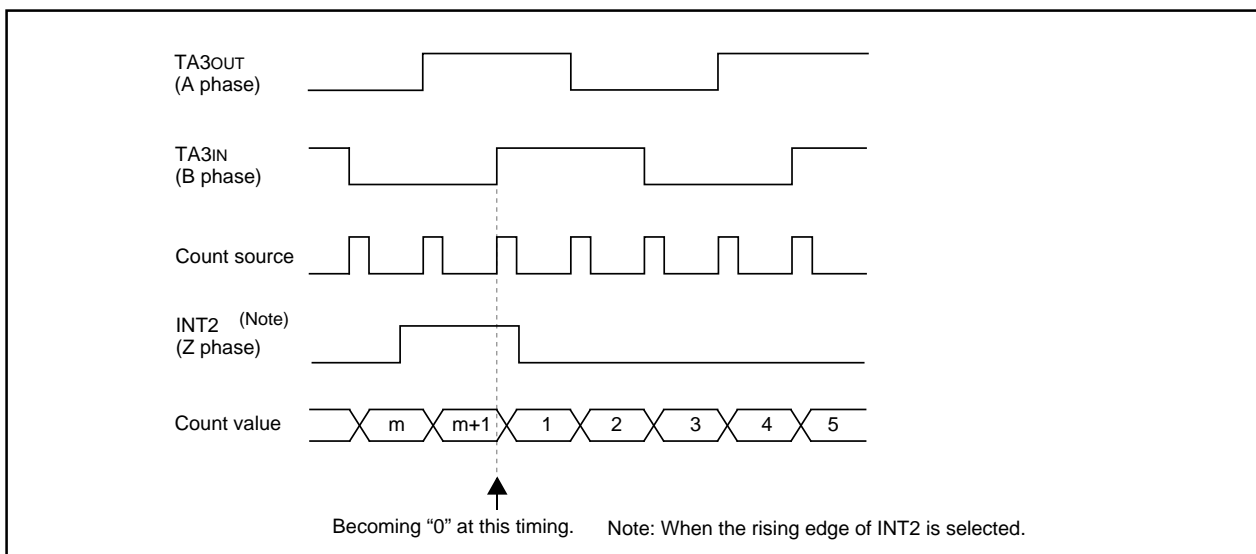


Figure 1.14.10. The counter reset timing

(3) One-shot timer mode

In this mode, the timer operates only once. (See Table 1.14.4.) When a trigger occurs, the timer starts up and continues operating for a given period. Figure 1.14.11 shows the timer Ai mode register in one-shot timer mode.

Table 1.14.4. Timer specifications in one-shot timer mode

Item	Specification
Count source	f1, f8, f2n, fC32
Count operation	<ul style="list-style-type: none"> • The timer counts down • When the count reaches 0000₁₆, 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	<ul style="list-style-type: none"> • An external trigger is input • The timer overflows • The one-shot start flag is set (= 1)
Count stop condition	<ul style="list-style-type: none"> • A new count is reloaded after the count has reached 0000₁₆ • The count start flag is reset (= 0)
Interrupt request generation timing	The count reaches 0000 ₁₆
TAiIN pin function	Programmable I/O port or trigger input
TAiOUT pin function	Programmable I/O port or pulse output (Setting by corresponding function select registers A, B and C)
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	<ul style="list-style-type: none"> • 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 A

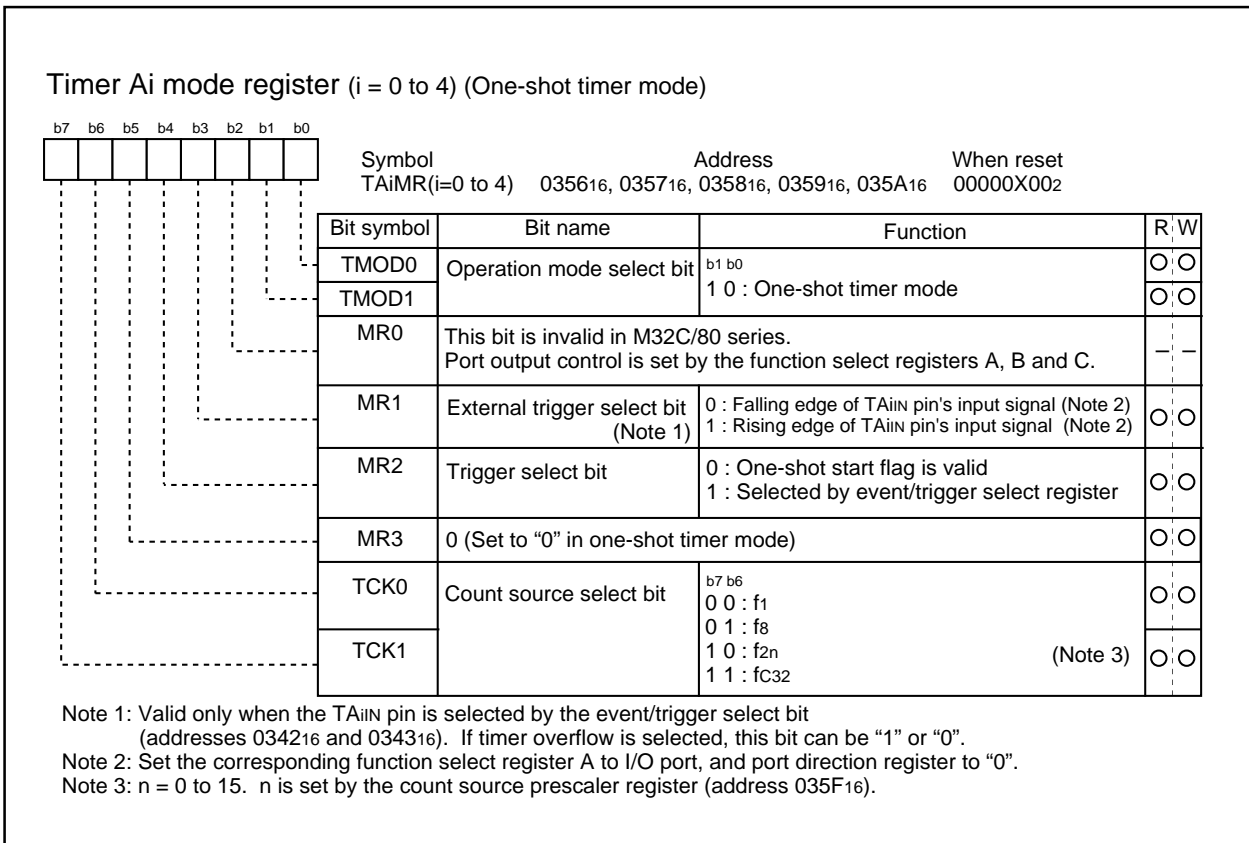


Figure 1.14.11. Timer Ai mode register in one-shot timer mode

(4) Pulse width modulation (PWM) mode

In this mode, the timer outputs pulses of a given width in succession. (See Table 1.14.5.) In this mode, the counter functions as either a 16-bit pulse width modulator or an 8-bit pulse width modulator. Figure 1.14.12 shows the timer Ai mode register in pulse width modulation mode. Figure 1.14.13 shows the example of how a 16-bit pulse width modulator operates. Figure 1.14.14 shows the example of how an 8-bit pulse width modulator operates.

Table 1.14.5. Timer specifications in pulse width modulation mode

Item	Specification
Count source	f1, f8, f2n, fC32
Count operation	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> High level width n / f_i n : Set value Cycle time $(2^{16}-1) / f_i$ fixed
8-bit PWM	<ul style="list-style-type: none"> High level width $n \times (m+1) / f_i$ n : values set to timer Ai register's high-order address Cycle time $(2^8-1) \times (m+1) / f_i$ m: values set to timer Ai register's low-order address
Count start condition	<ul style="list-style-type: none"> External trigger is input The timer overflows The count start flag is set (= 1)
Count stop condition	<ul style="list-style-type: none"> 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 is selected by corresponding function select registers A, B and C)
Read from timer	When timer Ai register is read, it indicates an indeterminate value
Write to timer	<ul style="list-style-type: none"> 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 A

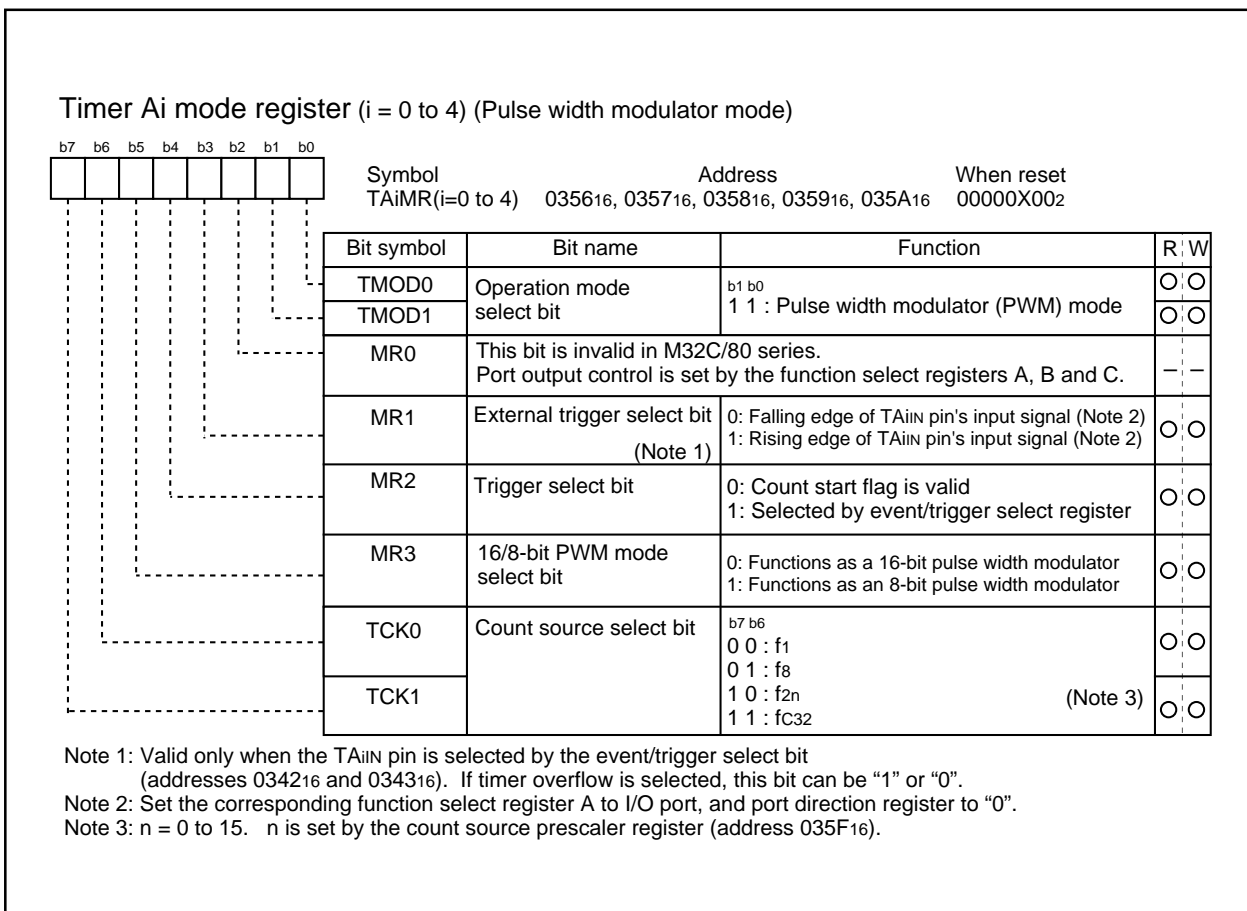


Figure 1.14.12. Timer Ai mode register in pulse width modulation mode

Timer A

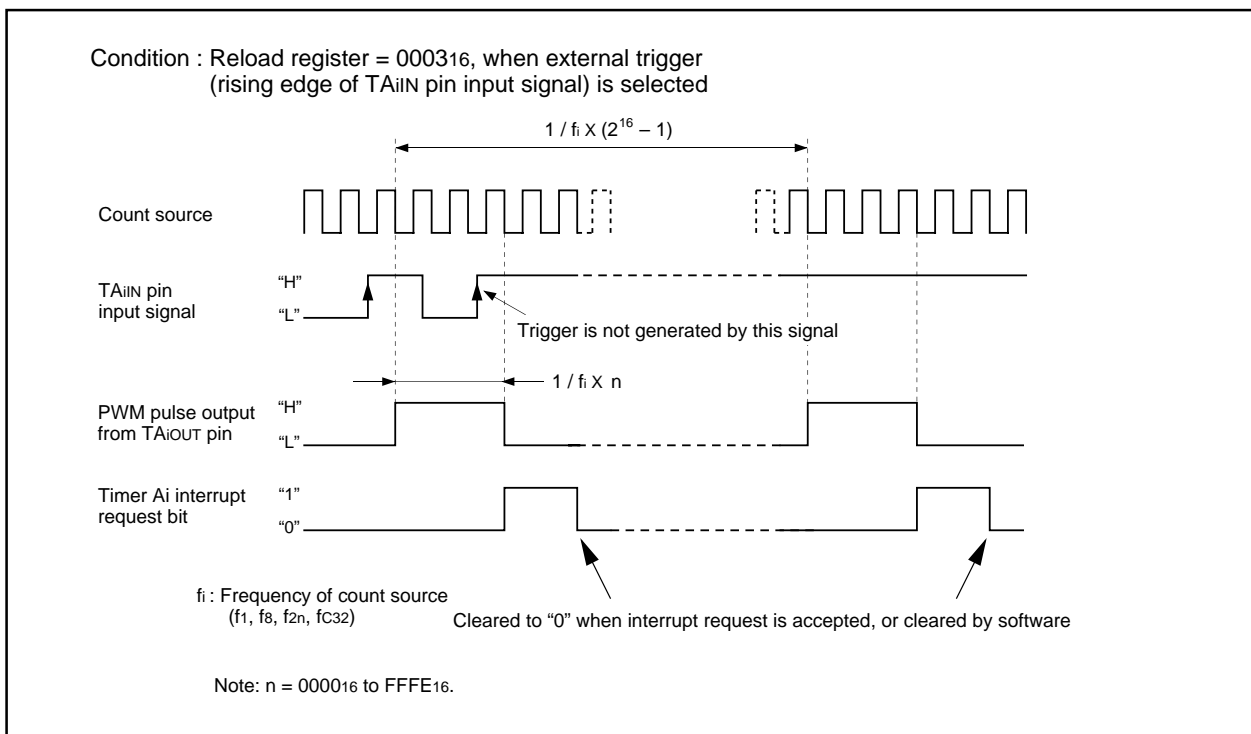


Figure 1.14.13. Example of how a 16-bit pulse width modulator operates

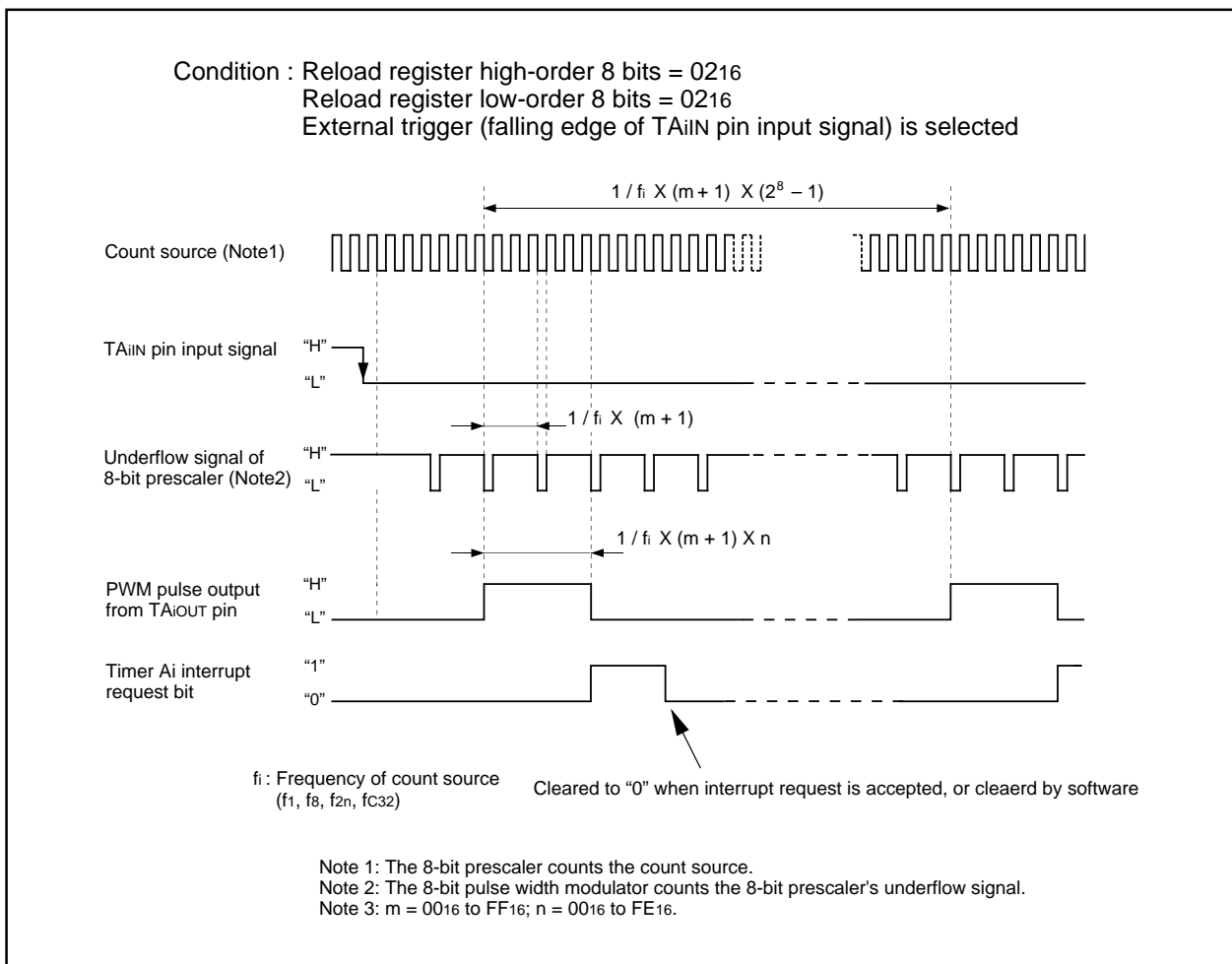


Figure 1.14.14. Example of how an 8-bit pulse width modulator operates

Timer B

Figure 1.15.1 shows the block diagram of timer B. Figures 1.15.2 and 1.15.4 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.

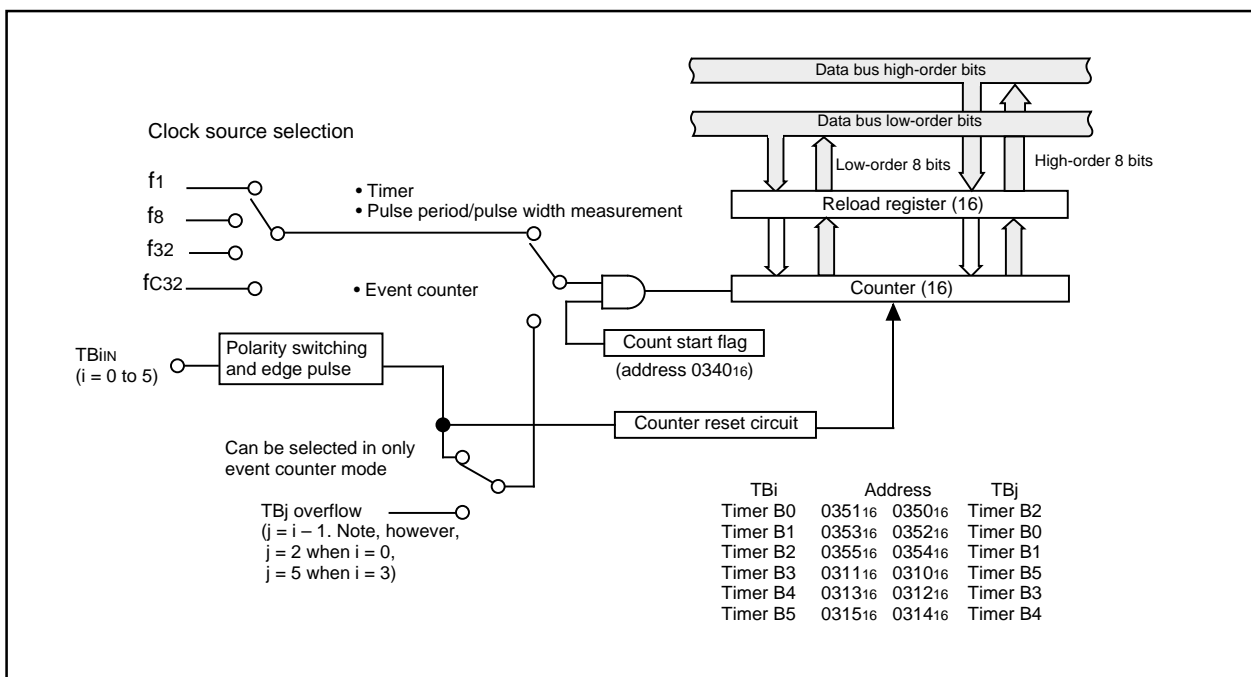


Figure 1.15.1. Block diagram of timer B

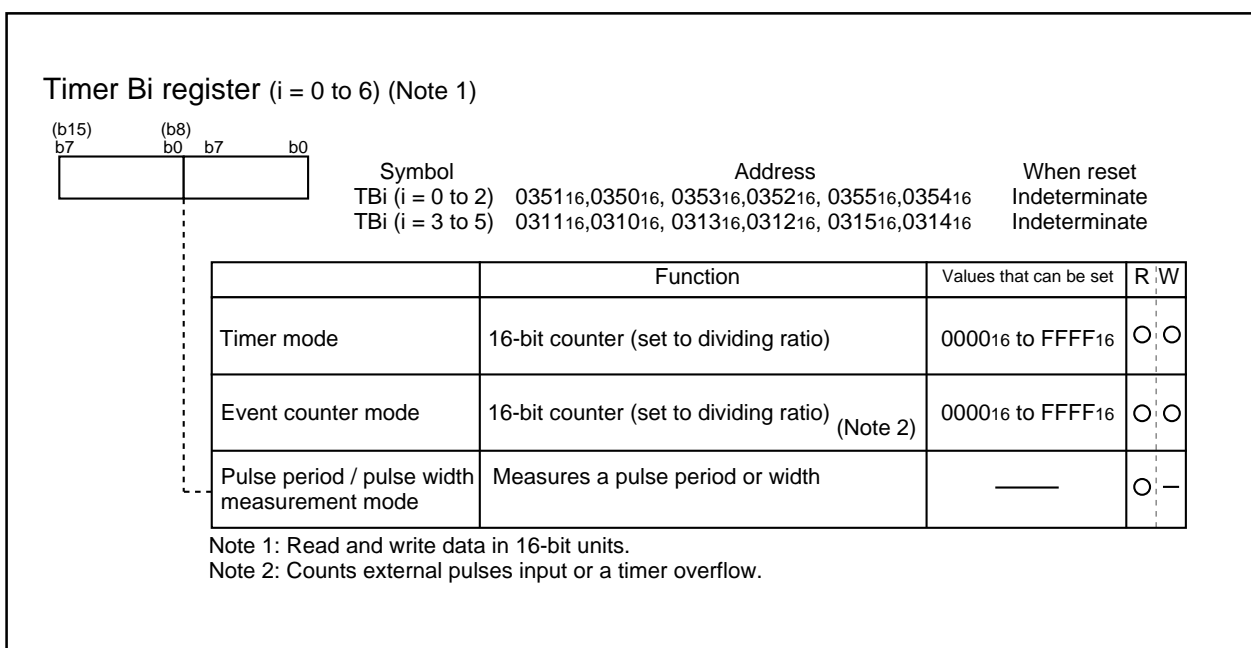


Figure 1.15.2. Timer B-related registers (1)

Timer B

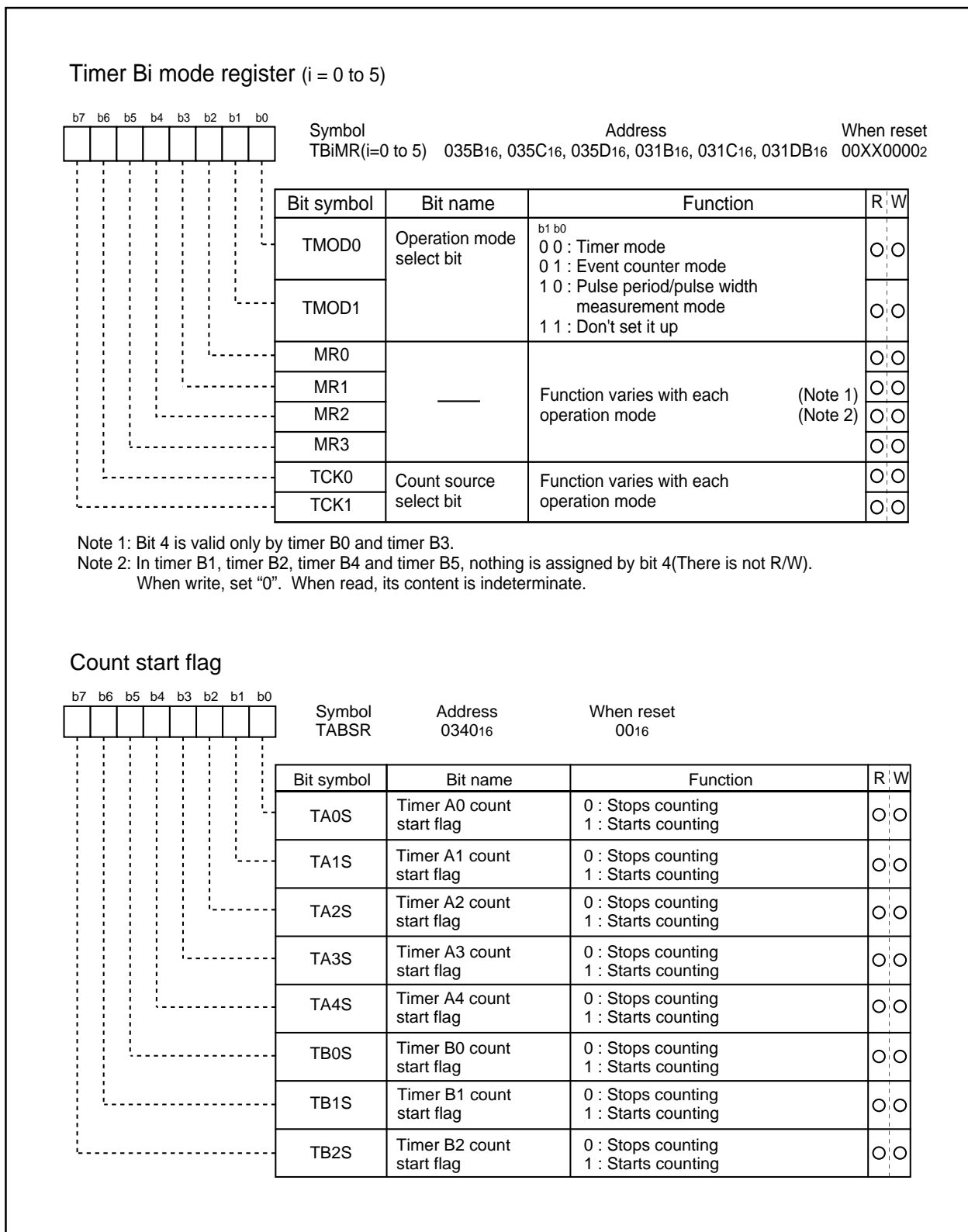


Figure 1.15.3. Timer B-related registers (2)

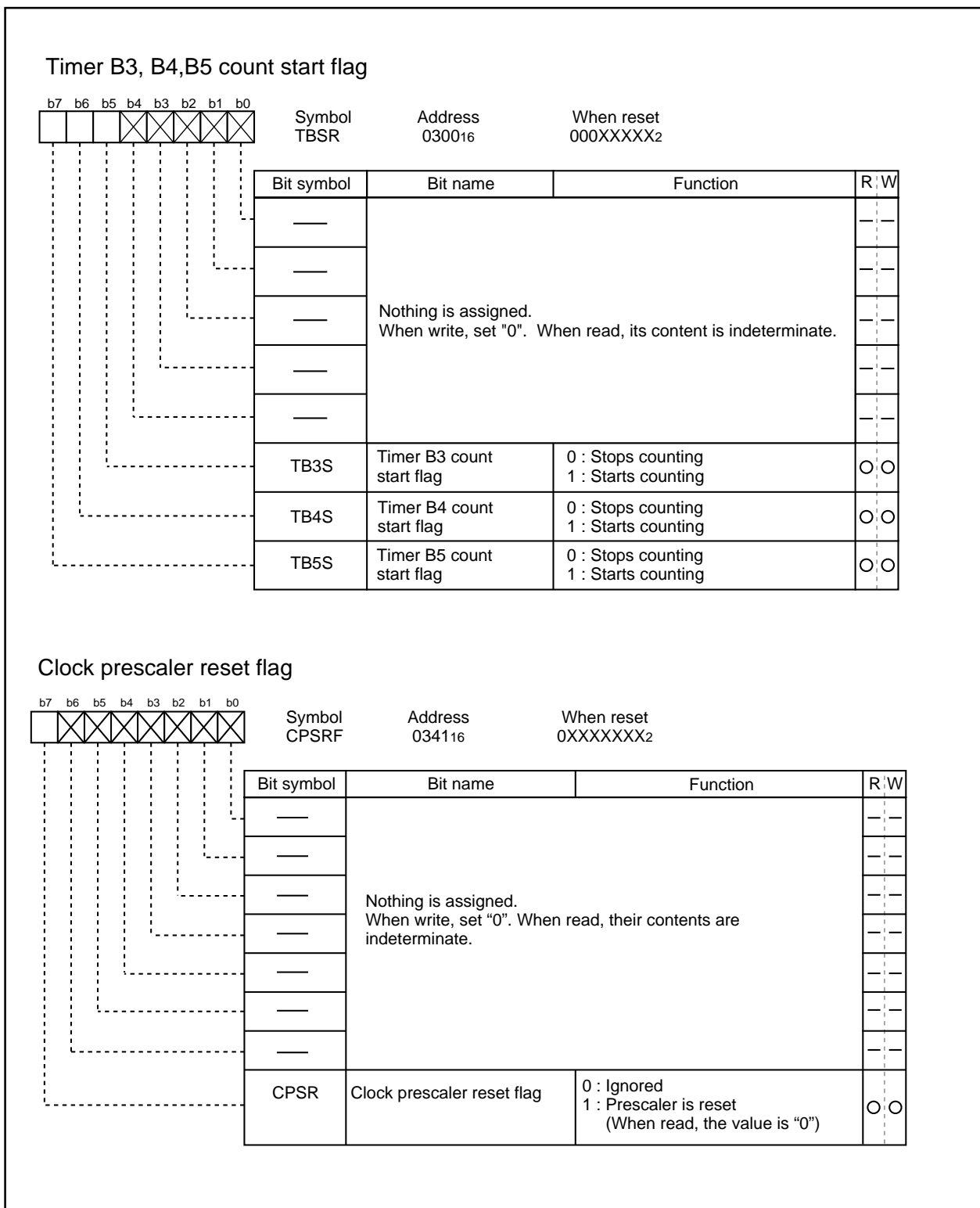


Figure 1.15.4. Timer B-related registers (3)

(1) Timer mode

In this mode, the timer counts an internally generated count source. (See Table 1.15.1.) Figure 1.15.5 shows the timer Bi mode register in timer mode.

Table 1.15.1. Timer specifications in timer mode

Item	Specification
Count source	f1, f8, f2n, fc32
Count operation	<ul style="list-style-type: none"> Counts down When the timer underflows, it reloads the reload register contents before continuing counting
Divide ratio	1/(m+1) ^m : 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	<ul style="list-style-type: none"> 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)

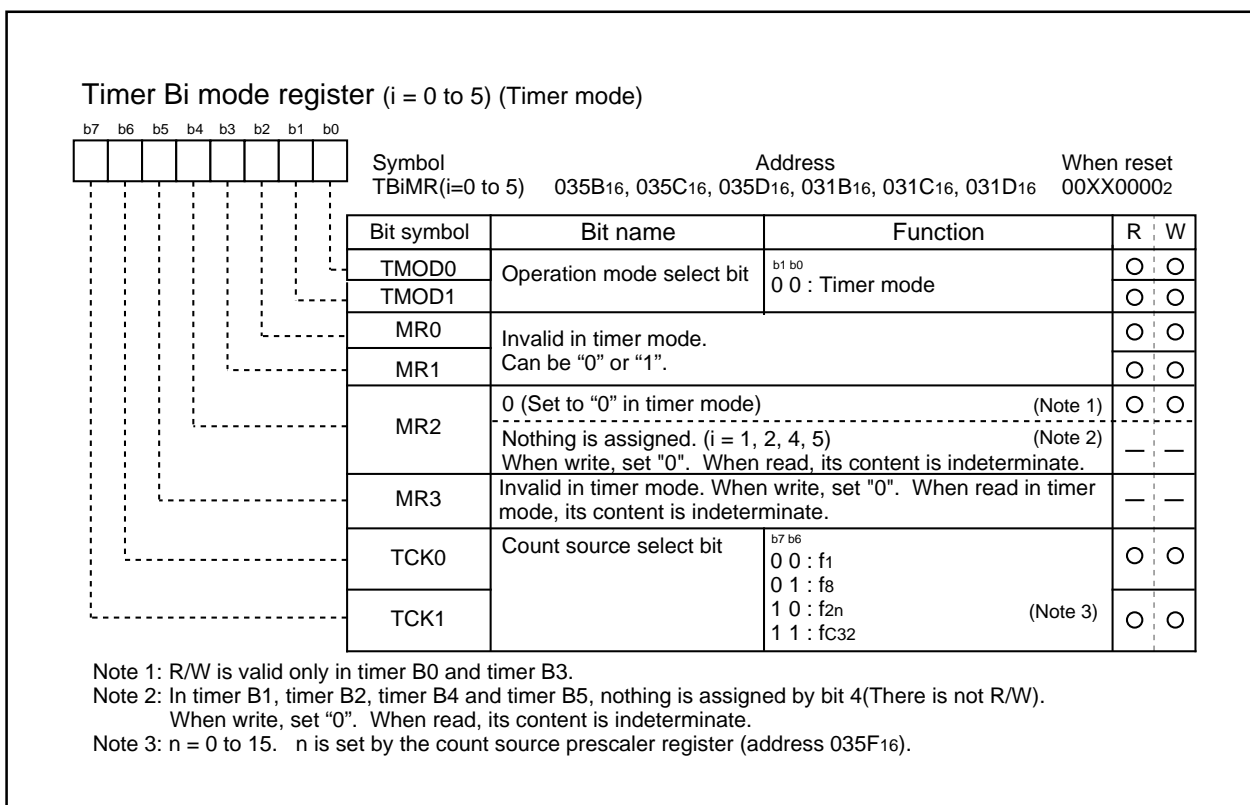


Figure 1.15.5. Timer Bi mode register in timer mode

(2) Event counter mode

In this mode, the timer counts an external signal or an internal timer's overflow. (See Table 1.15.2.)

Figure 1.15.6 shows the timer Bi mode register in event counter mode.

Table 1.15.2. Timer specifications in event counter mode

Item	Specification
Count source	<ul style="list-style-type: none"> 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 TBj overflows or underflows
Count operation	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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)

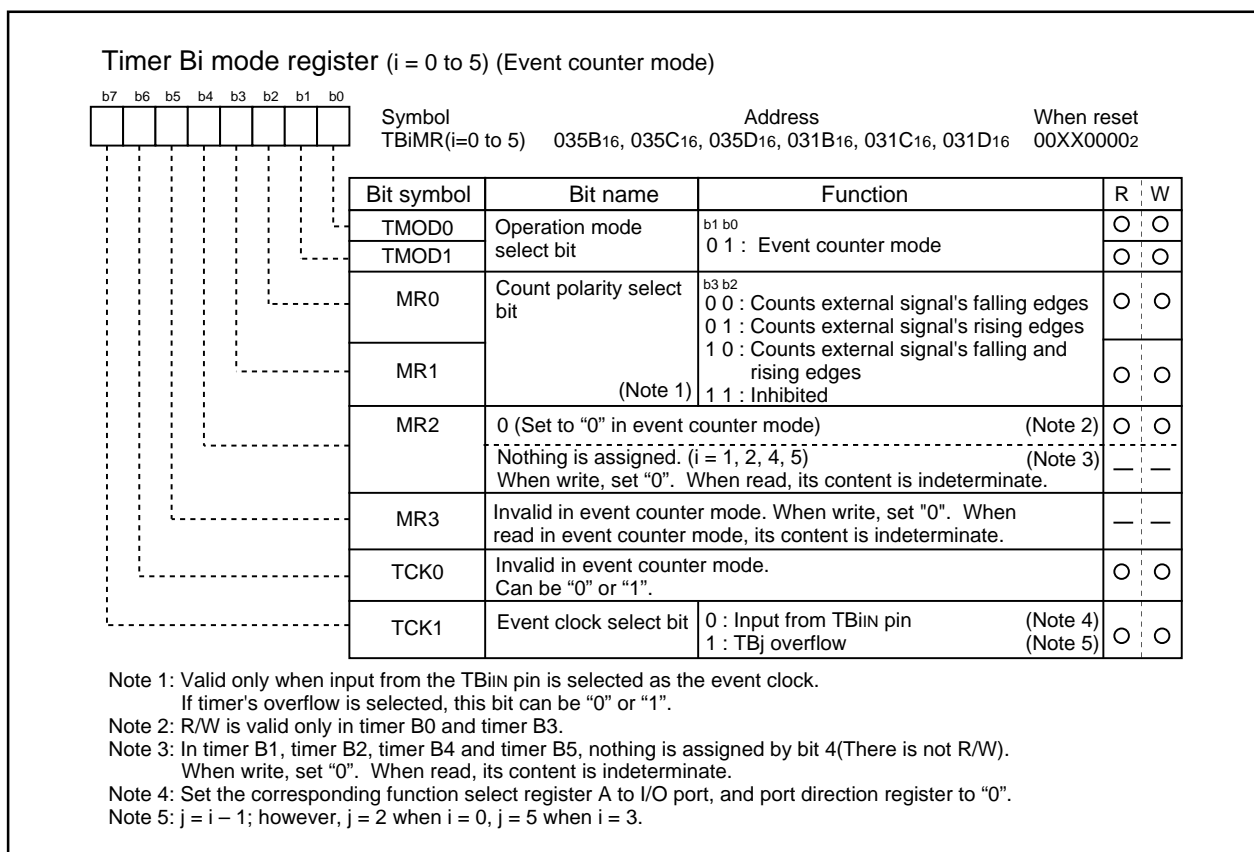


Figure 1.15.6. Timer Bi mode register in event counter mode

(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 1.15.3.)

Figure 1.15.7 shows the timer Bi mode register in pulse period/pulse width measurement mode. Figure

1.15.8 shows the operation timing when measuring a pulse period. Figure 1.15.9 shows the operation

timing when measuring a pulse width.

Table 1.15.3. Timer specifications in pulse period/pulse width measurement mode

Item	Specification
Count source	f1, f8, f2n, fC32
Count operation	<ul style="list-style-type: none"> Count up Counter value "0000₁₆" 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	<ul style="list-style-type: none"> 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.)
TBiIN 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.

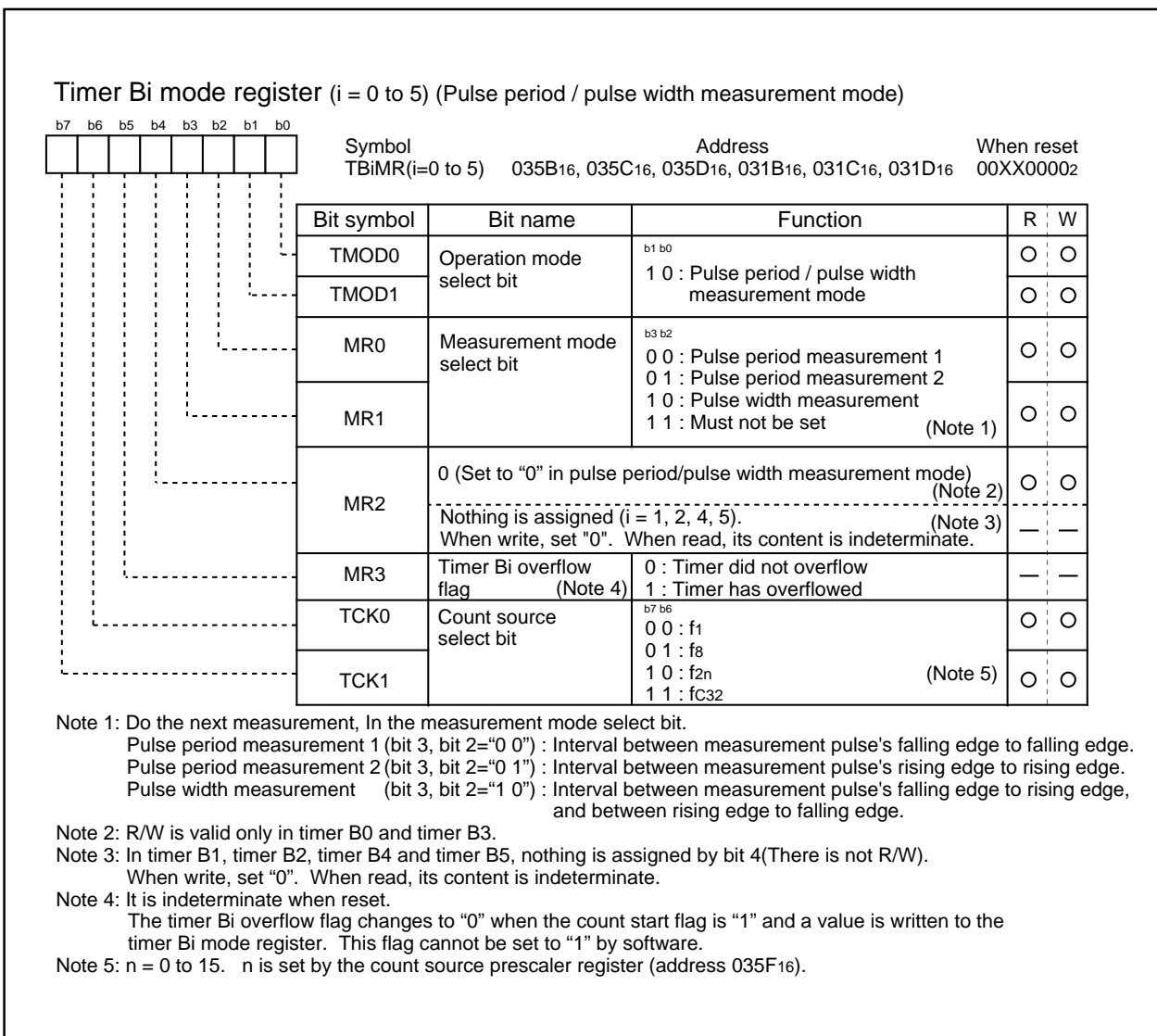


Figure 1.15.7. Timer Bi mode register in pulse period/pulse width measurement mode

Timer B

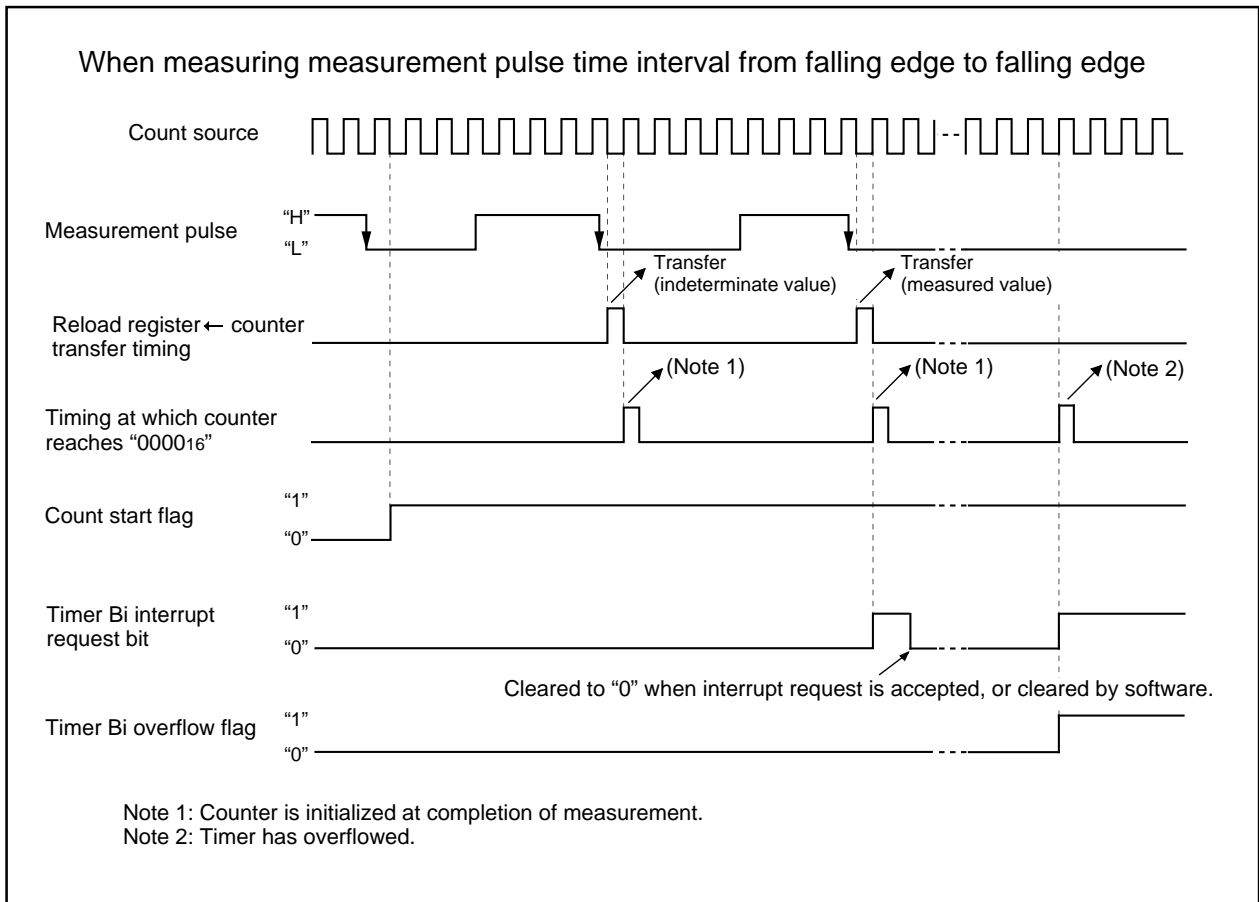


Figure 1.15.8. Operation timing when measuring a pulse period

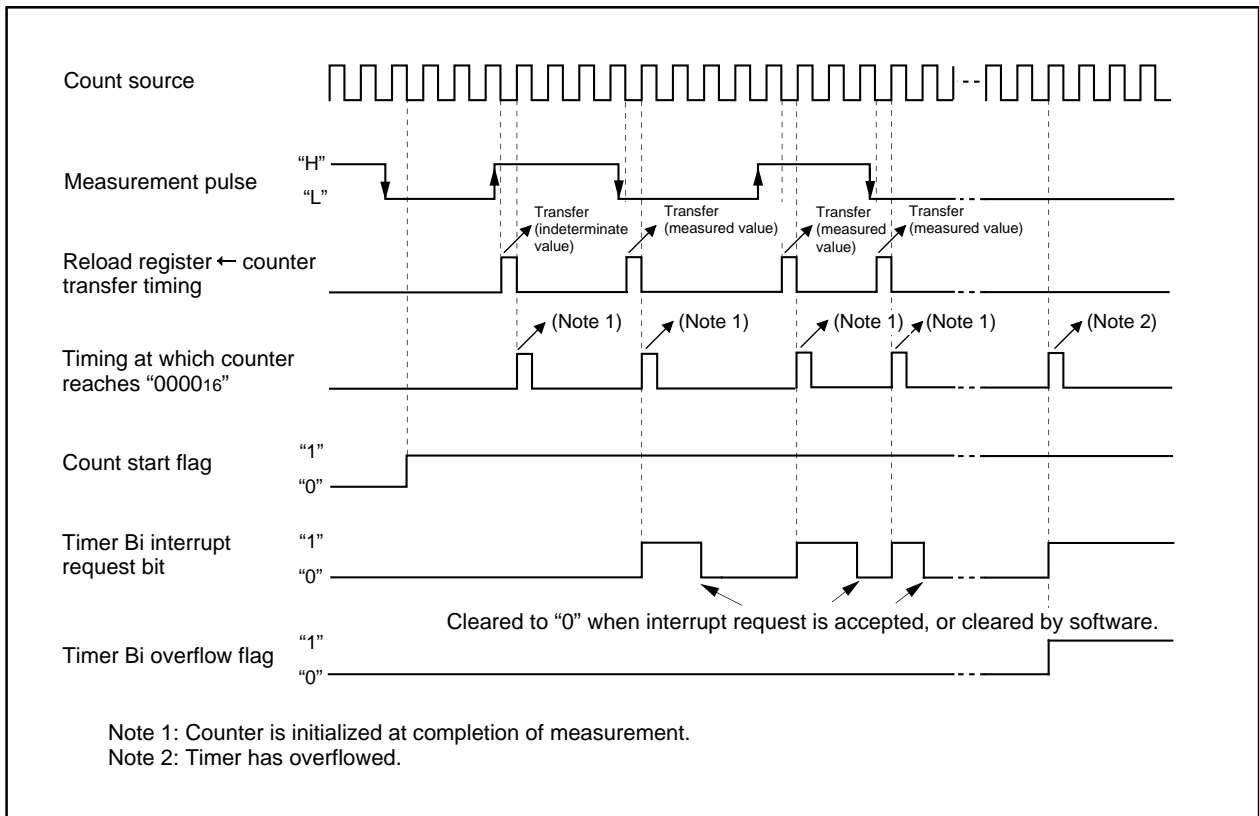


Figure 1.15.9. Operation timing when measuring a pulse width

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 1.16.1 through 1.16.5 show registers related to timers for three-phase motor control.

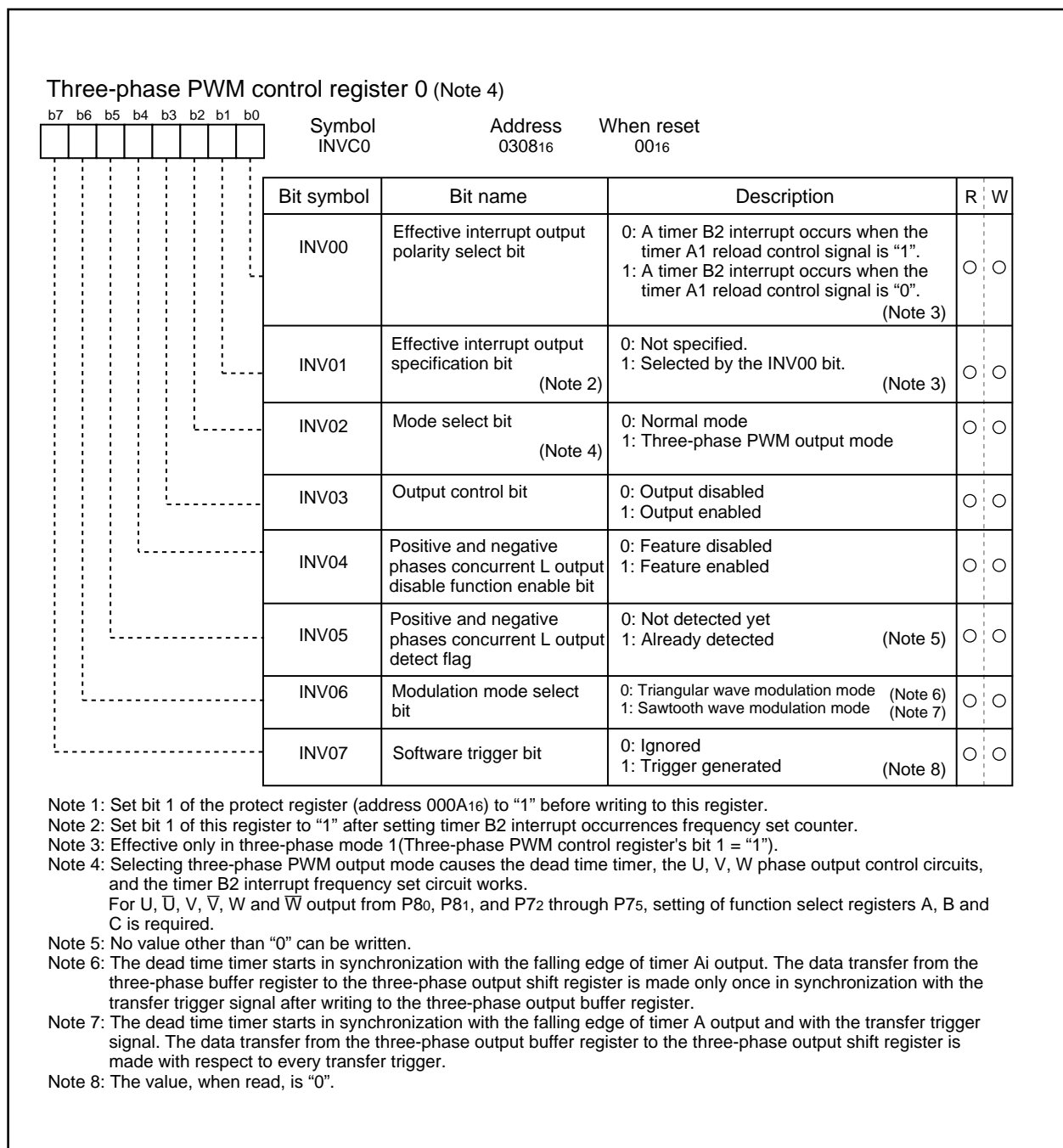


Figure 1.16.1. Registers related to timers for three-phase motor control

Three-phase motor control timers' functions

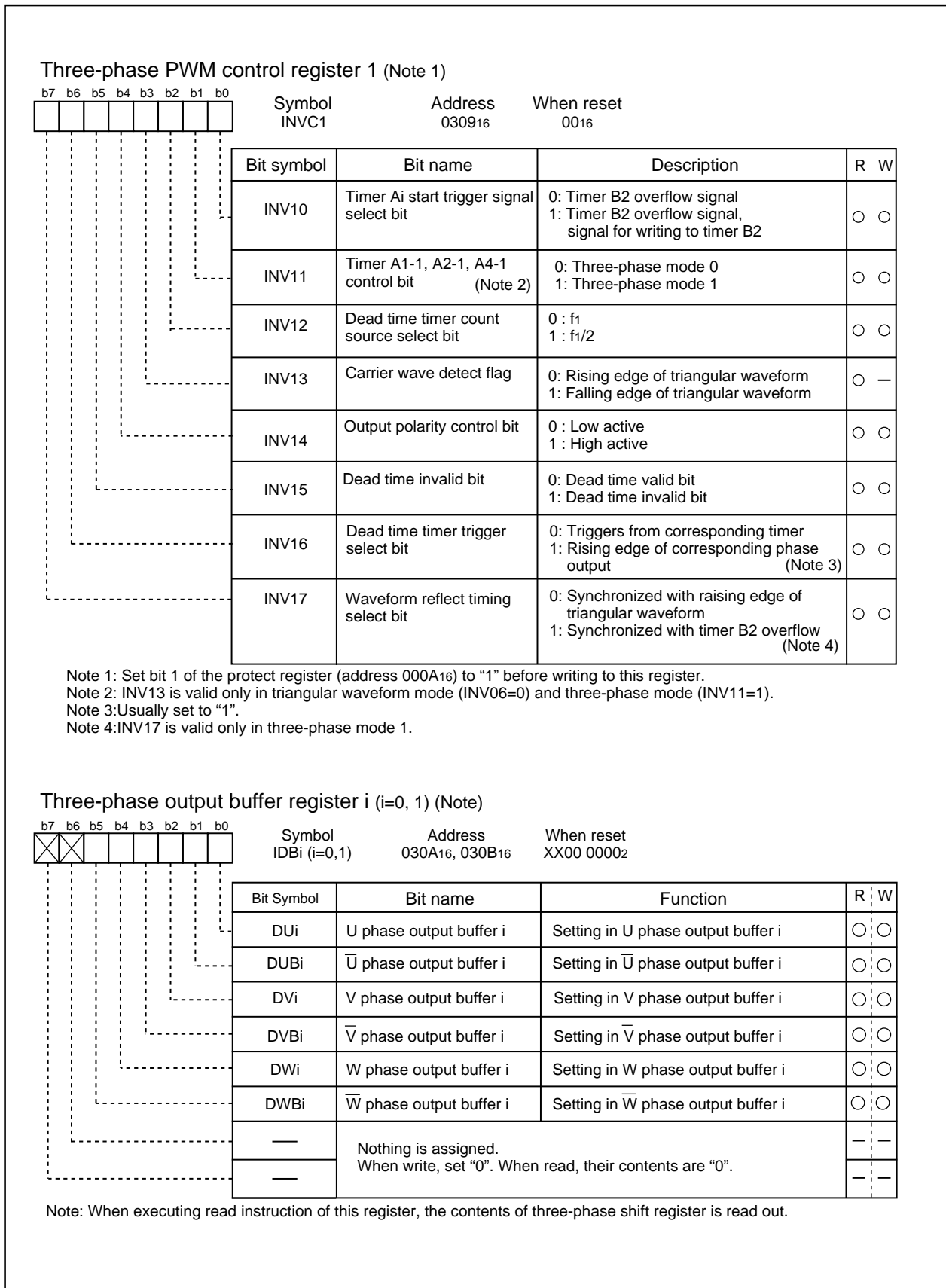


Figure 1.16.2. Registers related to timers for three-phase motor control

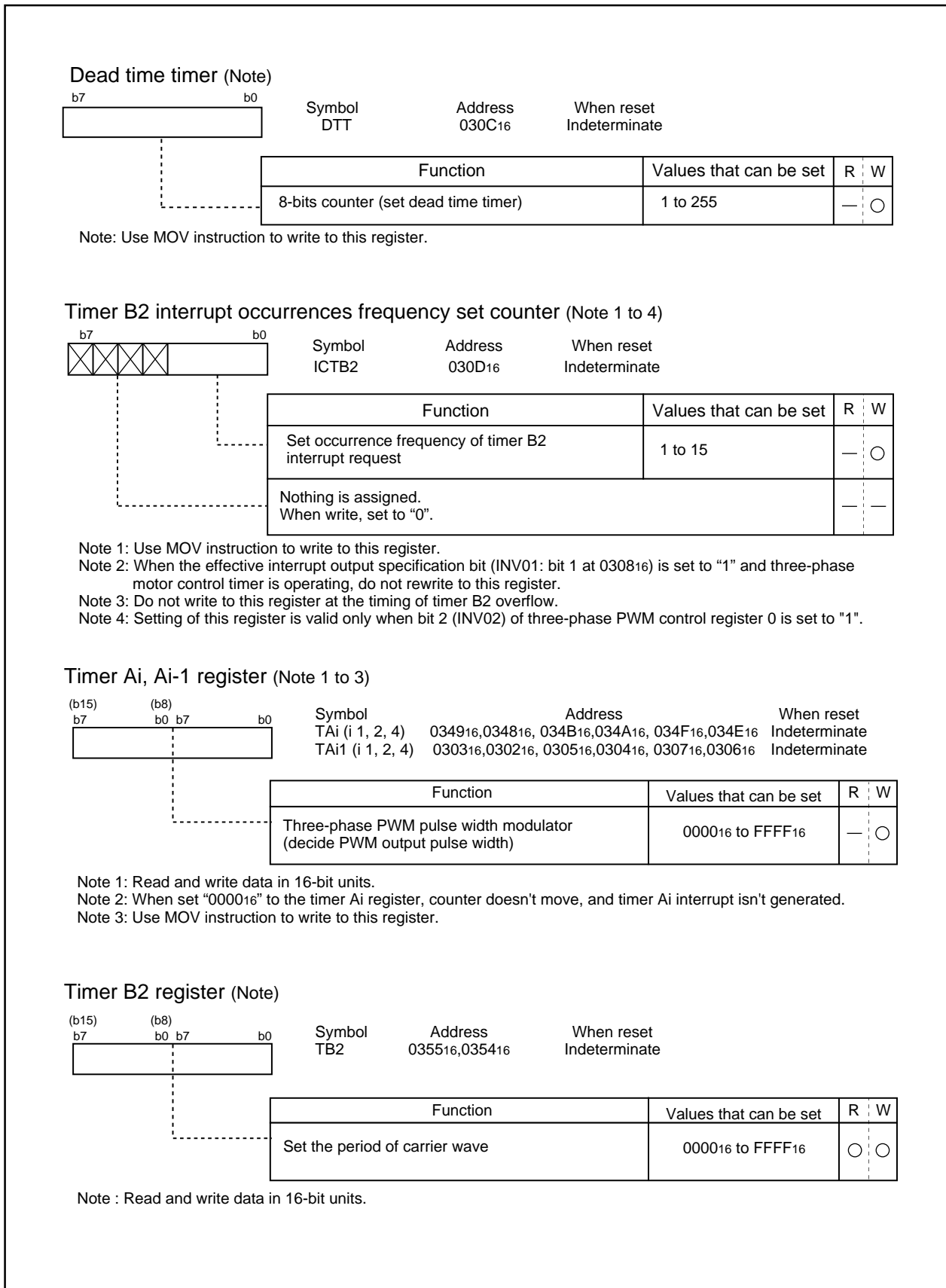


Figure 1.16.3. Registers related to timers for three-phase motor control

Three-phase motor control timers' functions

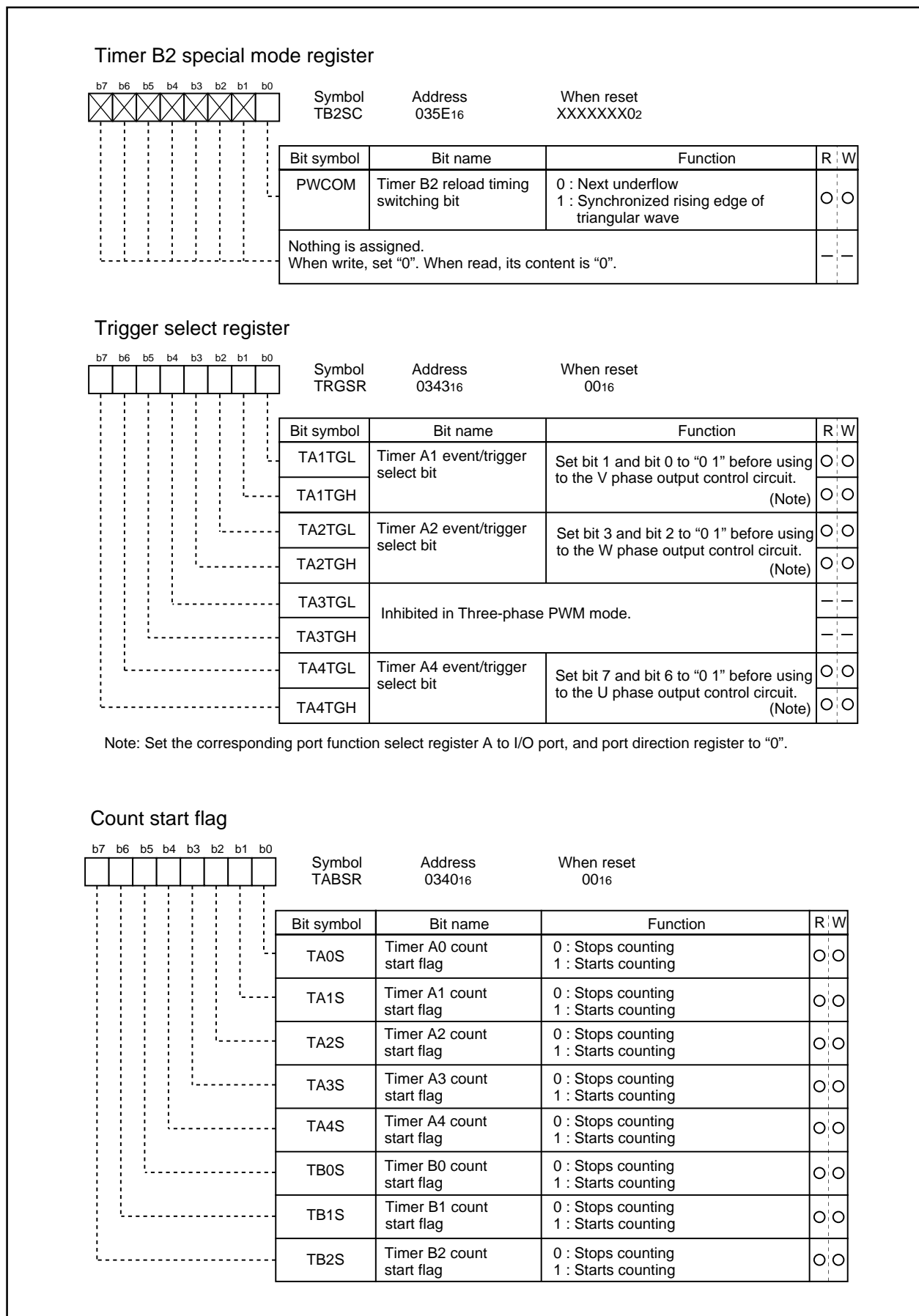


Figure 1.16.4. Registers related to timers for three-phase motor control

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

Setting "1" in the mode select bit (bit 2 at 030816) shown in Figure 1.16.1 causes three-phase PWM output mode that uses four timers A1, A2, A4, and B2. As shown in Figure 1.16.4 and 1.16.5 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.

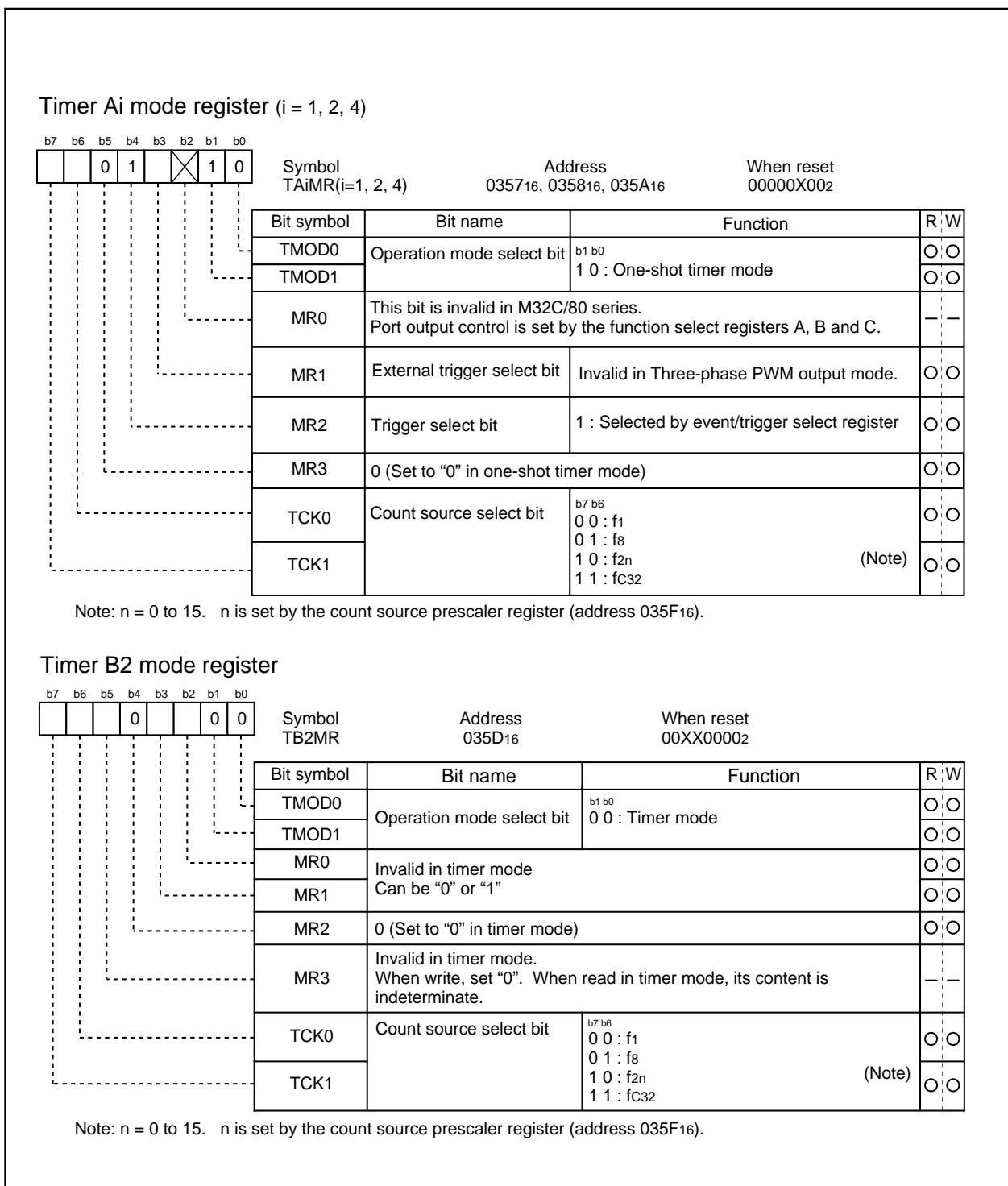


Figure 1.16.5. Timer mode registers in three-phase PWM output mode

Figure 1.16.6 shows the block diagram for three-phase waveform mode. The Low active output polarity in three-phase waveform mode, the positive-phase waveforms (U phase, V phase, and W phase) and negative waveforms (\bar{U} phase, \bar{V} phase, and \bar{W} phase), six waveforms in total, are output from P80, P81, P72, P73, P74, and P75 as active on the "L" level. Of the timers used in this mode, timer A4 controls the U phase and \bar{U} phase, timer A1 controls the V phase and \bar{V} phase, and timer A2 controls the W phase and \bar{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 (\bar{U} phase, \bar{V} phase, and \bar{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 (030C₁₆), 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 0309₁₆). 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 00₁₆, and waits for the next trigger to come.

The positive waveforms (U phase, V phase, and W phase) and the negative waveforms (\bar{U} phase, \bar{V} phase, and \bar{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 0308₁₆). 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 \bar{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 0308₁₆) causes one of the pairs of U phase and \bar{U} phase, V phase and \bar{V} phase, and W phase and \bar{W} phase concurrently go to "L", as a result, the output control bit becomes the high-impedance state.

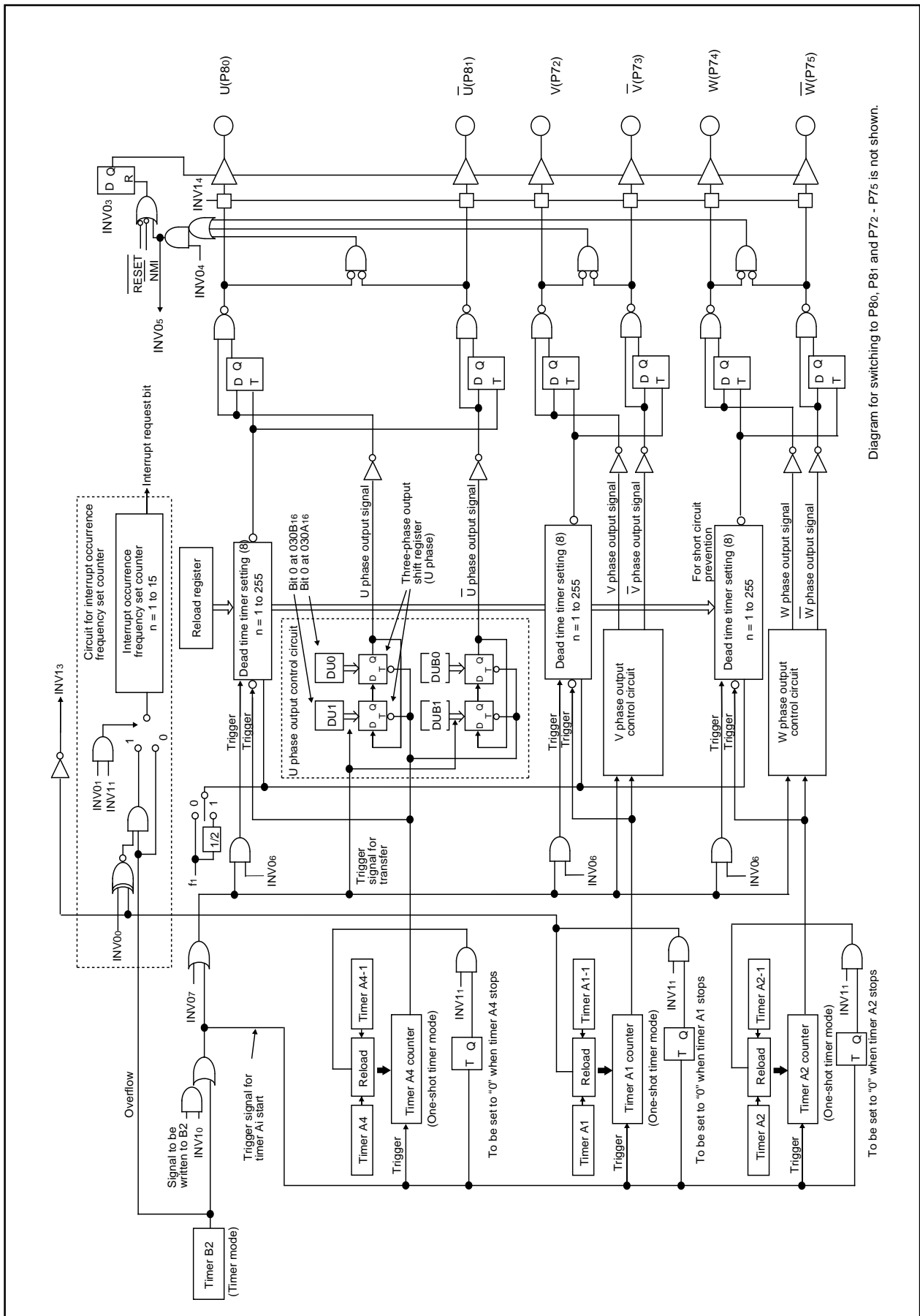


Diagram for switching to P80, P81 and P72 - P75 is not shown.

Figure 1.16.6. 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 0308₁₆). Also, set "1" in the timers A4-1, A1-1, A2-1 control bit (bit 1 at 0309₁₆). 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 0000₁₆. If "0" is set to the effective interrupt output specification bit (bit 1 at 0308₁₆), the frequency of interrupt requests that occur every time the timer B2 counter's value becomes 0000₁₆ can be set by use of the timer B2 counter (030D₁₆) for setting the frequency of interrupt occurrences. The frequency of occurrences is given by (setting; setting \neq 0).

Setting "1" in the effective interrupt output specification bit (bit 1 at 0308₁₆) 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 0308₁₆).

An example of U phase waveform is shown in Figure 1.16.7, and the description of waveform output workings is given below. Set "1" in DU0 (bit 0 at 030A₁₆). And set "0" in DUB0 (bit 1 at 030A₁₆). In addition, set "0" in DU1 (bit 0 at 030B₁₆) and set "1" in DUB1 (bit 1 at 030B₁₆). Also, set "0" in the effective interrupt output specification bit (bit 1 at 0308₁₆) 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 0000₁₆ as many as (setting) times. Furthermore, set "1" in the effective interrupt output specification bit (bit 1 at 0308₁₆), set in the effective interrupt polarity select bit (bit 0 at 0308₁₆) and set "1" in the interrupt occurrence frequency set counter (030D₁₆). 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 0000₁₆, timer A4 starts outputting one-shot pulses. In this instance, the content of DU1 (bit 0 at 030B₁₆) and that of DU0 (bit 0 at 030A₁₆) are set in the three-phase output shift register (U phase), the content of DUB1 (bit 1 at 030B₁₆) and that of DUB0 (bit 1 at 030A₁₆) are set in the three-phase shift register (\bar{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 0000₁₆.

The value of DU0 and that of DUB0 are output to the U terminal (P80) and to the \bar{U} terminal (P81) respectively. When the timer A4 counter counts the value written to timer A4 (034F₁₆, 034E₁₆) 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 \bar{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 overlap the Low level of the \bar{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 active, and the U phase waveform changes to the Low level. When the timer B2 counter's content becomes 0000₁₆, the timer A4 counter starts counting the value written to timer A4-1 (0307₁₆, 0306₁₆), 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 \bar{V} and \bar{W} phases, the latter are of opposite phase of the former, have the corresponding timers work similarly to dealing with the U and \bar{U} phases to generate an intended waveform.

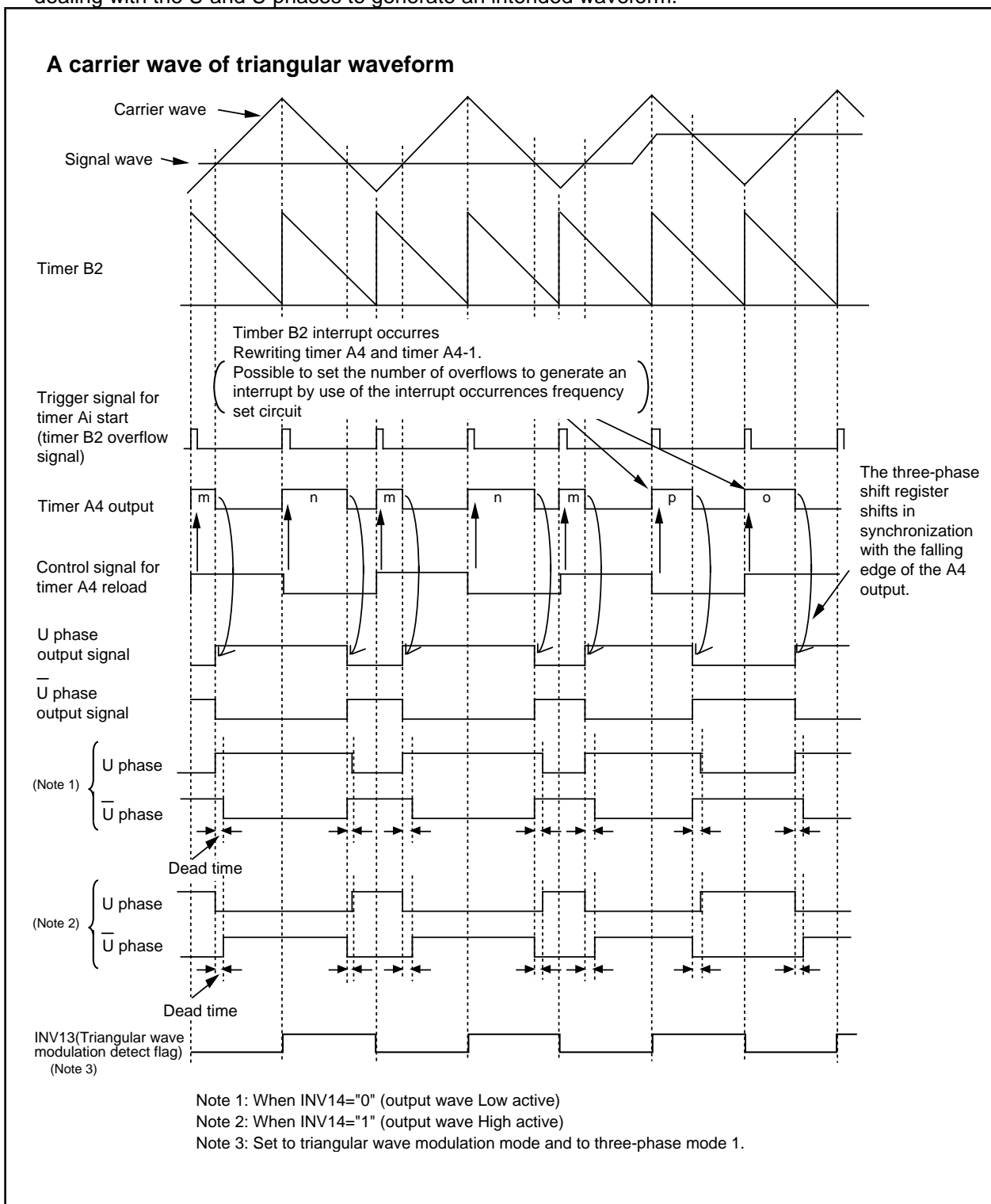


Figure 1.16.7. Timing chart of operation (1)

Assigning certain values to DU0 (bit 0 at 030A16) and DUB0 (bit 1 at 030A16), and to DU1 (bit 0 at 030B16) and DUB1 (bit 1 at 030B16) allows you to output the waveforms as shown in Figure 1.16.8, that is, to output the U phase alone, to fix \bar{U} phase to "H", to fix the U phase to "H," or to output the \bar{U} phase alone.

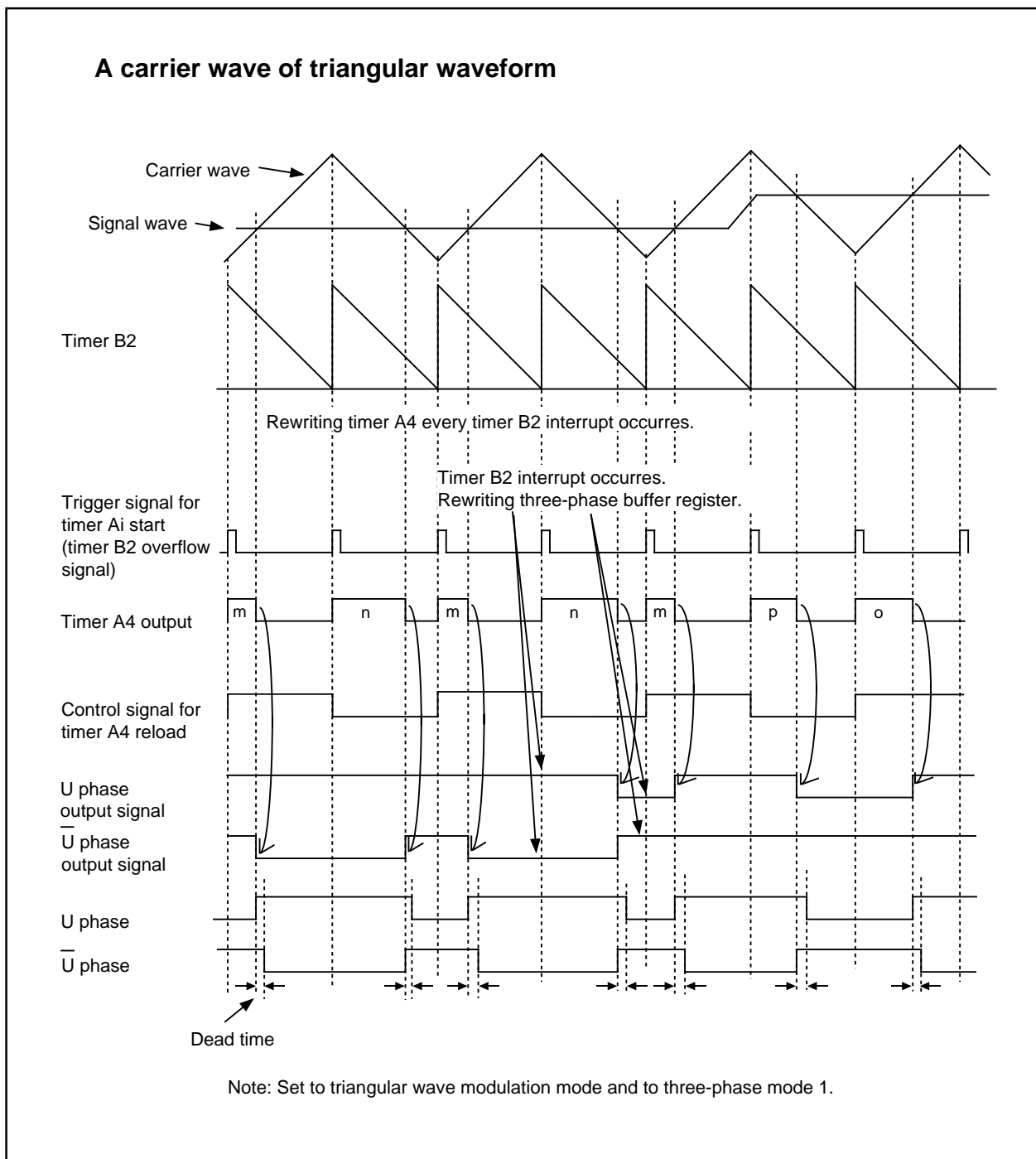


Figure 1.16.8. Timing chart of operation (2)

Sawtooth modulation

To generate a PWM waveform of sawtooth wave modulation, set "1" in the modulation mode select bit (bit 6 at 0308₁₆). Also, set "0" in the timers A4, A1, and A2-1 control bit (bit 1 at 0309₁₆). 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 0000₁₆. The effective interrupt output specification bit (bit 1 at 0308₁₆) and the effective interrupt output polarity select bit (bit 0 at 0308₁₆) go nullified.

An example of U phase waveform is shown in Figure 1.16.9, and the description of waveform output workings is given below. Set "1" in DU0 (bit 0 at 030A₁₆), and set "0" in DUB0 (bit 1 at 030A₁₆). In addition, set "0" in DU1 (bit 0 at 030B₁₆) and set "1" in DUB1 (bit 1 at 030B₁₆).

When the timer B2 counter's content becomes 0000₁₆, 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 0000₁₆.

The value of DU0 and that of DUB0 are output to the U terminal (P80) and to the \bar{U} terminal (P81) respectively. When the timer A4 counter counts the value written to timer A4 (034F₁₆, 034E₁₆) 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 \bar{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 \bar{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 0000₁₆, 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 (\bar{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 \bar{U} phase side is used, the workings in generating a \bar{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 can also be adjusted by varying the values of timer B2 and timer A4. In dealing with the V and W phases, and \bar{V} and \bar{W} phases, the latter are of opposite phase of the former, have the corresponding timers work similarly to dealing with the U and \bar{U} phases to generate an intended waveform.

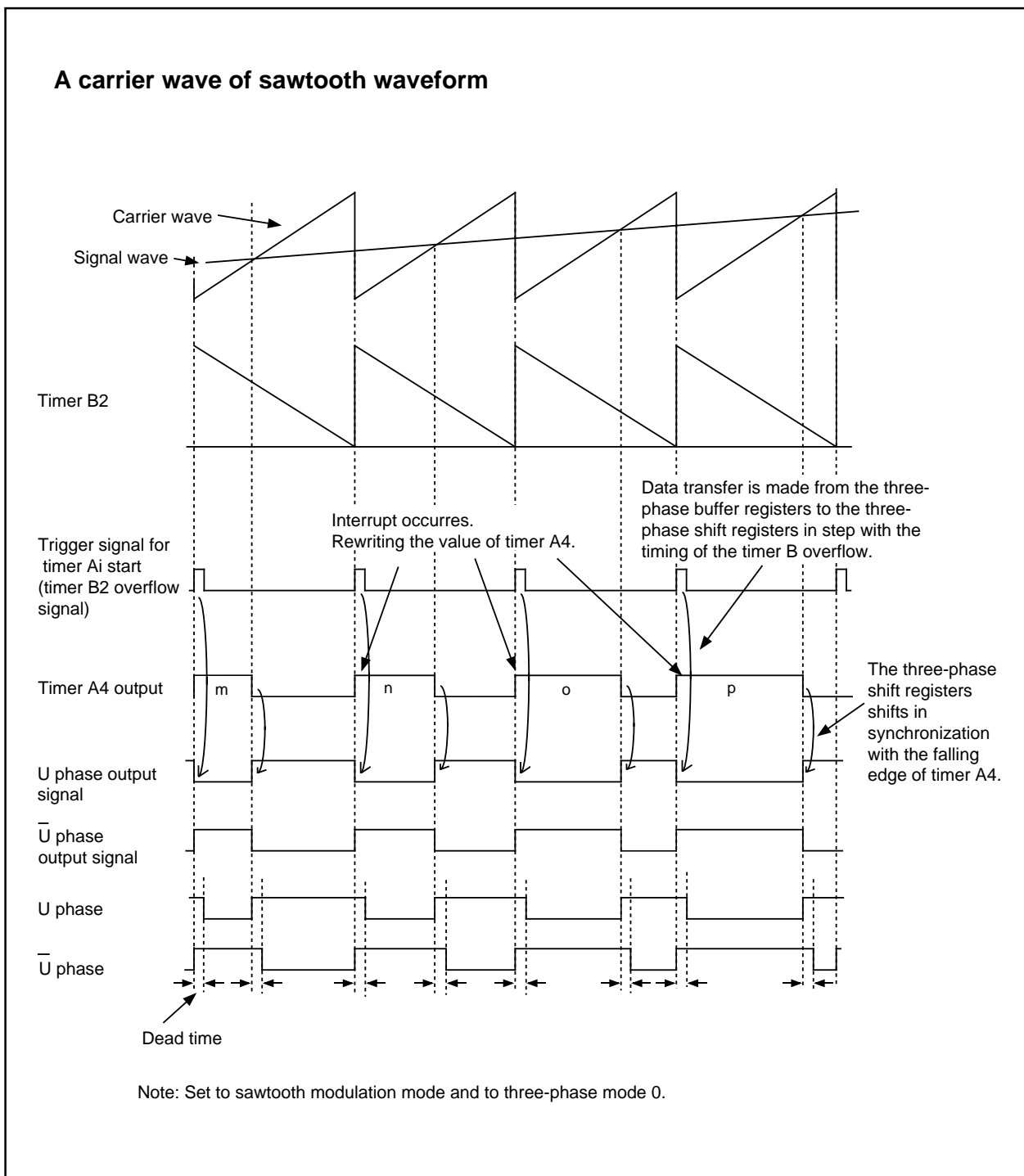


Figure 1.16.9. Timing chart of operation (3)

Setting "1" both in DUB0 and in DUB1 provides a means to output the U phase alone and to fix the \bar{U} phase output to "H" as shown in Figure 1.16.10.

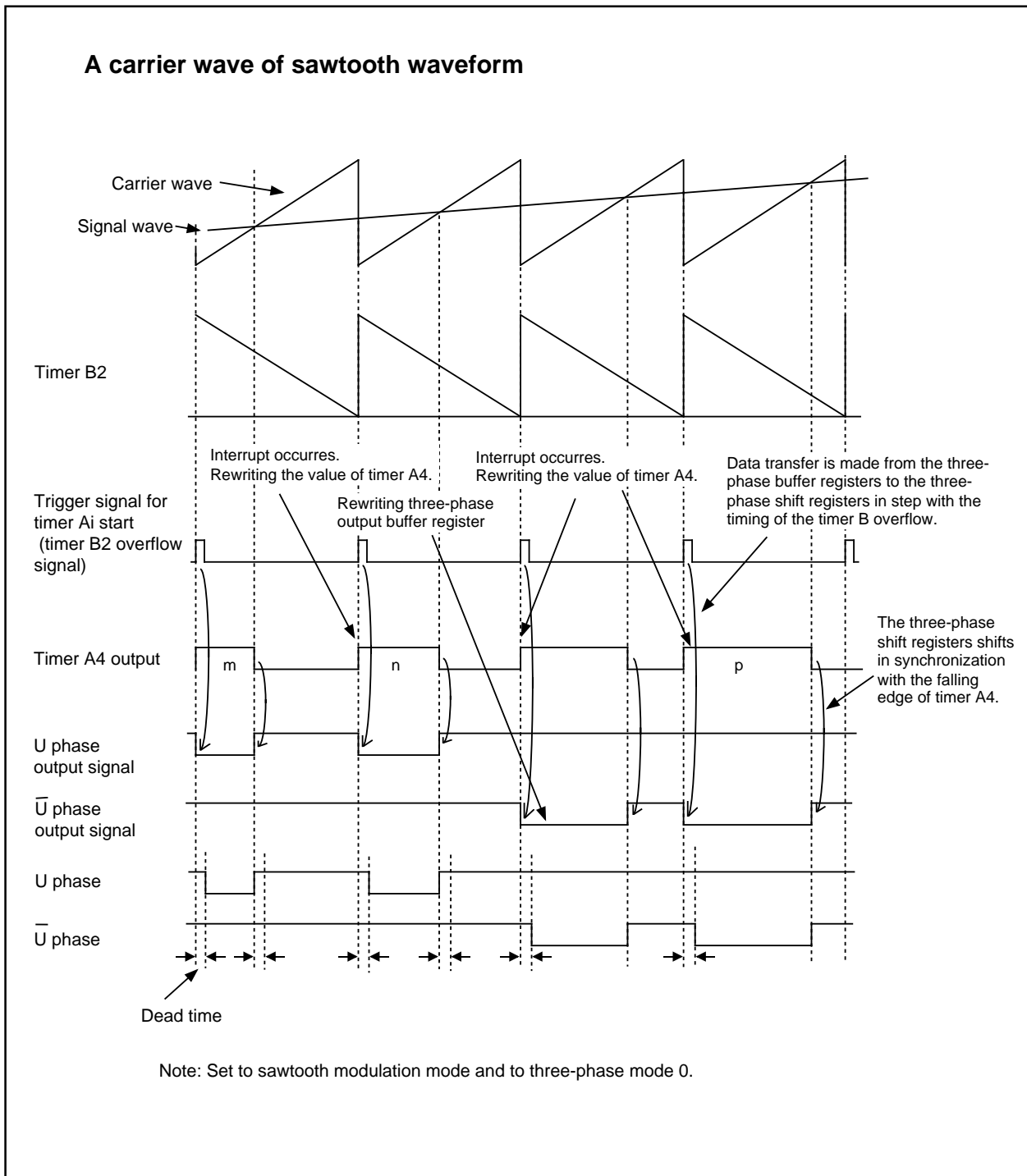


Figure 1.16.10. Timing chart of operation (4)

Serial I/O

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

UARTi (i=0 to 4) each have an exclusive timer to generate a transfer clock, so they operate independently of each other.

Figure 1.17.1 shows the block diagram of UARTi.

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 0368₁₆, 02E8₁₆, 0338₁₆, 0328₁₆ and 02F8₁₆) determine whether UARTi is used as a clock synchronous serial I/O or as a UART.

It has the bus collision detection function that generates an interrupt request if the TxD pin and the RxD pin are different in level.

Figures 1.17.2 through 1.17.8 show the registers related to UARTi.

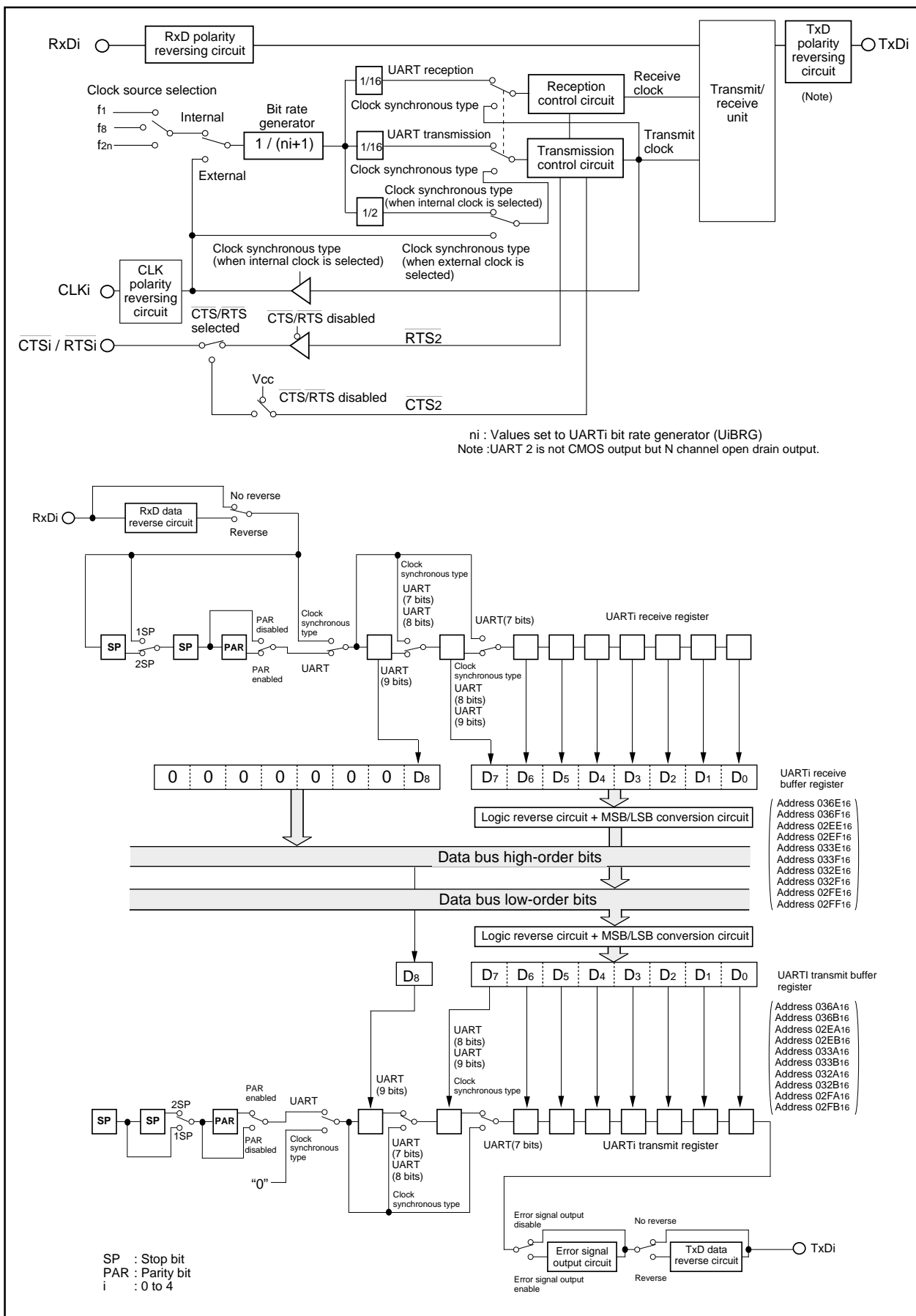
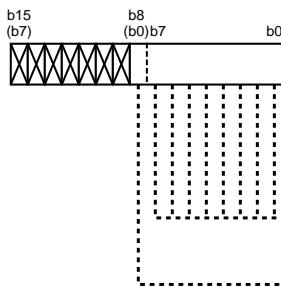


Figure 1.17.1. Block diagram of UARTi

UARTi transmit buffer register (i=0 to 4) (Note)

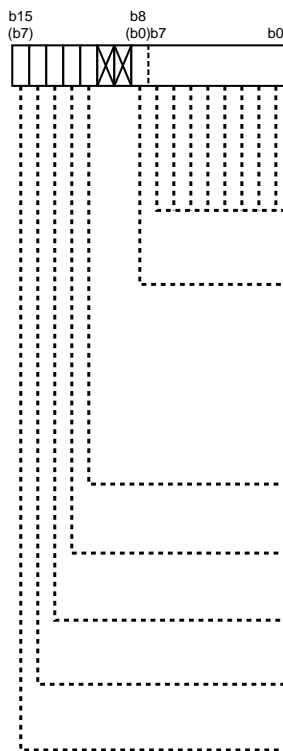


Symbol	Address	When reset
UiTB(i=0,1,2)	036B ₁₆ , 036A ₁₆ , 02EB ₁₆ , 02EA ₁₆ , 033B ₁₆ , 033A ₁₆	Indeterminate
UiTB(i=3,4)	032B ₁₆ , 032A ₁₆ , 02FB ₁₆ , 02FA ₁₆	Indeterminate

Bit symbol	Function (Clock synchronous serial I/O mode)	Function (UART mode)	R	W
—	Transmit data	Transmit data	—	○
—	—	Transmit data (9th bit)	—	○
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 (i = 0 to 4)



Symbol	Address	When reset
UiRB(i=0,1,2)	036F ₁₆ , 036E ₁₆ , 02EF ₁₆ , 02EE ₁₆ , 033F ₁₆ , 033E ₁₆	Indeterminate
UiRB(i=3,4)	032F ₁₆ , 032E ₁₆ , 02FF ₁₆ , 02FE ₁₆	Indeterminate

Bit symbol	Bit name	Function (Clock synchronous serial I/O mode)	Function (UART mode)	R	W
—	—	Receive data	Receive data	○	—
—	—	—	Receive data(9th bit)	○	—
Nothing is assigned. When write, set "0". When read, their contents are indeterminate.			—	—	—
ABT	Arbitration lost detecting flag (Note 1)	0: Not detected 1: Detected	Invalid	○	○
OER	Overrun error flag (Note 2)	0: No overrun error 1: Overrun error found	0: No overrun error 1: Overrun error found	○	—
FER	Framing error flag (Note 2)	Invalid	0: No framing error 1: Framing error found	○	—
PER	Parity error flag (Note 2)	Invalid	0: No parity error 1: Parity error found	○	—
SUM	Error sum flag (Note 2)	Invalid	0: No error 1: Error found	○	—

Note 1: Arbitration lost detecting flag must always write "0".

Note 2: Bits 15 through 12 are set to 000₂ when the serial I/O mode select bit (bits 2 to 0 at addresses 0368₁₆, 02E8₁₆, 0338₁₆, 0328₁₆, 02F8₁₆) are set to "000₂" 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 036E₁₆, 02EE₁₆, 033E₁₆, 032E₁₆, 02FE₁₆) is read.

Figure 1.17.2. Serial I/O-related registers (1)

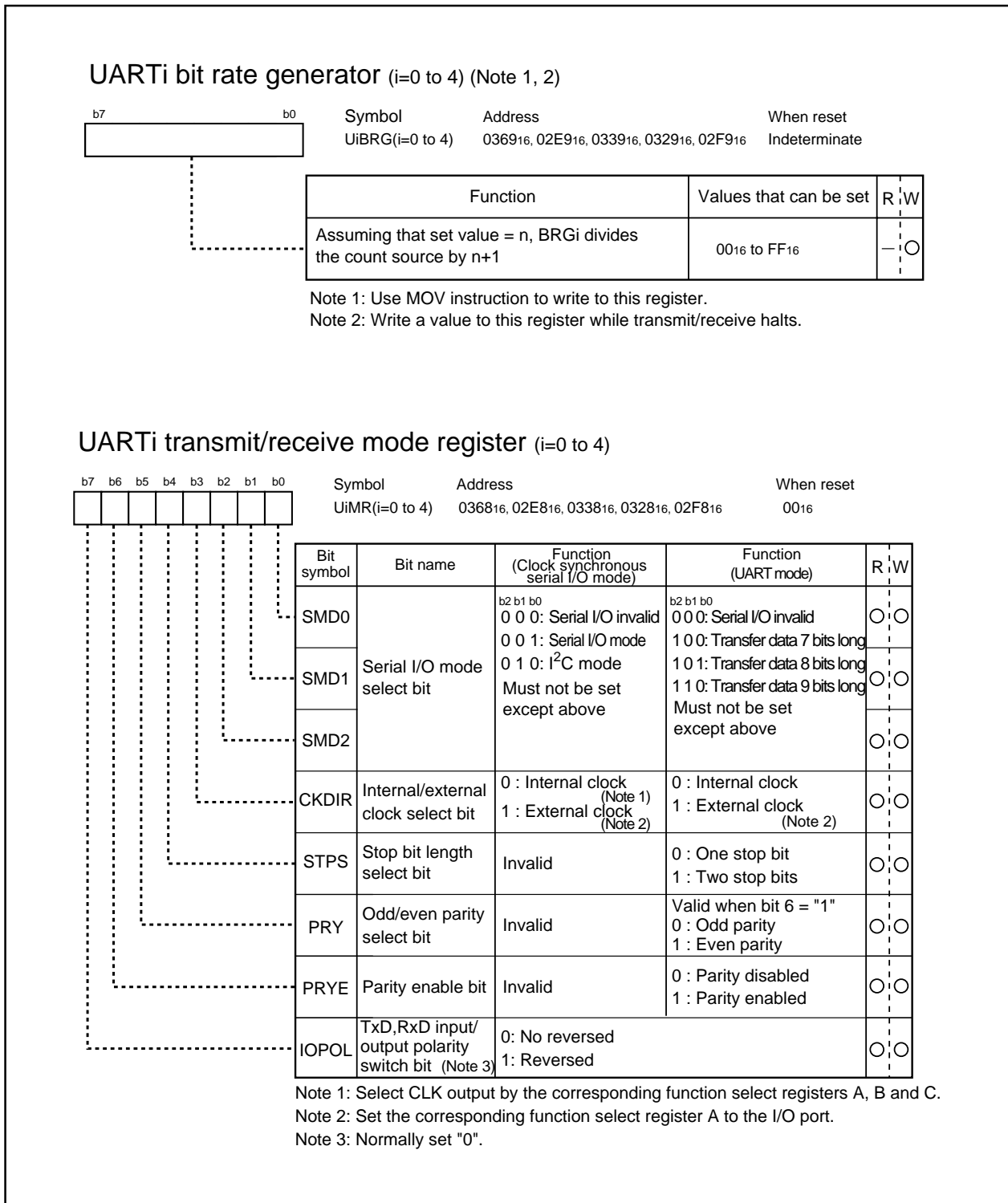


Figure 1.17.3. Serial I/O-related registers (2)

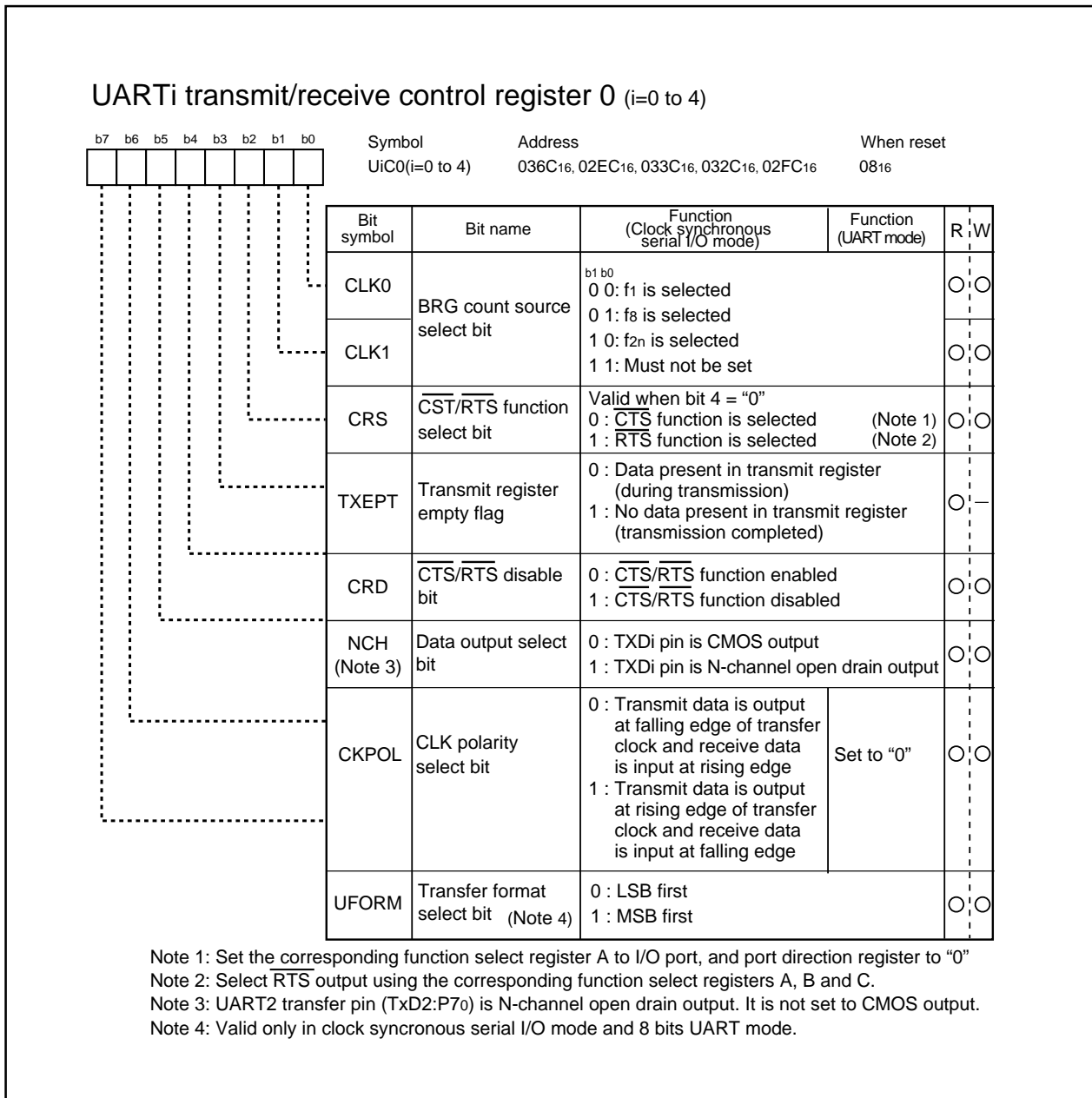


Figure 1.17.4. Serial I/O-related registers (3)

UARTi transmit/receive control register 1 (i=0 to 4)

Bit	Symbol	Address	When reset
b7	TE	036D ₁₆	021 ₆
b6	TI	02ED ₁₆	
b5	RE	033D ₁₆	
b4	RI	032D ₁₆	
b3	UiIRS	02FD ₁₆	
b2	UiRRM		
b1	UiLCH		
b0	SCLKSTPB / UIERE		

Bit symbol	Bit name	Function (Clock synchronous serial I/O mode)	Function (UART mode)	R	W
TE	Transmit enable bit	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		○	—
RE	Receive enable bit	0: Reception disabled 1: Reception enabled		○	○
RI	Receive complete flag	0: Data present in receive buffer register 1: No 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)		○	○
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		○	○
SCLKSTPB / UIERE	Clock divide synchronizing stop bit / error signal output enable bit	Clock divide synchronizing stop bit 0: Synchronizing stop 1: Synchronous start (Note)	Set to "0"	○	○

Note :When this bit and bit 7 of UARTi special mode register 2 are set, clock synchronizing function is used.

UARTi special mode register (i=0 to 4)

Bit	Symbol	Address	When reset
b7	IICM	0367 ₁₆	001 ₆
b6	ABC	02E7 ₁₆	
b5	BBS	0337 ₁₆	
b4	LSYN	0327 ₁₆	
b3	ABSCS	02F7 ₁₆	
b2	ACSE		
b1	SSS		
b0	SCLKDIV		

Bit symbol	Bit name	Function (Clock synchronous serial I/O mode)	Function (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"	○	○ (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 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	○	○
SSS	Transmit start condition select bit	Set to "0"	0: Ordinary 1: Falling edge of RxDi	○	○
SCLKDIV	Clock divide set bit	0: Divided-by-2 (Note 3) 1: No divided	Set to "0"	○	○

Note 1: Nothing but "0" may be written.

Note 2: UART0: timer A3 underflow signal, UART1: timer A4 underflow signal, UART2: timer A0 underflow signal, UART3: timer A3 underflow signal, UART4: timer A4 underflow signal.

Note 3: When this bit and bit 7 of UARTi transmit/receive control register 1 are set, clock synchronizing function is used.

Figure 1.17.5. Serial I/O-related registers (4)

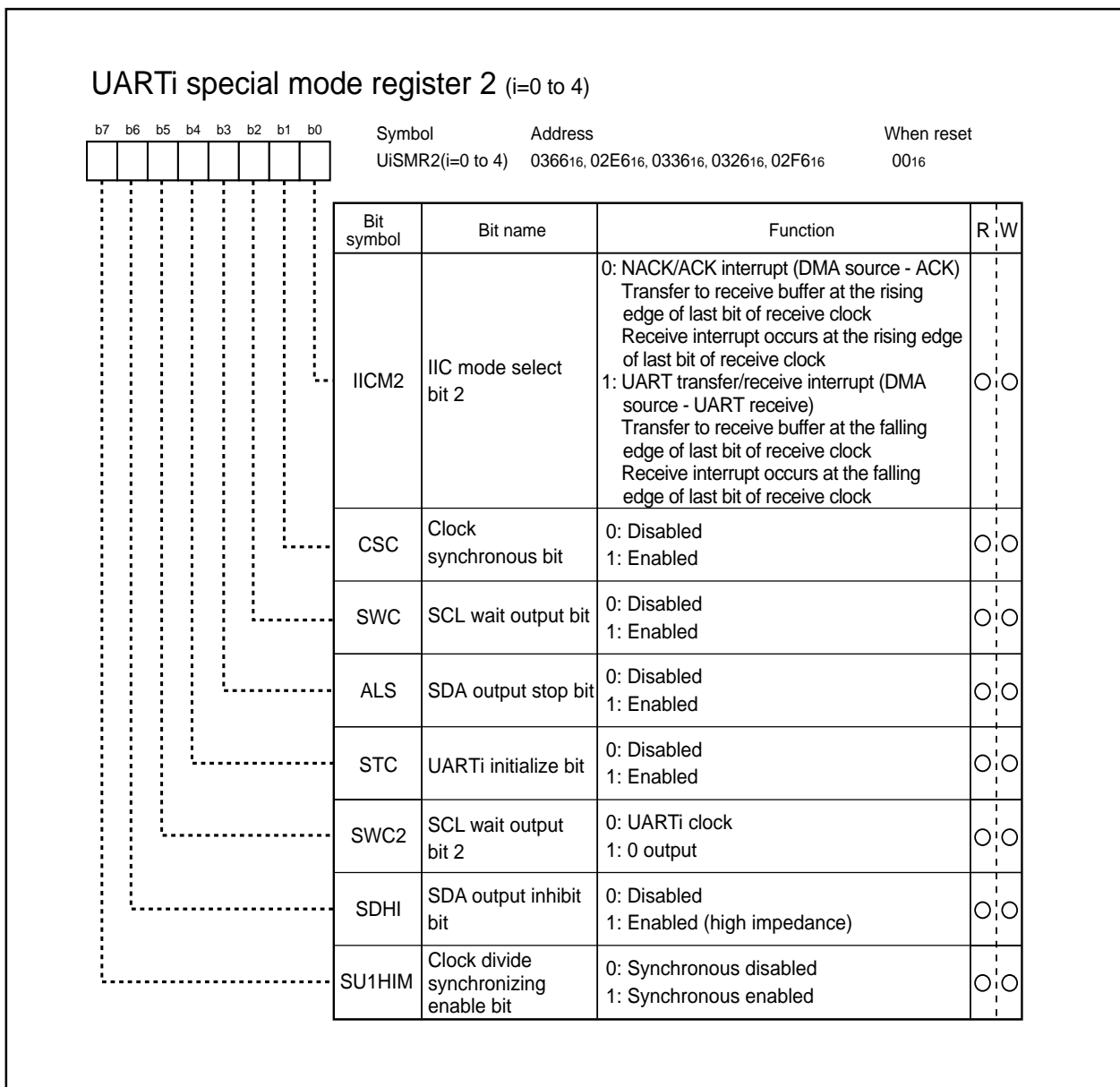


Figure 1.17.6. Serial I/O-related registers (5)

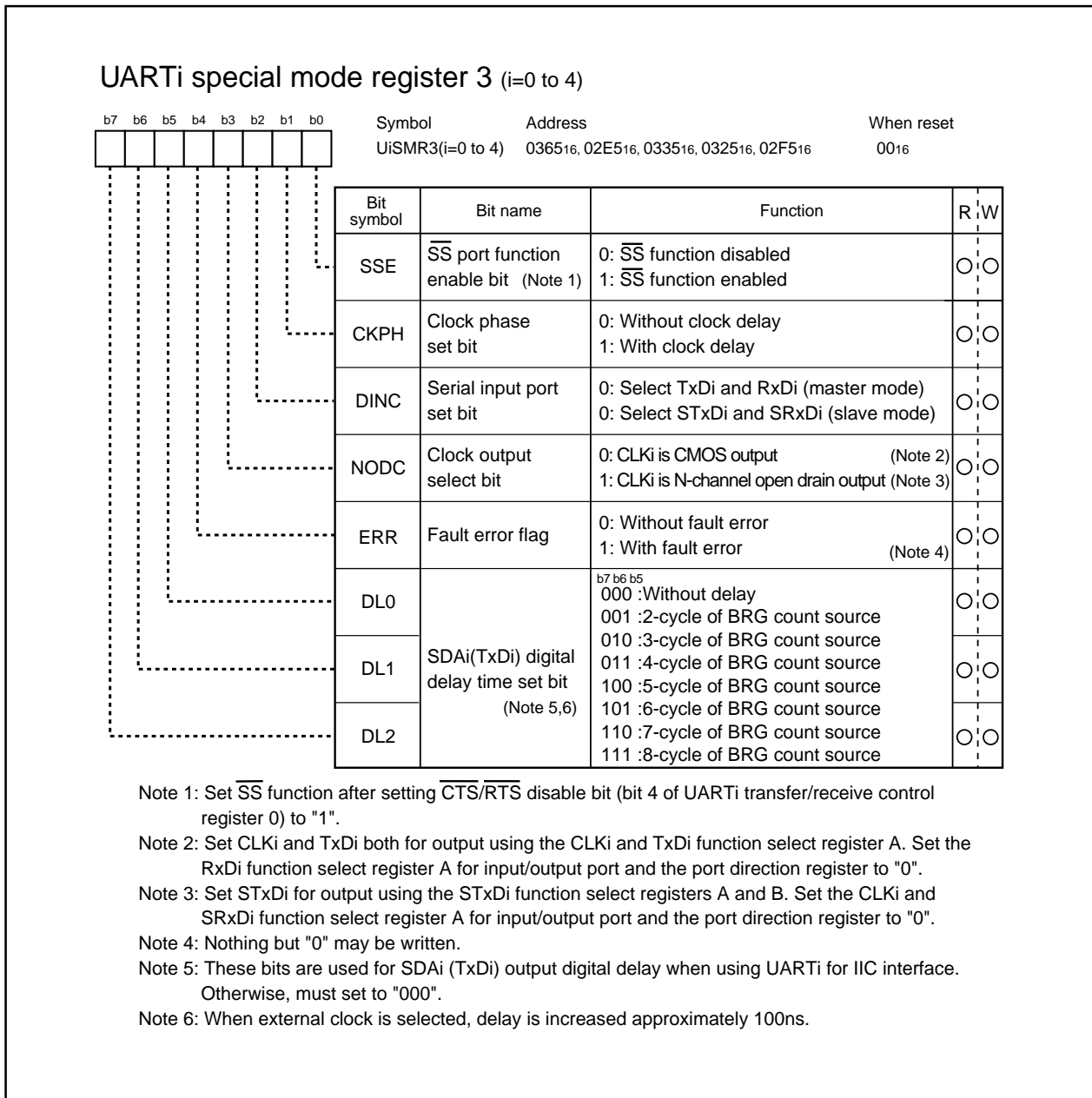


Figure 1.17.7. Serial I/O-related registers (6)

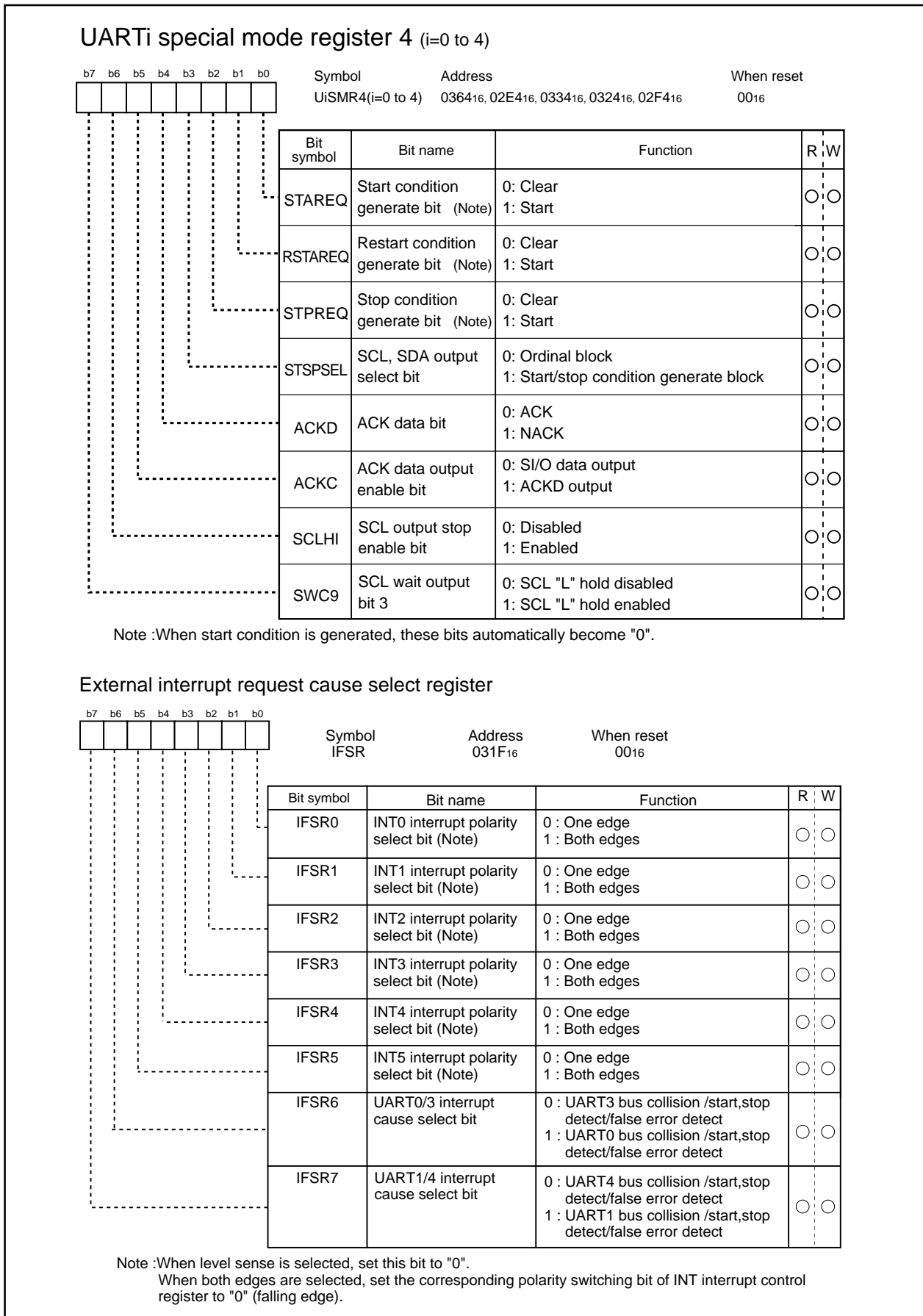


Figure 1.17.8. Serial I/O-related registers (7)

(1) Clock synchronous serial I/O mode

The clock synchronous serial I/O mode uses a transfer clock to transmit and receive data. Tables 1.18.1 and 1.18.2 list the specifications of the clock synchronous serial I/O mode.

Table 1.18.1. Specifications of clock synchronous serial I/O mode (1/2)

Item	Specification
Transfer data format	<ul style="list-style-type: none"> • Transfer data length: 8 bits
Transfer clock	<ul style="list-style-type: none"> • When internal clock is selected (bit 3 at addresses 0368₁₆, 02E8₁₆, 0338₁₆, 0328₁₆, 02F8₁₆ = "0") : $f_i / 2^{(m+1)}$ (Note 1) $f_i = f_1, f_8, f_{2n}$ (Note 2) <ul style="list-style-type: none"> – CLK is selected by the corresponding peripheral function select register A, B and C. • When external clock is selected (bit 3 at addresses 0368₁₆, 02E8₁₆, 0338₁₆, 0328₁₆, 02F8₁₆ = "1") : Input from CLKi pin <ul style="list-style-type: none"> – Set the corresponding function select register A to I/O port
Transmission/reception control	<ul style="list-style-type: none"> • $\overline{\text{CTS}}$ function/RTS function/$\overline{\text{CTS}}$, RTS function chosen to be invalid
Transmission start condition	<ul style="list-style-type: none"> • To start transmission, the following requirements must be met: <ul style="list-style-type: none"> – Transmit enable bit (bit 0 at addresses 036D₁₆, 02ED₁₆, 033D₁₆, 032D₁₆, 02FD₁₆) = "1" – Transmit buffer empty flag (bit 1 at addresses 036D₁₆, 02ED₁₆, 033D₁₆, 032D₁₆, 02FD₁₆) = "0" – When $\overline{\text{CTS}}$ function selected, $\overline{\text{CTS}}$ input level = "L" – TxD output is selected by the corresponding peripheral function select register A, B and C. • Furthermore, if external clock is selected, the following requirements must also be met: <ul style="list-style-type: none"> – CLKi polarity select bit (bit 6 at addresses 036C₁₆, 02EC₁₆, 033C₁₆, 032C₁₆, 02FC₁₆) = "0": CLKi input level = "H" – CLKi polarity select bit (bit 6 at addresses 036C₁₆, 02EC₁₆, 033C₁₆, 032C₁₆, 02FC₁₆) = "1": CLKi input level = "L"
Reception start condition	<ul style="list-style-type: none"> • To start reception, the following requirements must be met: <ul style="list-style-type: none"> – Receive enable bit (bit 2 at addresses 036D₁₆, 02ED₁₆, 033D₁₆, 032D₁₆, 02FD₁₆) = "1" – Transmit enable bit (bit 0 at addresses 036D₁₆, 02ED₁₆, 033D₁₆, 032D₁₆, 02FD₁₆) = "1" – Transmit buffer empty flag (bit 1 at addresses 036D₁₆, 02ED₁₆, 033D₁₆, 032D₁₆, 02FD₁₆) = "0" • Furthermore, if external clock is selected, the following requirements must also be met: <ul style="list-style-type: none"> – CLKi polarity select bit (bit 6 at addresses 036C₁₆, 02EC₁₆, 033C₁₆, 032C₁₆, 02FC₁₆) = "0": CLKi input level = "H" – CLKi polarity select bit (bit 6 at addresses 036C₁₆, 02EC₁₆, 033C₁₆, 032C₁₆, 02FC₁₆) = "1": CLKi input level = "L"
Interrupt request generation timing	<ul style="list-style-type: none"> • When transmitting <ul style="list-style-type: none"> – Transmit interrupt cause select bit (bit 4 at address 036D₁₆, 02ED₁₆, 033D₁₆, 032D₁₆, 02FD₁₆) = "0": Interrupts requested when data transfer from UARTi transfer buffer register to UARTi transmit register is completed – Transmit interrupt cause select bit (bit 4 at address 036D₁₆, 02ED₁₆, 033D₁₆, 032D₁₆, 02FD₁₆) = "1": Interrupts requested when data transmission from UARTi transfer register is completed • When receiving <ul style="list-style-type: none"> – Interrupts requested when data transfer from UARTi receive register to UARTi receive buffer register is completed

Note 1: "m" denotes the value 00₁₆ to FF₁₆ that is set to the UART bit rate generator.

Table 1.18.2. Specifications of clock synchronous serial I/O mode (2/2)

Item	Specification
Error detection	<ul style="list-style-type: none"> • Overrun error ^(Note) This error occurs when the next data is started to receive and 6.5 transfer clock is elapsed before UARTi receive buffer register are read out.
Select function	<ul style="list-style-type: none"> • 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 • Reversing serial data logic 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 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.

Table 1.18.3 lists the functions of the input/output pins during clock synchronous serial I/O mode. 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.)

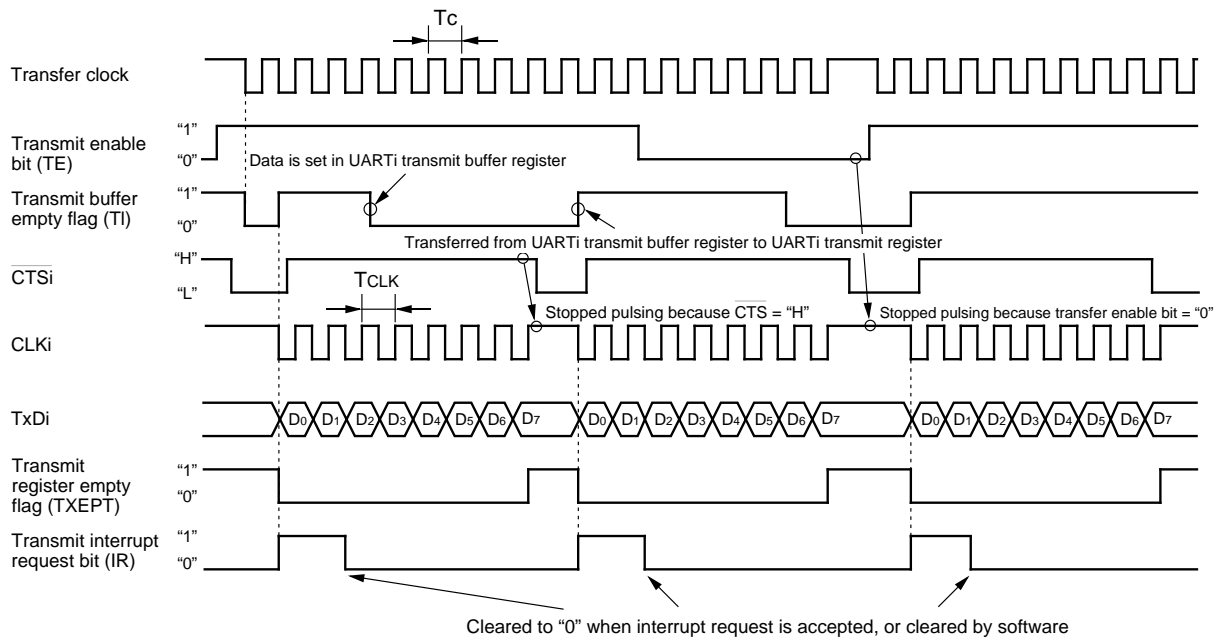
Table 1.18.3. Input/output pin functions in clock synchronous serial I/O mode

Pin name	Function	Method of selection
TxDi (P63, P67, P70, P92, P96)	Serial data output (Note 1)	(Outputs dummy data when performing reception only)
RxDi (P62, P66, P71, P91, P97)	Serial data input (Note 2)	Port P62, P66, P71, P91 and P97 direction register (bits 2 and 6 at address 03C216, bit 1 at address 03C316, bit 1 and 7 at address 03C716) = "0" (Can be used as an input port when performing transmission only)
CLKi (P61, P65, P72, P90, P95)	Transfer clock output (Note 1)	Internal/external clock select bit (bit 3 at addresses 036816, 02E816, 033816, 032816, 02F816) = "0"
	Transfer clock input (Note 2)	Internal/external clock select bit (bit 3 at addresses 036816, 02E816, 033816, 032816, 02F816) = "1" Port P61, P65, P72, P90 and P95 direction register (bits 1 and 5 at address 03C216, bit 2 at address 03C316, bit 0 and 5 at address 03C716) = "0"
CTS \bar /RTS \bar i (P60, P64, P73, P93, P94)	CTS input (Note 2)	CTS/RTS disable bit (bit 4 at addresses 036C16, 02EC16, 033C16, 032C16, 02FC16) = "0" CTS/RTS function select bit (bit 2 at addresses 036C16, 02EC16, 033C16, 032C16, 02FC16) = "0" Port P60, P64, P73, P93 and P94 direction register (bits 0 and 4 at address 03C216, bit 3 at address 03C316, bits 3 and 4 at address 03C716) = "0"
	RTS output (Note 1)	CTS/RTS disable bit (bit 4 at addresses 036C16, 02EC16, 033C16, 032C16, 02FC16) = "0" CTS/RTS function select bit (bit 2 at addresses 036C16, 02EC16, 033C16, 032C16, 02FC16) = "1"
	Programmable I/O port (Note 2)	CTS/RTS disable bit (bit 4 at addresses 036C16, 02EC16, 033C16, 032C16, 02FC16) = "1"

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)



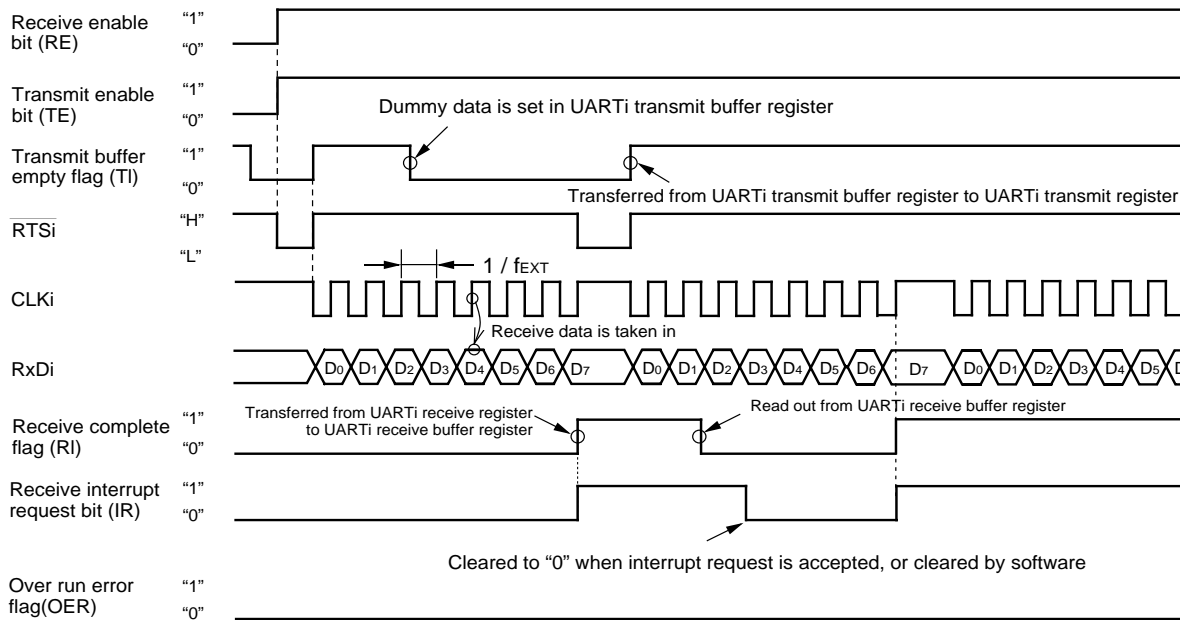
Shown in () are bit symbols.

- 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".

$$T_c = T_{CLK} = 2(m + 1) / f_i$$

f_i : frequency of BRGi count source (f_1, f_8, f_{2n})
 m : value set to BRGi

• Example of receive timing (when external clock is selected)



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".

f_{EXT} : frequency of external clock

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 1.18.1. Typical transmit/receive timings in clock synchronous serial I/O mode

(a) Polarity select function

As shown in Figure 1.18.2, the CLK polarity select bit (bit 6 at addresses 036C₁₆, 02EC₁₆, 033C₁₆, 032C₁₆, 02FC₁₆) allows selection of the polarity of the transfer clock.

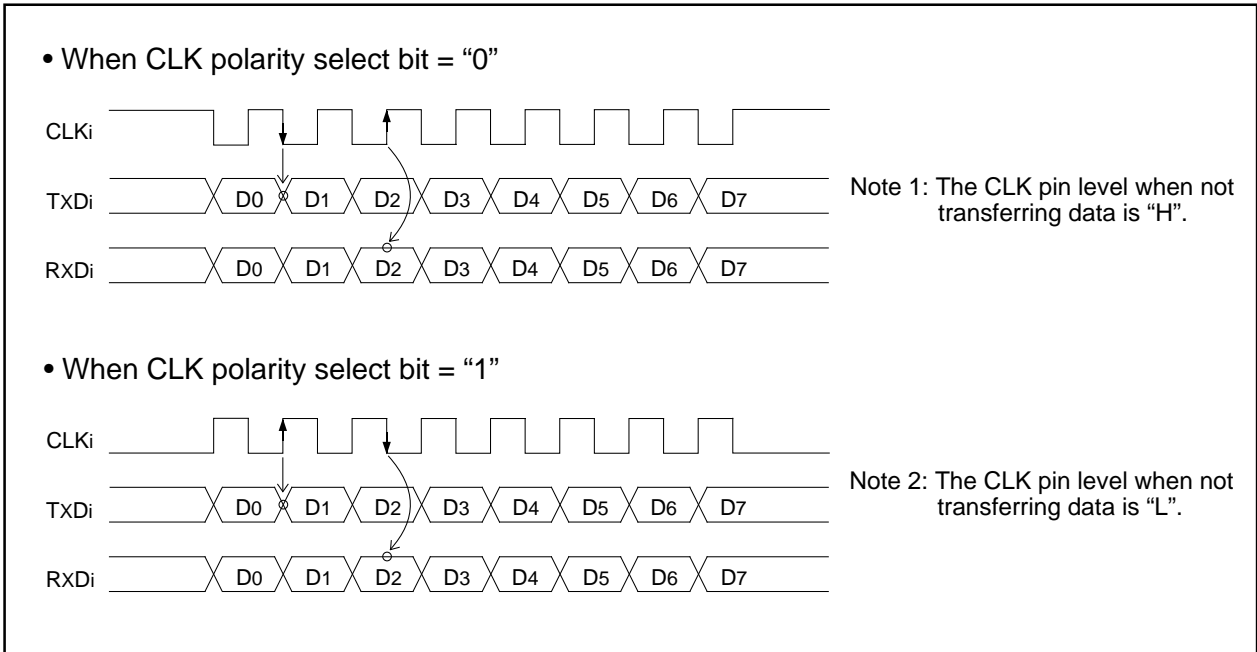


Figure 1.18.2. Polarity of transfer clock

(b) LSB first/MSB first select function

As shown in Figure 1.18.3, when the transfer format select bit (bit 7 at addresses 036C₁₆, 02EC₁₆, 033C₁₆, 032C₁₆, 02FC₁₆) = "0", the transfer format is "LSB first"; when the bit = "1", the transfer format is "MSB first".

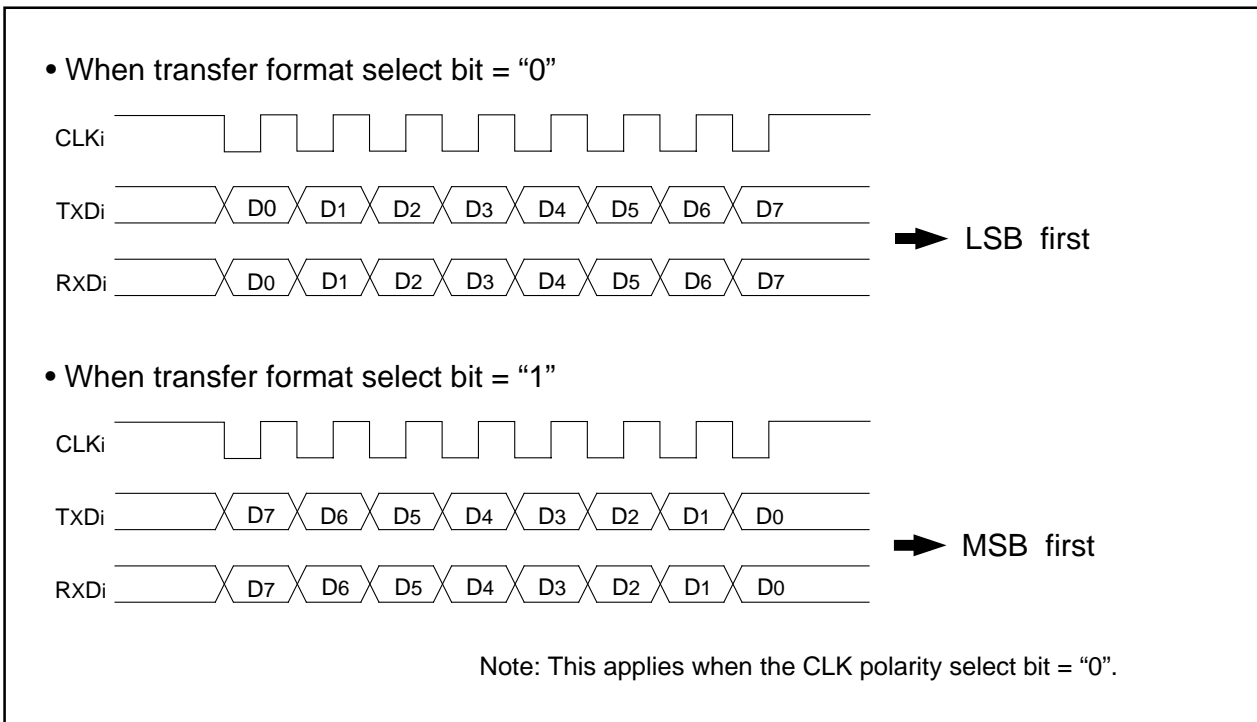


Figure 1.18.3. Transfer format

(c) Continuous receive mode

If the continuous receive mode enable bit (bit 5 at address 036D₁₆, 02ED₁₆, 033D₁₆, 032D₁₆, 02FD₁₆) 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 back to the transmit buffer register again.

(d) Serial data logic switch function

When the data logic select bit (bit6 at address 036D₁₆, 02ED₁₆, 033D₁₆, 032D₁₆, 02FD₁₆) = "1", and writing to transmit buffer register or reading from receive buffer register, data is reversed. Figure 1.18.4 shows the timing example of serial data logic switch.

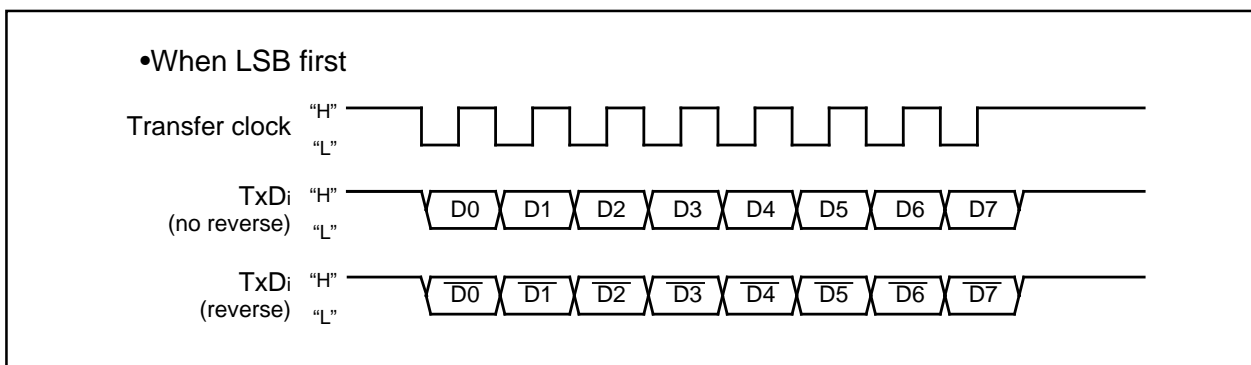


Figure 1.18.4. Timing for switching serial data logic

Clock asynchronous serial I/O (UART) mode

(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 1.19.1 and 1.19.2 list the specifications of the UART mode. Figure 1.19.1 shows the UART_i transmit/receive mode register.

Table 1.19.1. Specifications of UART Mode (1/2)

Item	Specification
Transfer data format	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • When internal clock is selected (bit 3 at addresses 0368₁₆, 02E8₁₆, 0338₁₆, 0328₁₆, 02F8₁₆ = "0") : $f_i/16(m+1)$ (Note 1) $f_i = f_1, f_8, f_{2n}$ • When external clock is selected (bit 3 at addresses 0368₁₆, 02E8₁₆, 0338₁₆, 0328₁₆, 02F8₁₆ = "1") : $f_{EXT}/16(m+1)$ (Note 1, 2)
Transmission/reception control	<ul style="list-style-type: none"> • CTS function, RTS function, CTS/RTS function chosen to be invalid
Transmission start condition	<ul style="list-style-type: none"> • To start transmission, the following requirements must be met: <ul style="list-style-type: none"> - Transmit enable bit (bit 0 at addresses 036D₁₆, 02ED₁₆, 033D₁₆, 032D₁₆, 02FD₁₆) = "1" - Transmit buffer empty flag (bit 1 at addresses 036D₁₆, 02ED₁₆, 033D₁₆, 032D₁₆, 02FD₁₆) = "0" - When CTS function selected, CTS input level = "L" - TxD output is selected by the corresponding peripheral function select register A, B and C.
Reception start condition	<ul style="list-style-type: none"> • To start reception, the following requirements must be met: <ul style="list-style-type: none"> - Receive enable bit (bit 2 at addresses 036D₁₆, 02ED₁₆, 033D₁₆, 032D₁₆, 02FD₁₆) = "1" - Start bit detection
Interrupt request generation timing	<ul style="list-style-type: none"> • When transmitting <ul style="list-style-type: none"> - Transmit interrupt cause select bits (bit 4 at address 036D₁₆, 02ED₁₆, 033D₁₆, 032D₁₆, 02FD₁₆) = "0": Interrupts requested when data transfer from UART_i transfer buffer register to UART_i transmit register is completed - Transmit interrupt cause select bits (bit 4 at address 036D₁₆, 02ED₁₆, 033D₁₆, 032D₁₆, 02FD₁₆) = "1": Interrupts requested when data transmission from UART_i transfer register is completed • When receiving <ul style="list-style-type: none"> - Interrupts requested when data transfer from UART_i receive register to UART_i receive buffer register is completed
Error detection	<ul style="list-style-type: none"> • Overrun error (Note 3) <p>This error occurs when the next data is started to receive and 6.5 transfer clock is elapsed before UART_i receive buffer register are read out.</p>

Note 1: 'm' denotes the value 00₁₆ to FF₁₆ that is set to the UART_i bit rate generator.

Note 2: f_{EXT} is input from the CLK_i pin.

Note 3: If an overrun error occurs, the UART_i receive buffer will be over written with the next data.

Clock asynchronous serial I/O (UART) mode

Table 1.19.2. Specifications of UART Mode (2/2)

Item	Specification
Error detection	<ul style="list-style-type: none"> • Framing error This error occurs when the number of stop bits set is not detected • Parity error If parity is enabled this error occurs when, 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	<ul style="list-style-type: none"> • Serial data logic switch This function reverses the logic value of transferring data. Start bit, parity bit and stop bit are not reversed. • Tx/D, Rx/D I/O polarity switch This function reverses the Tx/D port output and Rx/D port input. All I/O data level is reversed.

Table 1.19.3 lists the functions of the input/output pins in UART mode. 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 1.19.3. Input/output pin functions in UART mode

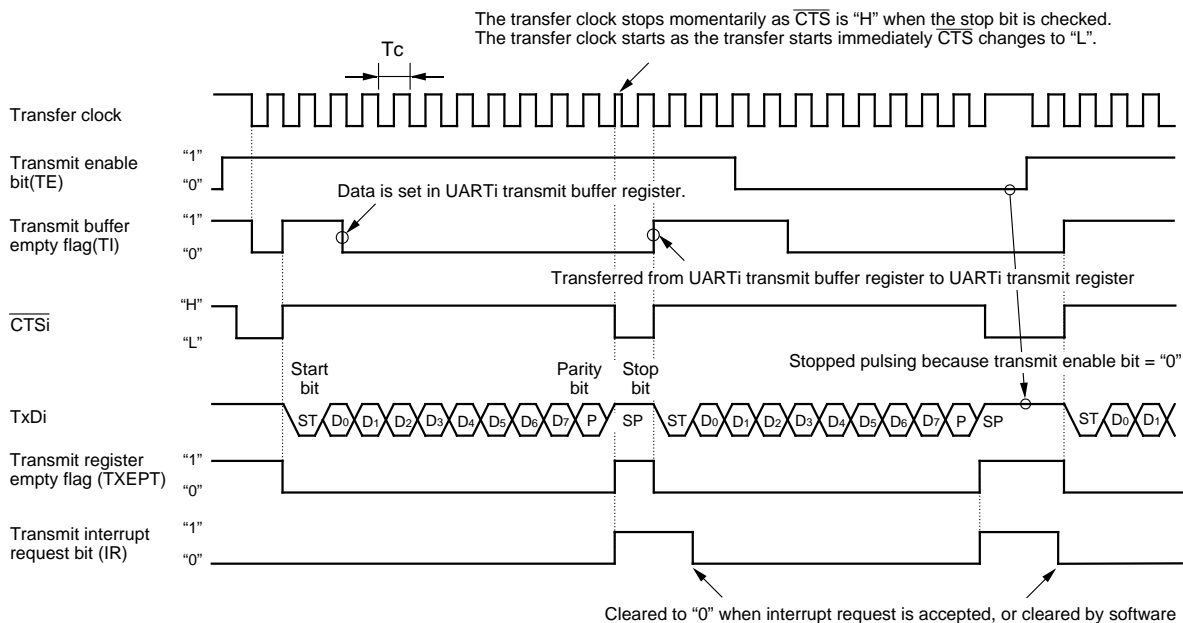
Pin name	Function	Method of selection
TxDi (P63, P67, P70, P92, P96)	Serial data output (Note 1)	
RxDi (P62, P66, P71, P91, P97)	Serial data input (Note 2)	Port P62, P66, P71, P91 and P97 direction register (bits 2 and 6 at address 03C216, bit 1 at address 03C316, bit 1 and 7 at address 03C716) = "0" (Can be used as an input port when performing transmission only)
CLKi (P61, P65, P72, P90, P95)	Programmable I/O port (Note 2)	Internal/external clock select bit (bit 3 at addresses 036816, 02E816, 033816, 032816, 02F816) = "0"
	Transfer clock input (Note 2)	Internal/external clock select bit (bit 3 at addresses 036816, 02E816, 033816, 032816, 02F816) = "1" Port P61, P65, P72, P90 and P95 direction register (bits 1 and 5 at address 03C216, bit 2 at address 03C316, bits 0 and 5 at address 03C716) = "0"
CTS \bar{i} /RTS \bar{i} (P60, P64, P73, P93, P94)	CTS input (Note 2)	CTS/RTS disable bit (bit 4 at addresses 036C16, 02EC16, 033C16, 032C16, 02FC16) = "0" CTS/RTS function select bit (bit 2 at addresses 036C16, 02EC16, 033C16, 032C16, 02FC16) = "0" Port P60, P64, P73, P93 and P94 direction register (bits 0 and 4 at address 03C216, bit 3 at address 03C316, bits 3 and 4 at address 03C716) = "0"
	RTS output (Note 1)	CTS/RTS disable bit (bit 4 at addresses 036C16, 02EC16, 033C16, 032C16, 02FC16) = "0" CTS/RTS function select bit (bit 2 at addresses 036C16, 02EC16, 033C16, 032C16, 02FC16) = "1"
	Programmable I/O port (Note 2)	CTS/RTS disable bit (bit 4 at addresses 036C16, 02EC16, 033C16, 032C16, 02FC16) = "1"

Note 1: Select Tx/D output, CLK output and 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.

Clock asynchronous serial I/O (UART) mode

• Example of transmit timing when transfer data is 8 bits long (parity enabled, one stop bit)



• Example of transmit timing when transfer data is 9 bits long (parity disabled, two stop bits)

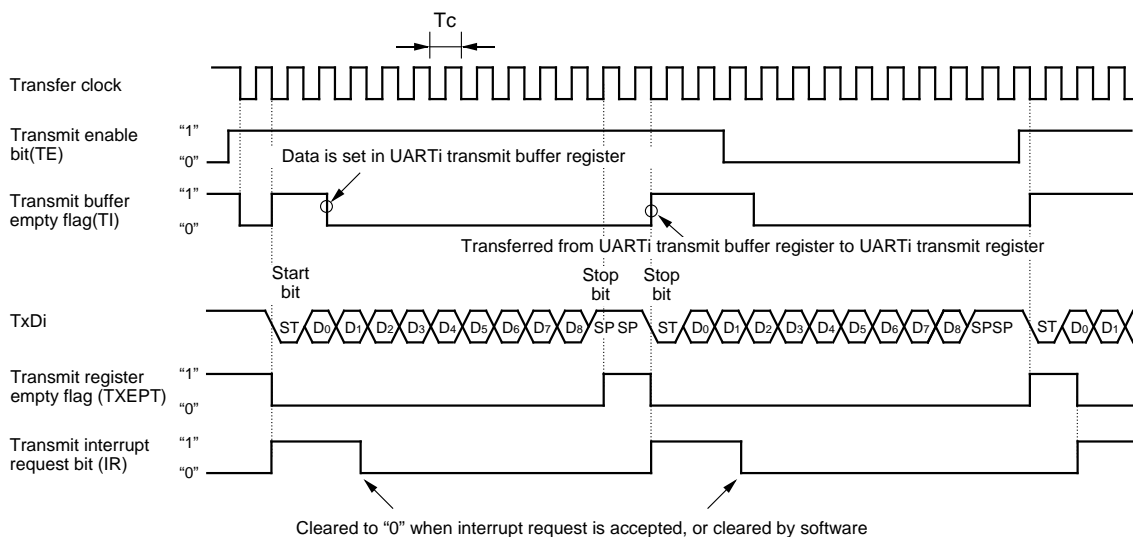


Figure 1.19.1. Typical transmit timings in UART mode

Clock asynchronous serial I/O (UART) mode

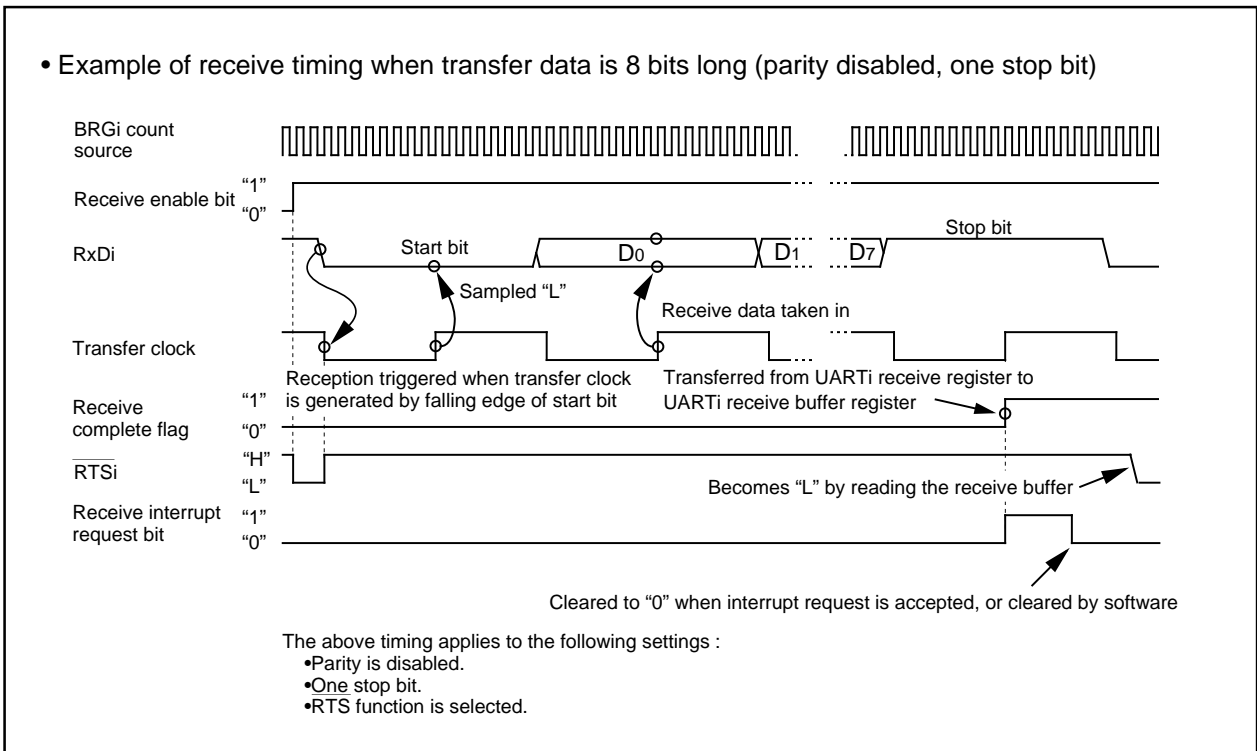


Figure 1.19.2. Typical receive timing in UART mode

(a) Function for switching serial data logic

When the data logic select bit (bit 6 of address 036D16, 02ED16, 033D16, 032D16, 02FD16) is assigned 1, data is inverted in writing to the transmission buffer register or reading the reception buffer register.

Figure 1.19.3 shows the example of timing for switching serial data logic.

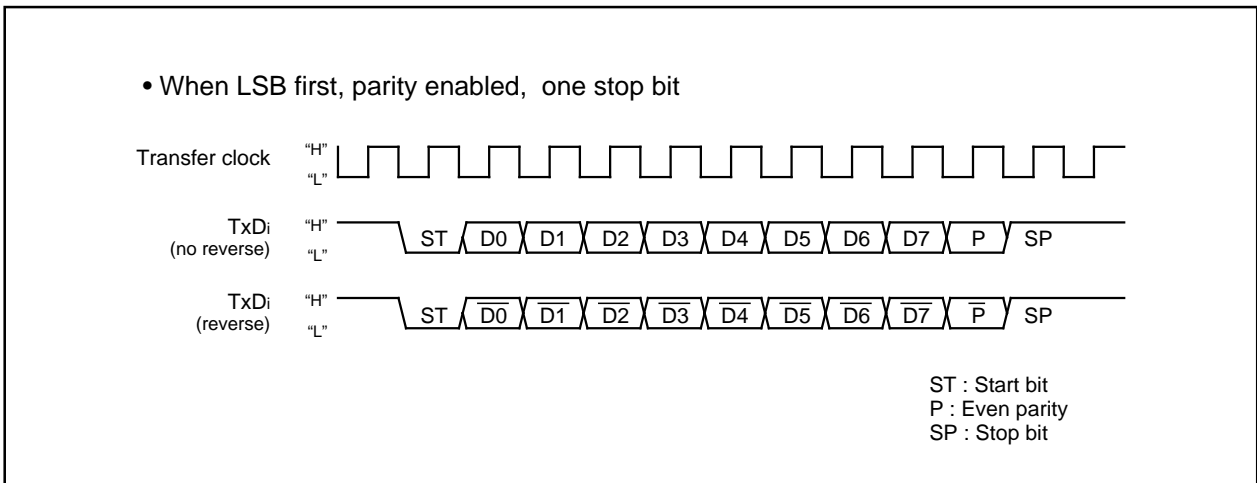


Figure 1.19.3. Timing for switching serial data logic

Clock asynchronous serial I/O (UART) mode

(b) TxD, RxD I/O polarity reverse function

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 normal use.

(c) Bus collision detection function

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 1.19.4 shows the example of detection timing of a bus collision (in UART mode).

UART0 and UART3 are allocated to software interrupt number 40. UART1 and UART4 are allocated to software interrupt number 41. When selecting UART 0, 3, 1 or 4 bus collision detect function, bit 6 or 7 of external interrupt cause select register (address 031F16) must be set.

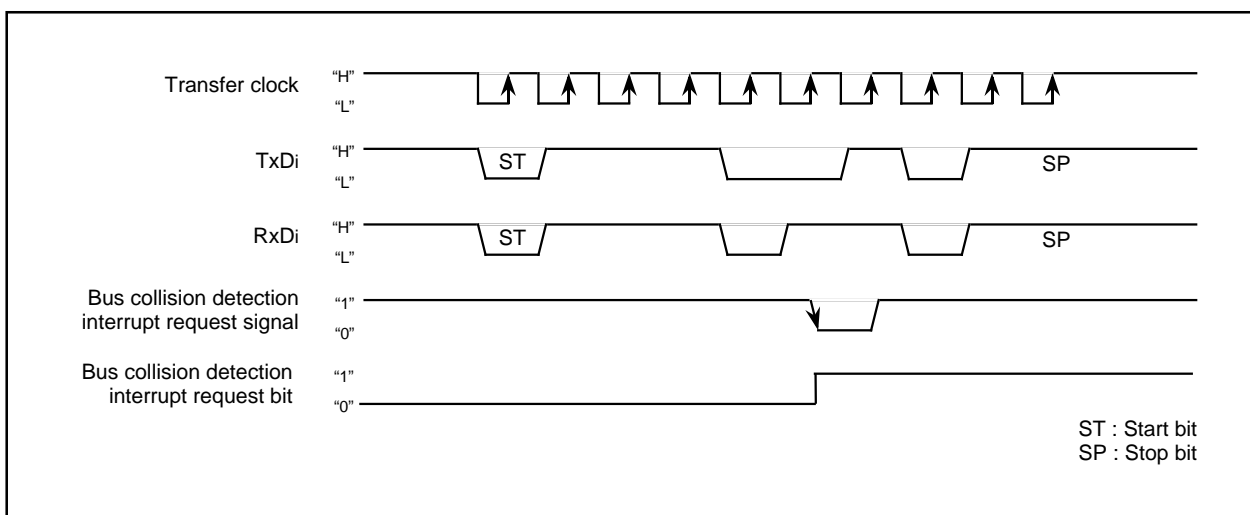


Figure 1.19.4. Detection timing of a bus collision (in UART mode)

UARTi Special Mode Register

UARTi Special Mode Register

UARTi (i=0 to 4) operate the IIC bus interface (simple IIC bus) using the UARTi special mode register (addresses 0367₁₆, 02E7₁₆, 0337₁₆, 0327₁₆ and 02F7₁₆) and UARTi special mode register 2 (addresses 0366₁₆, 02E6₁₆, 0336₁₆, 0326₁₆ and 02F6₁₆). UARTi add special functions using UARTi special mode register 3 (addresses 0365₁₆, 02E5₁₆, 0355₁₆, 0325₁₆ and 02F5₁₆).

(1) IIC Bus Interface Mode

The I²C bus interface mode is provided with UARTi.

Table 1.21.1 shows the construction of the UARTi special mode register and UARTi special mode register 2.

When the I²C mode select bit (bit 0 in addresses 0367₁₆, 02E7₁₆, 0337₁₆, 0327₁₆ and 02F7₁₆) is set to "1", the I²C bus (simple I²C bus) interface circuit is enabled.

To use the I²C bus, set the SCLi and the SDAi of both master and slave to output with the function select register. Also, set the data output select bit (bit 5 in address 036C₁₆, 02EC₁₆, 033C₁₆, 032C₁₆ and 02FC₁₆) to N-channel open drain output.

Table 1.21.1 shows the relationship of the IIC mode select bit to control. To use the chip in the clock synchronized serial I/O mode or UART mode, always set this bit to "0".

Table 1.21.1. Features in I²C mode

	Function	Normal mode (IICM=0)	I ² C mode (IICM=1) (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 17, 19, 33, 35, 37 ^(Note 2)	UARTi transmission	No acknowledgment detection (NACK)
3	Factor of interrupt number 18, 20, 34, 36, 38 ^(Note 2)	UARTi reception	Acknowledgment detection (ACK)
4	UARTi transmission output delay	Not delayed	Delayed
5	P63, P67, P70, P92, P96 at the time when UARTi is in use	TxDi (output)	SDAi (input/output)
6	P62, P66, P71, P91, P97 at the time when UARTi is in use	RxDi (input)	SCLi (input/output)
7	P61, P65, P72, P90, P95 at the time when UARTi is in use	CLKi	P61, P65, P72, P90, P95 (Note 3)
8	DMA factor at the time	UARTi reception	Acknowledgment detection (ACK)
9	Noise filter width	15ns	50ns
10	Reading P62, P66, P71, P91, P97	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 P63, P67, P70, P92, P96 when the port is selected (Note 3)

Note 1: Make the settings given below when I²C mode is used.

Set 0 1 0 in bits 2, 1, 0 of the UARTi transmission/reception mode register.

Disable the RTS/CTS function. Choose the MSB First function.

Note 2: Follow the steps given below to switch from one 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

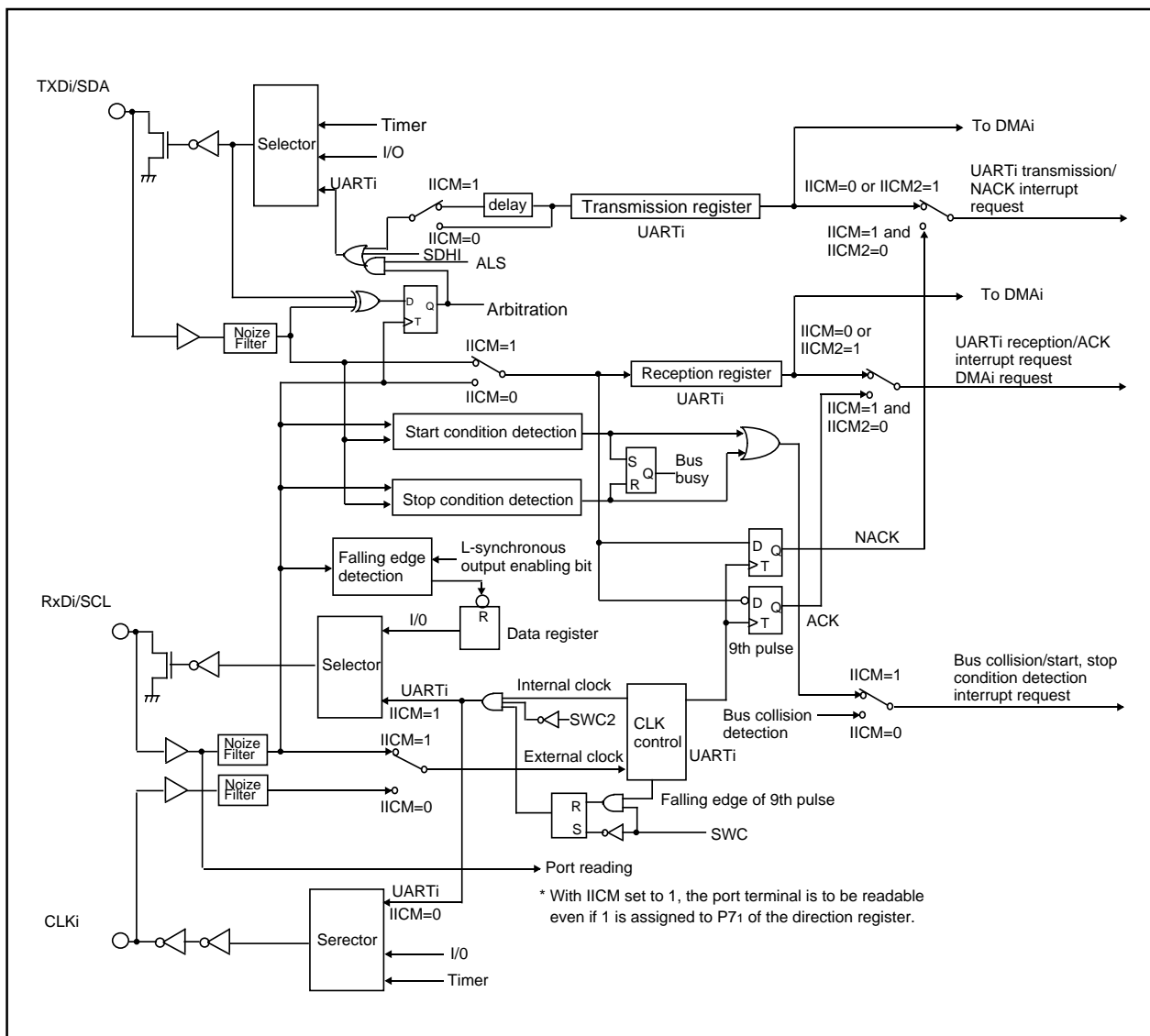


Figure 1.21.1. Functional block diagram for I²C mode

Figure 1.21.1 is a block diagram of the IIC bus interface.

The control bits of the IIC bus interface is explained as follow:

UARTi Special Mode Register (UiSMR:Addresses 0367₁₆, 02E7₁₆, 0337₁₆, 0327₁₆, 02F7₁₆)

Bit 0 is the IIC mode select bit. When set to “1”, ports operate respectively as the SDAi data transmission-reception pin, SCLi clock I/O pin and port. A delay circuit is added to SDAi transmission output, therefore after SCLi is sufficiently L level, SDAi output changes. Port (SCLi) is designed to read pin level regardless of the content of the port direction register. SDAi transmission output is initially set to port in this mode. Furthermore, interrupt factors for the bus collision detection interrupt, UARTi transmission interrupt and UARTi 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 falling edge at the SDAi pin is detected while the SCLi pin is in the H state. The stop condition detection interrupt is generated when the rising edge at the SDAi pin is detected while the SCLi pin is in the H state.

The acknowledge non-detection interrupt is generated when the H level at the SDAi pin is detected at the 9th rise of the transmission clock.

The acknowledge detection interrupt is generated when the L level at the SDAi pin is detected at the 9th rise of the transmission clock. Also, DMA transfer can be started when the acknowledge is detected and UARTi transmission is selected as the DMAi request factor.

Bit 1 is the arbitration lost detection flag control bit (ABC). Arbitration detects a conflict between data transmitted at SCLi rise and data at the SDAi pin. This detection flag is allocated to bit 11 in UARTi transmission buffer register (addresses 036F₁₆, 02EF₁₆, 033F₁₆, 032F₁₆, 02FF₁₆). 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 2 is the bus busy flag (BBS). It is set to "1" when the start condition is detected, and reset to "0" when the stop condition is detected.

Bit 3 is the SCLi L synchronization output enable bit (LSYN). When this bit is set to "1", the port data register is set to "0" in sync with the L level at the SCLi pin.

Bit 4 is the bus collision detection sampling clock select bit (ABSCS). 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 A3 with UART0, timer A4 with UART1, timer A0 with UART2, timer A3 with UART3 and timer A4 with UART4) underflows. Operation is shown in Figure 1.21.2.

Bit 5 is the transmission enable bit automatic clear select bit (ACSE). 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 (SSS). By setting this bit to "1", TxDi transmission starts in sync with the rise at the RxDi pin.

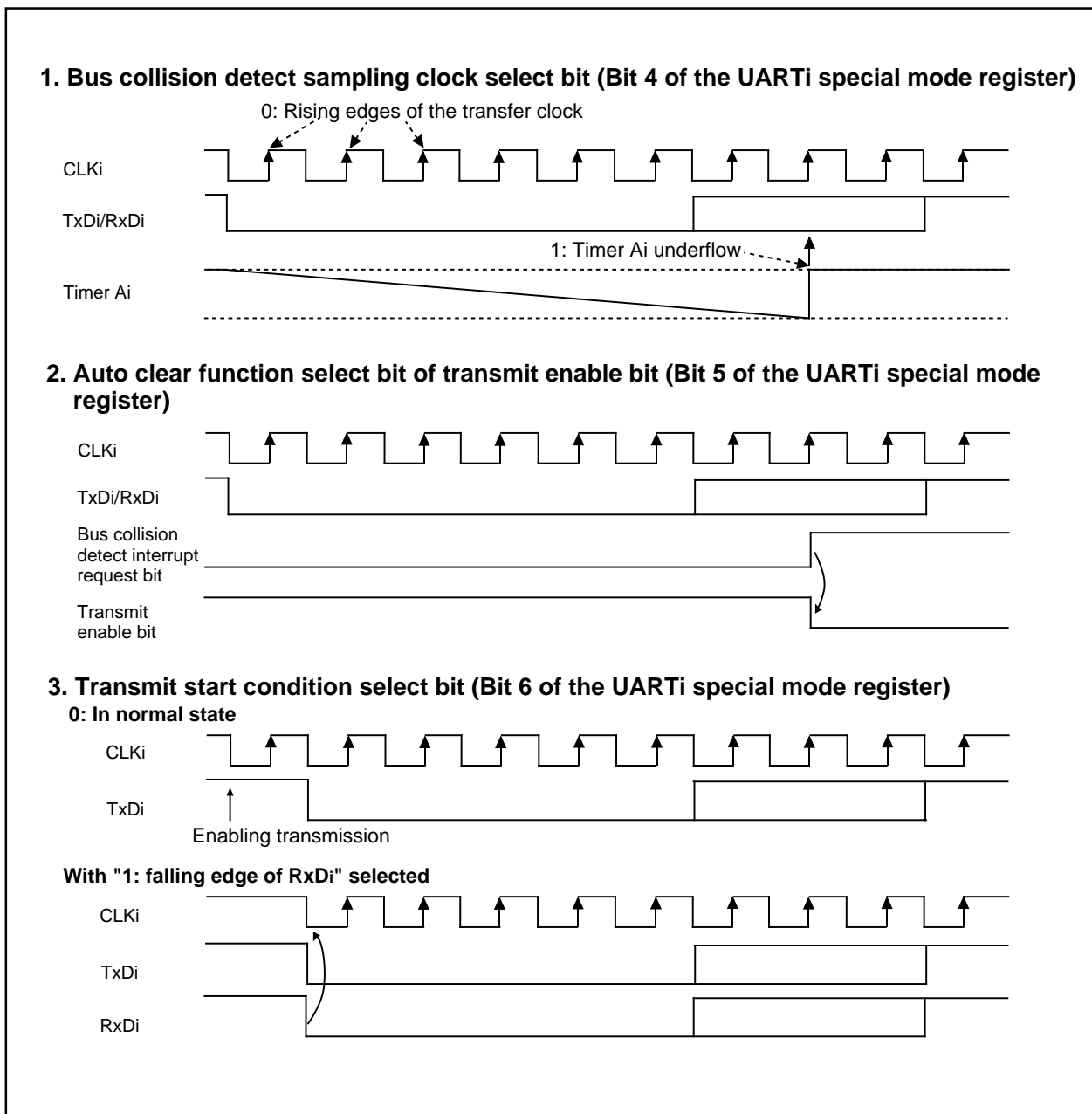


Figure 1.21.2. Some other functions added

UARTi Special Mode Register 2 (UiSMR2:Addresses 0366₁₆, 02E6₁₆, 0336₁₆, 0326₁₆, 02F6₁₆)

Bit 0 is the IIC mode select bit 2 (IICM2). Table 1.21.2 gives control changes by bit when the IIC mode select bit is "1". Start and stop condition detection timing characteristics are shown in Figure 1.21.4. Always set bit 7 (start/stop condition control bit) to "1".

Bit 1 is the clock synchronizing bit (CSC). When this bit is set to "1", and the rising edge is detected at pin SCLi while the internal SCL is High level, the internal SCL is changed to Low level, the baud rate generator value is reloaded and the Low sector count starts. Also, while the SCLi pin is Low level, and the internal SCL changes from Low level to High, baud rate generator stops counting. If the SCLi pin is H level, counting restarts. Because of this function, the UARTi transmission-reception clock takes the AND condition for the internal SCL and SCLi pin signals. This function operates from the clock half period before the 1st rise of the UARTi 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 (SWC). When this bit is set to "1", output from the SCLi pin is fixed to L level at the clock's 9th rise. When set to "0", the Low output lock is released.

Bit 3 is the SDA output stop bit (ALS). When this bit is set to "1", an arbitration lost is generated. If the arbitration lost detection flag is "1", then the SDAi pin simultaneously becomes high impedance.

Bit 4 is the UARTi initialize bit (STC). 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 UARTi 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 SCLi pin becomes Low level at the rise of the 9th bit of the clock.

When UART transmission-reception has 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 (SWC2). When this bit is set to "1" and serial I/O is selected, a Low level can be forcefully output from the SCLi pin even during UART operation. When this bit is set to "0", the Low output from the SCLi pin is canceled and the UARTi clock is input and output.

Bit 6 is the SDA output disable bit (SDHI). When this bit is set to "1", the SDAi pin is forced to high impedance. To overwrite this bit, do so at the rise of the UARTi transfer clock. The arbitration lost detection flag may be set.

UARTi Special Mode Register

Table 1.21.2. Functions changed by I²C mode select bit 2

Function	IICM2 = 0	IICM2 = 1
Interrupt no. 17, 19, 33, 35, 37 factor	Acknowledge not detect (NACK)	UARTi transfer (rising edge of the last bit)
Interrupt no. 18, 20, 34, 36, 38 factor	Acknowledge detect (ACK)	UARTi receive (falling edge of the last bit)
DMA factor	Acknowledge detect (ACK)	UARTi receive (falling edge of the last bit)
Data transfer timing from UART receive shift register to receive buffer	Rising edge of the last bit of receive clock	Rising edge of the last bit of receive clock
UART receive / ACK interrupt request generation timing	Rising edge of the last bit of receive clock	Rising edge of the last bit of receive clock

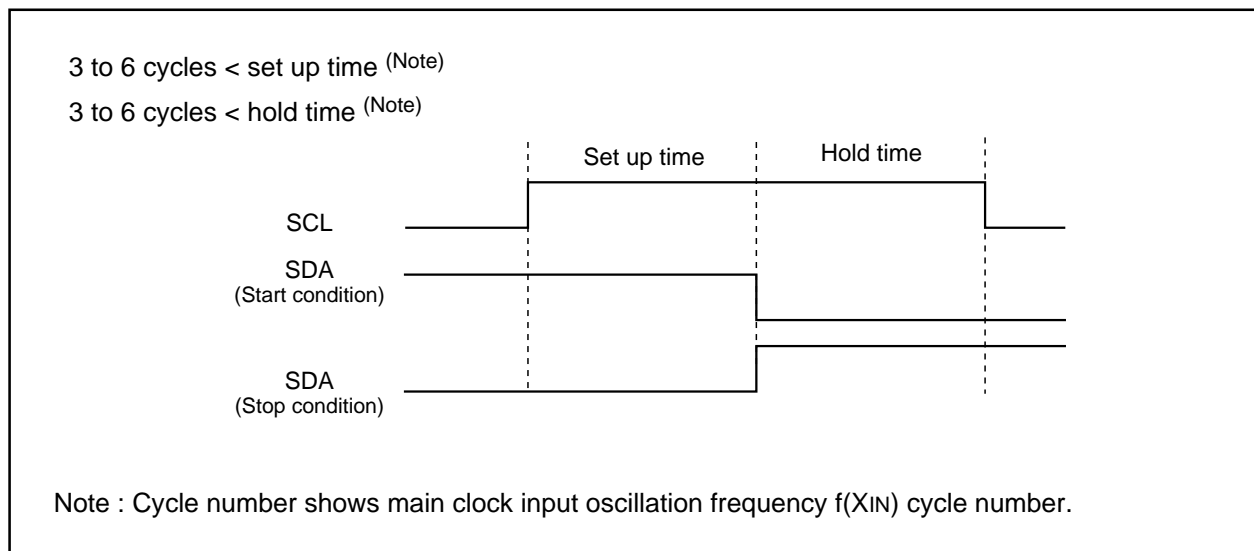


Figure 1.21.3. Start/stop condition detect timing characteristics

UARTi Special Mode Register 3 (UiSMR3:Addresses 0365₁₆, 02E5₁₆, 0335₁₆, 0325₁₆, 02F5₁₆)

Bit 1 is clock phase set bit (CKPH). When both the IIC mode select bit (bit 0 of UARTi special mode select register) and the IIC mode select bit 2 (bit 0 of UiSMR2 register) are "1", functions changed by these bits are shown in table 1.21.3 and figure 1.21.4.

Bits 5 to 7 are SDAi digital delay setting bits (DL0 to DL2). By setting these bits, it is possible to turn the SDAi delay OFF or set the BRG count source delay to 2 to 8 cycles.

Table 1.21.3. Functions changed by clock phase set bits

Function	CKPH = 0, IICM = 1, IICM2 = 1	CKPH = 1, IICM = 1, IICM2 = 1
SCL initial and last value	Initial value = H, last value = L	Initial value = L, last value = L
Transfer interrupt factor	Rising edge of 9th bit	Falling edge of 10th bit
Data transfer times from UART receive shift register to receive buffer register	Falling edge of 9th bit	Two times :falling edge of 9th bit and rising edge of 9th bit

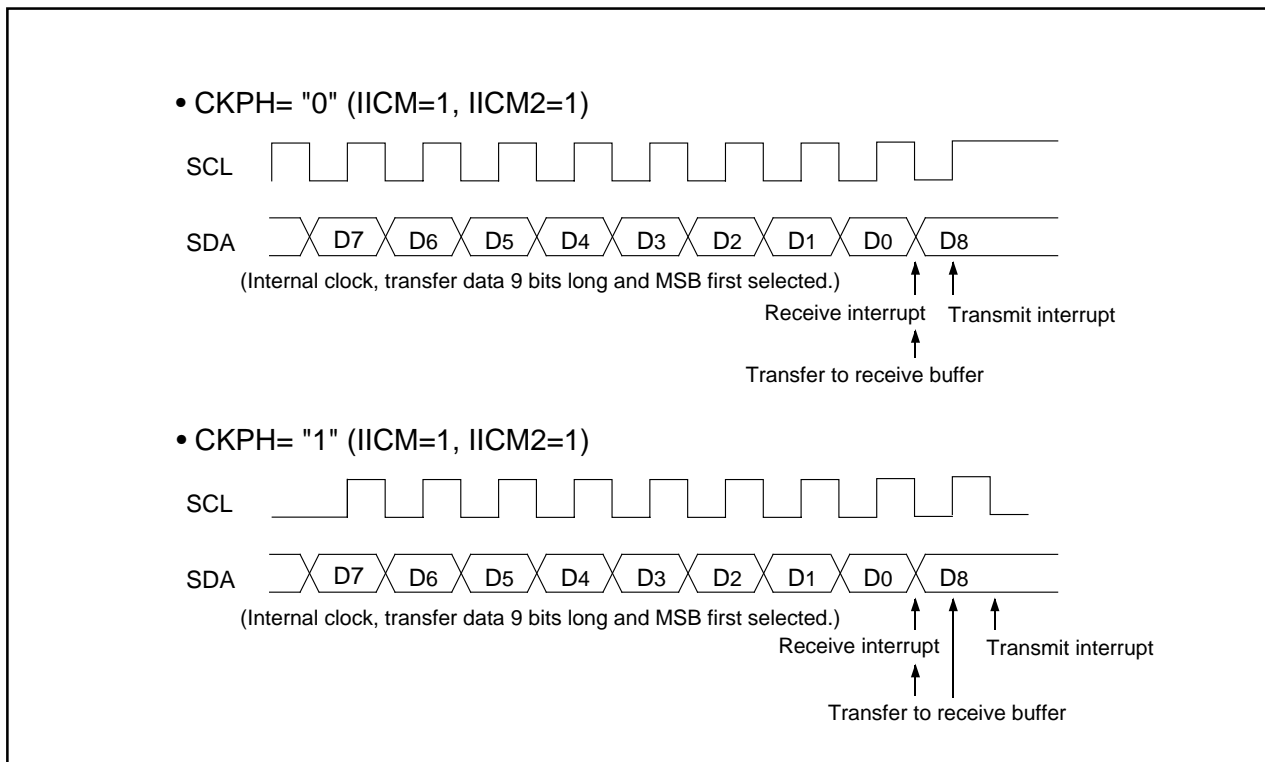


Figure 1.21.4. Functions changed by clock phase set bits

UARTi Special Mode Register 4 (UiSMR4:Addresses 0364₁₆, 02E4₁₆, 0334₁₆, 0324₁₆, 02F4₁₆)

Bit 0 is the start condition generate bit (STAREQ). When the SCL, SDA output select bit (bit 3 of UiSMR4 register) is "1" and this bit is "1", then the start condition is generated.

Bit 1 is the restart condition generate bit (RSTAREQ). When the SCL, SDA output select bit (bit 3 of UiSMR4 register) is "1" and this bit is "1", then the restart condition is generated.

Bit 2 is the stop condition generate bit (STPREQ). When the SCL, SDA output select bit (bit 3 of UiSMR4 register) is "1" and this bit is "1", then the stop condition is generated.

Bit 3 is SCL, SDA output select bit (STSPSEL). Functions changed by these bits are shown in table 1.21.4 and figure 1.21.5.

Table 1.21.4. Functions changed by SCL, SDA output select bit

Function	STSPSEL = 0	STSPSEL = 1
SCL, SDA output	Output of SI/O control circuit	Output of start/stop condition control circuit
Star/stop condition interrupt factor	Start/stop condition detection	Completion of start/stop condition generation

UARTi Special Mode Register

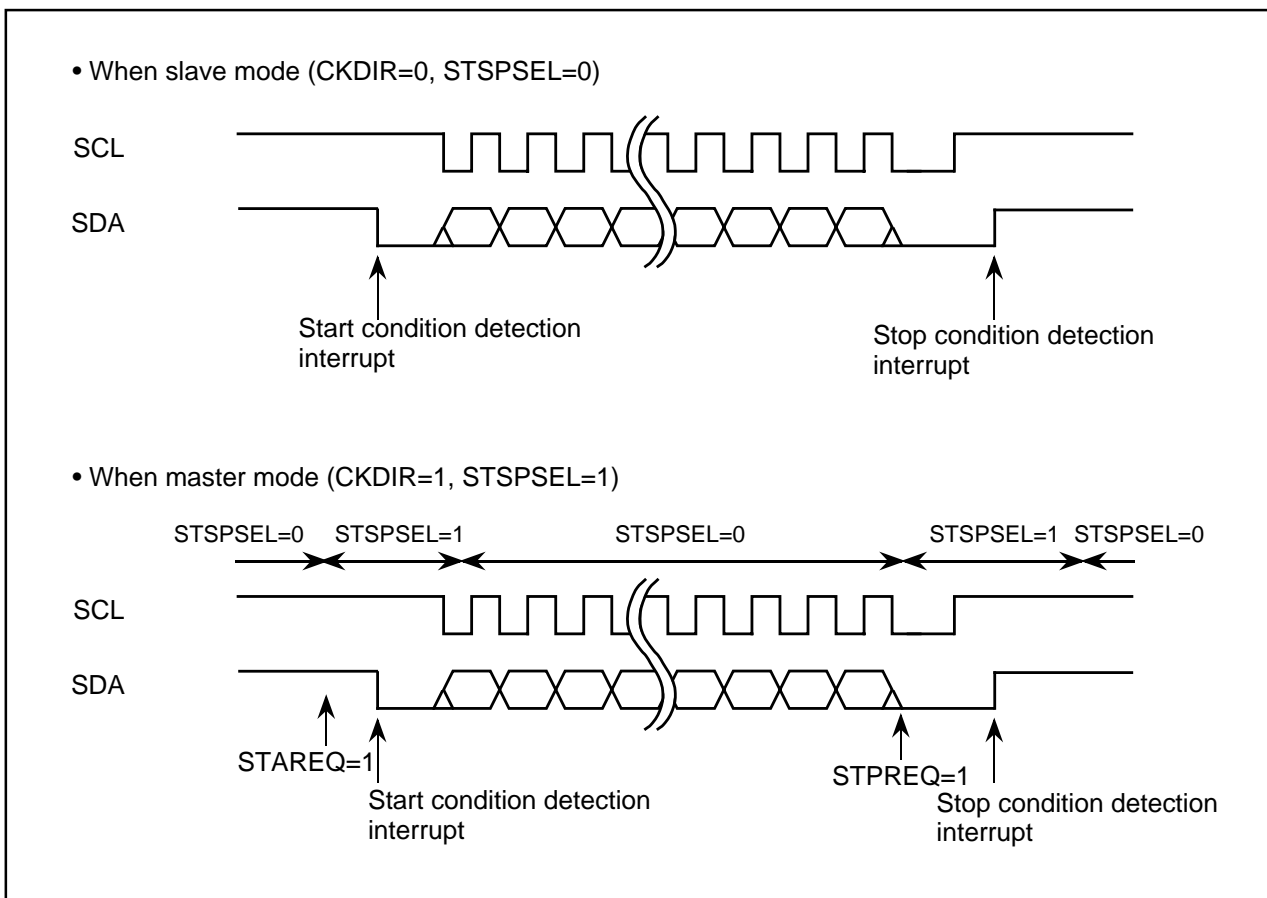


Figure 1.21.5 Functions changed by SCL, SDA output select bit

Bit 4 is ACK data bit (ACKD). When the SCL, SDA output select bit (bit 3 of UiSMR4 register) is "0" and the ACK data output enable bit (bit 5 of UiSMR4 register) is "1", then the content of ACK data bit is output to SDAi pin.

Bit 5 is ACK data output enable bit (ACKC). When the SCL, SDA output select bit (bit 3 of UiSMR4 register) is "0" and this bit is "1", then the content of ACK data bit is output to SDAi pin.

Bit 6 is SCL output stop bit (SCLHI). When this bit is "1", SCLi output is stopped at stop condition detection. (Hi-impedance status).

Bit 7 is SCL wait output bit 3 (SWC9). When this bit is "1", SCLi output is fixed to "L" at falling edge of 10th bit of clock. When this bit is "0", SCLi output fixed to "L" is released.

(2) Serial Interface Special Function

UARTi can control communications on the serial bus using the \overline{SS}_i input pins (Figure 1.21.6). 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{SS}_i input pins.

\overline{SS}_i input pins function between the master and slave are as follows.

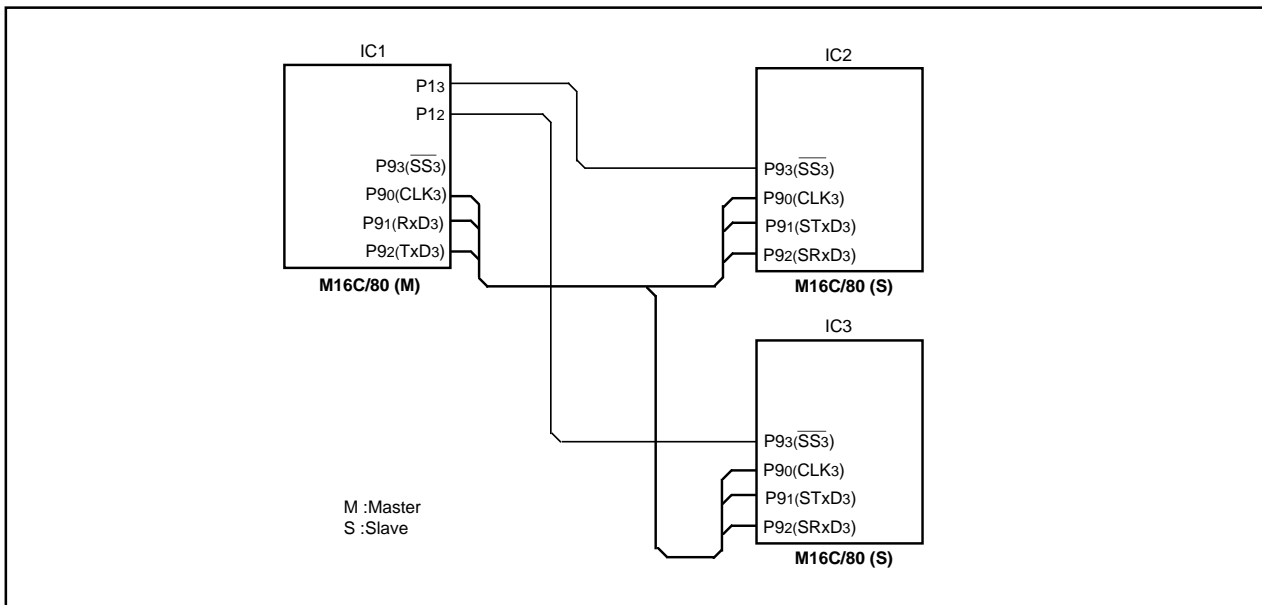


Figure 1.21.6. Serial bus communication control example using the SS input pins

< Slave Mode (STxDi and SRxDi are selected, DINC = 1) >

When an H level signal is input to an \overline{SS}_i input pin, the STxDi and SRxDi pins both become high impedance, hence the clock input is ignored. When an "L" level signal is input to an \overline{SS}_i input pin, the clock input becomes effective and serial communications are enabled.

< Master Mode (TxDi and RxDi are selected, DINC = 0) >

The \overline{SS}_i input pins are used with a multiple master system. When an \overline{SS}_i input pin is H level, transmission has priority and serial communications are enabled. When an L signal is input to an \overline{SS}_i 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 (addresses 036816, 02E816, 033816, 032816 and 02F816) to "0".

UARTi Special Mode Register

■ Clock Phase Setting

With bit 1 of UARTi special mode register 3 (UiSMR3:addresses 0365₁₆, 02E5₁₆, 0335₁₆, 0325₁₆, 02F5₁₆) and bit 6 of UARTi transmission-reception control register 0 (addresses 036C₁₆, 02EC₁₆, 033C₁₆, 032C₁₆, 02FC₁₆), four combinations of transfer clock phase and polarity can be selected.

Bit 6 of UARTi transmission-reception control register 0 sets transfer clock polarity, whereas bit 1 of UiSMR3 register 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 1.21.7 shows the transmission and reception timing.

< Slave (External Clock) (DINC = 1) >

- With "0" for CKPH bit (bit 1 of UiSMR3 register), when an \overline{SSi} input pin is H level, output data is high impedance. When an \overline{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 1.21.8 shows the timing.
- With "1" for CKPH bit, when an \overline{SSi} input pin is H level, output data is high impedance. When an \overline{SSi} input pin is L level, the first data is output. After that, serial transmission is synchronized with the clock. Figure 1.21.9 shows the thing.

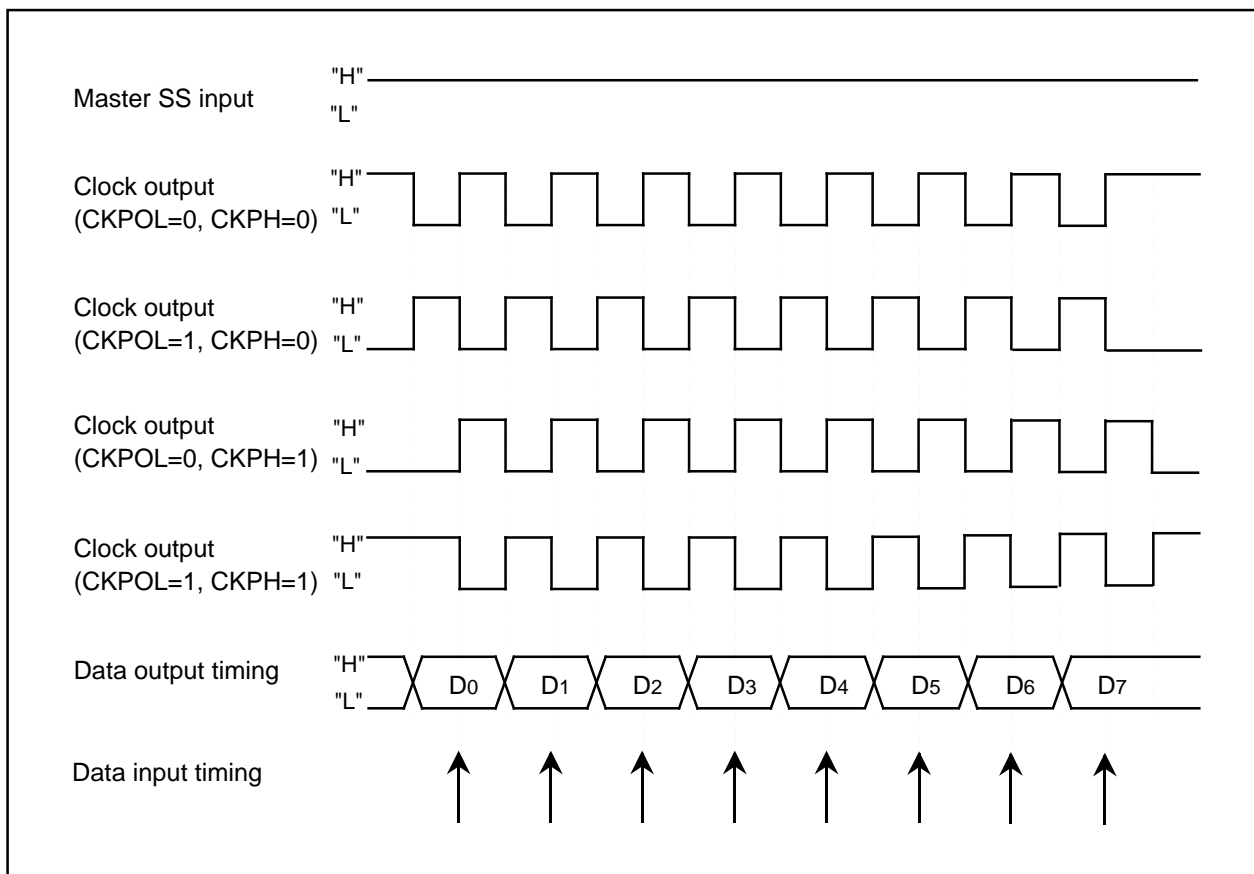


Figure 1.21.7. The transmission and reception timing in master mode (internal clock)

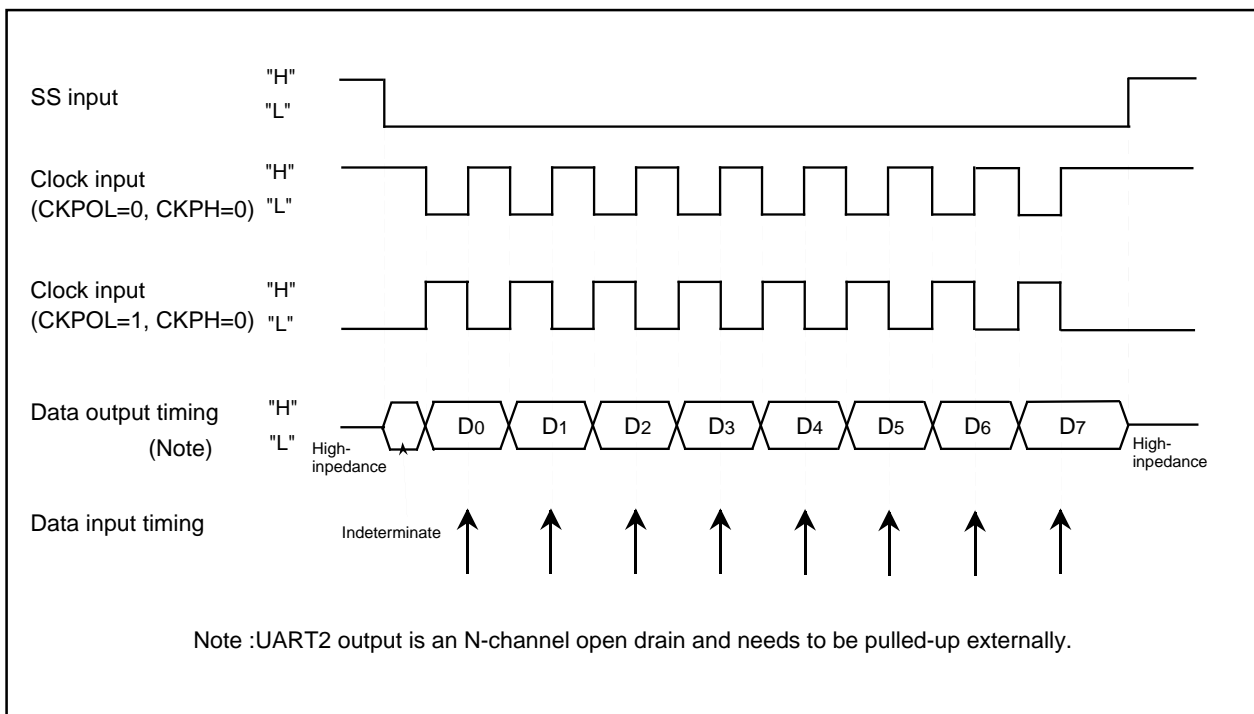


Figure 1.21.8. The transmission and reception timing (CKPH=0) in slave mode (external clock)

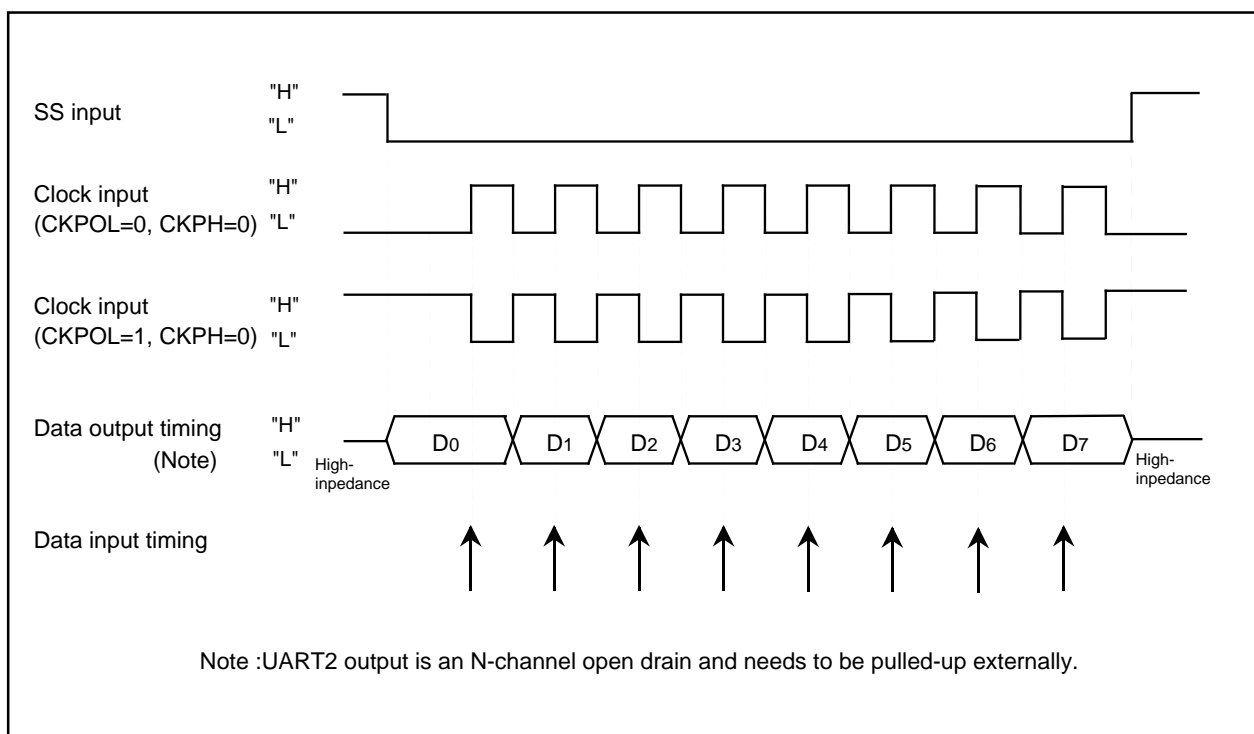


Figure 1.21.9. The transmission and reception timing (CKPH=1) in slave mode (external clock)

CAN Module

The microcomputer incorporates Full-CAN modules compliant with CAN (Controller Area Network) 2.0B specification.

These Full-CAN modules are outlined below.

Table 1.22.1 Outline of the CAN module

Item	Description
Protocol	Compliant with CAN 2.0B specification
Number of message slots	16 slots
Polarity	0: Dominant 1: Recessive
Acceptance filter	Global mask: 1 mask (for message slots 0–13) Local mask: 2 masks (for message slots 14 and 15 each)
Baud rate	1 time quantum (Tq) = (BRP + 1) / CPU clock ^(Note) (BRP = baud rate prescaler set value) Baud rate = 1 / (Tq period x number of Tq's in one bit) ---Max. 1 Mbps BRP: 1-255 (0: Inhibited) Number of Tq's in one bit = Synchronization Segment + Propagation Time Segment + Phase Buffer Segment 1 + Phase Buffer Segment 2 Synchronization Segment : 1 Tq (fixed) Propagation Time Segment : 1 to 8 Tq Phase Buffer Segment 1 : 2 to 8 Tq Phase Buffer Segment 2 : 2 to 8 Tq
Remote frame automatic answering function	The message slot that received a remote frame automatically transmits it.
Timestamp function	This timestamp function is based on a 16-bit counter. A count period can be derived from the CAN bus bit period (as the fundamental period) by dividing it by 1, 2, 3, or 4.
BasicCAN mode	The BasicCAN function is realized by using message slots 14 and 15.
Transmit abort function	This function is used to cancel a transmit request.
Loopback function	The data the CAN module itself transmitted is received.
Return from bus-off function	Forcibly placed into an error active state from a bus-off state.

Note: Use a specification conforming resonator whose maximum permissible error of oscillation is not greater than 1.58%

CAN Module

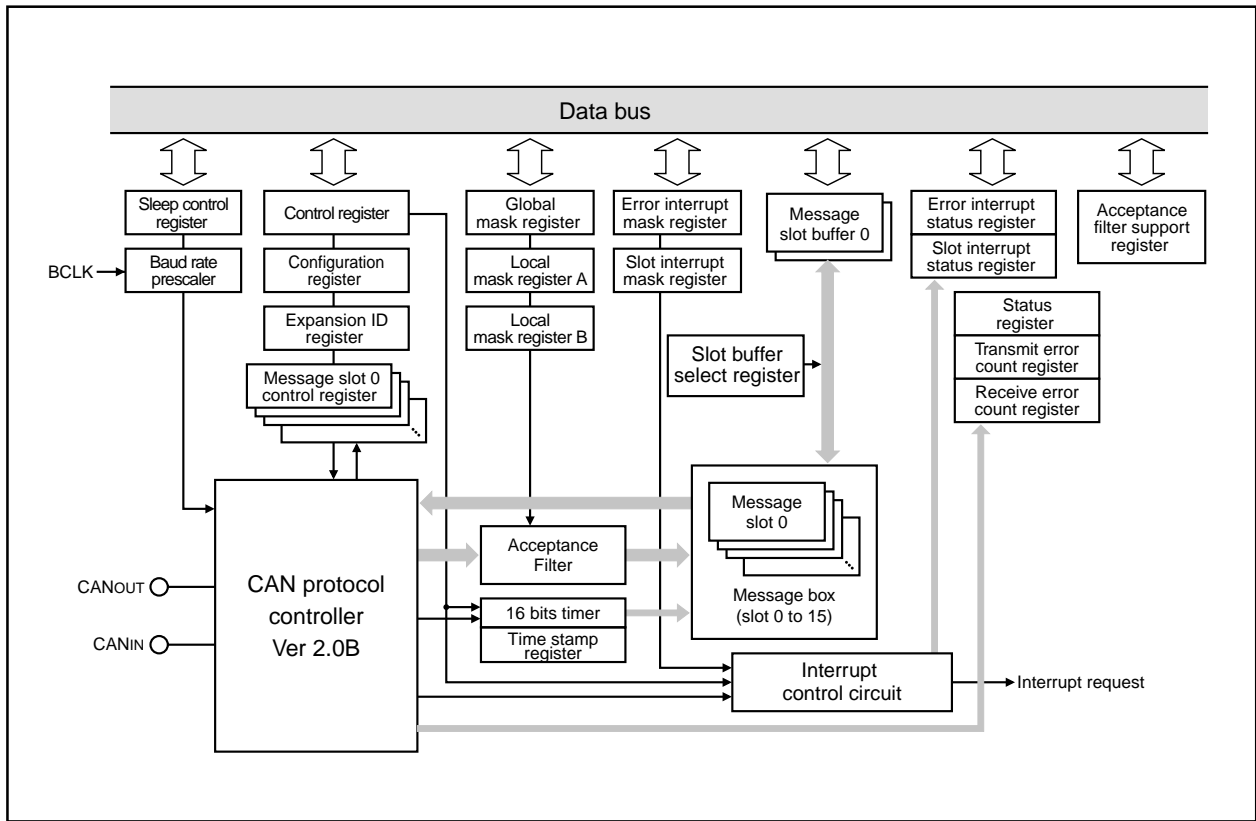


Figure 1.22.1 CAN module block diagram

CAN0 message slot buffer 0 and 1 can be selected by setting of slot buffer select register. Figure 1.22.2 shows the message slot buffer and 16 bytes of message slots. Figure 1.22.26 to 1.22.30 show related registers.

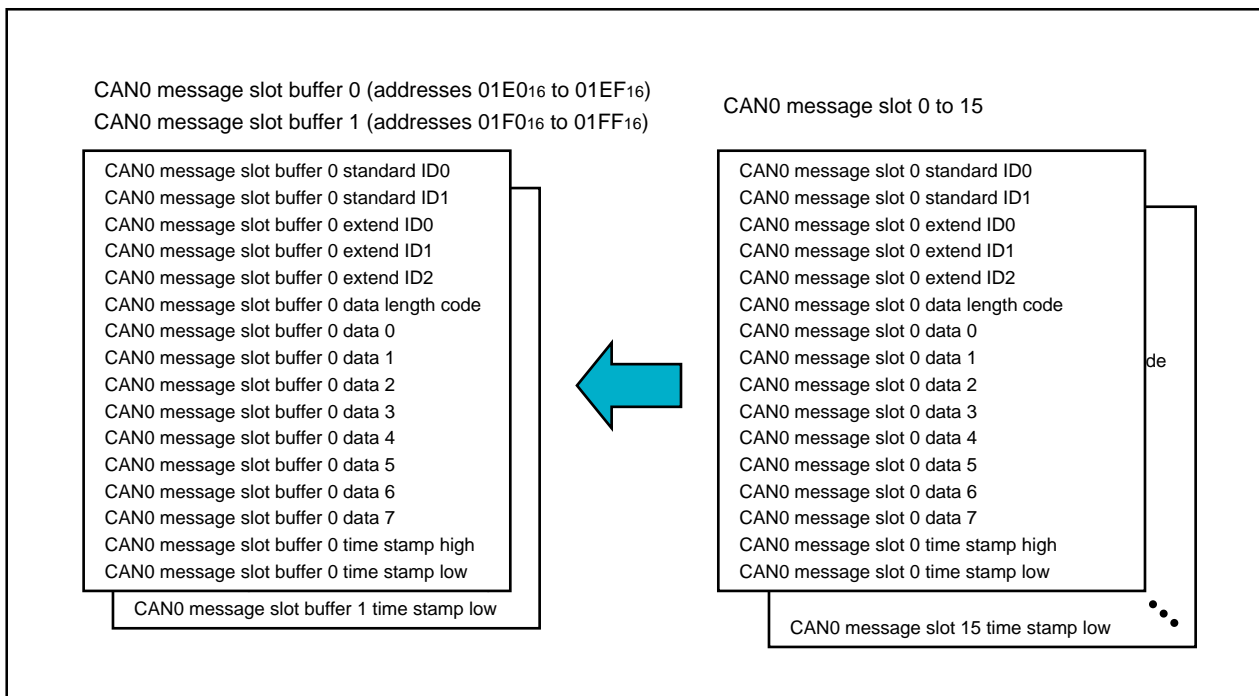


Figure 1.22.2. Message slot buffer and message slots

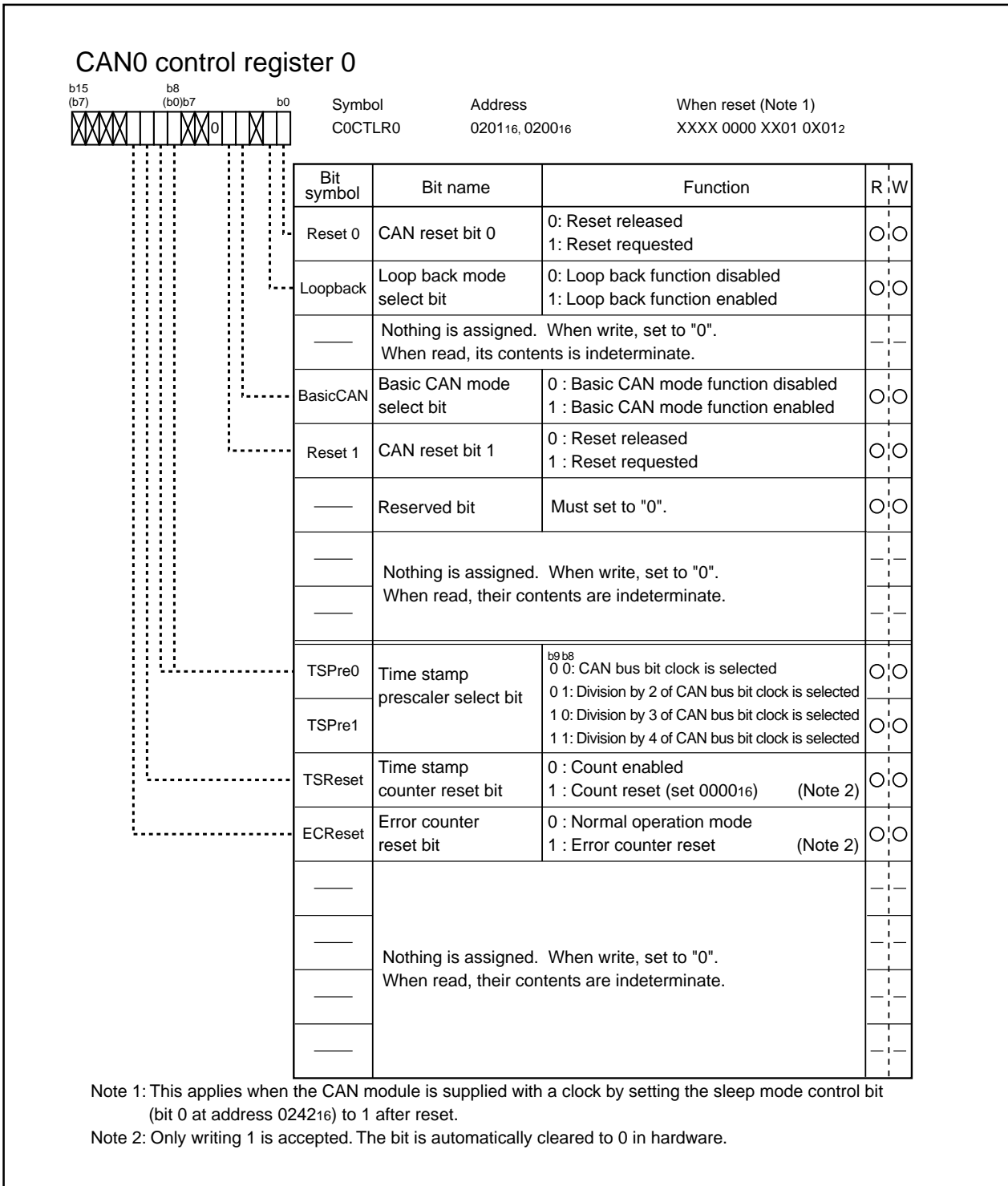


Figure 1.22.3 CAN0 control register 0

1. CAN0 control register 0

Bit 0: CAN reset bits 0 and 1 (Reset0 and Reset1)

If the Reset0 and Reset1 bits both are set from 1 to 0, CAN communication is enabled after detecting 11 consecutive recessive bits. The CAN Timestamp Register starts counting at the same time communication is enabled.

In no case will the CAN be reset unless transmission of all messages are completed.

Note 1: Reset0 and Reset1 bits must both be cleared to "0" or set to "1" simultaneously.

Note 2: Setting a new transmit request is inhibited before the CAN Status Register State_Reset bit is set to 1 and the CAN module is reset after setting the Reset0 and Reset1 bits to 1.

Note 3: When the CAN module is reset by setting the Reset0 and Reset1 bits to 1, the CAN Timestamp Register (C0TSR), CAN Transmit Error Counter (C0TEC), and CAN Receive Error Counter (C0REC) are initialized to 0.

Note 4: If Reset0 and Reset1 bits are set to "1" during communication, the CANOUT pin output goes "H" immediately after that. Therefore, setting these bits to 1 while the CAN module is sending a frame may cause a CAN bus error.

Note 5: To CAN communication, function select register A1 (PS1), function select register A2 (PS2), function select register B1 (PSL1), function select register B2 (PSL2), function select register C (PSC) and input function select register (IPS) must be set. These registers must be set when CAN module is reset.

Bit 1: Loopback mode select bit (LoopBack)

Setting the LoopBack bit to 1 enables loopback mode, so that if any receive slot whose ID matches that of a frame the CAN module itself transmitted exists, the frame is received.

Note 1: ACK is not returned for the transmit frame.

Note 2: Do not set or reset the LoopBack bit while the CAN module is operating (CAN Status Register State_Reset bit = 0).

Bit 3: BasicCAN mode select bit (BasicCAN)

If this bit is set to 1, message slots 14 and 15 operate in BasicCAN mode.

• Operation during BasicCAN mode

In BasicCAN mode, message slots 14 and 15 are used with a dual-structured buffer. The received frames whose IDs are found matching by acceptance filtering are stored in slots 14 and 15 alternately. When slot 14 is active (i.e., the next received frame is to be stored in slot 14), this acceptance filtering is accomplished using the ID that is set in slot 14 and local mask A; when slot 15 is active, it is accomplished using the ID that is set in slot 15 and local mask B. Frame types of both data frame and remote frame can be received.

When using BasicCAN mode, setting the IDs of two slots and the mask registers the same way helps to reduce the possibility of causing an overrun error.

• Procedure for entering BasicCAN mode

Make the following settings during initialization.

- (1) Set the BasicCAN bit to 1.
- (2) Set the IDs of slots 14 and 15 and Local Mask Registers A and B. (We recommend setting the same value)
- (3) Set the frame format to be handled with slots 14 and 15 (standard or extended) in the CAN Extended ID Register. (We recommend setting the same format)

(4) Set the Message Slot Control Registers for slots 14 and 15 to receive data frames.

Note 1: Do not set or reset the BasicCAN bit while the CAN module is operating (CAN Status Register State_Reset bit = 0).

Note 2: Slot 14 is the first slot to become active after clearing the Reset0 bit.

Note 3: Even during BasicCAN mode, slot 0 through slot 13 can be used in the same way as when operating normally.

Bit 8, 9: Timestamp prescaler select bits (TSPre0, 1)

These bits select the count clock source for the timestamp counter.

Note 1: Do not set or reset these TSPre0, 1 bits while the CAN module is operating (CAN Status Register State_Reset bit = 0).

Bit 10: Timestamp counter reset bit (TSReset)

Setting this bit to 1 clears the value of the CAN Timestamp Register (C0TSR) to 0000₁₆. This bit is automatically cleared after the CAN Timestamp Register (C0TSR) has its value cleared to 0000₁₆.

Bit 11: Error counter reset bit (ECReset)

Setting this bit to 1 clears the Receive Error Counter Register (C0REC) and Transmit Error Counter Register (C0TEC), with the CAN module forcibly placed in an error active state. This bit is automatically cleared upon entering an error active state.

Note 1: When in an error active state, the CAN module becomes ready to communicate when it detects 11 consecutive recessive bits on the CAN bus.

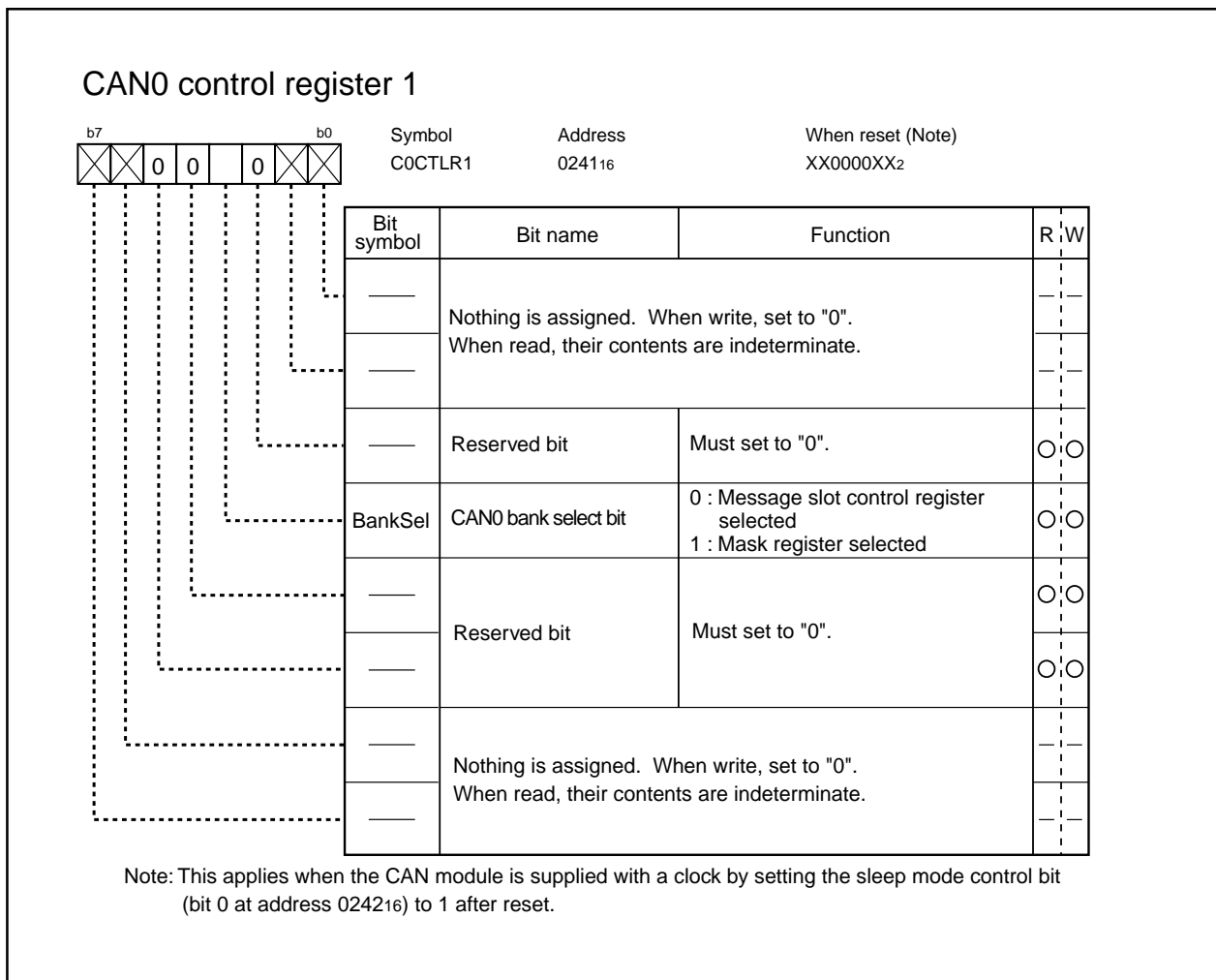


Figure 1.22.4. CAN0 control register 1

2. CAN0 control register 1

Bit 3: CAN0 bank select bit (BankSel)

This bit selects between registers allocated to the addresses 0220₁₆ through 023F₁₆.

Setting the BankSel bit to 0 selects the CAN0 Message Slot Control Register. Setting the BankSel bit to 1 selects the CAN0 Mask Register.

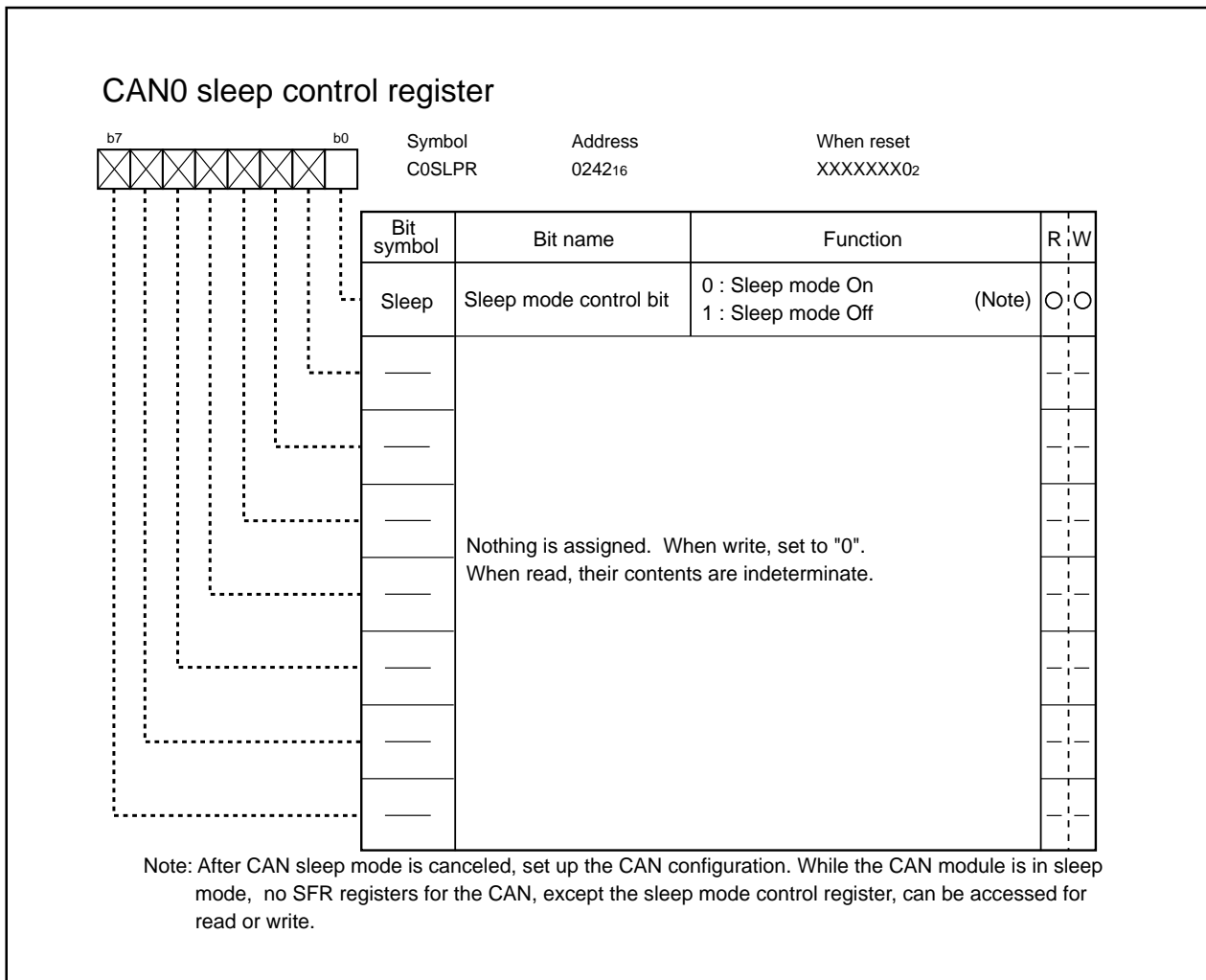


Figure 1.22.5. CAN0 sleep control register

3. CAN0 sleep control register

Bit 0: Sleep mode control bit (Sleep)

The CAN module isn't supplied with a clock by setting the Sleep bit to 0, and is shifted to sleep mode. The CAN module is supplied with a clock by setting the Sleep bit to 1, and is released from sleep mode.

Note: Sleep mode can be shifted to only after CAN is reset (State_Reset bit = 1).

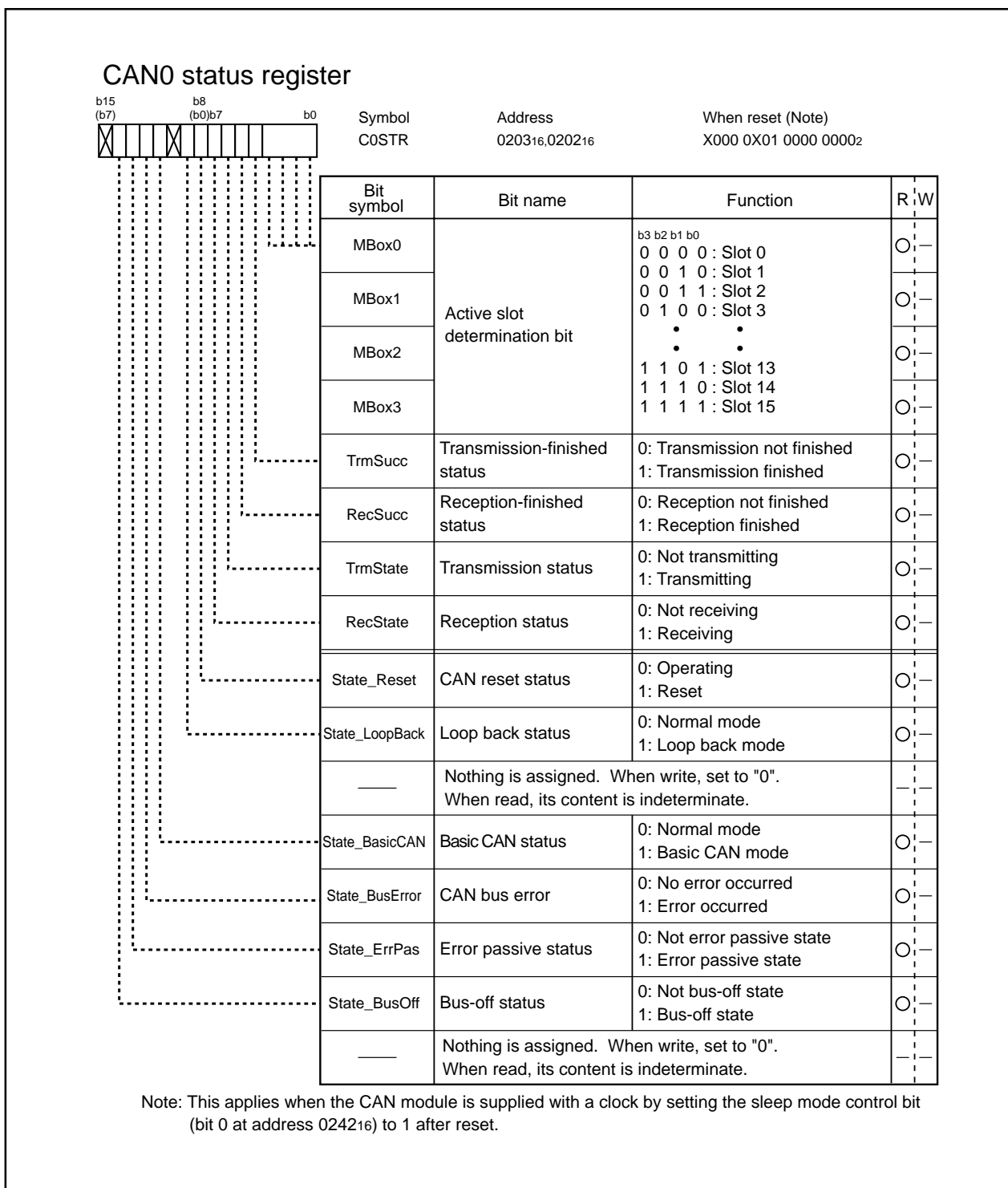


Figure 1.22.6. CAN0 status register

4. CAN0 status register

Bits 0–3: Active slot determination bits (MBox)

When the CAN module finished transmitting data or finished storing received data, the relevant slot number is stored in these bits.

The MBox bits cannot be cleared to 0 in software.

Bit 4: Transmission-finished status (TrmSucc)

[Set condition]

This bit is set to 1 when the CAN module finished transmitting data normally.

[Clear condition]

This bit is cleared when the CAN module finished receiving data normally.

Bit 5: Reception-finished status (RecSucc)

[Set condition]

This bit is set to 1 when the CAN module finished receiving data normally (regardless of whether the received message has been stored in a message slot). However, this bit is not set if the received message is one that was transmitted in loopback mode.

[Clear condition]

This bit is cleared when the CAN module finished transmitting data normally.

Bit 6: Transmission status (TrmState)

[Set condition]

This bit is set to 1 when the CAN module is operating as a transmit node.

[Clear condition]

This bit is cleared when the CAN module goes to a bus-idle state or starts operating as a receive node.

Bit 7: Reception status (RecState)

[Set condition]

This bit is set to 1 when the CAN module is operating as a receive node.

[Clear condition]

This bit is cleared when the CAN module goes to a bus-idle state or starts operating as a transmit node.

Bit 8: CAN reset status (State_Reset)

When the State_Reset bit = 1, it means that the CAN module is in a reset state.

[Set condition]

This bit is set to 1 when CAN module is in a reset state.

[Clear condition]

This bit is cleared by clearing the Reset0 or Reset1 bits to 0.

Bit 9: Loopback status (State_loopBack)

When the State_loopBack bit = 1, it means that the CAN module is operating in loopback mode.

[Set condition]

This bit is set to 1 by setting the CAN control register LoopBack bit to 1.

[Clear condition]

This bit is cleared by clearing the LoopBack bit to 0.

Bit 11: BasicCAN status (State_BasicCAN)

When the State_BasicCAN bit = 1, it means that the CAN module is operating in BasicCAN mode.

[Set condition]

This bit is set to 1 when the CAN module is operating in BasicCAN mode.

Conditions for the CAN module to operate in BasicCAN mode are as follows:

- The CAN Control Register BasicCAN bit is set to 1.
- Slots 14 and 15 both are set for data frame reception.

[Clear condition]

This bit is cleared by clearing the BasicCAN bit to 0.

Bit 12: CAN bus error (State_BusError)

[Set condition]

This bit is set to 1 when an error on the CAN bus is detected.

[Clear condition]

This bit is cleared when the CAN module finished transmitting or receiving normally. Clearing of this bit does not depend on whether the received message has been stored in a message slot.

Note :When this bit is 1, although CAN module is reset, this bit does not become to 0.

Bit 13: Error passive status (State_ErrPas)

When the State_ErrPas bit = 1, it means that the CAN module is in an error-passive state.

[Set condition]

This bit is set to 1 when the value of C0TEC register or C0REC register exceeds 127, with the CAN module in an error-passive state.

[Clear condition]

This bit is cleared when the CAN module goes from the error-passive state to any other error state.

Note :When this bit is 1, then CAN module is reset, this bit becomes 0 automatically.

Bit 14: Bus-off status (State_BusOff)

When the State_BusOff bit = 1, it means that the CAN module is in a bus-off state.

[Set condition]

This bit is set to 1 when the value of the C0TEC register exceeds 255, with the CAN module in a bus-off state.

[Clear condition]

This bit is cleared when the CAN module returns from the bus-off state.

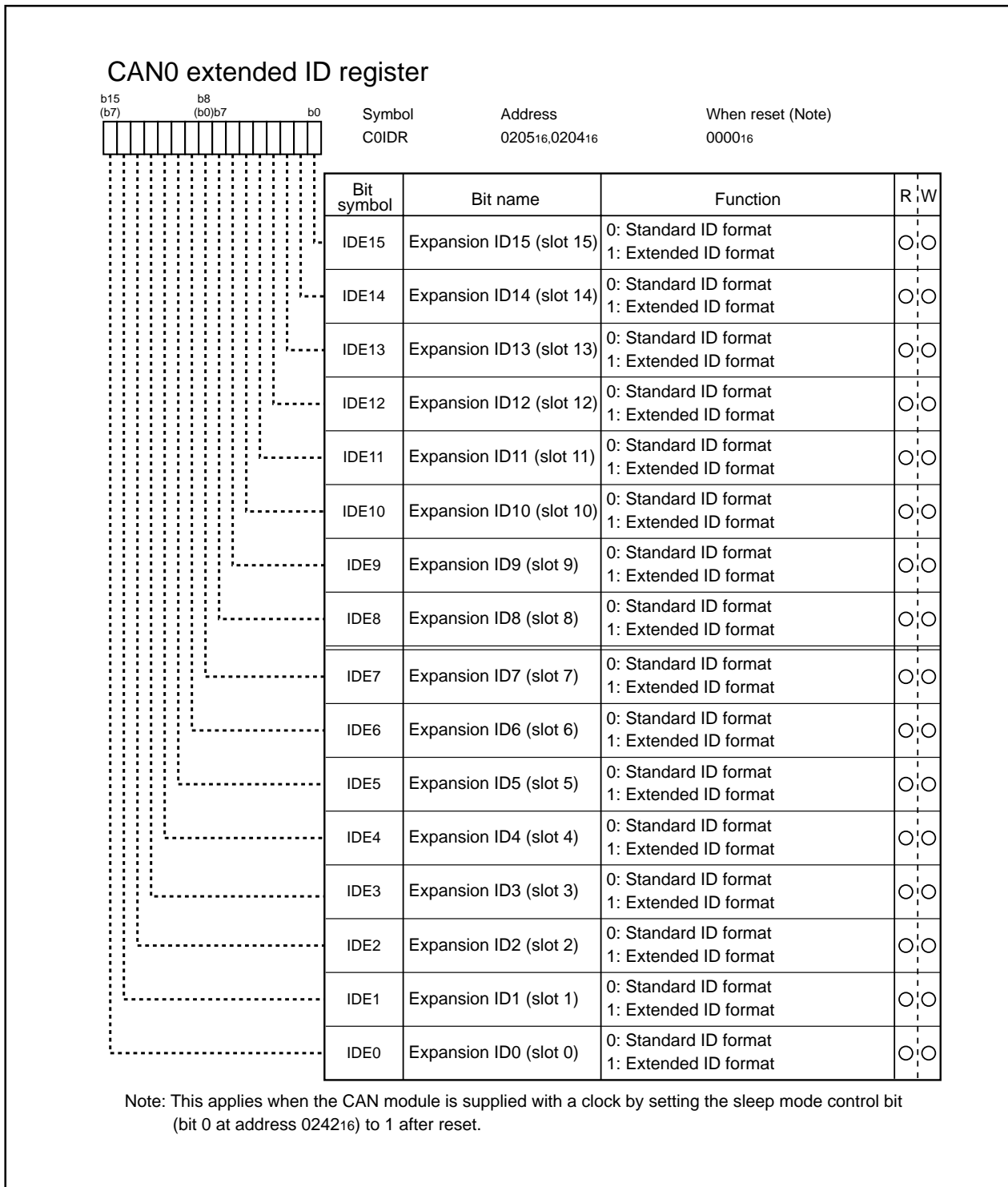


Figure 1.22.7. CAN0 extended ID register

5. CAN0 extended ID register

This register selects the format of a frame handled by the message slot that corresponds to each bit in this register.

Setting any bit to 0 selects the standard (Standard ID) format.

Setting any bit to 1 selects the extended (Extended ID) format.

Note 1: When setting or resetting any bit in this register, make sure the corresponding slot has no transmit or receive request.

CAN Module

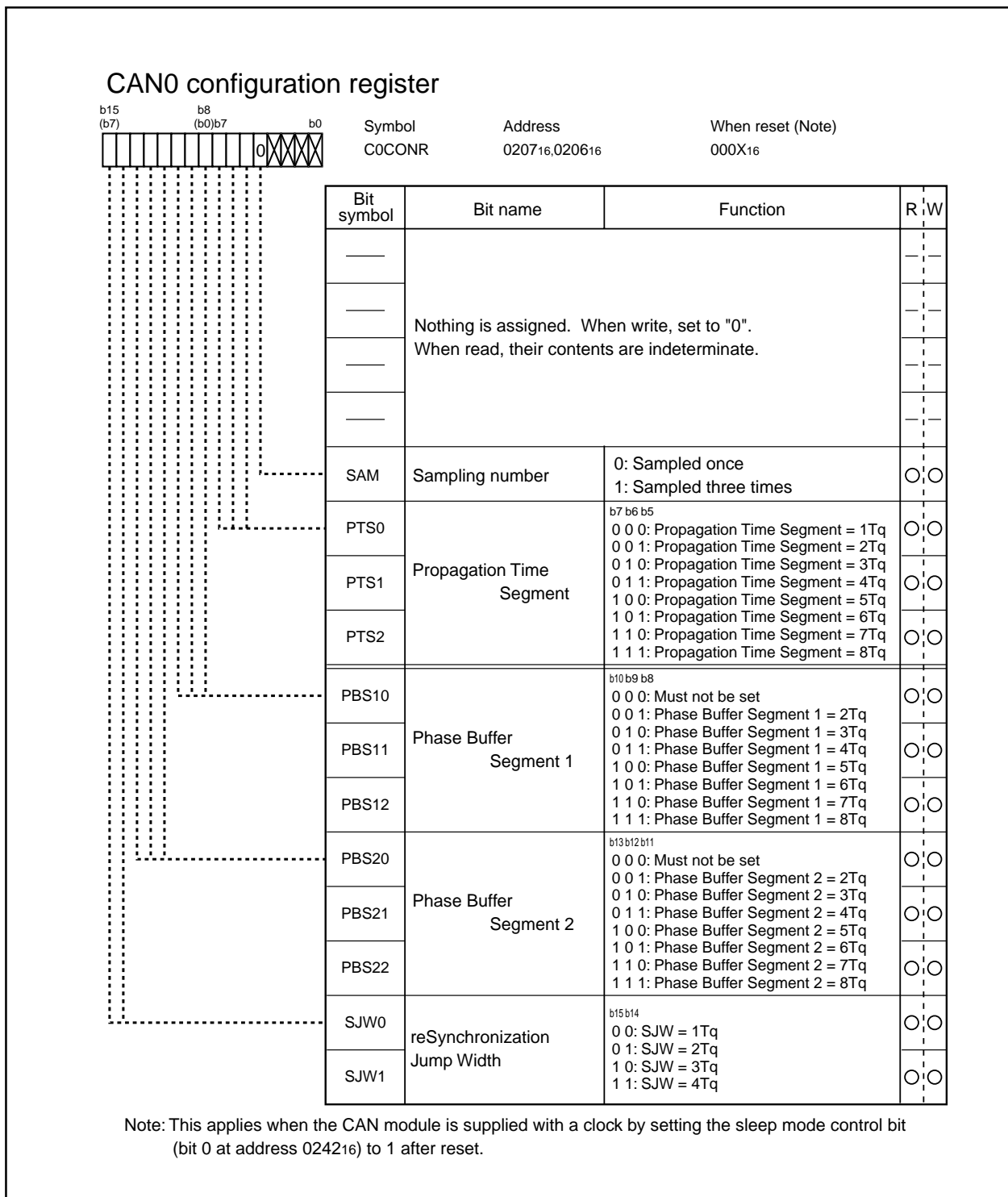


Figure 1.22.8. CAN0 configuration register

6. CAN0 configuration register

Bit 4: SAM bit (SAM)

This bit sets the sampling number per one bit.

0: The value sampled at the last of the Phase Buffer Segment 1 becomes the bit value.

1: The bit value is determined by the majority operation circuit using values sampled at the following three points: the last of the Phase Buffer Segment 1, before 1T_q, and before 2T_q.

Bits 5–7: PTS bits (RTS00-RTS02)

These bits set the width of Propagation Time Segment.

Bits 8–10: PBS1 bits (PBS10-PBS12)

These bits set the width of Phase Buffer Segment 1. The PBS1 bits must be set to 1 or greater.

Bits 11–13: PBS2 bits (PBS20-PBS22)

These bits set the width of Phase Buffer Segment 2. The PBS2 bits must be set to 1 or greater.

Bits 14, 15: SJW bits (SJW0, SJW1)

These bits set the width of reSynchronization Jump Width. The SJW bits must be set to a value equal to or less than PBS2.

Table 1.22.2 Bit Timing Setup Example when the CPU Clock = 30 MHz

Baud rate	BRP	Tq period (ns)	1 bit's Tq number	PTS+PBS1	PBS2	Sample point
1Mbps	1	66.7	15	12	2	87%
	1	66.7	15	11	3	80%
	1	66.7	15	10	4	73%
	2	100	10	7	2	80%
	2	100	10	6	3	70%
	2	100	10	5	4	60%
500Kbps	2	100	20	16	3	85%
	2	100	20	15	4	80%
	2	100	20	14	5	75%
	3	133.3	15	12	2	87%
	3	133.3	15	11	3	80%
	3	133.3	15	10	4	73%
	4	166.7	12	9	2	83%
	4	166.7	12	8	3	75%
	4	166.7	12	7	4	67%
	5	200	10	7	2	80%
	5	200	10	6	3	70%
	5	200	10	5	4	60%

CAN Module

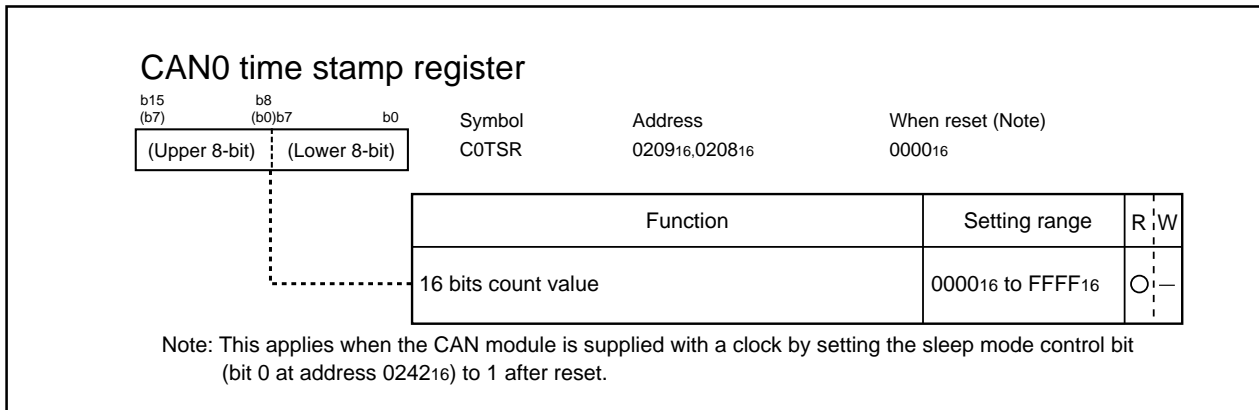


Figure 1.22.9. CAN0 time stamp register

7. CAN0 Timestamp register

The CAN module incorporates a 16-bit counter. The count period for this counter can be derived from the CAN bus bit period by dividing it by 1, 2, 3, or 4 using the CAN0 control register0 (C0CTRL0)'s TSPre0, 1 bits.

When the CAN module finishes transmitting or receiving, the CAN0 Timestamp Register (C0TSR) value is captured and the value is automatically stored in a message slot.

The C0TSR register starts counting upon clearing the C0CTRL register's Reset and Reset1 bits to 0.

Note 1: Setting the C0CTRL0 register's Reset0 and Reset1 bits to 1 resets CAN, and the C0TSR register thereby initialized to 0000₁₆. Also, setting the TSReset (timestamp counter reset) bit to 1 initializes the C0TSR register to 0000₁₆ on-the-fly (while the CAN remains operating; CAN0 status register's State_Reset bit is "0").

Note 2: During loopback mode, if any receive slot exists in which a message can be stored, the C0TSR register value is stored in the corresponding slot when the CAN module finished receiving. (This storing of the C0TSR register value does not occur at completion of transmission.)

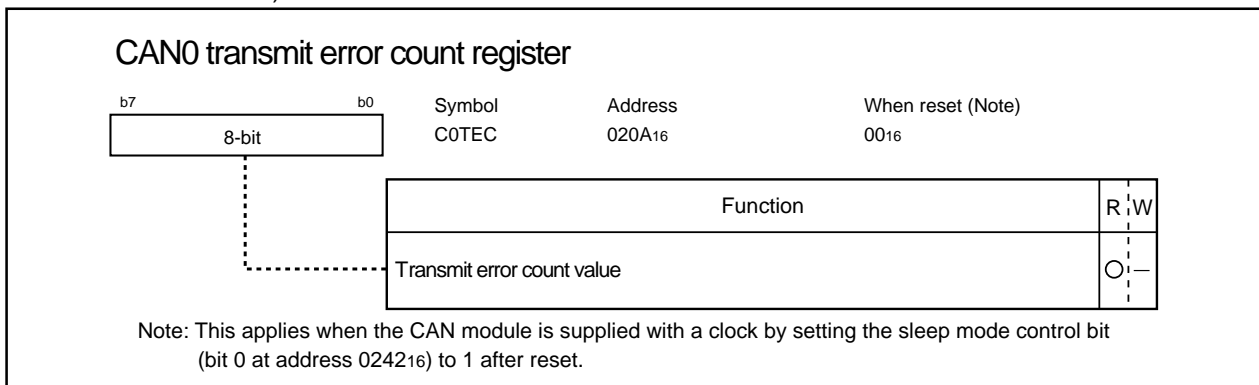


Figure 1.22.10. CAN0 transmit error count register

8. CAN0 transmit error count register

When in an error active or an error passive state, the transmit error count value is stored in this register. The count is decremented when the CAN module finished transmitting normally or incremented when an error occurred while transmitting.

When in a bus-off state, an indeterminate value is stored in this register. The register is reset to 00₁₆ upon returning to an error active state.

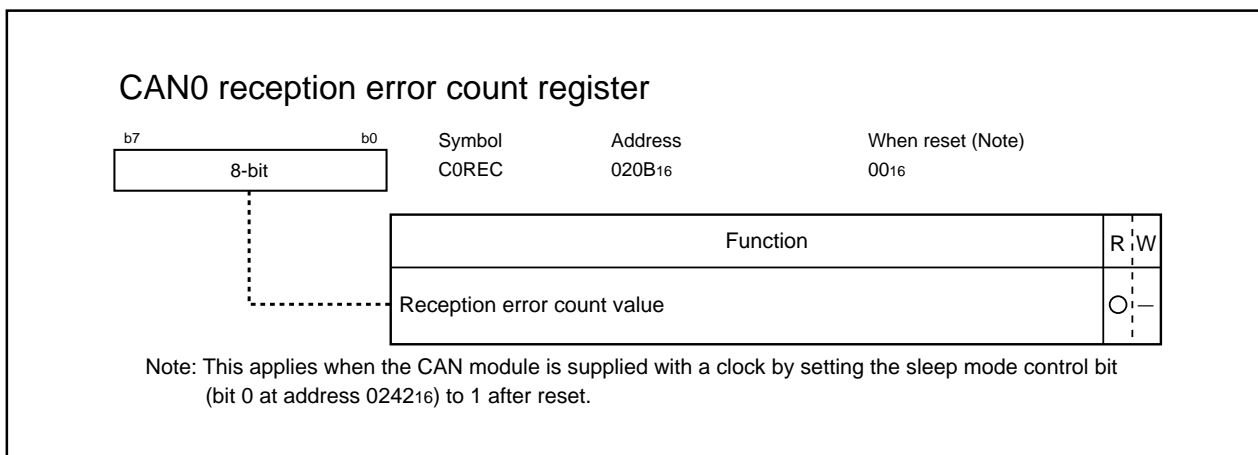


Figure 1.22.11. CAN0 reception error count register

9. CAN0 reception error count register

When in an error active or an error passive state, the receive error count value is stored in this register. The count is decremented when the CAN module finished receiving normally or incremented when an error occurred while receiving.

When $C0REC \geq 128$ (error passive state) at the time the CAN module finished receiving normally, the C0REC register is set to 127.

When in a bus-off state, an indeterminate value is stored in this register. The register is reset to 00₁₆ upon returning to an error active state.

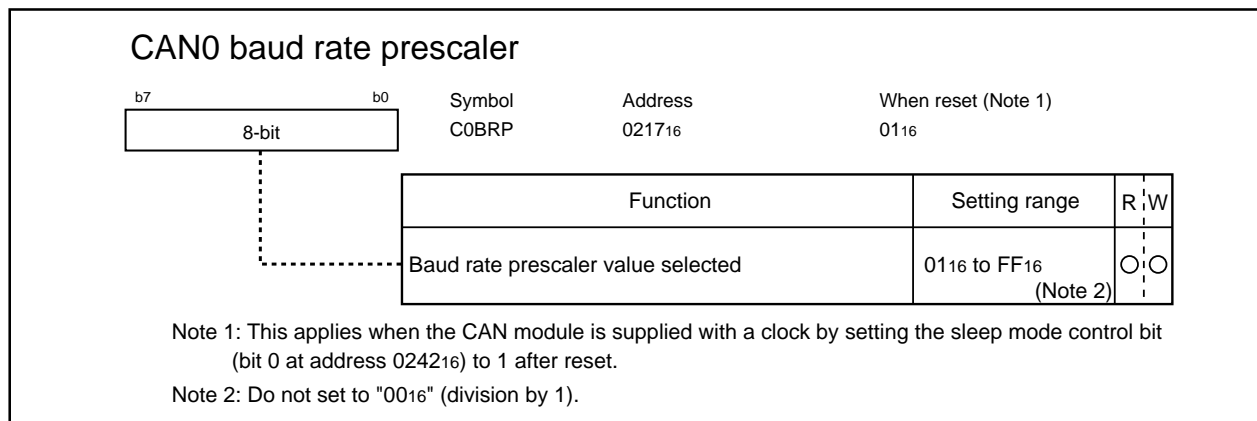


Figure 1.22.12. CAN0 baud rate register

10. CAN0 baud rate prescaler

This register is used to set the T_q period, the CAN bit time. The CAN baud rate is determined by (T_q period x number of T_q's in one bit).

$$T_q \text{ period} = (C0BRP+1)/\text{CPU clock}$$

$$\text{CAN baud rate} = 1 / (T_q \text{ period} \times \text{number of } T_q\text{'s in one bit})$$

$$\begin{aligned} \text{Number of } T_q\text{'s in one bit} = & \text{Synchronization Segment} + \\ & \text{Propagation Time Segment} + \\ & \text{Phase Buffer Segment 1} + \\ & \text{Phase Buffer Segment 2} \end{aligned}$$

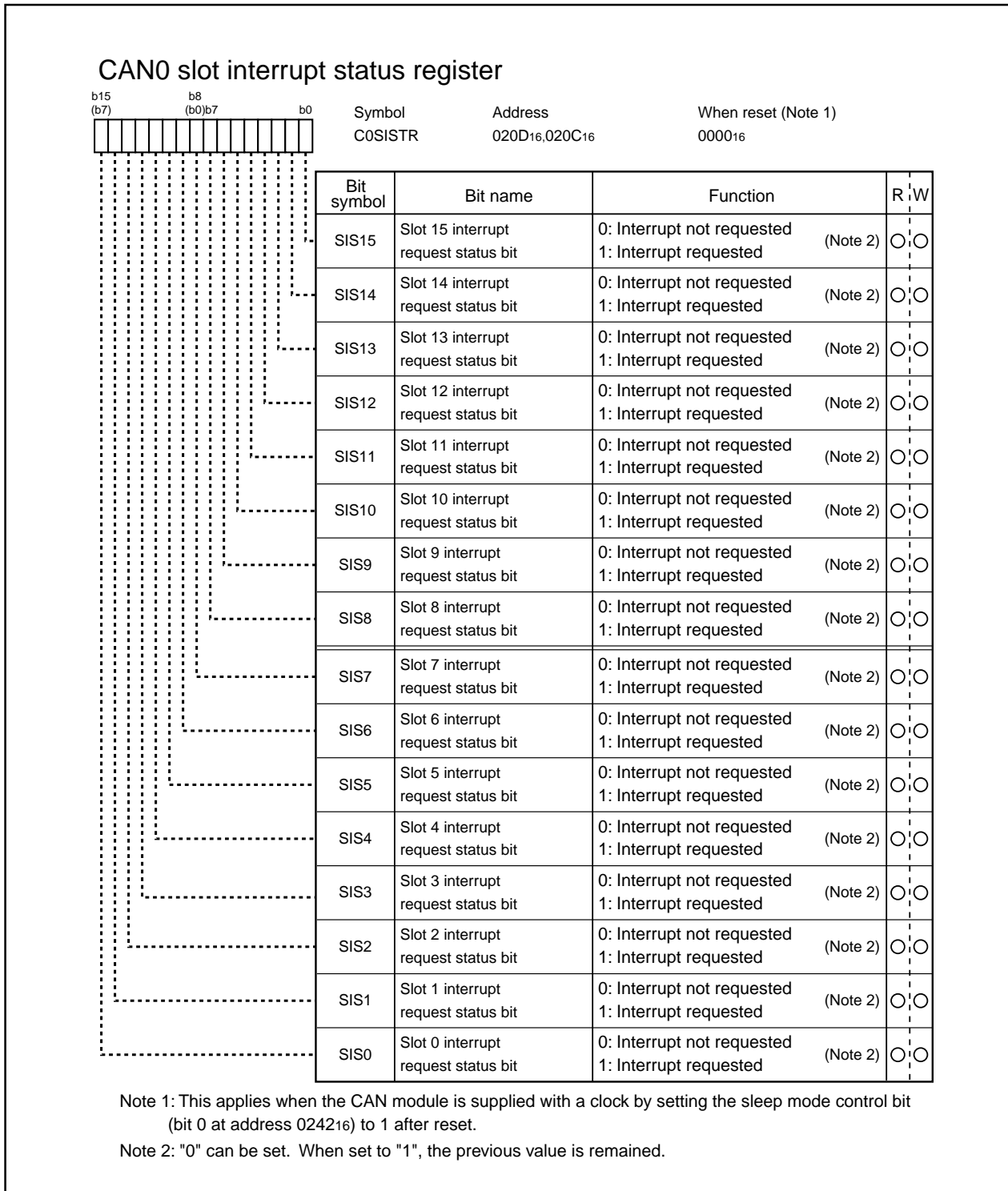


Figure 1.22.13. CAN slot interrupt status register

9. CAN0 slot interrupt status register

When using CAN interrupts, the CAN0 Slot Interrupt Status Register helps to know which slot requested an interrupt.

- **For transmit slots**

The status is set to 1 when the CAN module finished storing the CAN Timestamp Register value in the message slot after completing transmission.

To clear this bit, write 0 in software (Note 1).

- **For receive slots**

The status is set to 1 when the CAN module finished storing the received message in the message slot after completing reception.

To clear this bit, write 0 in software (Note 1).

Note 1: To clear any bit of the CAN Interrupt Status Register, write 0 to the bit to be cleared and 1 to all other bits, without using bit clear instructions.

Example : Assembler language `mov.w #07FFFh, C0SISTR`
 C language `c0sister = 0x7FFF;`

Note 2: For remote frame receive slots whose automatic answering function is enabled, the slot interrupt status bit is set when the CAN module finished receiving a remote frame and when it finished transmitting a data frame.

Note 3: For remote frame transmit slots, the slot interrupt status bit is set when the CAN module finished transmitting a remote frame and when it finished receiving a data frame.

Note 4: If the slot interrupt status bit is set by an interrupt request at the same time it is cleared by writing in software, the former has priority, i.e., the slot interrupt status bit is set.

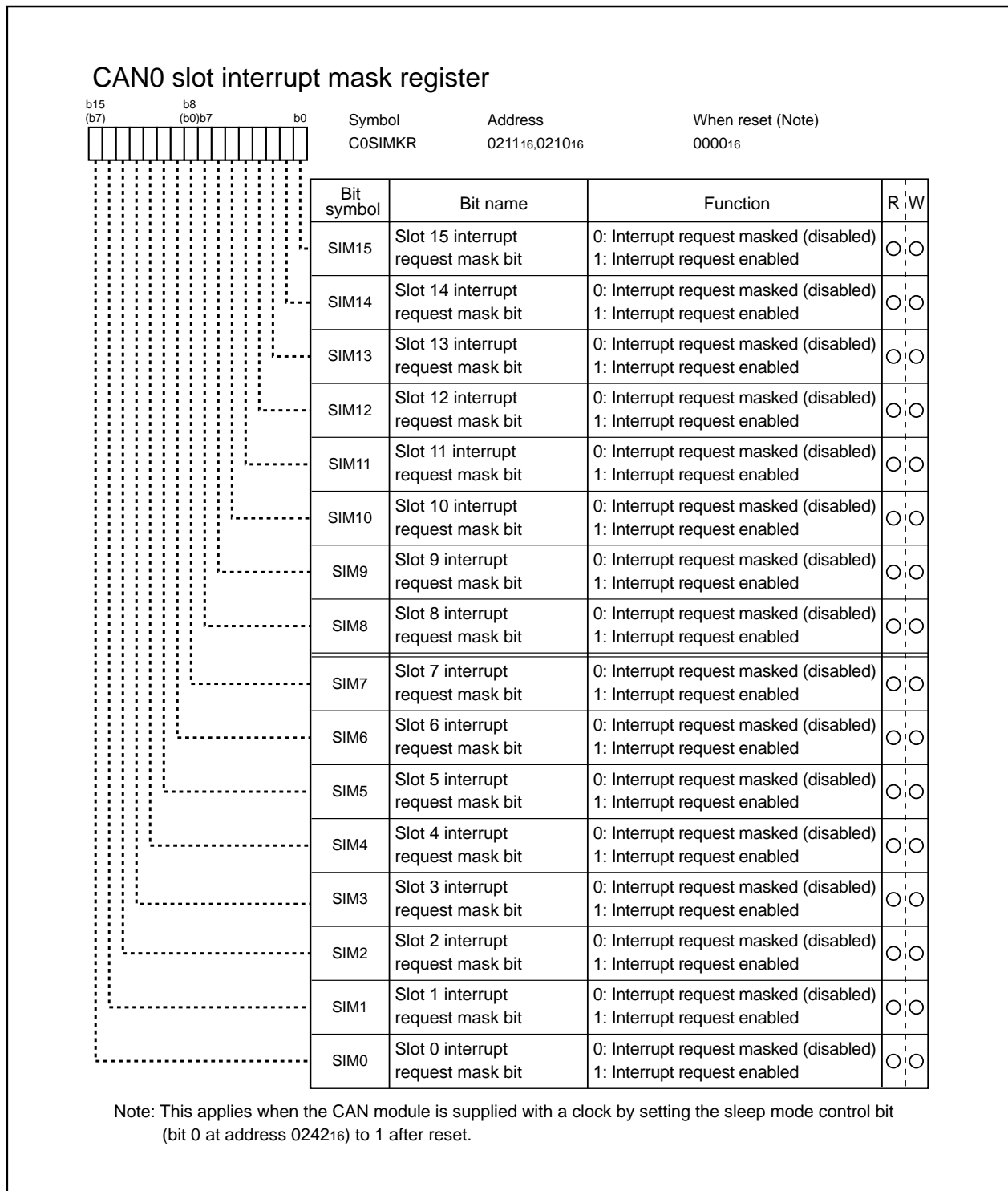


Figure 1.22.14. CAN0 slot interrupt mask register

12. CAN0 slot interrupt mask register

This register controls CAN interrupts by enabling or disabling interrupt requests generated by each corresponding slot at completion of transmission or reception. Setting any bit of this register (SIM_n where $n = 0-15$) to 1 enables the interrupt request to be generated by the corresponding slot at completion of transmission or reception.

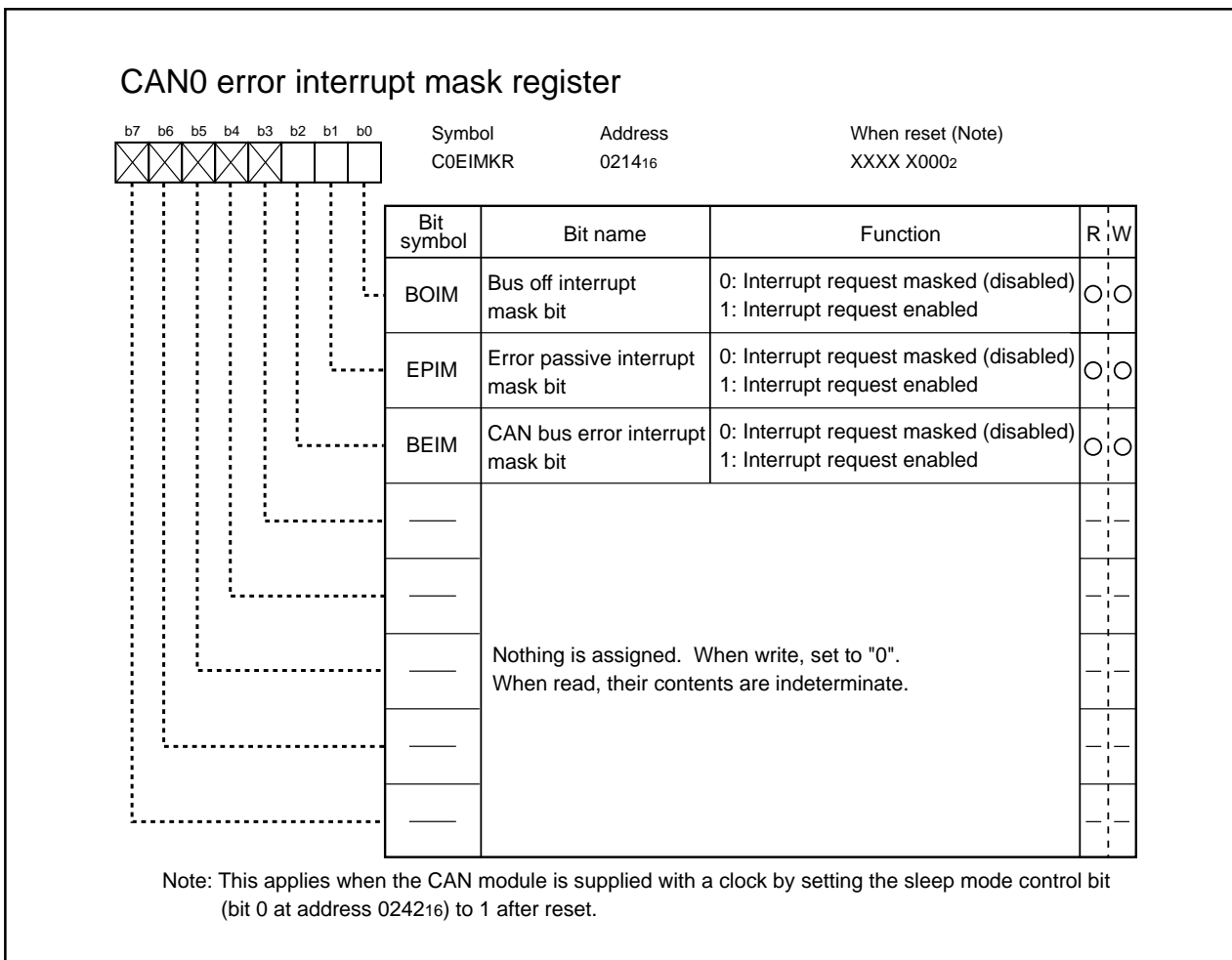


Figure 1.22.15. CAN0 error interrupt mask register

13. CAN0 error interrupt mask register

Bit 0: Bus-off interrupt mask bit (BOIM)

This bit controls CAN interrupts by enabling or disabling interrupt requests generated when the CAN module goes to a bus-off state. Setting this bit to 1 enables a bus-off interrupt request.

Bit 1: Error passive interrupt mask bit (EPIM)

This bit controls CAN interrupts by enabling or disabling interrupt requests generated when the CAN module goes to an error passive state. Setting this bit to 1 enables an error passive interrupt request.

Bit 2: CAN bus error interrupt mask bit (BEIM)

This bit controls CAN interrupts by enabling or disabling interrupt requests generated by occurrence of a CAN bus error. Setting this bit to 1 enables a CAN bus error interrupt request.

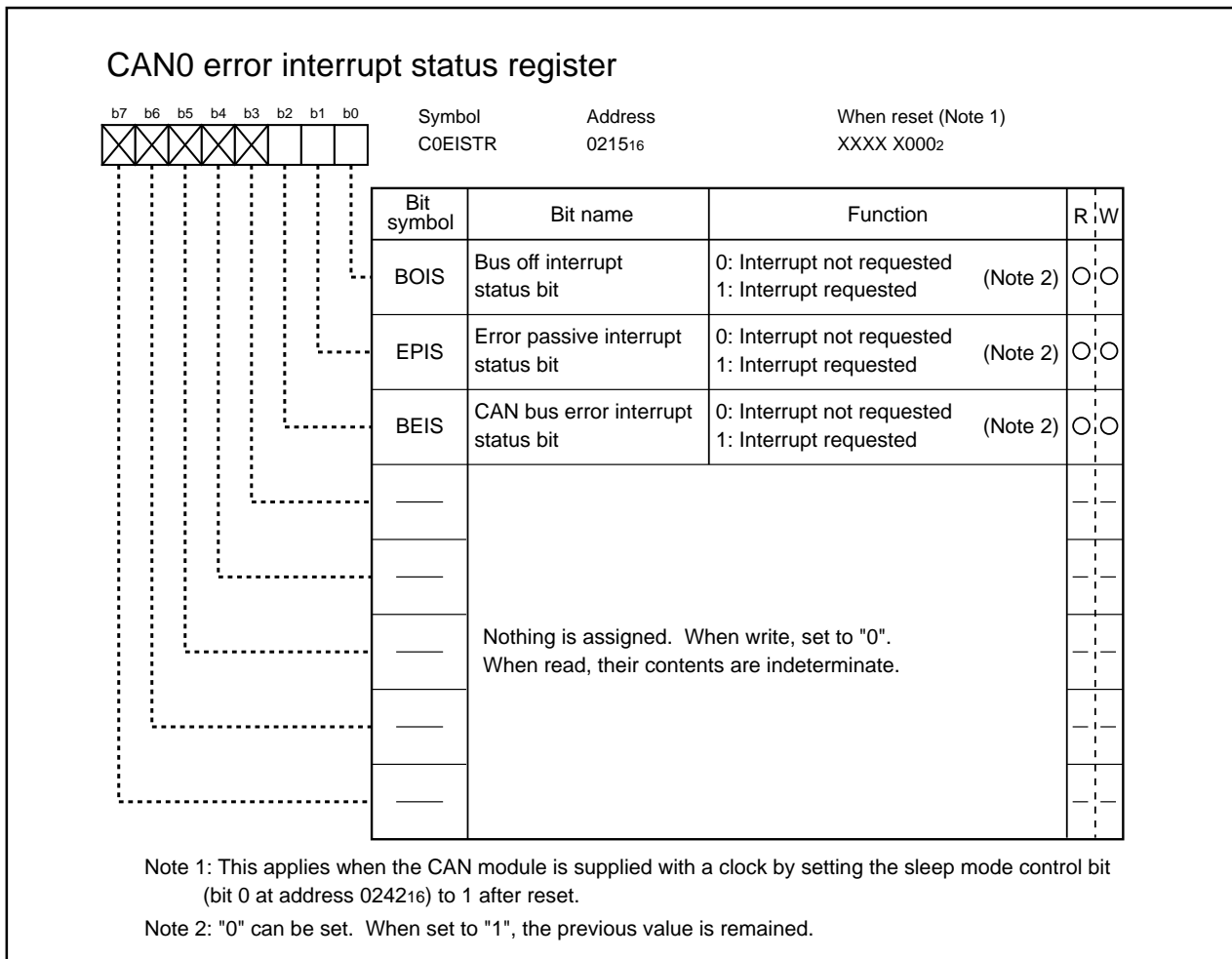


Figure 1.22.16. CAN0 error interrupt status register

14. CAN0 error interrupt status register

When using CAN interrupts, the CAN Error Interrupt Status Register helps to verify the causes of error-derived interrupts.

Bit 0: Bus-off interrupt status bit (BOIS)

This bit is set to 1 when the CAN module goes to a bus-off state.
To clear this bit, write 0 in software (Note 1).

Bit 1: Error passive interrupt status bit (EPIS)

This bit is set to 1 when the CAN module goes to an error passive state.
To clear this bit, write 0 in software (Note 1).

Bit 2: CAN bus error interrupt status bit (BEIS)

This bit is set to 1 when a CAN communication error is detected.
To clear this bit, write 0 in software (Note 1).

Note 1: To clear any bit of the CAN Error Interrupt Status Register, write 0 to the bit to be cleared and 1 to all other bits, without using bit clear instructions.

```
Example: Assembler language   mov.B   #006h, C0EISTR
          C language           c0eistr = 0x06;
```

CAN Module

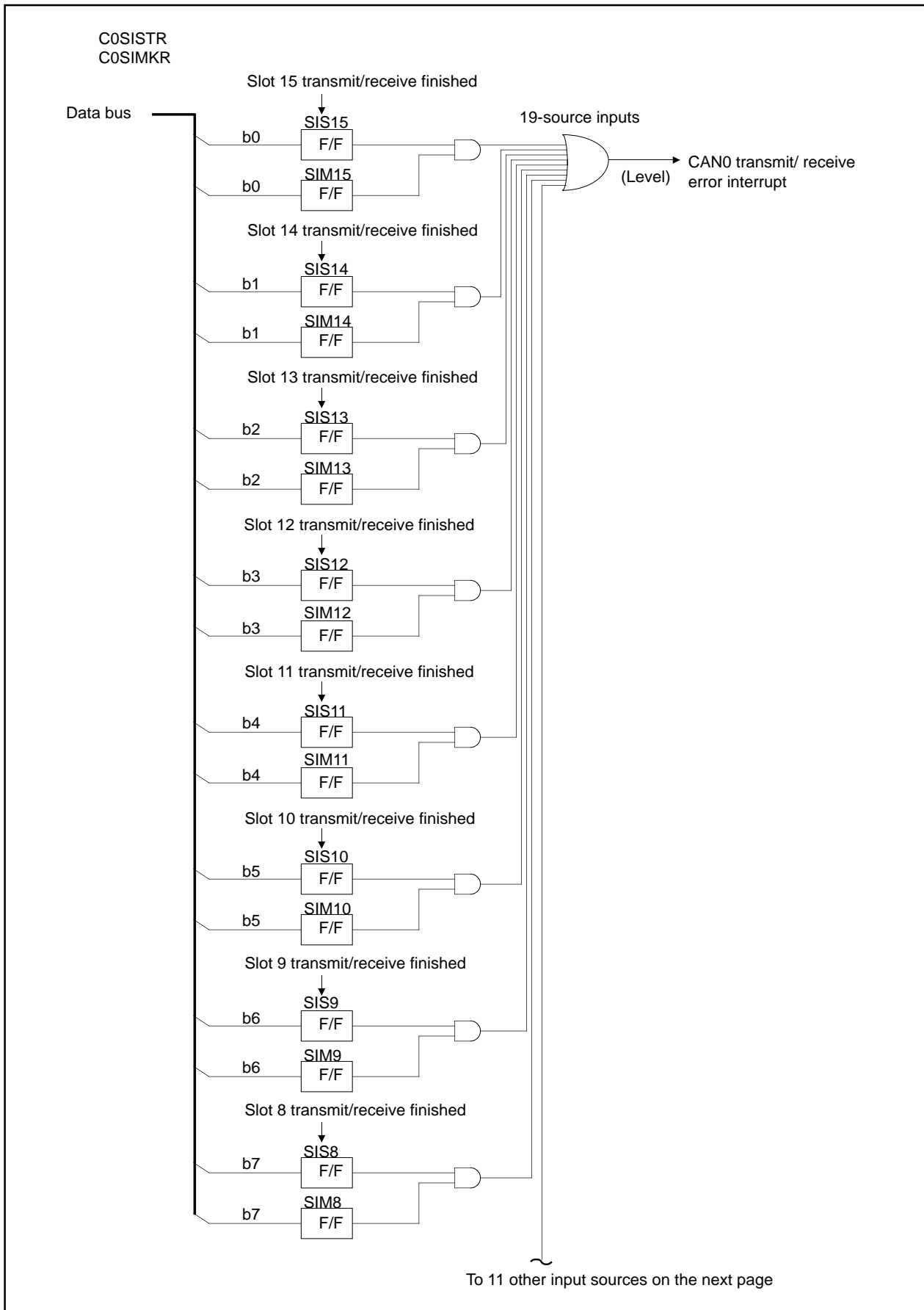


Figure 1.22.17. CAN0 transmit, receive and error interrupt block diagram (1/3)

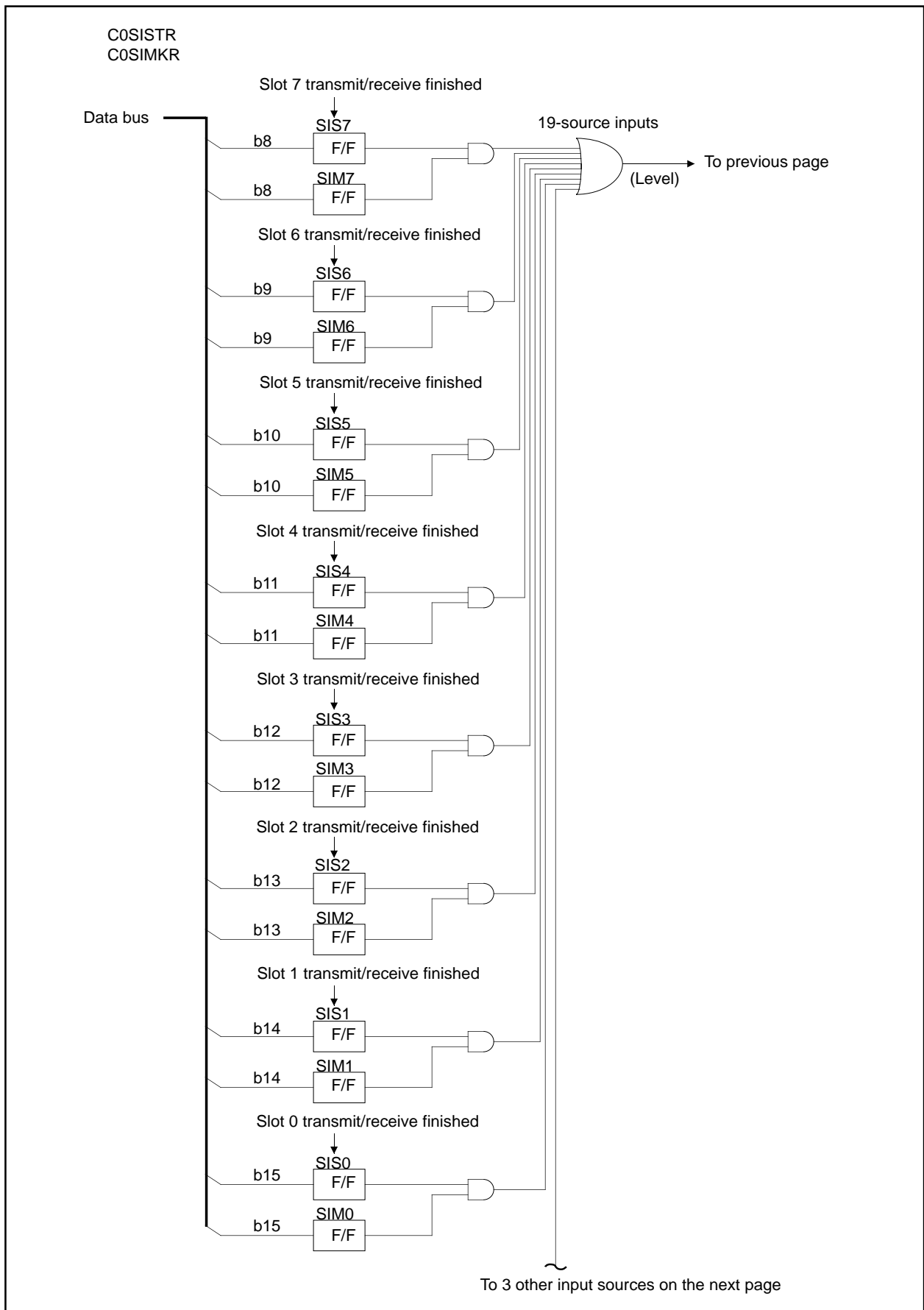


Figure 1.22.18. CAN0 transmit, receive and error interrupt block diagram (2/3)

CAN Module

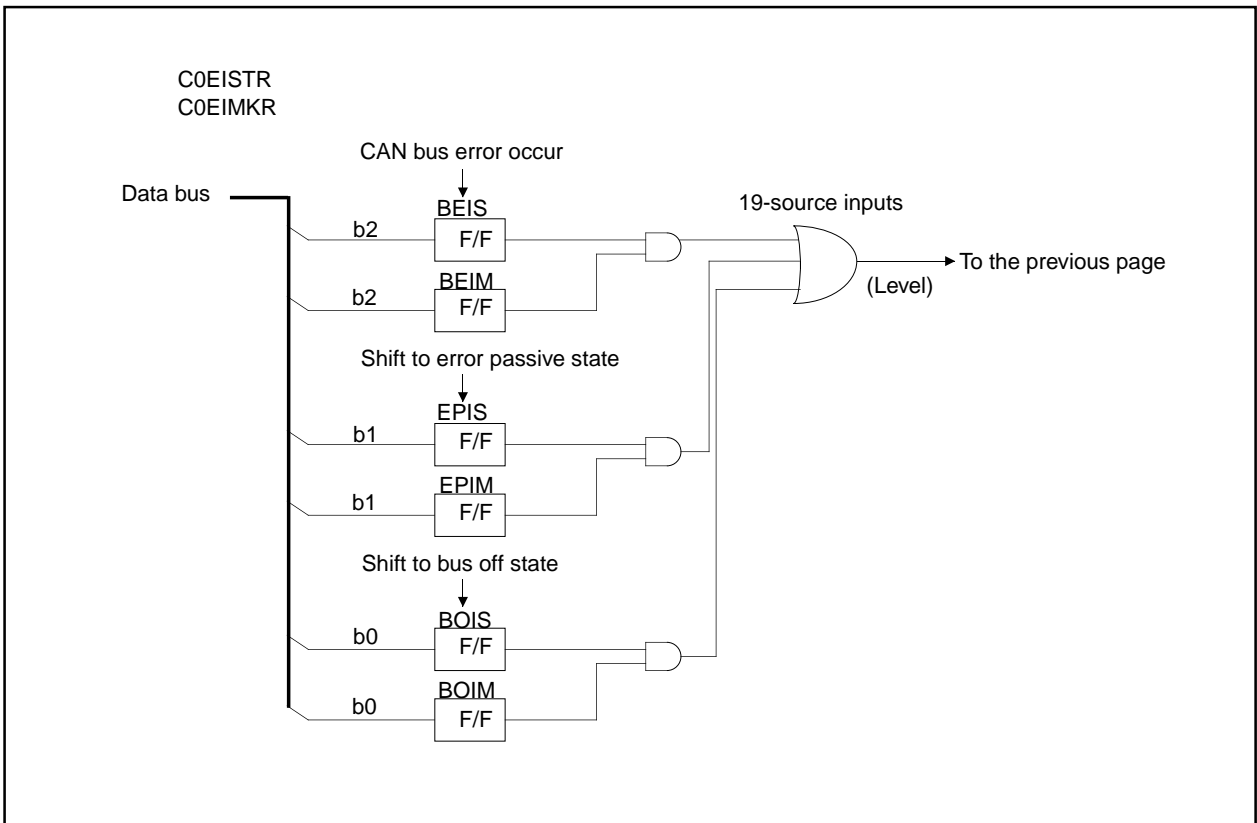


Figure 1.22.19. CAN0 transmit, receive and error interrupt block diagram (3/3)

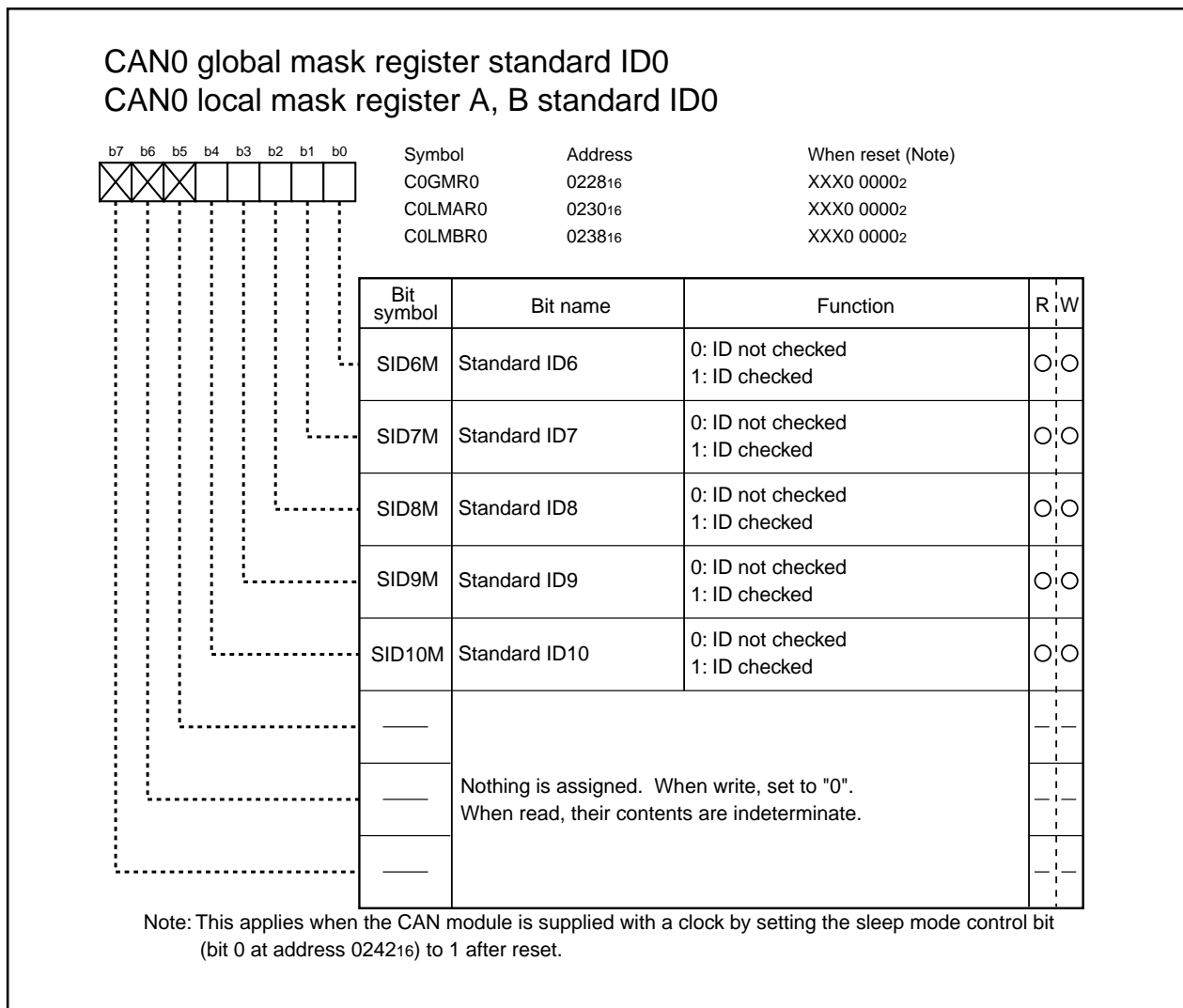


Figure 1.22.20. CAN0 global mask register standard ID0 and CAN0 local mask register A, B standard ID0

15. CAN0 global mask register standard ID0

CAN0 local mask register A, B standard ID0

The mask registers used for acceptance filtering consist of the global mask register, local mask register A, and local mask register B.

The global mask register takes care of message slots 0–13 whereas local mask registers A and B are used for message slots 14 and 15, respectively.

- If any bit of this register is set to 0, its corresponding ID bit is masked during acceptance filtering. (The masked bit is not checked for ID; the ID is assumed to be matching.)
- If any bit of this register is set to 1, its corresponding ID bit is compared with the received ID during acceptance filtering. If it matches the ID that is set in any message slot, the received data is stored in that slot.

Note 1: The global mask register can only be modified when none of the slots 0–13 has receive requests set.

Note 2: The local mask register A can only be modified when slot 14 has no receive requests set.

Note 3: The local mask register B can only be modified when slot 15 has no receive requests set.

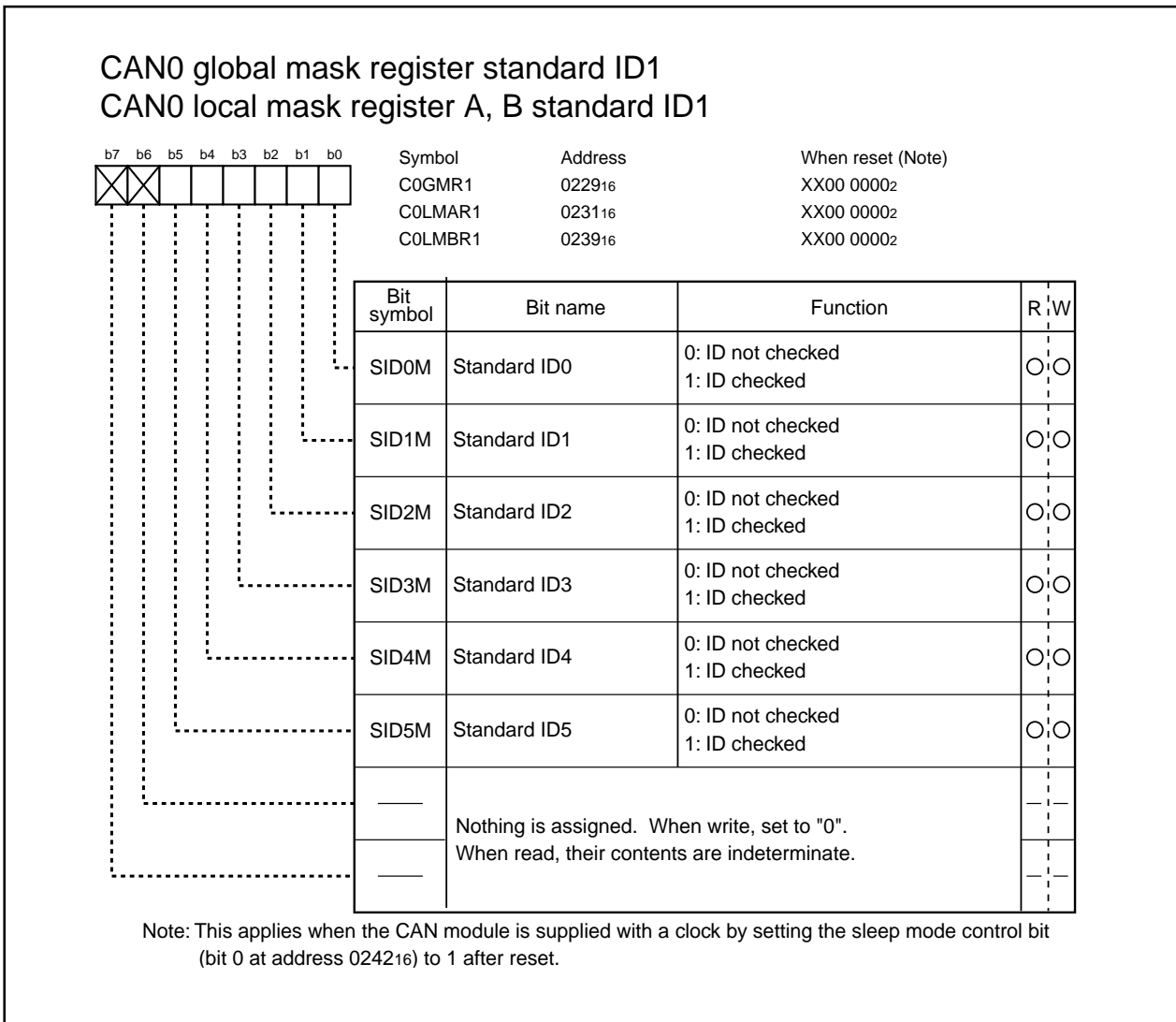


Figure 1.22.21. CAN0 global mask register standard ID1 and CAN0 local mask register A, B standard ID1

16. CAN0 global mask register standard ID1

CAN0 local mask register A, B standard ID1

The mask registers used for acceptance filtering consist of the global mask register, local mask register A, and local mask register B.

The global mask register takes care of message slots 0–13 whereas local mask registers A and B are used for message slots 14 and 15, respectively.

- If any bit of this register is set to 0, its corresponding ID bit is masked during acceptance filtering. (The masked bit is not checked for ID; the ID is assumed to be matching.)
- If any bit of this register is set to 1, its corresponding ID bit is compared with the received ID during acceptance filtering. If it matches the ID that is set in any message slot, the received data is stored in that slot.

Note 1: The global mask register can only be modified when none of the slots 0–13 has receive requests set.

Note 2: The local mask register A can only be modified when slot 14 has no receive requests set.

Note 3: The local mask register B can only be modified when slot 15 has no receive requests set.

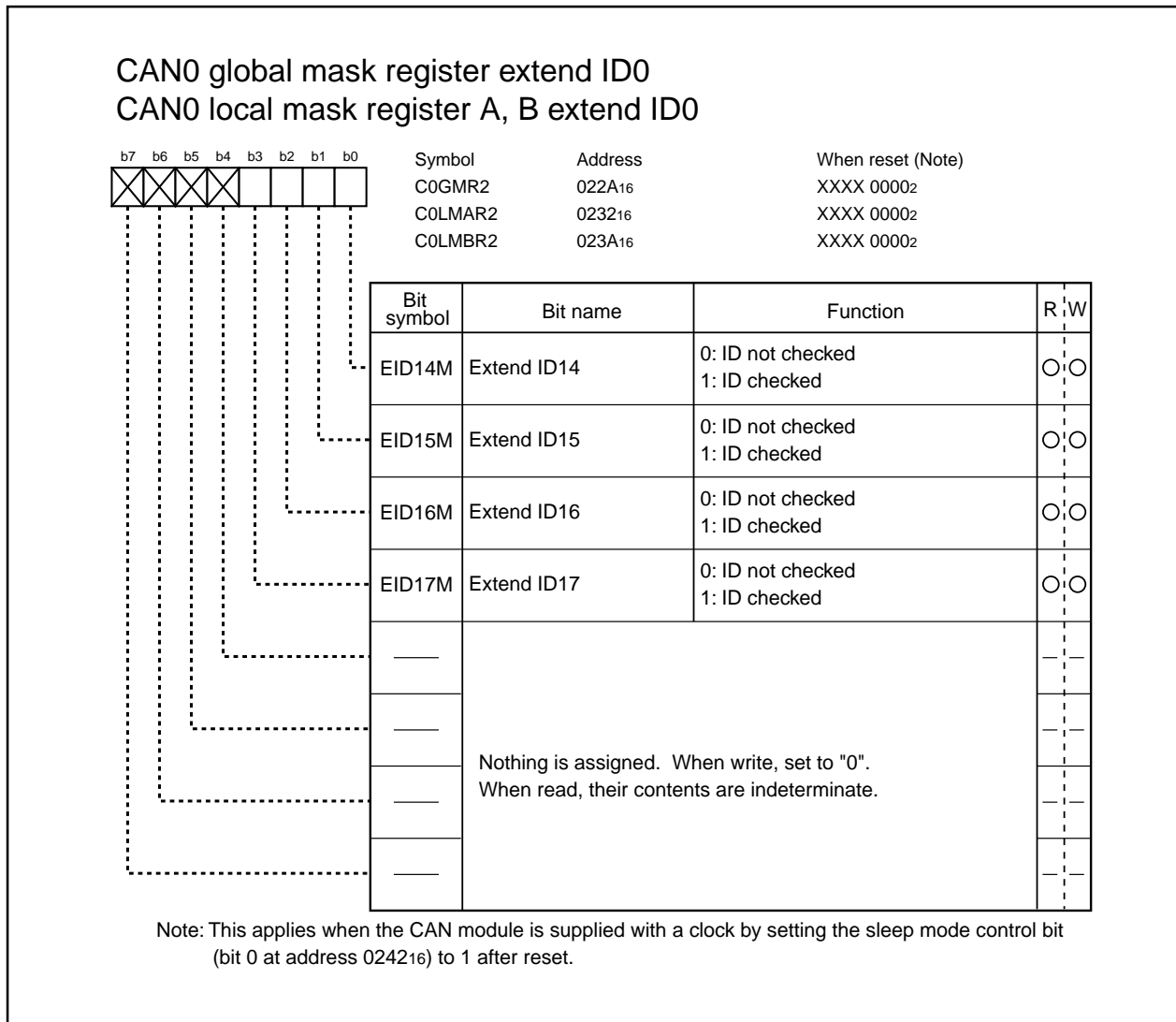


Figure 1.22.22. CAN0 global mask register extend ID0 and CAN0 local mask register A, B extend ID0

17. CAN0 global mask register extend ID0

CAN0 local mask register A, B extend ID0

The mask registers used for acceptance filtering consist of the global mask register, local mask register A, and local mask register B.

The global mask register takes care of message slots 0–13 whereas local mask registers A and B are used for message slots 14 and 15, respectively.

- If any bit of this register is set to 0, its corresponding ID bit is masked during acceptance filtering. (The masked bit is not checked for ID; the ID is assumed to be matching.)
- If any bit of this register is set to 1, its corresponding ID bit is compared with the received ID during acceptance filtering. If it matches the ID that is set in any message slot, the received data is stored in that slot.

Note 1: The global mask register can only be modified when none of the slots 0–13 has receive requests set.

Note 2: The local mask register A can only be modified when slot 14 has no receive requests set.

Note 3: The local mask register B can only be modified when slot 15 has no receive requests set.

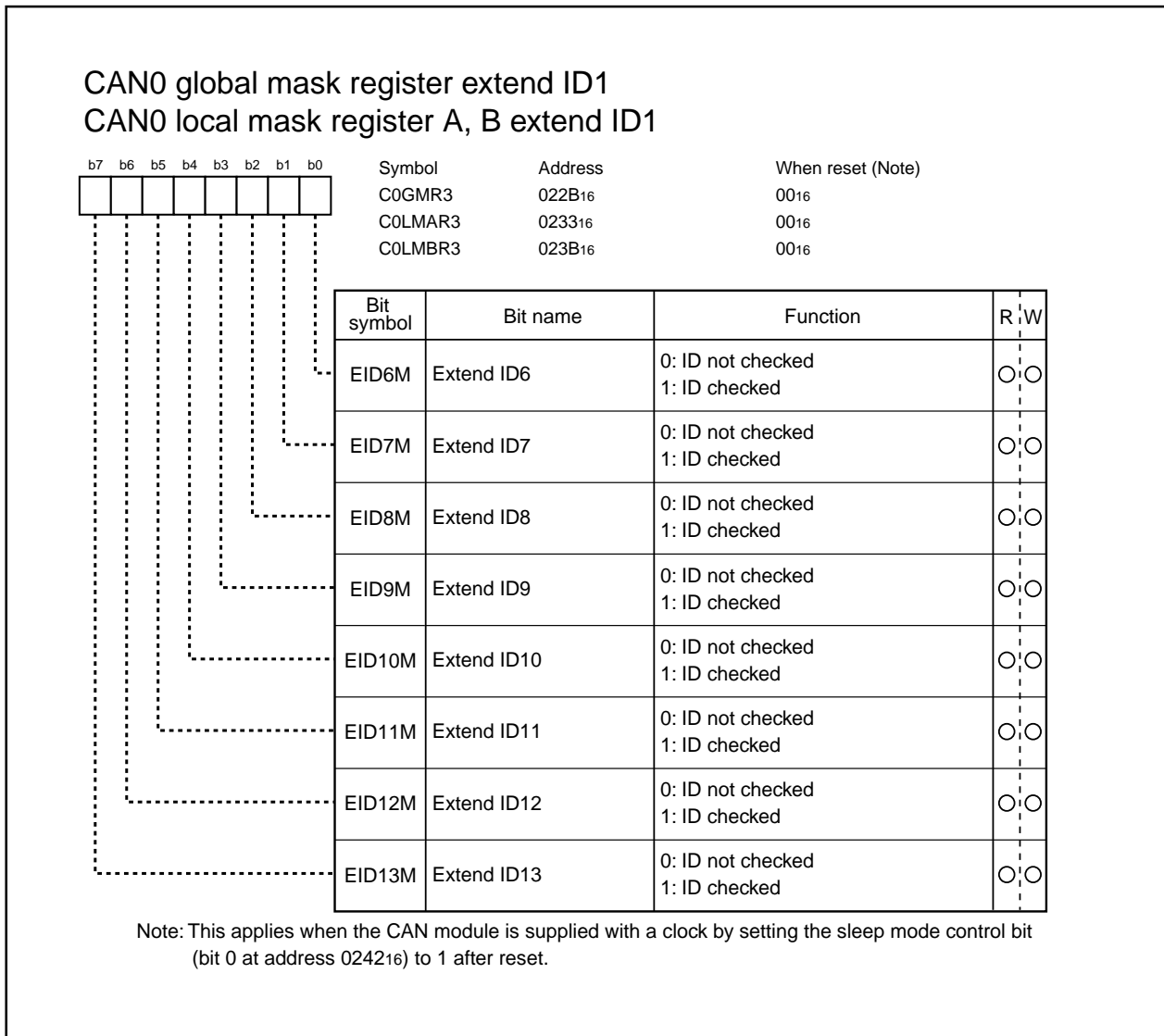


Figure 1.22.23. CAN0 global mask register extend ID1 and CAN0 local mask register A, B extend ID1

18. CAN0 global mask register extend ID1

CAN0 local mask register A, B extend ID1

The mask registers used for acceptance filtering consist of the global mask register, local mask register A, and local mask register B.

The global mask register takes care of message slots 0–13, whereas local mask registers A and B are used for message slots 14 and 15, respectively.

- If any bit of this register is set to 0, its corresponding ID bit is masked during acceptance filtering. (The masked bit is not checked for ID; the ID is assumed to be matching.)
- If any bit of this register is set to 1, its corresponding ID bit is compared with the received ID during acceptance filtering. If it matches the ID that is set in any message slot, the received data is stored in that slot.

Note 1: The global mask register can only be modified when none of the slots 0–13 has receive requests set.

Note 2: The local mask register A can only be modified when slot 14 has no receive requests set.

Note 3: The local mask register B can only be modified when slot 15 has no receive requests set.

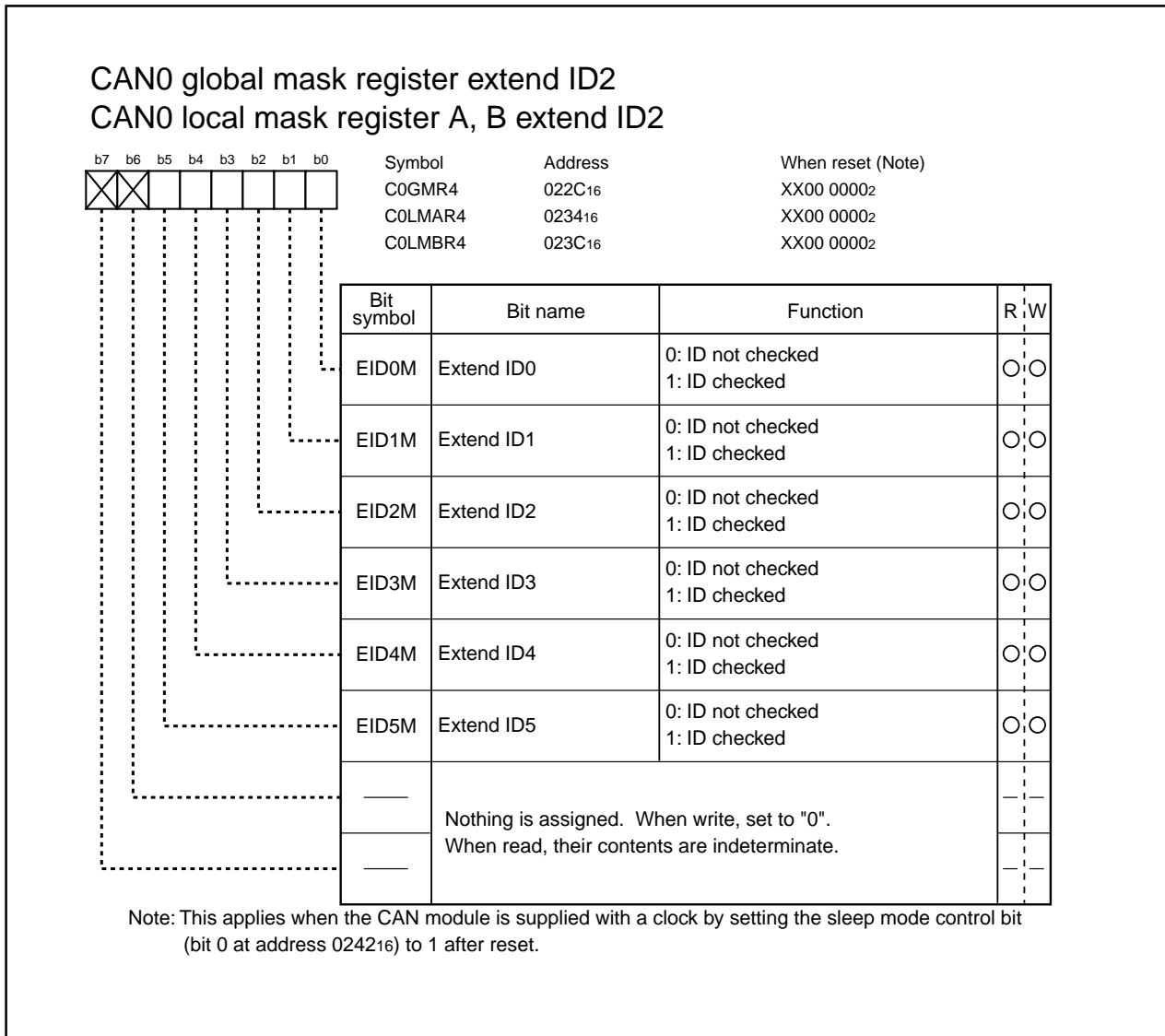


Figure 1.22.24. CAN0 global mask register extend ID2 and CAN0 local mask register A, B extend ID2

19. CAN0 global mask register extend ID2

CAN0 local mask register A, B extend ID2

The mask registers used for acceptance filtering consist of the global mask register, local mask register A, and local mask register B.

The global mask register takes care of message slots 0–13, whereas local mask registers A and B are used for message slots 14 and 15, respectively.

- If any bit of this register is set to 0, its corresponding ID bit is masked during acceptance filtering. (The masked bit is not checked for ID; the ID is assumed to be matching.)
- If any bit of this register is set to 1, its corresponding ID bit is compared with the received ID during acceptance filtering. If it matches the ID that is set in any message slot, the received data is stored in that slot.

Note 1: The global mask register can only be modified when none of the slots 0–13 has receive requests set.

Note 2: The local mask register A can only be modified when slot 14 has no receive requests set.

Note 3: The local mask register B can only be modified when slot 15 has no receive requests set.

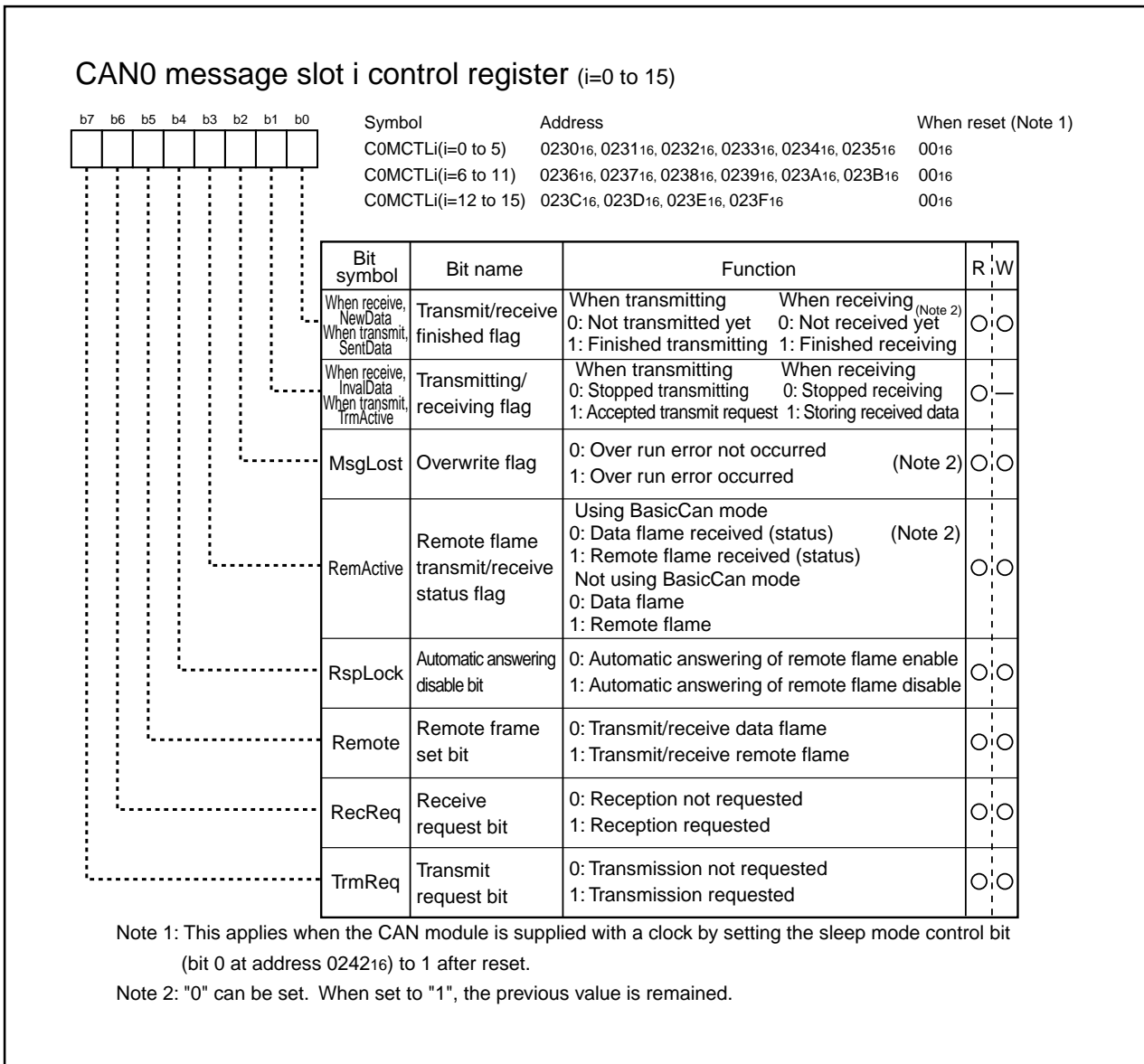


Figure 1.22.25. CAN0 message slot i control register

20. CAN0 message slot i control register

Bit 0: Transmission finished flag /reception finished flag (SentData, NewData)

This bit indicates that the CAN module finished transmitting or receiving a message.

- For transmit slots

The bit is set to 1 when the CAN module finished transmitting from the message slot.

This bit is cleared by writing 0 in software. However, it cannot be cleared when the TrmActive (transmit/receive status) bit = 1.

- For receive slots

The bit is set to 1 when the CAN module finished receiving a message normally that is to be stored in the message slot.

This bit is cleared by writing 0 in software. However, it cannot be cleared when the InvalData (transmit/receive status) bit = 1.

Note 1: Before reading received data from the message slot, be sure to clear the NewData (transmission/reception finished status) bit. Also, if the NewData bit is set to 1 after readout, it means that new received data has been stored in the message slot while reading out from the slot, and that the read data contains an indeterminate value. In this case, discard the read data and clear the NewData bit before reading out from the slot again.

Note 2: The NewData bit is not set by a completion of remote frame transmission or reception.

Bit 1: Transmitting flag /receiving flag (TrmActive, InvalData)

This bit indicates that the CAN module is transmitting or receiving a message, with the message slot being accessed. The bit is set to 1 when the CAN module is accessing the message slot and set to 0 when not accessing the message slot.

- For transmit slots

This bit is set to 1 when the message slot has its transmit request accepted. If the message slot failed in arbitration, this bit is cleared to 0 by occurrence of a CAN bus error or completion of transmission.

- For receive slots

This bit is set to 1 when the CAN module is receiving a message, with the received message being stored in the message slot. Note that the value read out from the message slot while this bit remains set is indeterminate.

Bit 2: Overwrite flag (MsgLost)

This bit is useful for the receive slots, those that are set for reception. This bit is set to 1 when while the message slot contains an unread received message, it is overwritten by a new received message.

This bit is cleared by writing 0 in software.

Bit 3: Remote frame transmit/receive status flag (RemActive)

This bit functions differently for slots 0–13 and slots 14, 15.

- For slots 0–13

If the slot is set for remote frame transmission (or reception), this bit is set to 1. Then, when the slot finished transmitting (or receiving) a remote frame, this bit is cleared to 0.

- For slots 14 and 15

The RemActive bit functions differently depending on how the CAN Control Register's BasicCAN (BasicCAN mode) bit is set.

When BasicCAN = 0 (operating normally), if the slot is set for remote frame transmission (or reception), the RemActive bit is set to 1.

When BasicCAN = 1 (operating in BasicCAN mode), the RemActive bit indicates which frame type of message was received. During BasicCAN mode, slots 14 and 15 store the received data whether it be a data frame or a remote frame.

If RemActive = 0, it means that the message stored in the slot is a data frame.

If RemActive = 1, it means that the message stored in the slot is a remote frame.

Bit 4: Automatic answering disable bit (RspLock)

This bit is useful for the slots set for remote frame reception, indicating the processing to be performed after receiving a remote frame.

If this bit is set to 0, the slot automatically changes to a transmit slot after receiving a remote frame and the message stored in the slot is transmitted as a data frame.

If this bit is set to 1, the slot stops operating after receiving a remote frame.

Note 1: This bit must always be set to 0 for any slots other than those set for remote frame reception.

Bit 5: Remote frame set bit (Remote)

Set this bit to 1 for the message slots that handle a remote frame.

Message slots can be set to handle a remote frame in the following two ways.

- Set to transmit a remote frame and receive a data frame

The message stored in the message slot is transmitted as a remote frame. The slot automatically changes to a data frame receive slot after it finished transmitting.

However, if it receives a data frame before it finishes transmitting a remote frame, the data frame is stored in the message slot and the remote frame is not transmitted.

- Set to receive a remote frame and transmit a data frame

The slot receives a remote frame. The processing to be performed after receiving a remote frame depends on how the RspLock (automatic answering disable) bit is set.

Bit 6: Receive request bit (RecReq)

Set this bit to 1 when using any message slot as a receive slot.

Set this bit to 0 when using any message slot as a data frame transmit or remote frame transmit slot.

If the TrmReq (transmit request) bit and RecReq (receive request) bit both are set to 1, the operation of the CAN module is indeterminate.

Bit 7: Transmit request bit (TrmReq)

Set this bit to 1 when using any message slot as a transmit slot.

Set this bit to 0 when using any message slot as a data frame receive or remote frame receive slot.

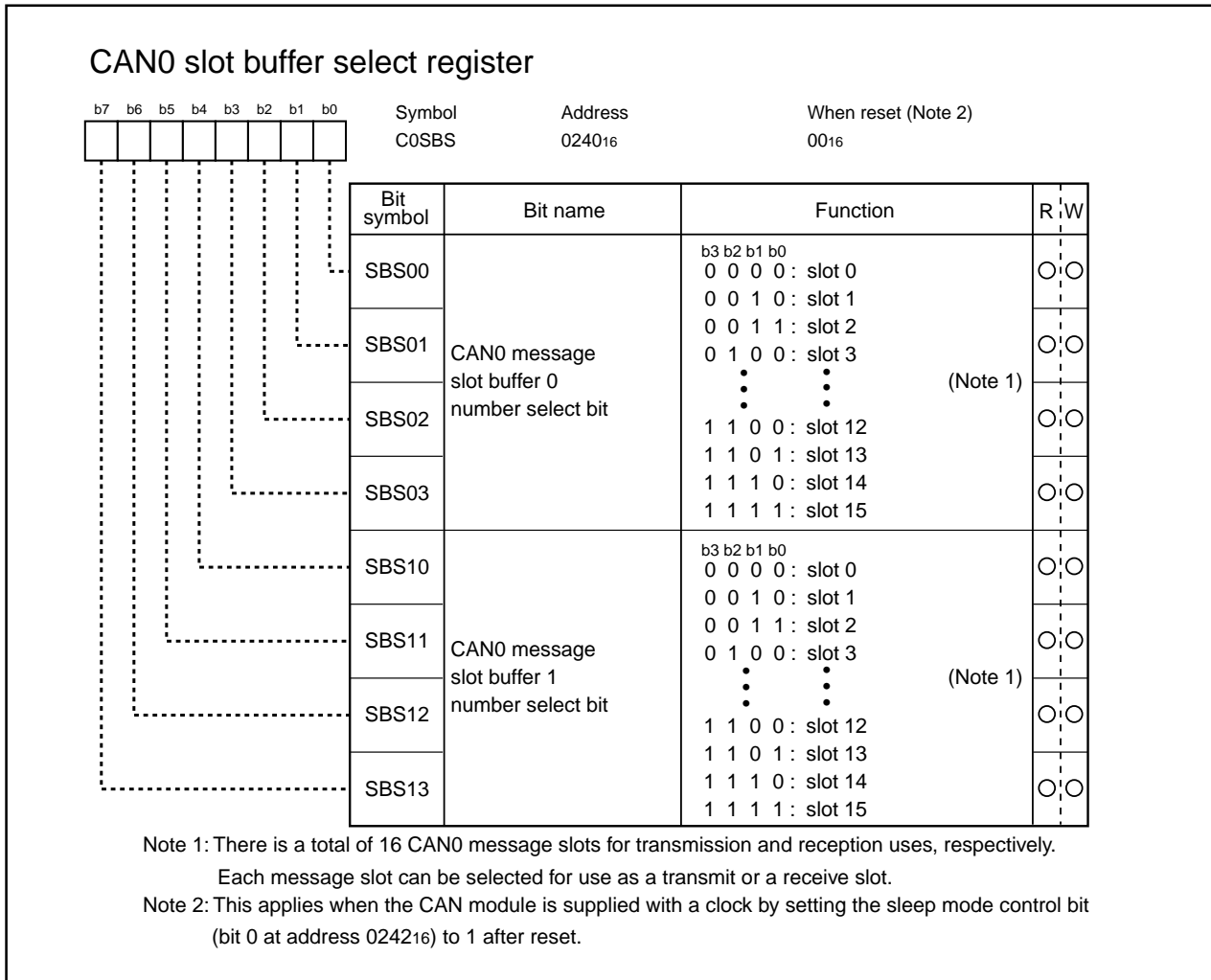


Figure 1.22.26. CAN0 slot buffer select register

21. CAN0 slot buffer select register

Bits 0-3: CAN0 message slot buffer 0 slot number select bits (SBS0)

The message slot whose number is selected with these bits appears in CAN0 message slot buffer 0.

Bits 4-7: CAN0 message slot buffer 1 slot number select bits (SBS1)

The message slot whose number is selected with these bits appears in CAN0 message slot buffer 1.

The selected message slot can be identified by reading the message slot buffer.

A message written to the message slot buffer is stored in the selected message slot.

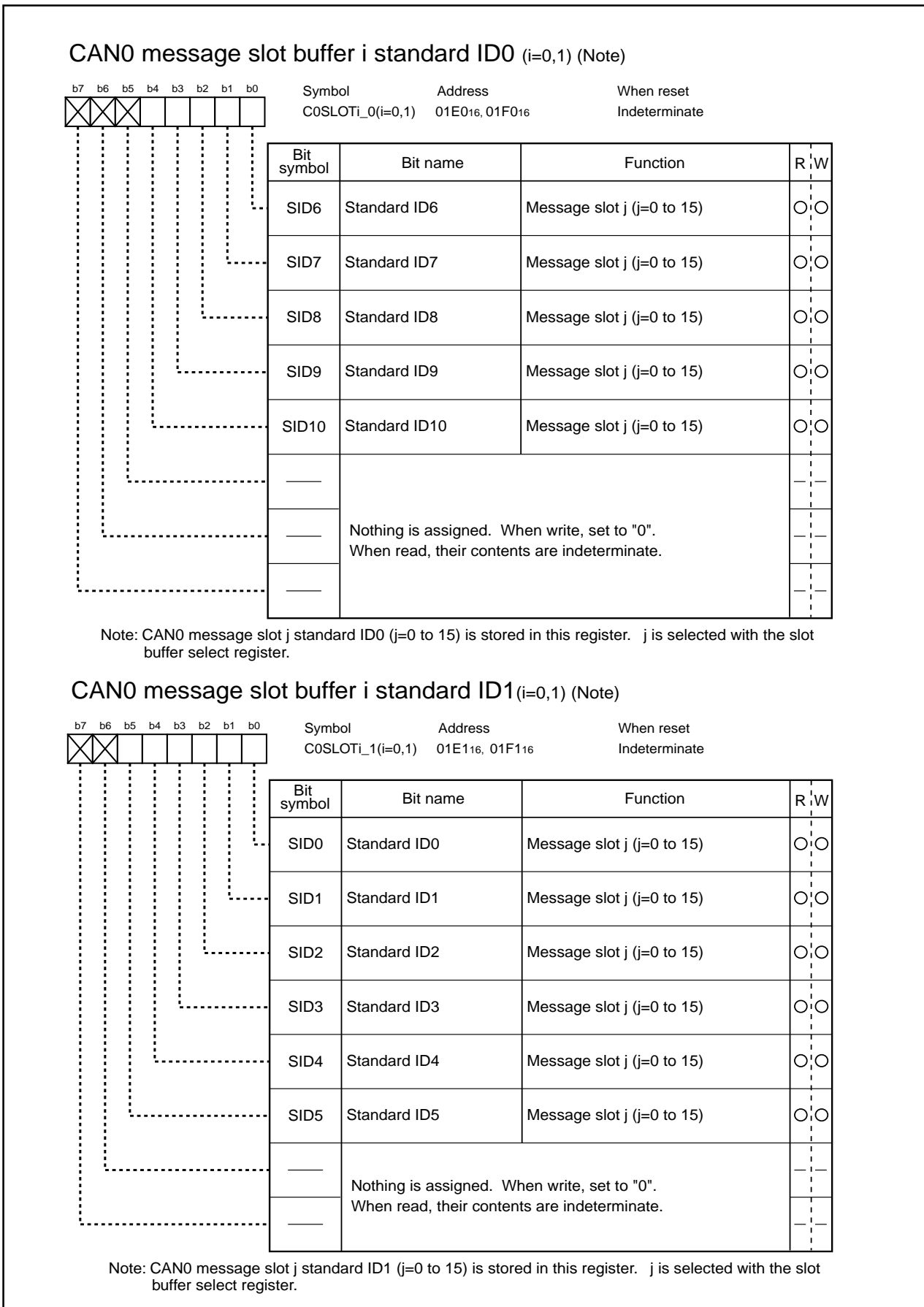


Figure 1.22.27. CAN0 message slot buffer i standard ID0 and ID1

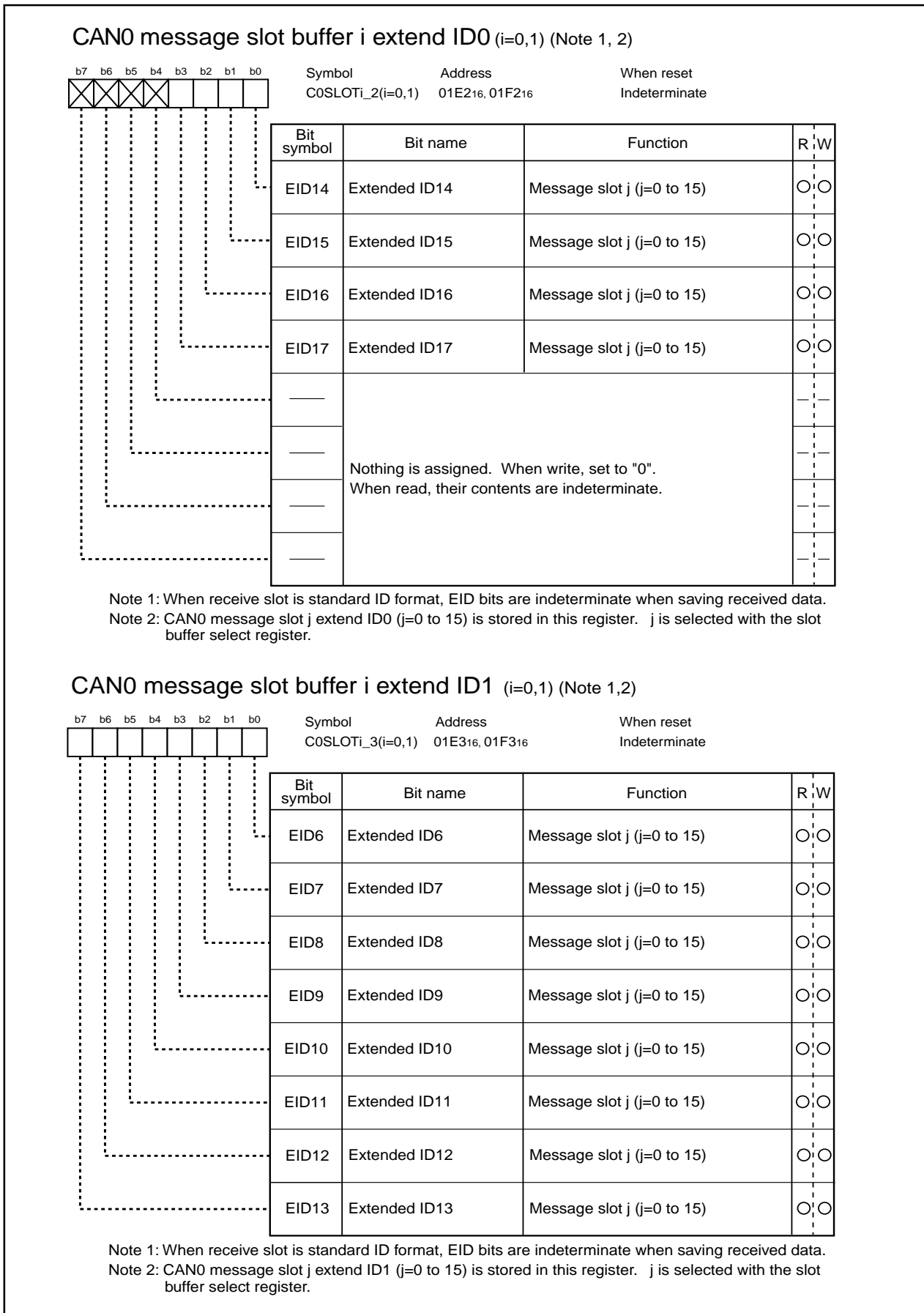


Figure 1.22.28. CAN0 message slot buffer i extended ID0 and ID1

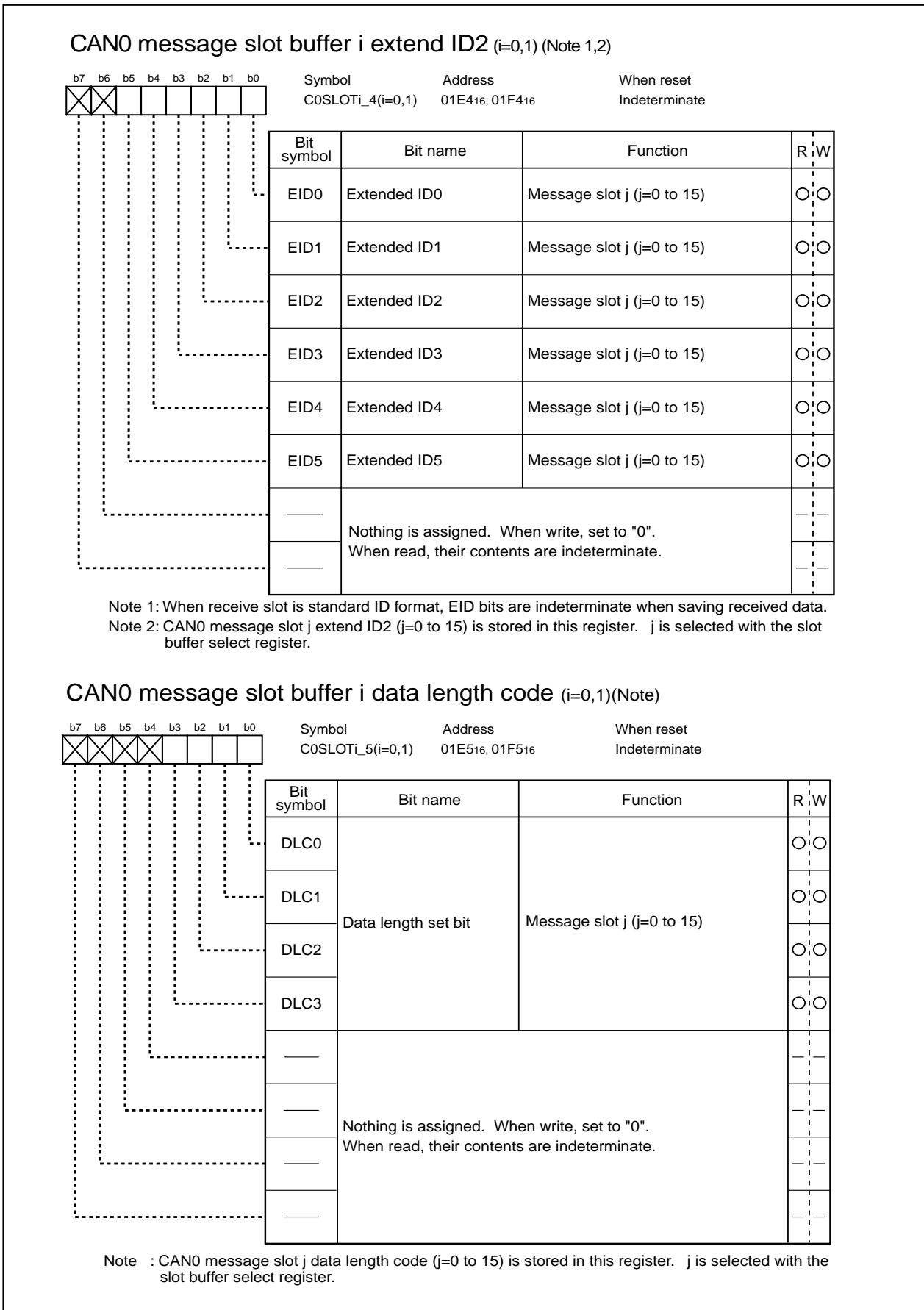


Figure 1.22.29. CAN0 message slot buffer i extended ID2 and CAN0 message slot buffer i data lengthcode

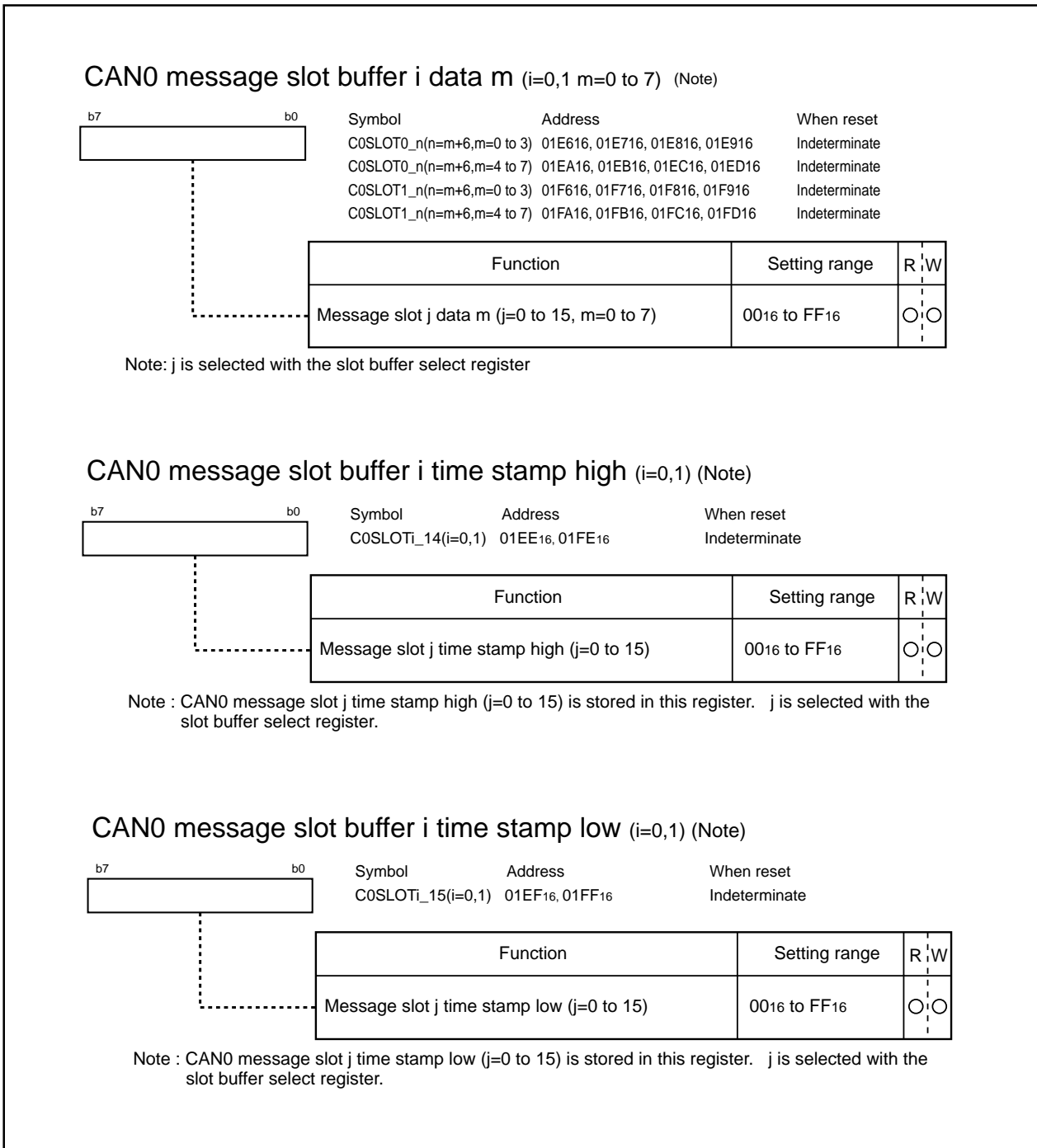


Figure 1.22.30. CAN0 message slot buffer i data m and CAN0 message slot buffer i time stamp

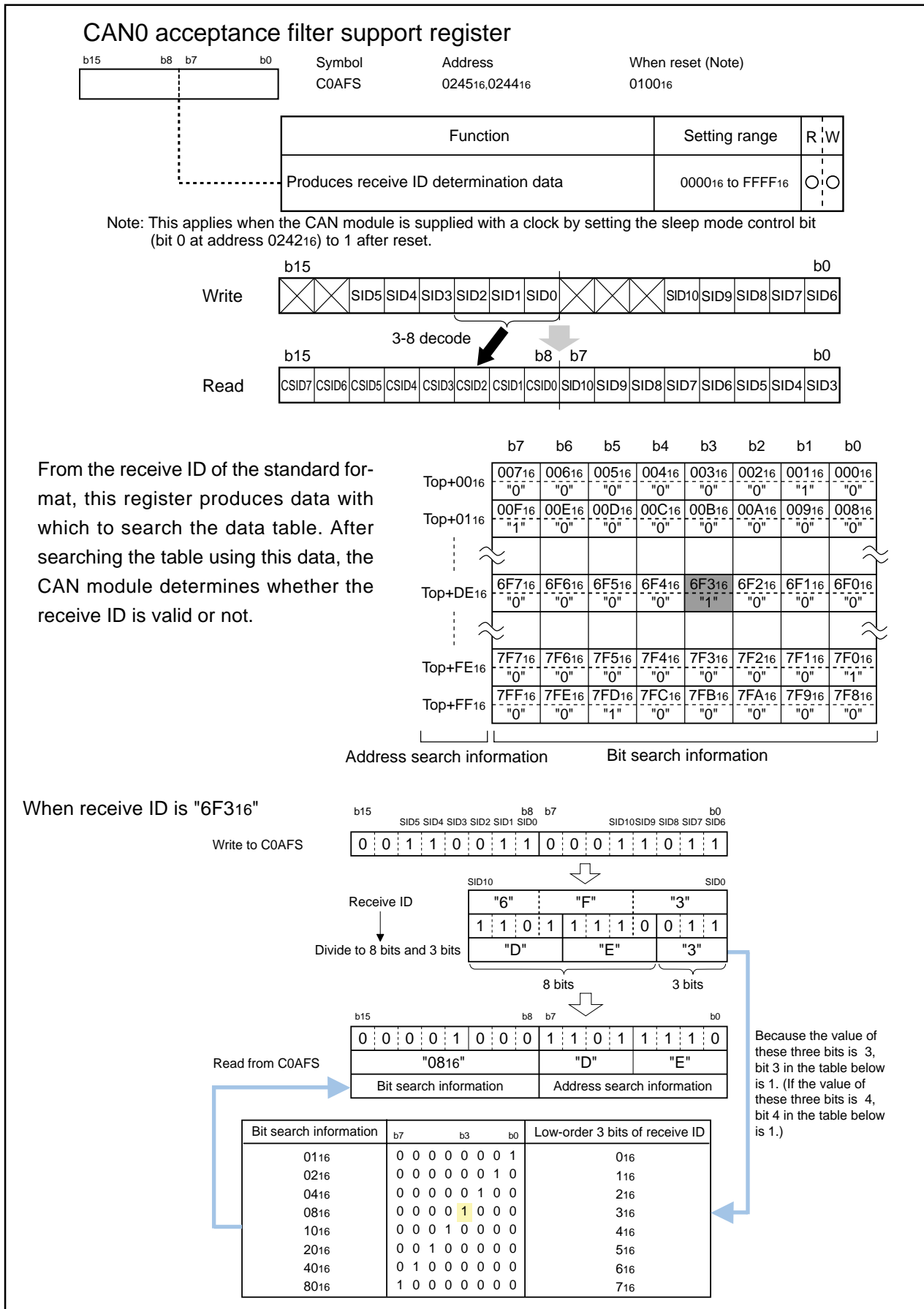


Figure 1.22.31. CAN0 acceptance filter support register

Intelligent I/O

Intelligent I/O uses multifunctional I/O ports for time measurement, waveform generation, clock-synchronous/asynchronous (UART) serial I/O, IE bus ^(Note) communications, HDLC data processing and more. A single Intelligent I/O group comes with one 16-bit base timer for free running, eight 16-bit registers for time measurement and waveform generation, and two shift registers for 8-bit and 16-bit communications.

The M32C/83 has four internal Intelligent I/O groups. Table 1.23.1 lists functions by group.

Table 1.23.1. List of functions of intelligent I/O

Function	Group 0	Group 1	Group 2	Group 3	Group 0,1 cascaded
Configuration					
•Base timer	1	1	1	1	1
•TM	4ch(2ch)	–	–	–	–
•TM/WG register (shared)	4chs(1ch)	4chs(2chs)	–	–	8chs(3chs)
•WG register	–	4chs(1ch)	8chs	8chs(3chs)	8chs(2chs)
•Communication shift register	8bits X 2chs	8bits X 2chs	8bits X 2chs	–	–
Time measurement functions	Max. 8chs (3chs)	Max. 4chs (2chs)	–	–	Max. 8chs (3chs)
•Digital filter function	√	√	–	–	√
•Trigger input prescale function	2chs	2chs	–	–	2chs
•Gate function for trigger input	2chs	2chs	–	–	2chs
WG function	Max. 4chs (1ch)	Max. 8chs (3chs)	Max. 8chs (3chs)	Max. 8chs (2chs)	Max. 8chs (1ch)
•Single phase waveform output	√	√	√	√	√
•Phase delayed waveform output	√	√	√	√	√
•Set/reset waveform output	√	√	√	√	√
•Bit modulation PWM output	–	–	√	√	–
•Real-time port output	–	–	√	√	–
•Parallel real-time port output	–	–	√	√	–
Communication functions					
•Bit length	8 bits fixed	8 bits fixed	Variable length	–	–
•Communication mode					
1. Clock synchronous serial I/O	√	√	√	–	–
2. UART	√	√	–	–	–
3. HDLC data processing	√	√	–	–	–
4. IE Bus sub set	–	–	√	–	–

Note 1: IE Bus is a trademark of NEC.

Note 2: 100-pin specification are in parentheses.

√ : Present

– : Not present

TM:Time Measurement

WG:Waveform Generation

Block diagrams for groups 0 to 3 are given in Figures 1.23.1 to 1.23.4.

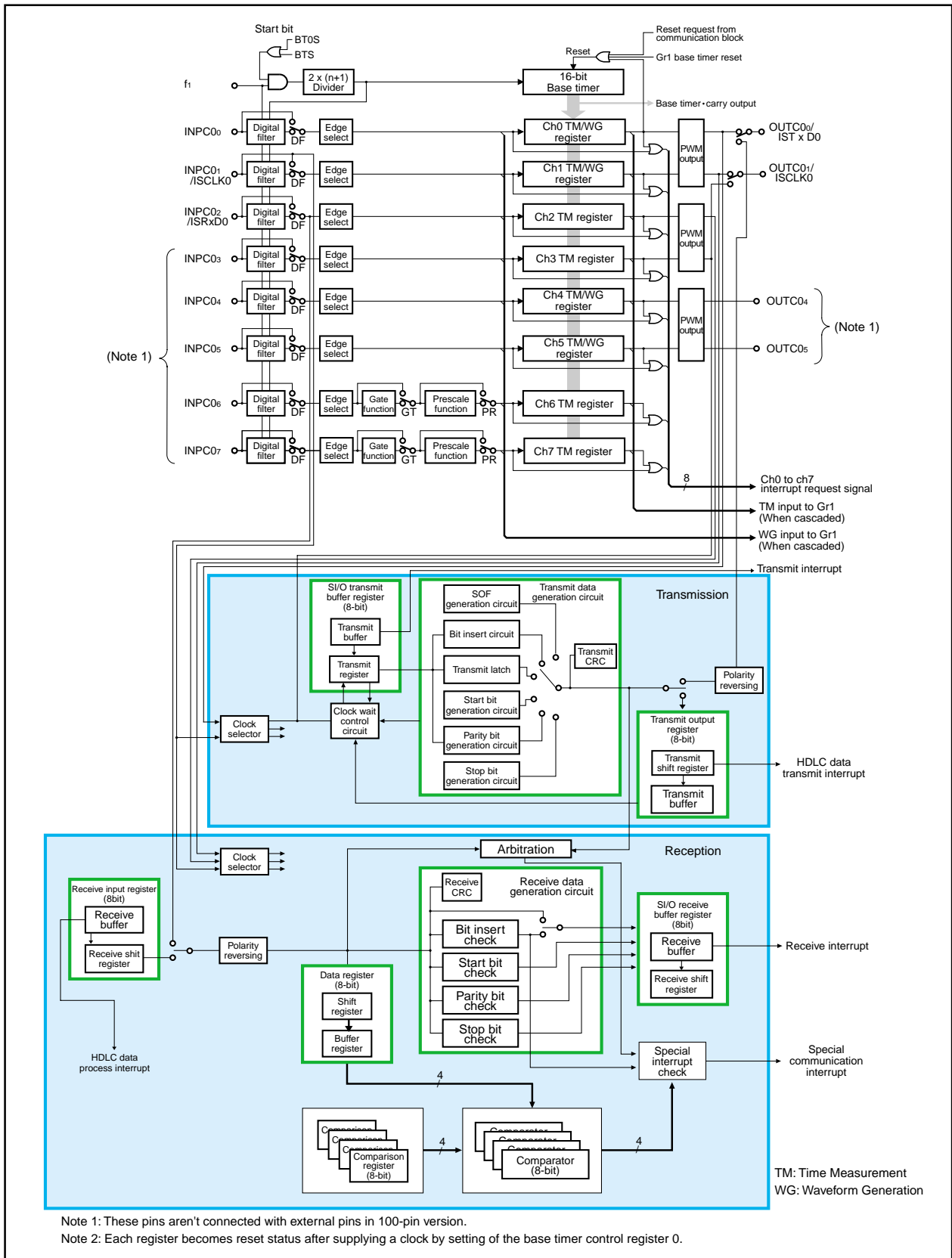


Figure 1. 23. 1. Block diagram of intelligent I/O group 0

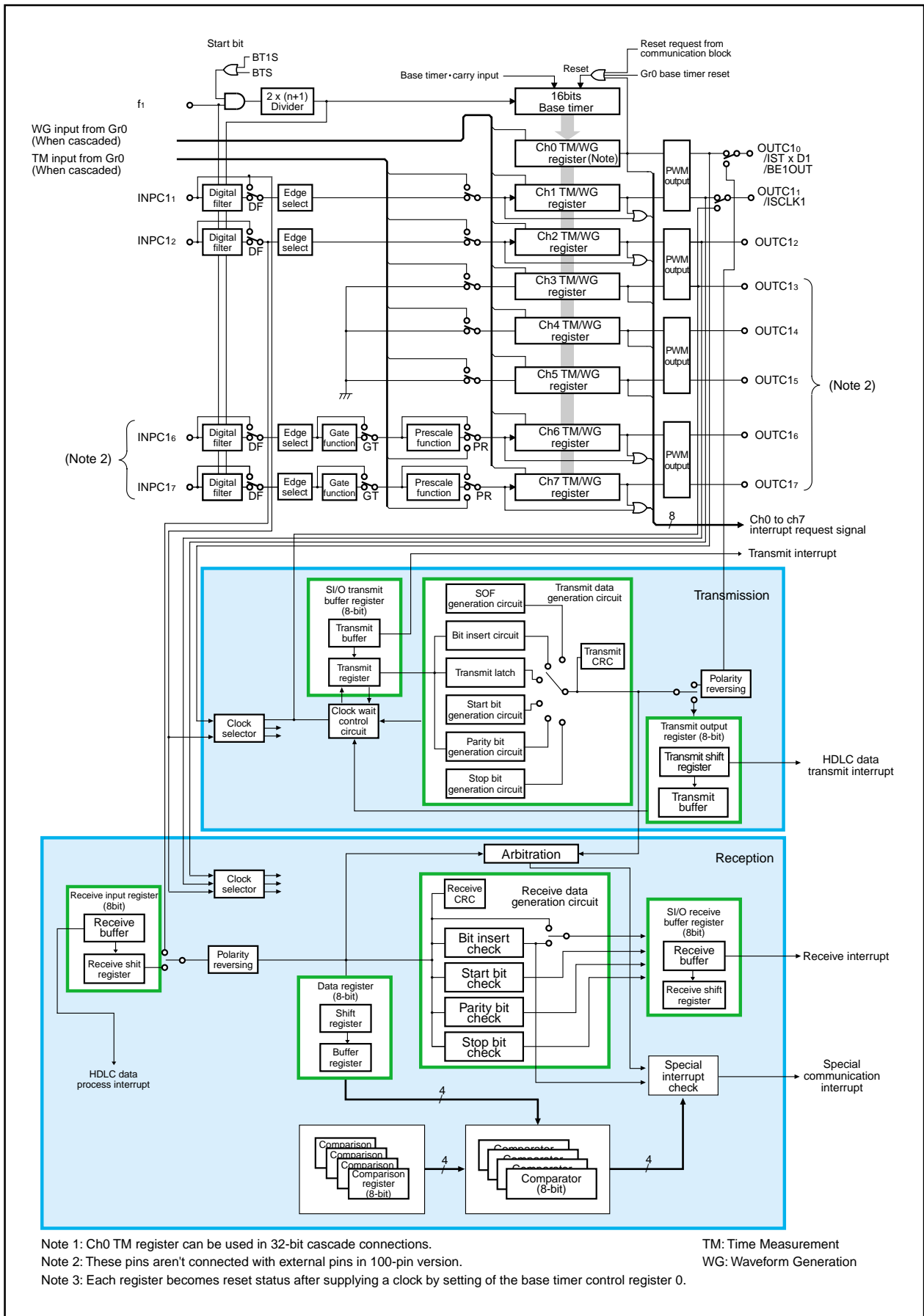


Figure 1. 23. 2. Block diagram of intelligent I/O group 1

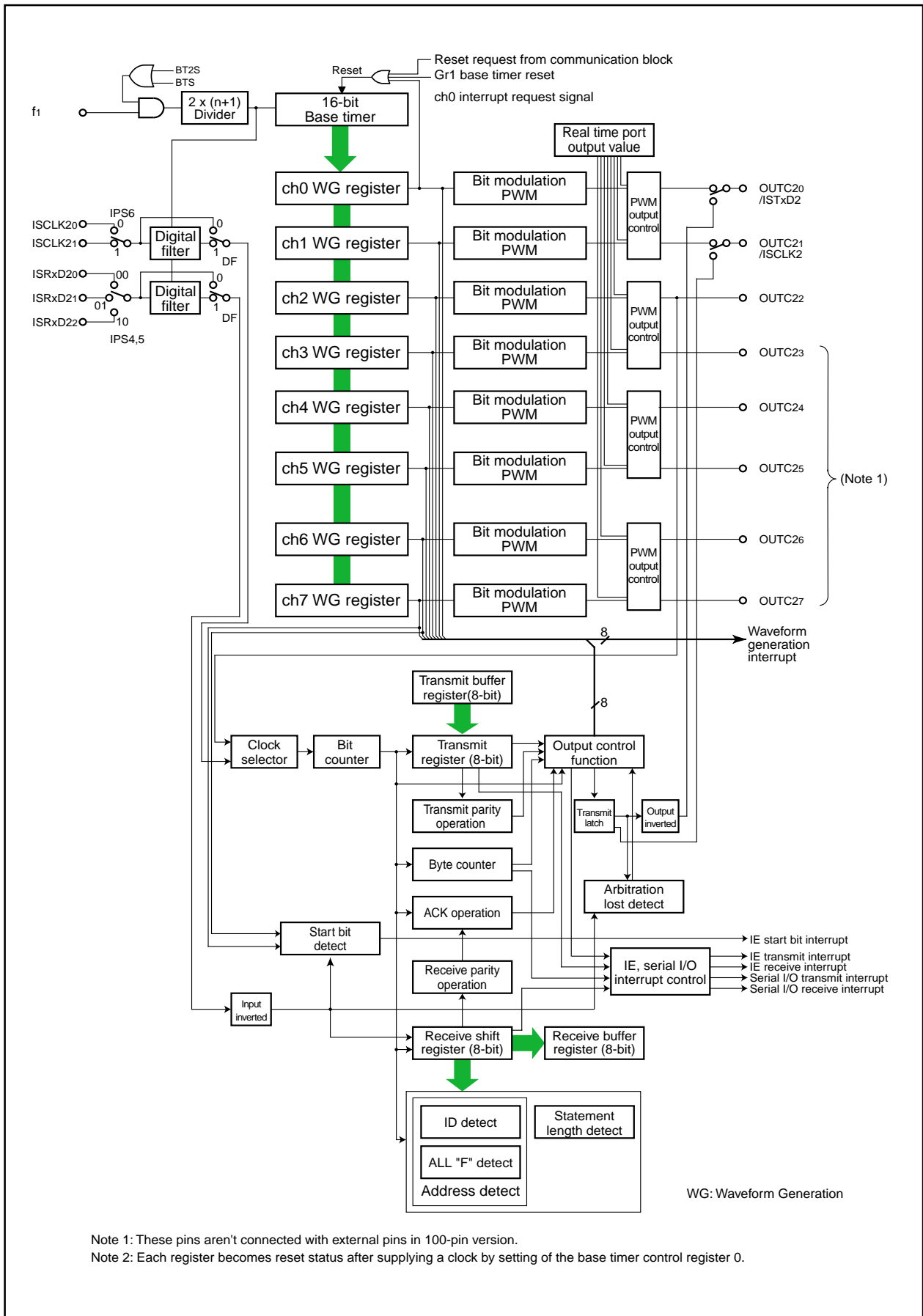


Figure 1. 23. 3. Block diagram of intelligent I/O group 2

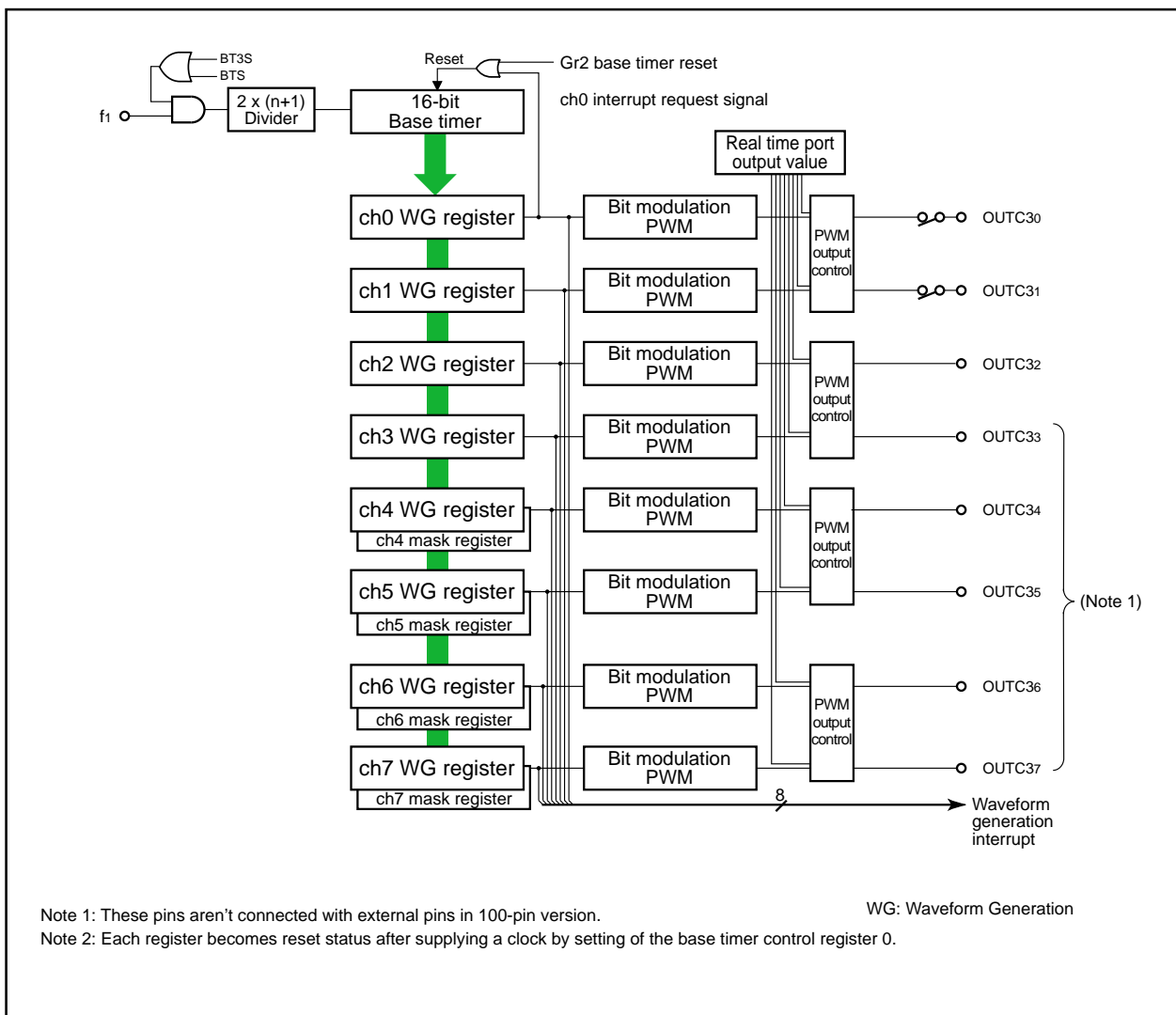


Figure 1. 23 . 4. Block diagram of intelligent I/O group 3

Base timer (group 0 to 3)

The internally generated count source is a free run source. Base timer specifications are given in Table 1.23.2, base timer registers in Figures 1.23.5 to 1.23.9 and a block diagram in Figure 1.23.10.

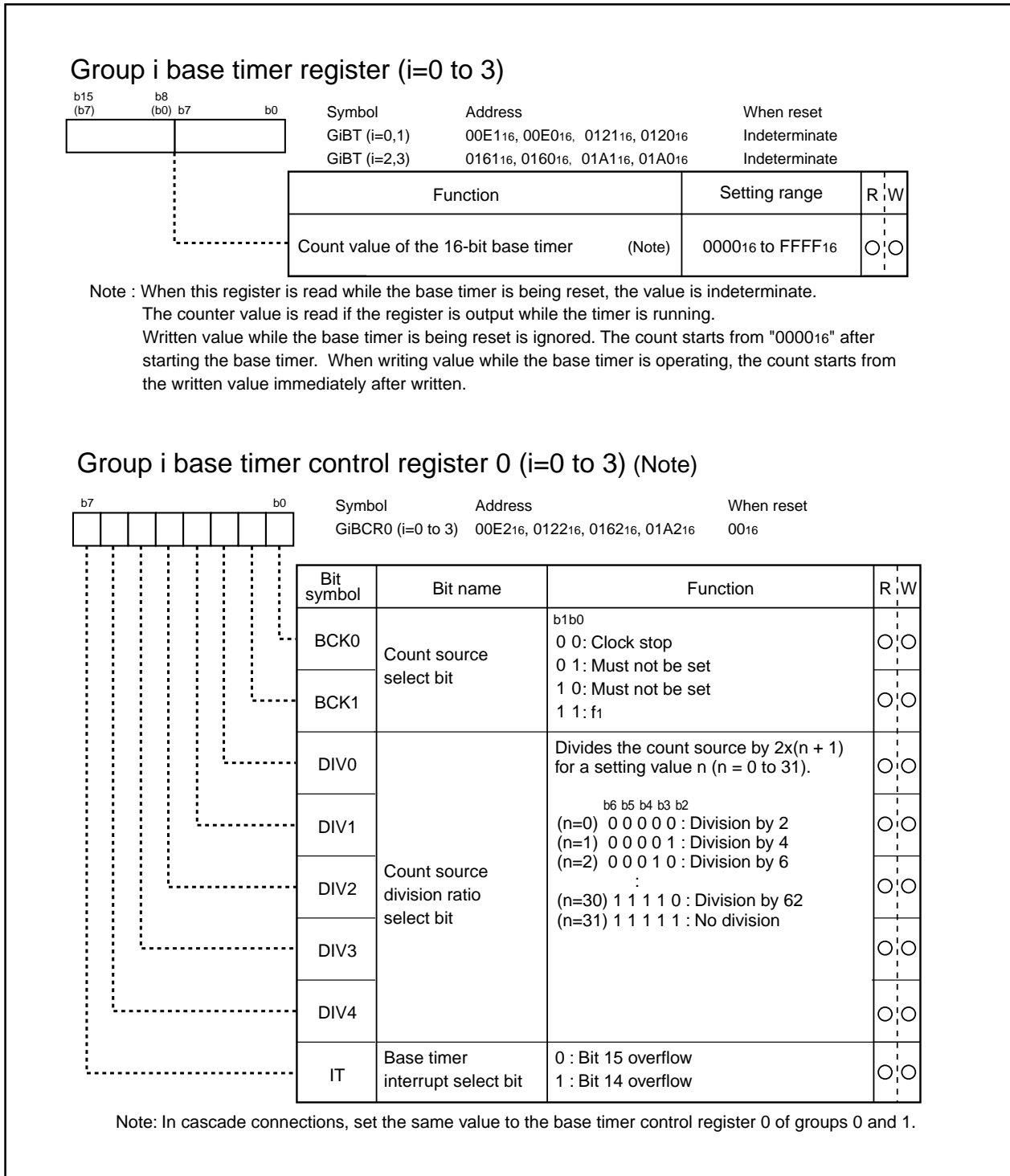


Figure 1. 23. 5. Base timer-related register (1)

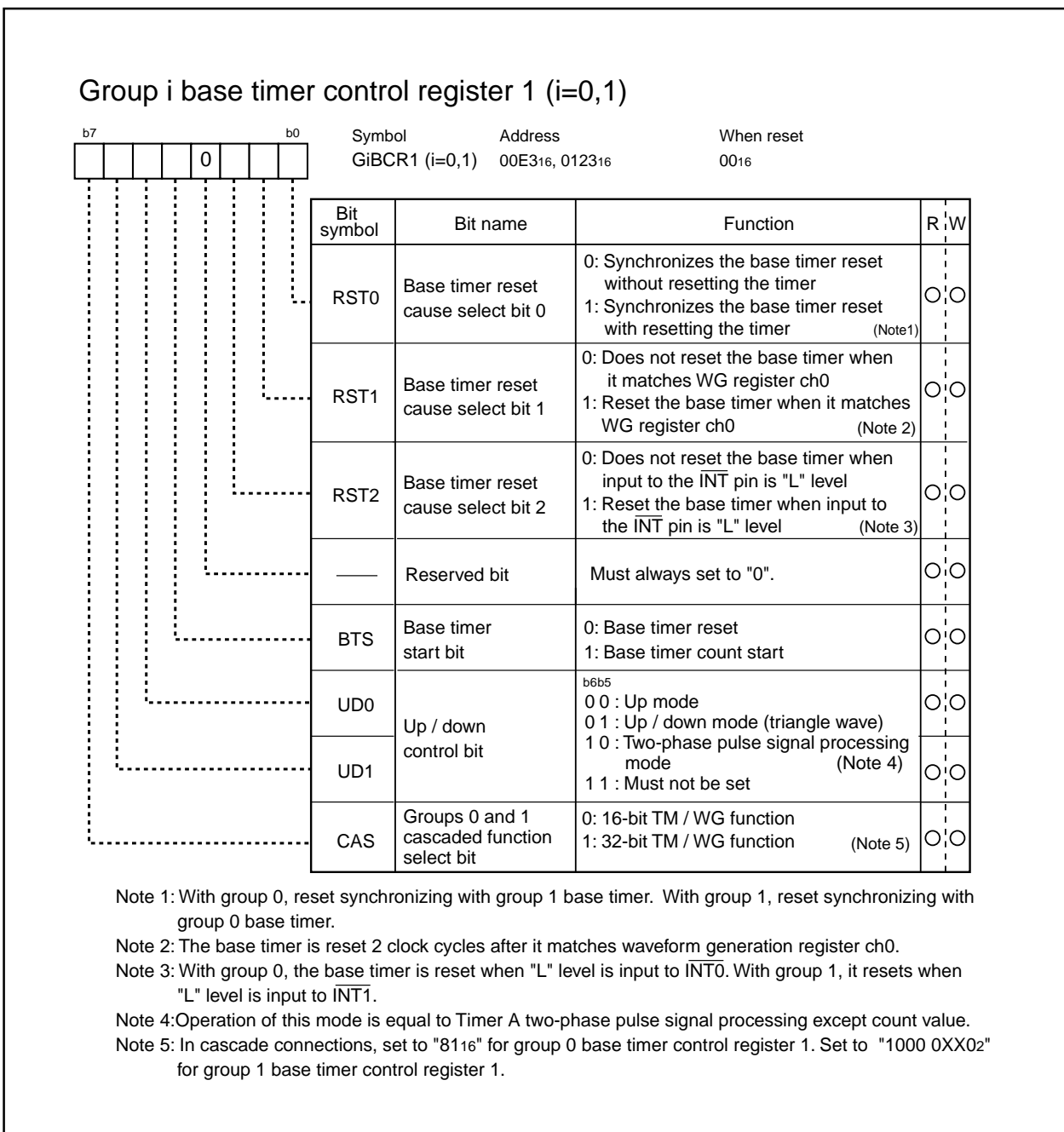


Figure 1. 23. 6. Base timer-related register (2)

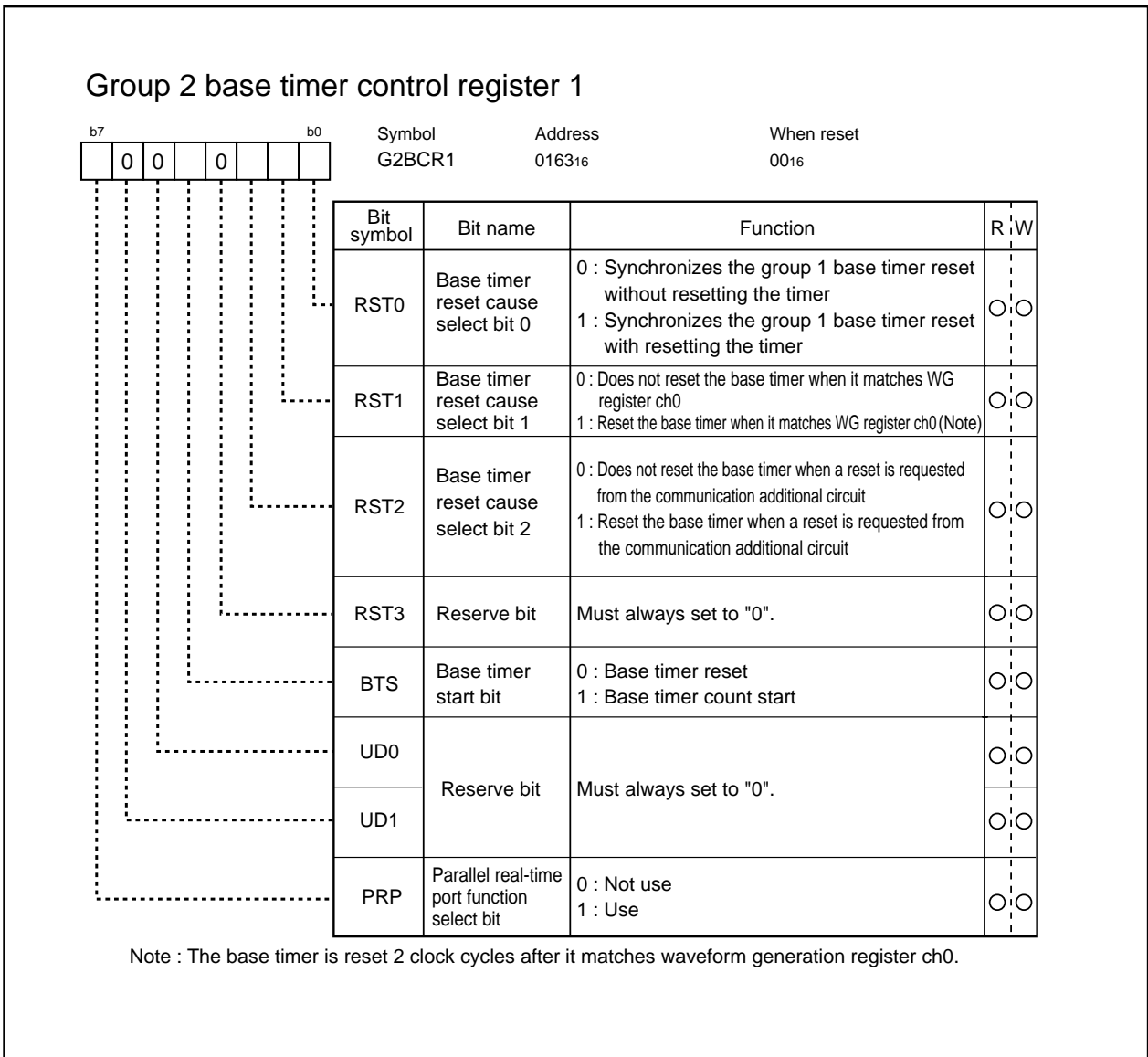


Figure 1. 23. 7. Base timer-related register (3)

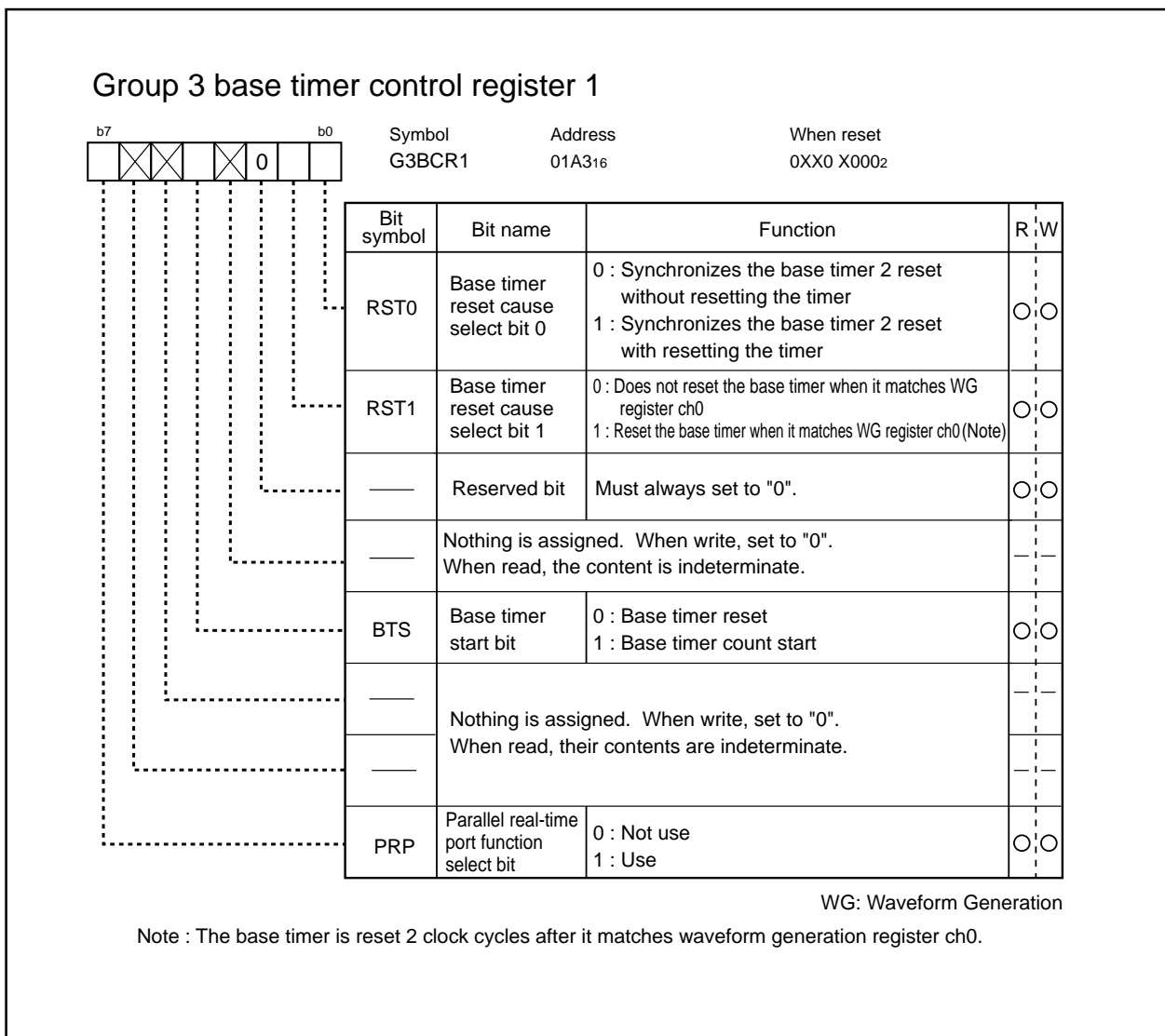


Figure 1. 23. 8. Base timer-related register (4)

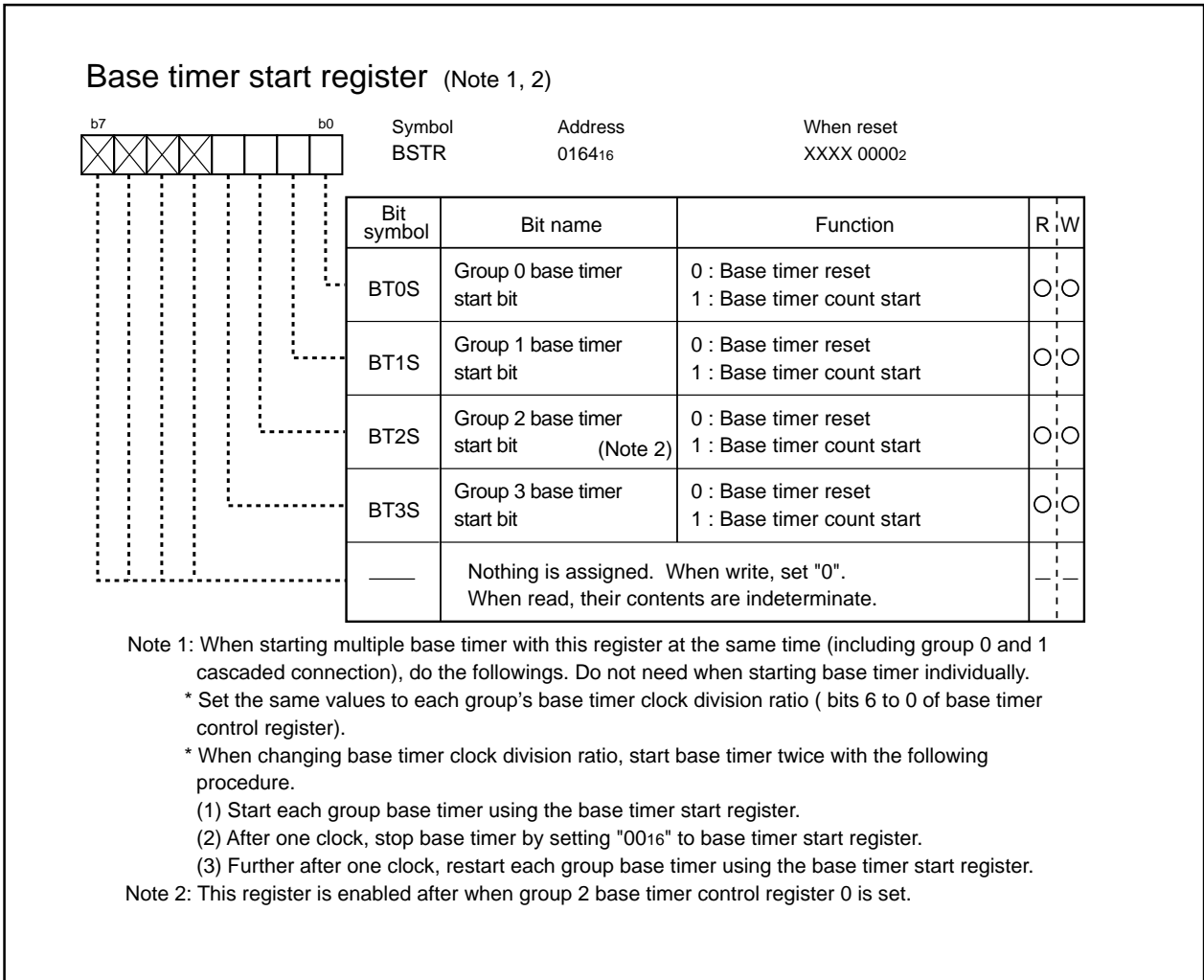


Figure 1. 23. 9. Base timer-related register (5)

Table 1. 23.2. Base timer specifications

Item	Specifications
Count source	$f_1/2(n+1)$ n: Set by count source division ratio select bit (n=0 to 31, however, please note when n=31, the counter source is not divided.)
Count operation	Up count / down count
Count start condition	Writes "1" for the start bit in the base timer start register or base timer control register 1. (After writing the bit, the base timer resets to "0000 ₁₆ " and counting starts.)
Count stop condition	Writes "0" for both the start bit in the base timer start register and base timer control register 1.
Count reset condition	Group 0, 1 (1) Synchronizes and resets the base timer with that of another group. Group 0: Synchronizes base timer reset with the group 1 base timer. Group 1: Synchronizes base timer reset with the group 0 base timer. (2) Matches the value of the base timer to the value of WG register 0. (3) Input "L" to INT pin Group 0 : INT 0 pin Group 1 : INT 1 pin The above 3 factors can be used in conjunction with one another.
	Group 2, 3 (1) Synchronizes and resets the base timer with that of another group. Group 2: Synchronizes base timer reset with the group 1 base timer. Group 3: Synchronizes base timer reset with the group 2 base timer. (2) Matches the value of the base timer to the value of WG register 0. (3) Reset request from communication additional circuit (group 2 only) The above 3 factors can be used in conjunction with one another.
Interrupt request generation timing	When bit 14 or bit 15 overflows
Read from timer	•When the base timer is running The count is output when the base timer is read. •When the base timer not running An undefined value is output when the base timer is read.
Write to timer	Possible. Values that are written while the base timer is resetting are ignored. If values are written while the base timer is running, counting continues after the values are written.

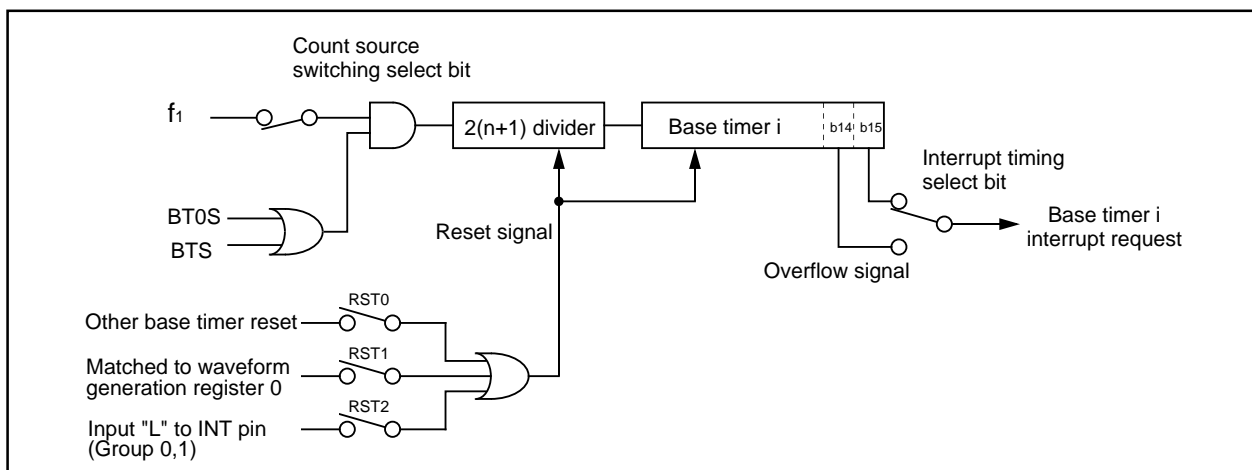


Figure 1. 23.10. Base timer block diagram

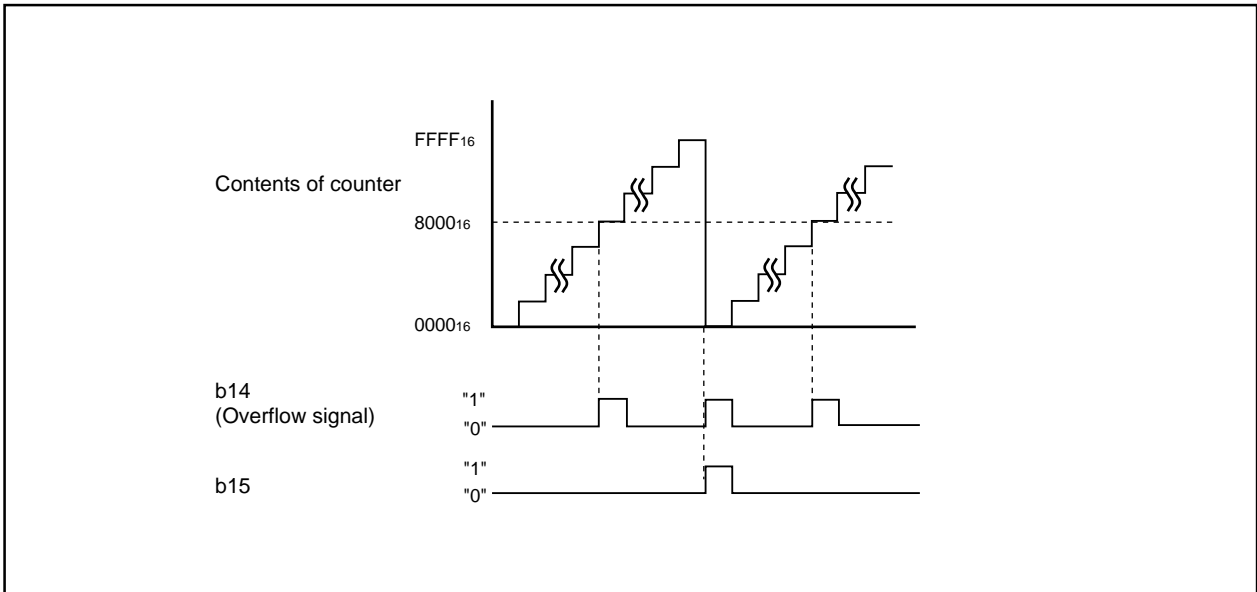


Figure 1. 23.11. Operation timing of base timer

Time measurement (group 0 and 1)

Synchronizes external trigger input and stores the base timer value in the time measurement register j. Specifications for the time measurement function are given in Table 1.23.3, the time measurement control registers in Figures 1.23.12 to 1.23.13, and the operating timing of the time measurement function in Figure 1.23.14 and 15.

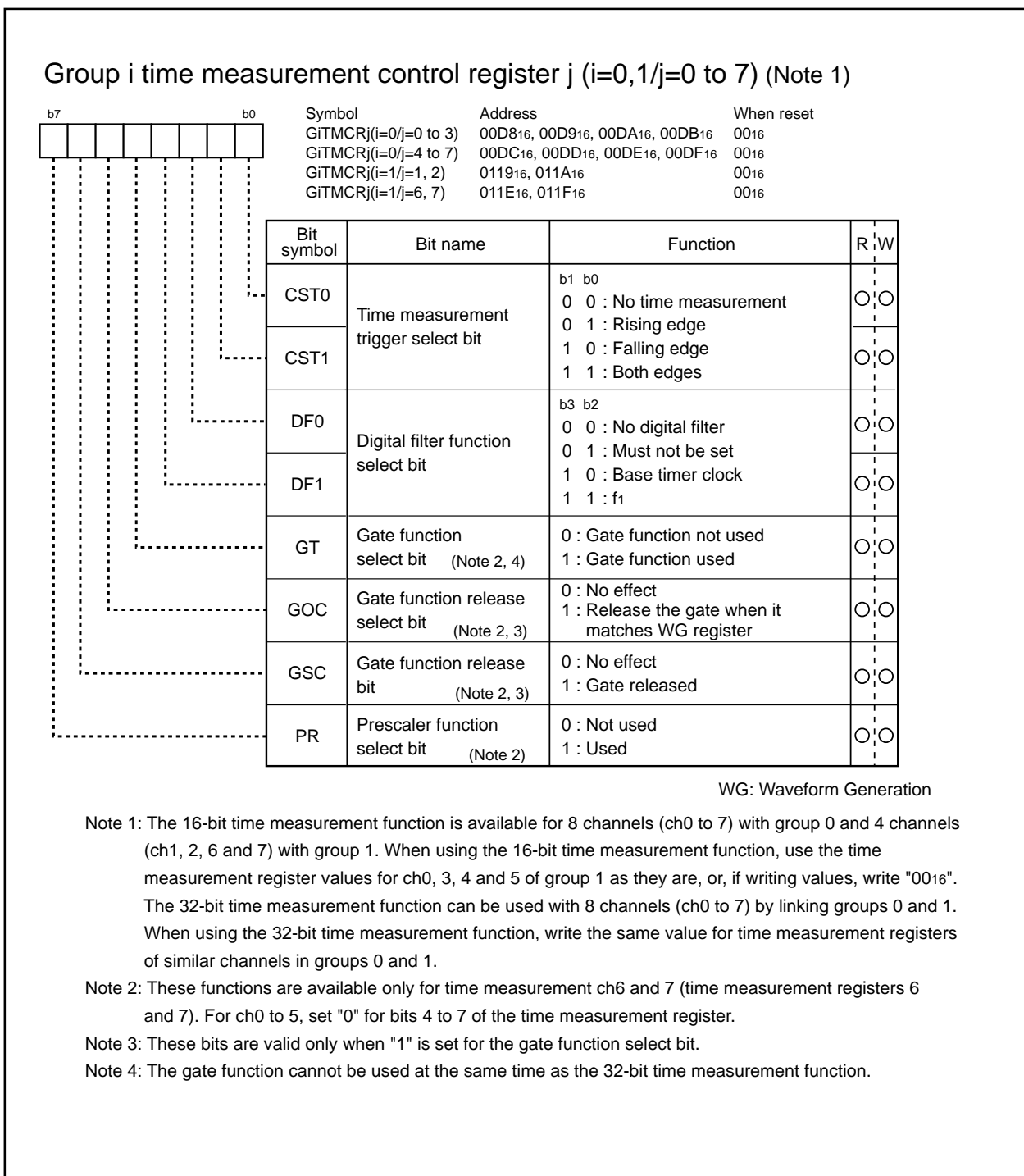


Figure 1. 23. 12. Time measurement-related register (1)

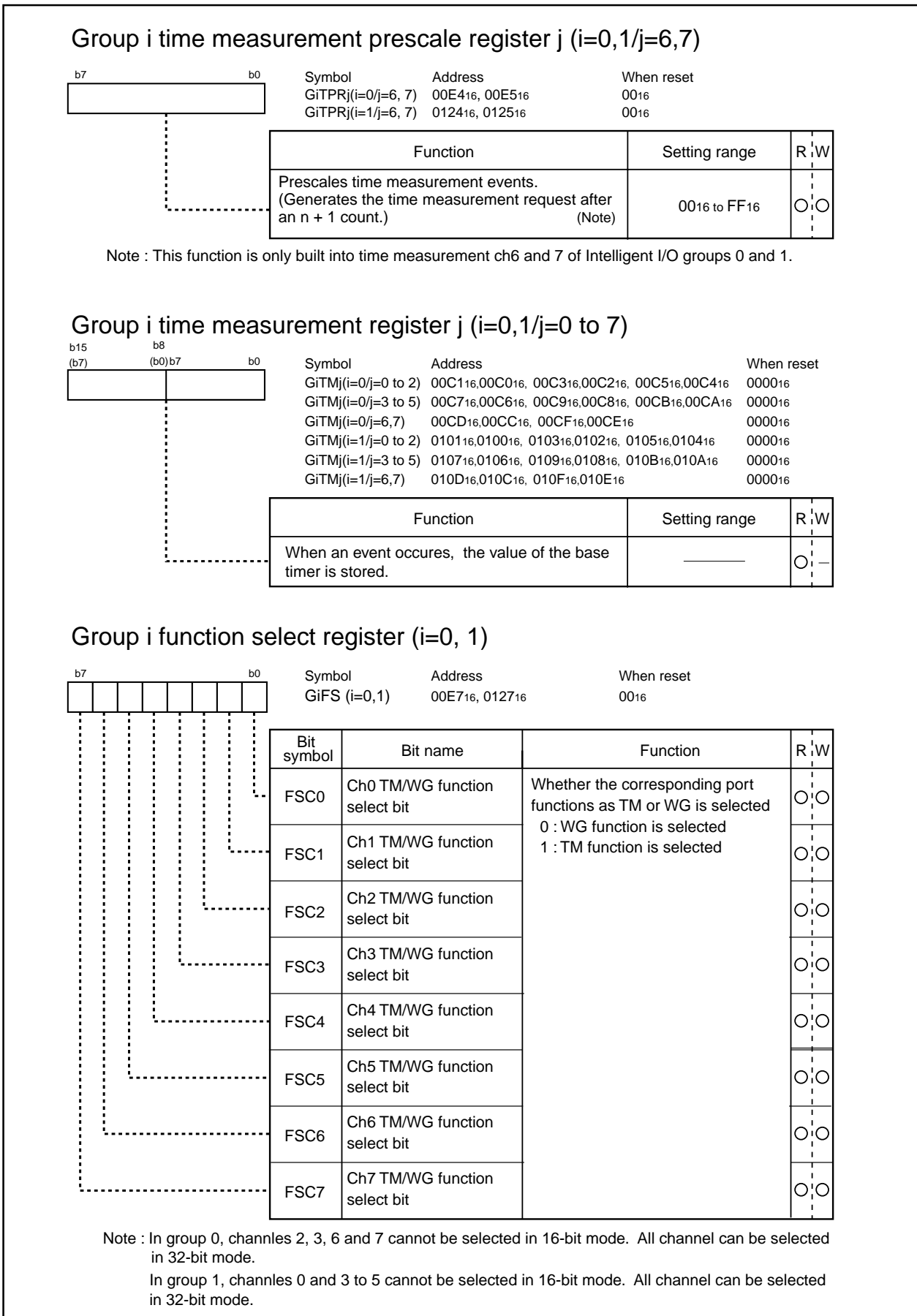


Figure 1. 23. 13. Time measurement-related register (2)

Table 1. 23.3. Specifications of time measurement function

Item	Specifications
Time resolution	$t=1/(\text{base timer count source})$
Trigger input polarity select	•Rising edge •Falling edge •Both edges
Measurement start condition (Note)	Write "1" to the function enable bit
Measurement stop condition	Write "0" to the function enable bit
Time measurement timing	•Prescaler (only ch6 and ch7) : Every the (m+1) trigger input •No prescaler : Every trigger input
Interrupt request generation timing	Same timing as time measurement
INPC pin function	Trigger input pin (Set the corresponding pin to input with the function select register)
Select function	<ul style="list-style-type: none"> •Digital filter function Pulses will pass when they match either f1 or the base timer clock 3 times . •Prescaler function (only for ch6 and ch7) Counts trigger inputs and measures time by inputting a trigger of +1 the value of the time measurement prescale register. •Gate function (only for ch6 and ch7) Prohibits the reception of trigger inputs after the time measurement starts for the first trigger input. Trigger input is newly enabled when the below conditions are satisfied. <ul style="list-style-type: none"> (1) When the base timer i matches the value in WG register j (2) When "1" is written for the gate function release bit This bit automatically becomes "0" after the gate function is released.

Note: On channels where both the time measurement function and waveform output function can be used, select the time measurement function for the function select register (addresses 00E716 and 012716).

Table 1. 23.4. List of time measurement channels with prescaler function and gate function

Group	Channel	TM register	WG register matches signal to release gate function
Group 0	ch6	TM register 6	Base timer 0 matches to WG register 4
	ch7	TM register 7	Base timer 0 matches to WG register 5
Group 1	ch6	TM register 6	Base timer 1 matches to WG register 4
	ch7	TM register 7	Base timer 1 matches to WG register 5

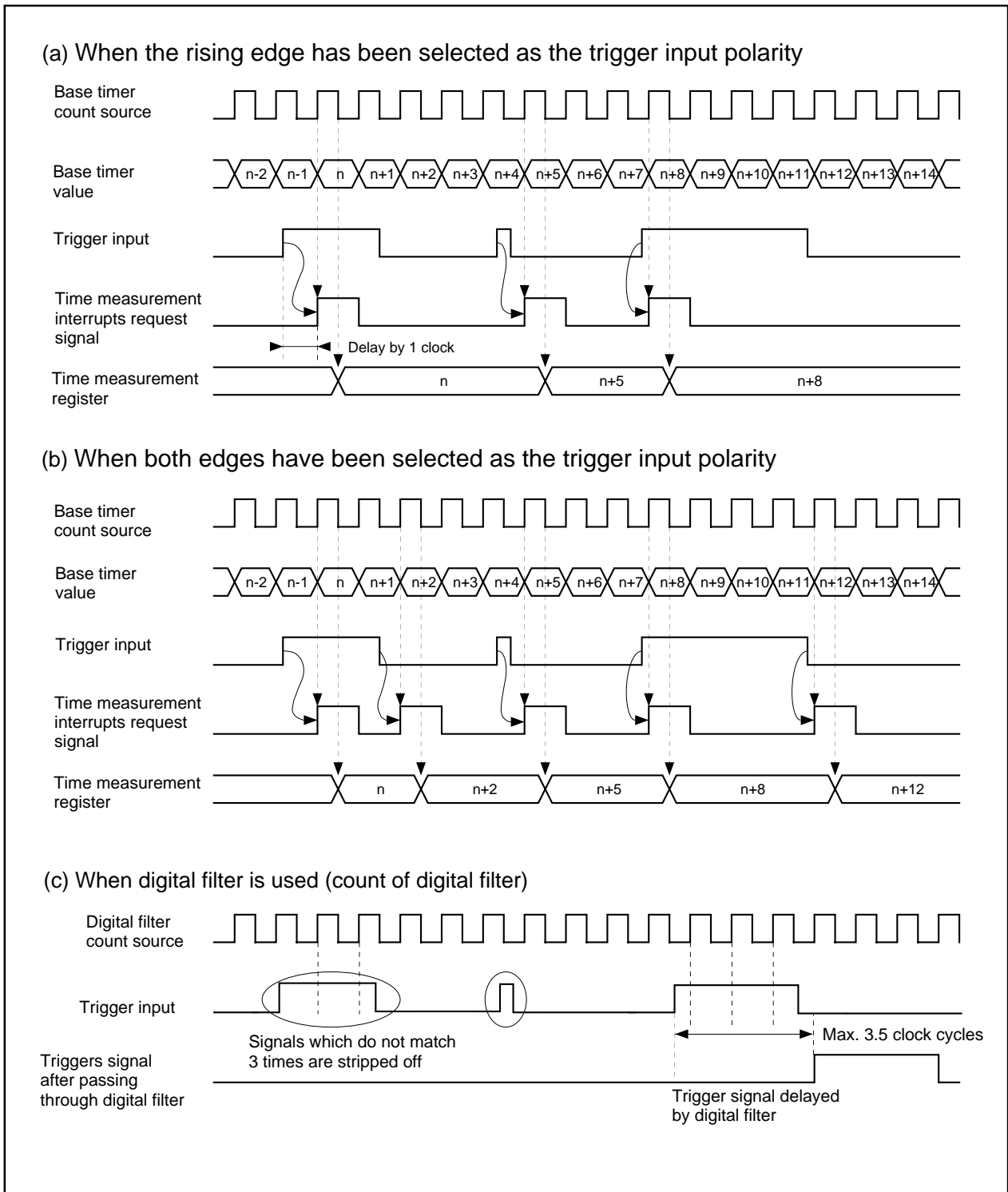


Figure 1. 23. 14 Operation timing of time measurement function

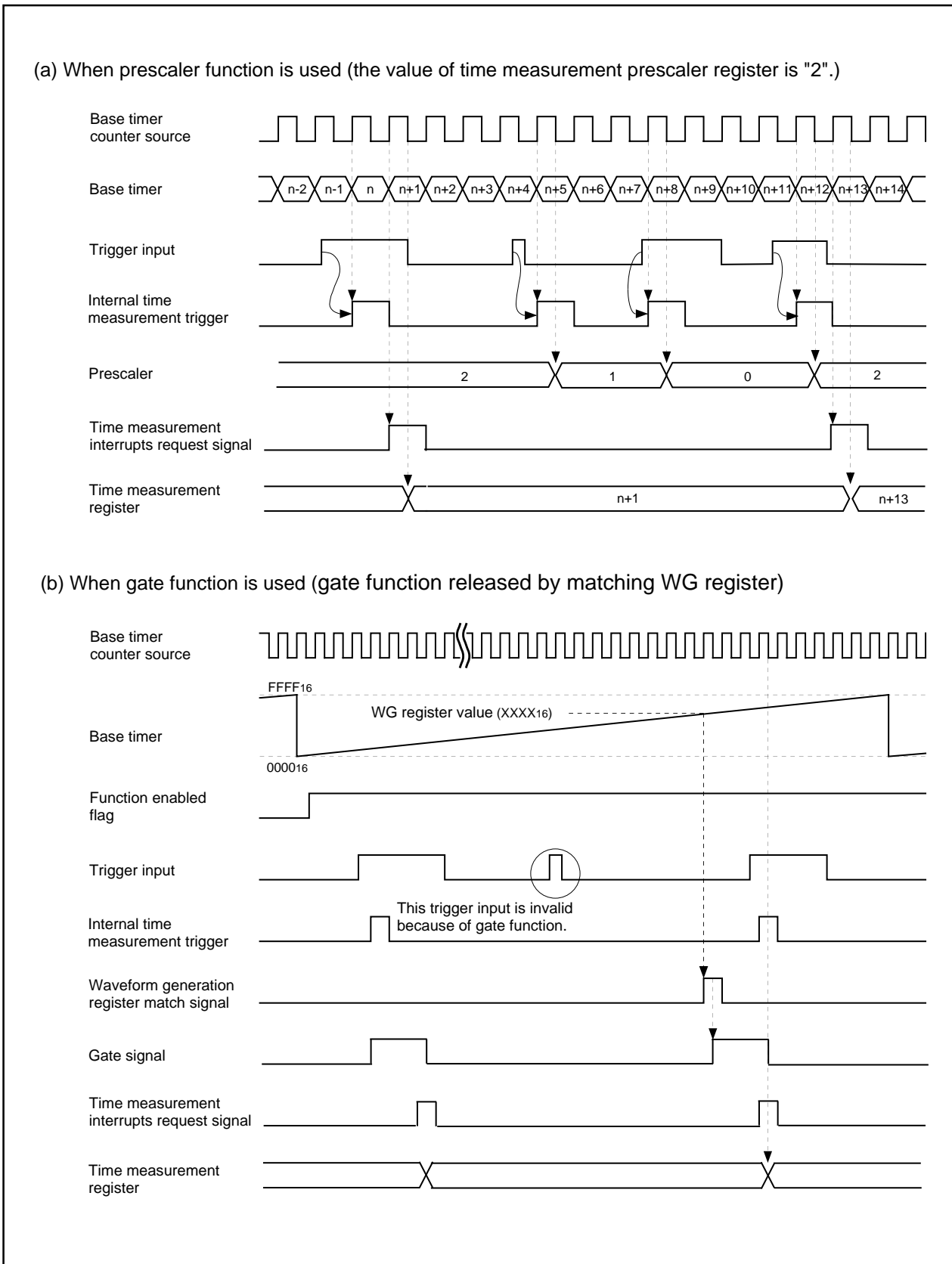


Figure 1. 23. 15. Operation timing when gate function and prescaler function is used

Waveform generation (WG) function (group 0 to 3)

Waveforms are generated when the base timer value matches the value of WG register j.

There are five mode in WG function: single phase waveform output mode (group 0 to 3), phase delayed waveform output mode (group 0 to 3), SR (Set/Reset) waveform output mode (group 0 to 3), bit modulation PWM output mode (group 2 and 3) and parallel real-time port output mode (group 2 and 3).

The WG function related registers are shown in Figures 1.23.16 to 1.23.19.

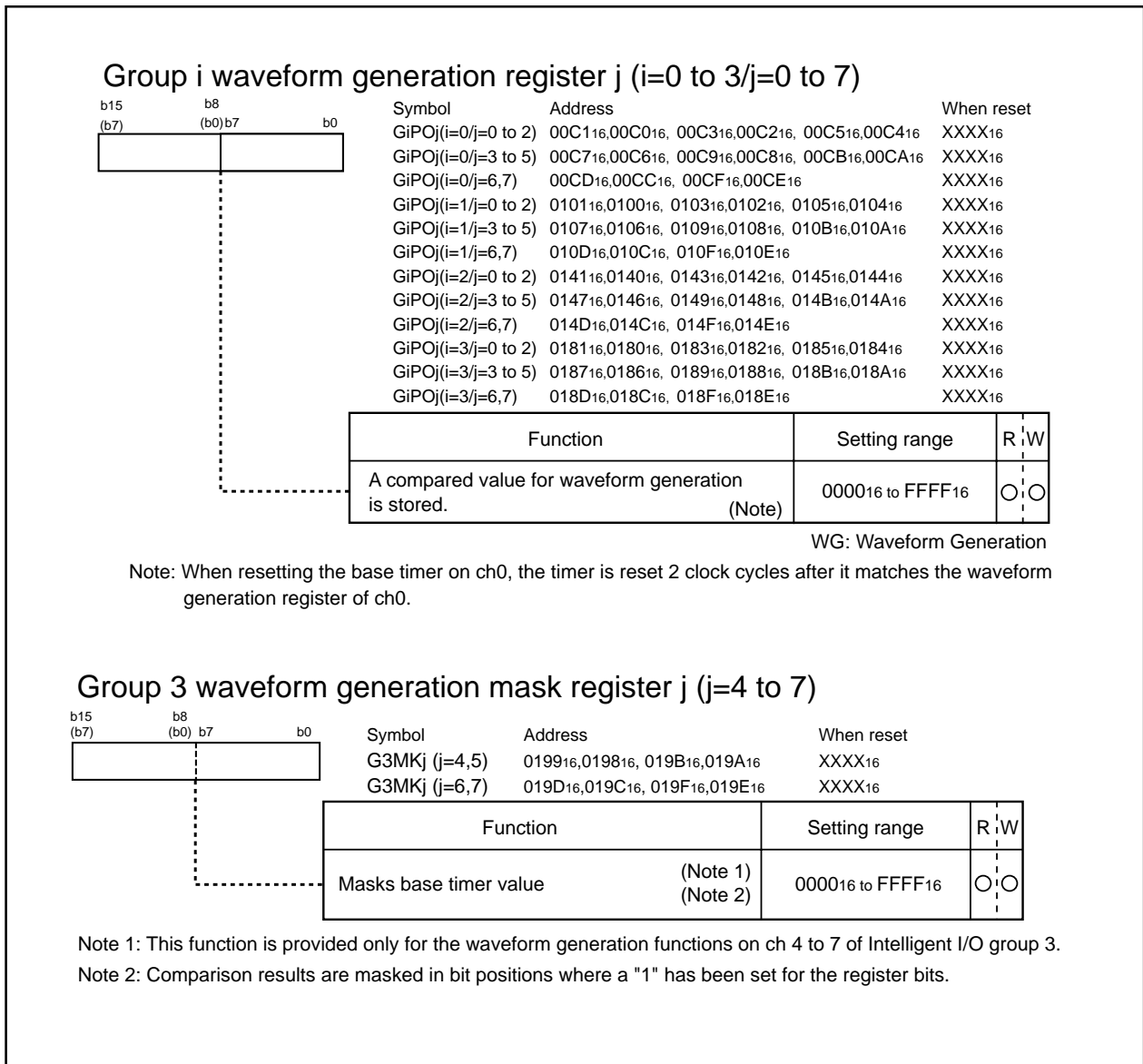


Figure 1. 23. 16. WG-related register (1)

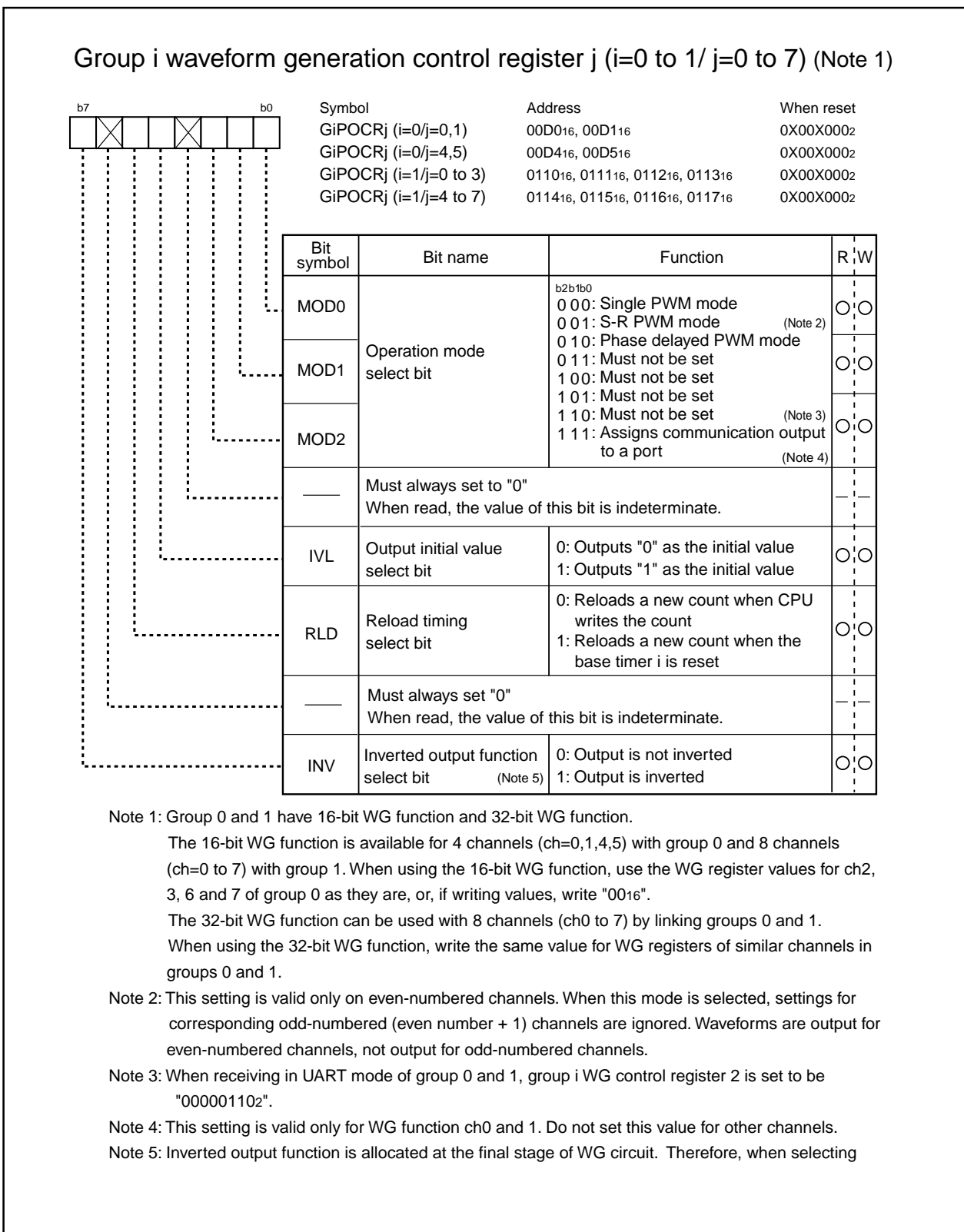


Figure 1. 23. 17. WG-related register (2)

Group i waveform generation control register j (i=2 to 3/ j=0 to 7)

b7	b0	Symbol	Address	When reset
□	□	GiPOCRj (i=2/j=0 to 3)	0150 ₁₆ , 0151 ₁₆ , 0152 ₁₆ , 0153 ₁₆	0X00 X000 ₂
□	□	GiPOCRj (i=2/j=4 to 7)	0154 ₁₆ , 0155 ₁₆ , 0156 ₁₆ , 0157 ₁₆	0X00 X000 ₂
□	□	GiPOCRj (i=3/j=0 to 3)	0190 ₁₆ , 0191 ₁₆ , 0192 ₁₆ , 0193 ₁₆	0X00 X000 ₂
□	□	GiPOCRj (i=3/j=4 to 7)	0194 ₁₆ , 0195 ₁₆ , 0196 ₁₆ , 0197 ₁₆	0X00 X000 ₂

Bit symbol	Bit name	Function	R	W
MOD0	Operation mode select bit	b2b1b0 0 0 0: Single PWM mode 0 0 1: S-R PWM mode (Note 1) 0 1 0: Phase delayed PWM mode 0 1 1: Must not be set 1 0 0: Bit modulation PWM mode 1 0 1: Must not be set 1 1 0: Must not be set 1 1 1: Assigns communication output to a port (Note 2)	○	○
MOD1			○	○
MOD2			○	○
PRT	Parallel RTP output trigger select bit	0: Match of WG register j isn't trigger 1: Match of WG register j is trigger	○	○
IVL	Output initial value select bit	0: Outputs "0" as the initial value 1: Outputs "1" as the initial value	○	○
RLD	Reload timing select bit	0: Reloads a new count when CPU writes the count 1: Reloads a new count when the base timer i is reset	○	○
RTP	RTP port function select bit	0: Not use 1: Use	○	○
INV	Inverted output function select bit (Note 3)	0: Output is not inverted 1: Output is inverted	○	○

Note 1: This setting is valid only on even-numbered channels. When this mode is selected, settings for corresponding odd-numbered (even number + 1) channels are ignored. Waveforms are output for even-numbered channels, not output for odd-numbered channels.

Note 2: This setting is valid only for group 2 WG function ch0 and 1. Do not set this value for other channels.

Note 3: Inverted output function is allocated at the final stage of WG circuit. Therefore, when selecting "0" output by IVL bit and inverted output by INV bit, "1" is output.

Group i function enable register (i=0 to 3)

b7	b0	Symbol	Address	When reset
□	□	GiFE (i=0 to 3)	00E6 ₁₆ , 0126 ₁₆ , 0166 ₁₆ , 01A6 ₁₆	00 ₁₆

Bit symbol	Bit name	Function	R	W
IFE0	Ch0 function enable bit	Whether the corresponding port functions is selected 0 : Disables function on ch i 1 : Enables function on ch i	○	○
IFE1	Ch1 function enable bit		○	○
IFE2	Ch2 function enable bit		○	○
IFE3	Ch3 function enable bit		○	○
IFE4	Ch4 function enable bit		○	○
IFE5	Ch5 function enable bit		○	○
IFE6	Ch6 function enable bit		○	○
IFE7	Ch7 function enable bit		○	○

Figure 1. 23. 18. WG-related register (3)

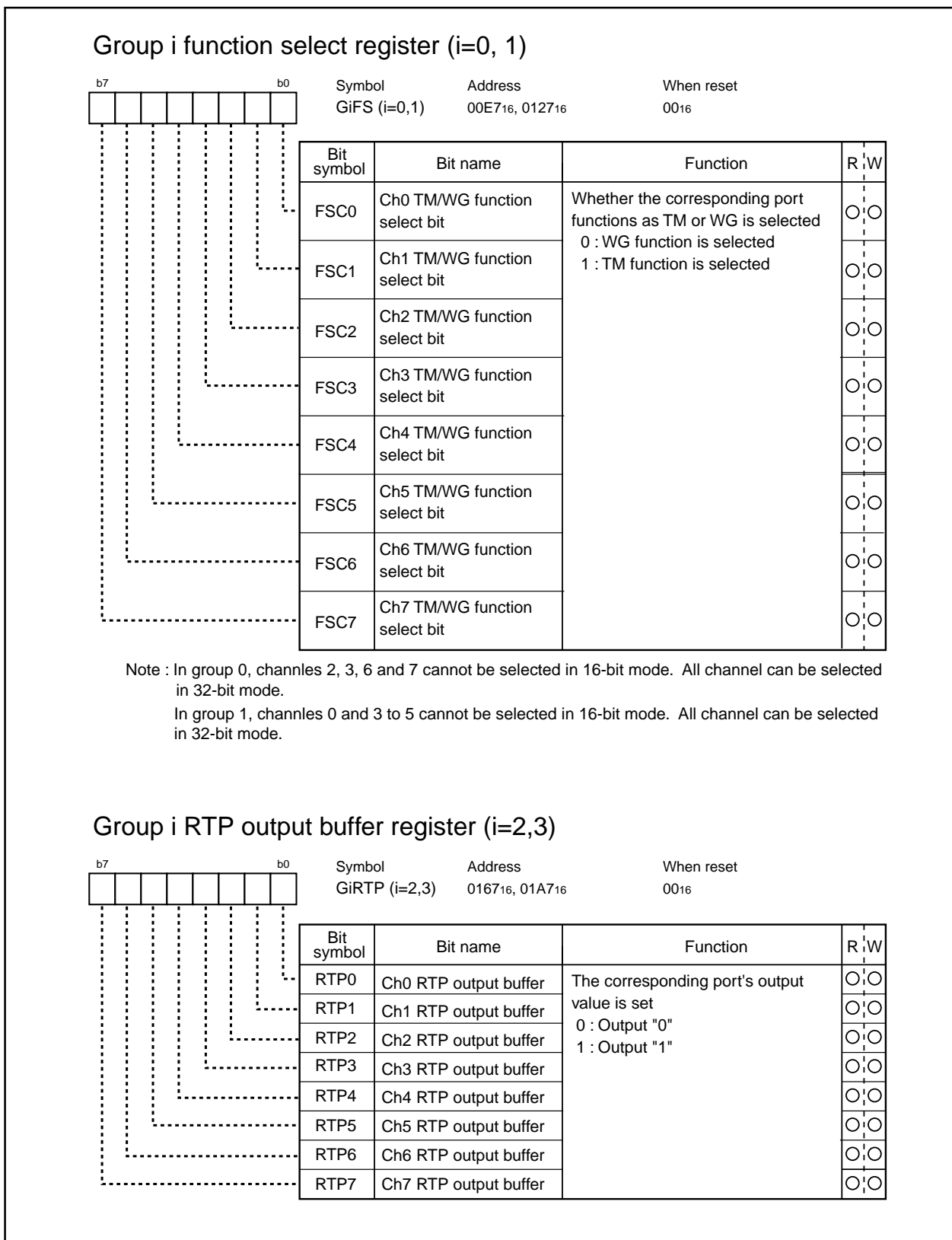


Figure 1. 23. 19. WG-related register (4)

(1) Single phase waveform output mode (group 0 to 3)

This mode is set when the base timer value matches the value of WG register j, and reset when the base timer overflows or the count is reset. Specifications for the single phase waveform output mode are given in Table 1.23.5 and an operating chart for the single phase waveform output mode in Figure 1.23.20.

Table 1. 23.5. Specifications of single phase waveform output mode

Item	Specifications
Output waveform	<ul style="list-style-type: none"> When free run operation <ul style="list-style-type: none"> Period : Base timer count source x 1/65536 "H" level width : 1/base timer count source x (65536 - m) Resetting when the base timer matches WG register 0 (ch0) <ul style="list-style-type: none"> Period : Base timer count source x 1/(k+2) "H" level width : 1/base timer count source x (k+2-m) m : values set to WG register j k: values set to WG register 0
Waveform output start condition	Write "1" to the function enable bit ^(Note)
Waveform output stop condition	Write "0" to the function enable bit
Interrupt generation timing	When the base timer value matches the WG register j
OUTC pin	Pulse output (Corresponding pins are set with the function select register.)
Read from the WG register 0	The set value is output
Write to the WG register 0	Can always write
Select function	<ul style="list-style-type: none"> Initial value setting function <ul style="list-style-type: none"> Sets output level used at waveform output start Inverted output function <ul style="list-style-type: none"> Inverts waveform output level and outputs the waveform from the OUTC pin

Note: On channels where both the time measurement function and waveform output function can be used, select the waveform output function for the function select register (addresses 00E7₁₆ and 0127₁₆).

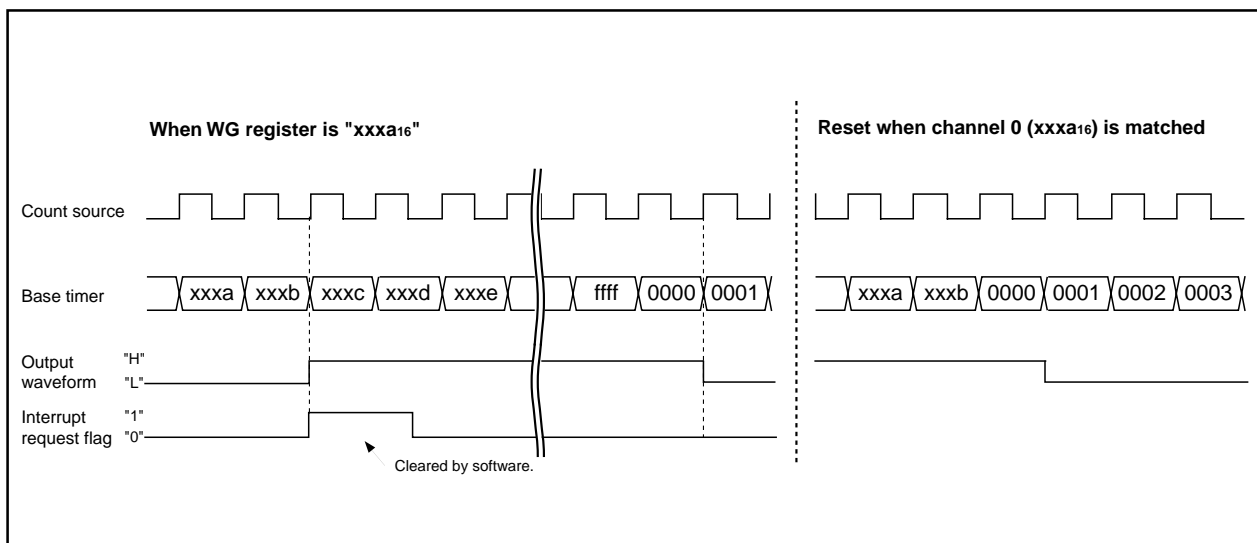


Figure 1. 23. 20. Operation timing in single phase waveform output mode

(2) Phase delayed waveform output mode (group 0 to 3)

This mode is repeatedly set and reset when the base timer value matches the value of WG register j. Specifications for the phase delayed waveform output mode are given in Table 1.23.6 and an operation timing in phase delayed waveform output mode in Figure 1.23.21.

Table 1. 23.6. Specifications of phase delayed waveform output mode

Item	Specifications
Output waveform	<ul style="list-style-type: none"> When free run operation <ul style="list-style-type: none"> Period : Base timer count source x 1/65536 x 1/2 "H" and "L" level width : 1/base timer count source x 65536 Resetting when group i base timer matches WG register 0 (ch0) <ul style="list-style-type: none"> Period : Base timer count source x 1/(k+2) x 1/2 "H" and "L" level width : 1/base timer count source x (k+2) k : values set to WG register 0
Waveform output start condition	Write "1" to the function enable bit (Note)
Waveform output stop condition	Write "0" to the function enable bit
Interrupt generation timing	When the base timer value matches the WG register j
OUTCij pin	Pulse output (Corresponding pins are set with the function select register.)
Read from the WG register	The set value is output
Write to the WG register	Can always write
Select function	<ul style="list-style-type: none"> Initial value setting function <ul style="list-style-type: none"> Sets output level used at waveform output start Inverted output function <ul style="list-style-type: none"> Inverts waveform output level and outputs the waveform from the OUTC pin

Note : On channels where both the time measurement function and waveform output function can be used, select the waveform output function for the function select register (addresses 00E716 and 012716).

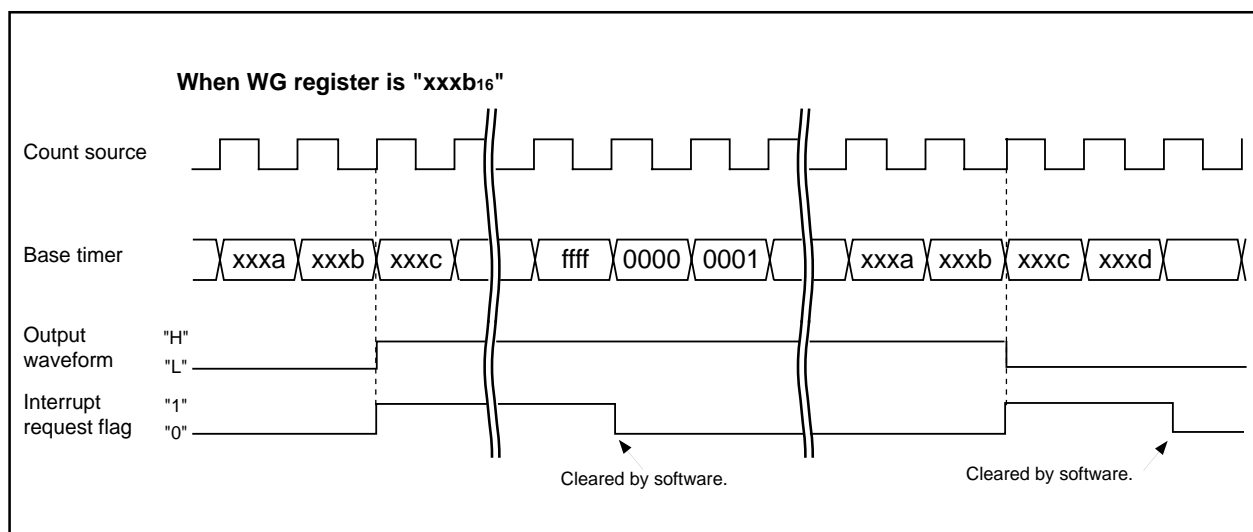


Figure 1. 23. 21. Operation timing in phase delayed waveform output mode

(3) SR (Set/Reset) waveform output mode (group 0 to 3)

This mode is set when the base timer value matches the value of WG register j (j is an even-numbered channel), and reset when the base timer matches the WG register (j + 1) or the base timer value is "0". Specifications for the SR waveform output mode are given in Table 1.23.7 and an operating chart for the SR waveform output mode in Figure 1.23.22.

Table 1. 23.7. Specifications of SR waveform output mode

Item	Specifications
Output waveform	<ul style="list-style-type: none"> •When free run operation <ul style="list-style-type: none"> Period : Base timer count source x 1/65536 "H" level width : 1/base timer count source x (m-p) •Resetting when base timer matches WG register 0 (ch0) <ul style="list-style-type: none"> Period : Base timer count source x 1/(k+2) (Note 1) "H" level width : 1/base timer count source x (m-p) m : values set to WG register j p : values set to WG register i(j+1) k : values set to WG register 0 (j is an even-numbered channel) (Note 2)
Waveform output start condition	Write "1" to the function enable bit (Note 3)
Waveform output stop condition	Write "0" to the function enable bit
Interrupt generation timing	When the base timer value matches the WG register j
OUTC pin (Note 4)	Pulse output (Corresponding pins are set with the function select register.)
Read from the WG register	The set value is output
Write to the WG register	Can always write
Select function (Note 5)	<ul style="list-style-type: none"> •Initial value setting function <ul style="list-style-type: none"> Sets output level used at waveform output start •Inverted output function <ul style="list-style-type: none"> Inverts waveform output level and outputs the waveform from the OUTC pin

Note 1: The SR waveform output function that sets and resets the mode on ch0 and 1 cannot be used when the base timer is reset by WG register 0 (ch0).

Note 2: Set WG register values for odd-numbered channels that are lower than even-numbered channels.

Note 3: On channels where both the time measurement function and waveform output function can be used, select the waveform output function for the function select register (addresses 00E7₁₆ and 0127₁₆).

Note 4: SR waveforms are output for even-numbered channels only.

Note 5: Settings for the WG control register on the odd-numbered channels are ignored.

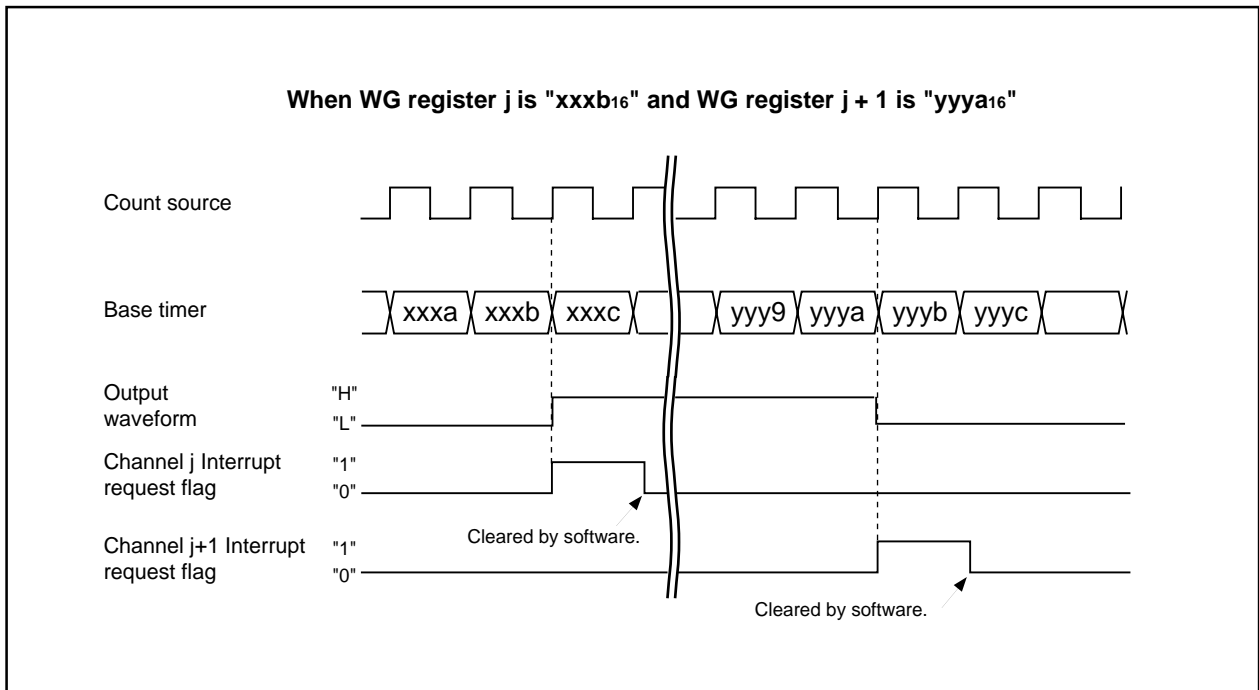


Figure 1. 23. 22. Operation timing in SR waveform output mode

(4) Bit modulation PWM output mode (group 2 and 3)

This mode performs PWM to improve output resolution. Specifications for the bit modulation PWM mode are given in Table 1.23.8 and an operating chart for the bit modulation PWM mode in Figure 1.23.23.

Table 1. 23.8. Specifications of bit modulation PWM mode

Item	Specifications
Output waveform	Period : Base timer count source x 1/64 "H" level width (average) : 1/base timer count source x [k+(m/1024)] k : values set to WG register j (six high-order bits) m : values set to WG register j (ten lower-order bits)
Waveform output start condition	Write "1" to the function enable bit
Waveform output stop condition	Write "0" to the function enable bit
Interrupt generation timing	When the base timer value matches the WG register j
OUTC pin	Pulse output (Corresponding pins are set with the function select register.)
Read from the WG register j	The set value is output
Write to the WG register j	Can always write
Select function	<ul style="list-style-type: none"> Initial value setting function Sets output level used at waveform output start Inverted output function Inverts waveform output level and outputs the waveform from the OUTC pin

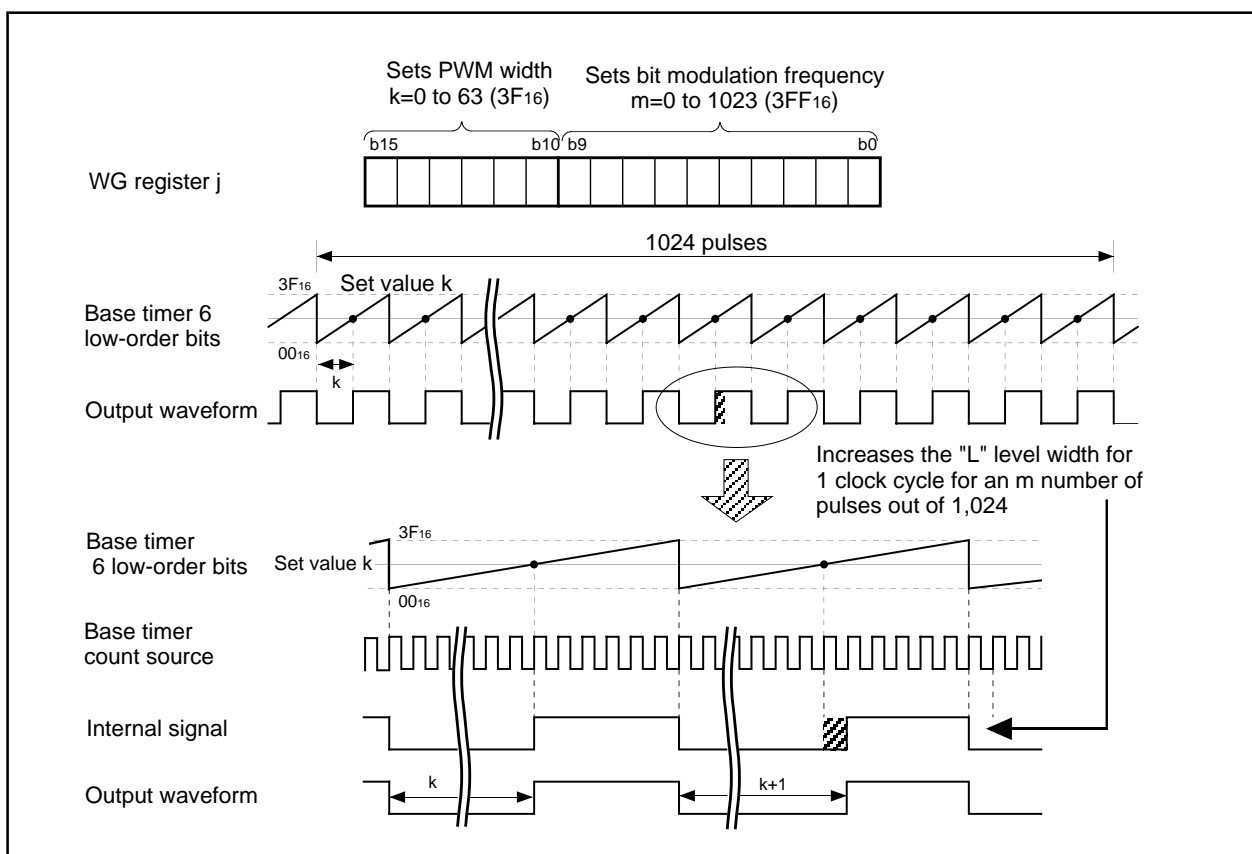


Figure 1. 23. 23. Operation timing in bit modulation PWM mode

(5) Real-time port output mode (group 2 and 3)

This mode outputs the value set in the real-time port register from the OUTC pin when the base timer value matches the value of WG register j. Specifications for the real-time port output mode are given in Table 1.23.9 and a block diagram and timing chart of the real-time port output function in Figure 1.23.24.

Table 1. 23.9. Specifications of real-time port output mode

Item	Specifications
Waveform output start condition	Write "1" to the function enable bit
Waveform output stop condition	Write "0" to the function enable bit
Interrupt generation timing	When the base timer value matches the WG register j
OUTC pin	RTP output (Corresponding pins are set with the function select register.)
Read from the WG register j	The set value is output
Write to the WG register j	Can always write
Read from the RTP output buffer register	The set value is output
Write to the RTP output buffer register	Can always write
Select function	<ul style="list-style-type: none"> •Initial value setting function Sets output level used at waveform output start •Inverted output function Inverts waveform output level and outputs the waveform from the OUTC pin

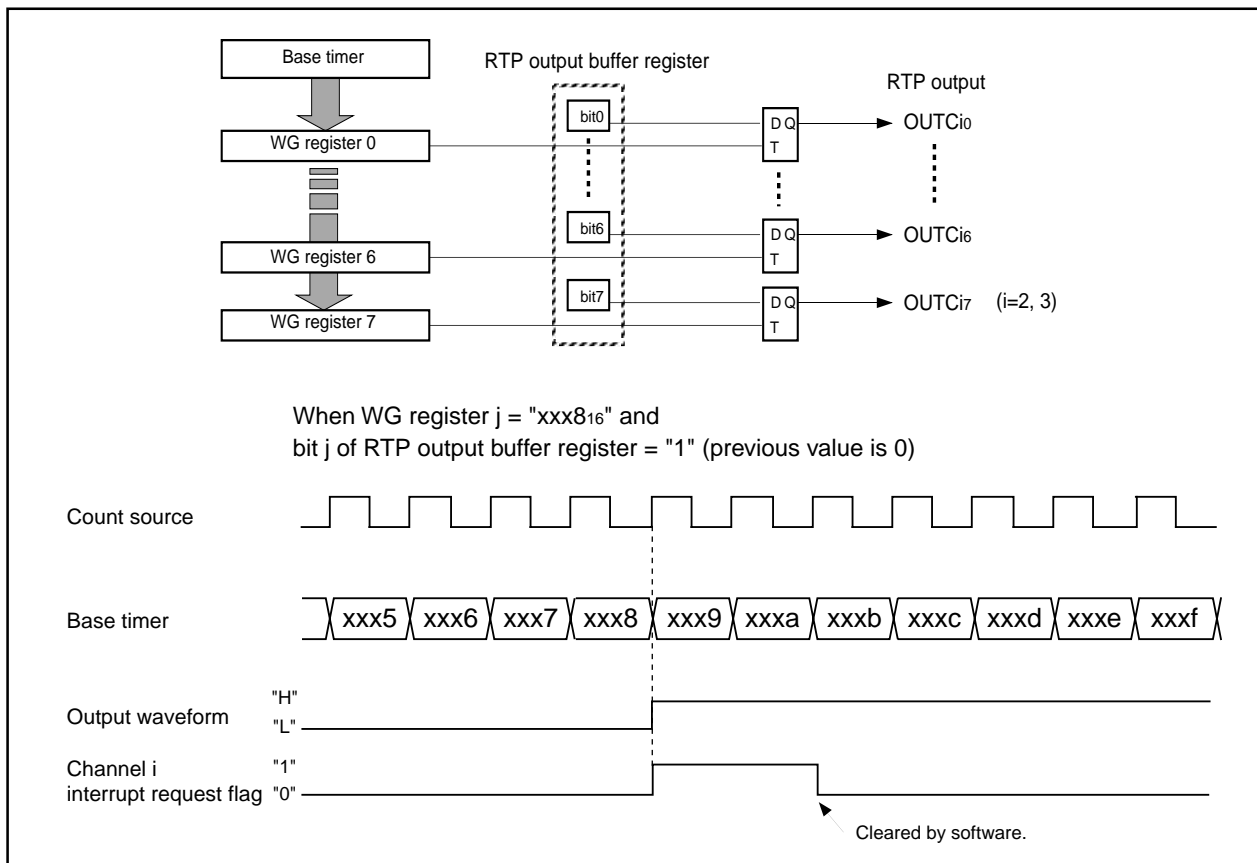


Figure 1. 23. 24. Block diagram and operation timing of real-time port output function

(6) Parallel real-time port output mode (group 2 and 3)

This mode outputs the value set in the real-time port register from the OUTC pin when the base timer value matches the value of WG register j. Specifications for the parallel real-time port output mode are given in Table 1.23.10 and a block diagram and timing chart of the real-time port output function in Figure 1.23.25.

Table 1. 23.10. Specifications of parallel real-time port output mode

Item	Specifications
Waveform output start condition	Write "1" to the function enable bit
Waveform output stop condition	Write "0" to the function enable bit
Interrupt generation timing	When the base timer value matches the WG register
OUTC pin	RTP output (Corresponding pins are set with the function select register.)
Read from the WG register	The set value is output
Write to the WG register	Can always write
Read from the RTP output buffer register	The set value is output
Write to the RTP output buffer register	Can always write
Select function	<ul style="list-style-type: none">•Initial value setting function Sets output level used at waveform output start•Inverted output function Inverts waveform output level and outputs the waveform from the OUTC pin

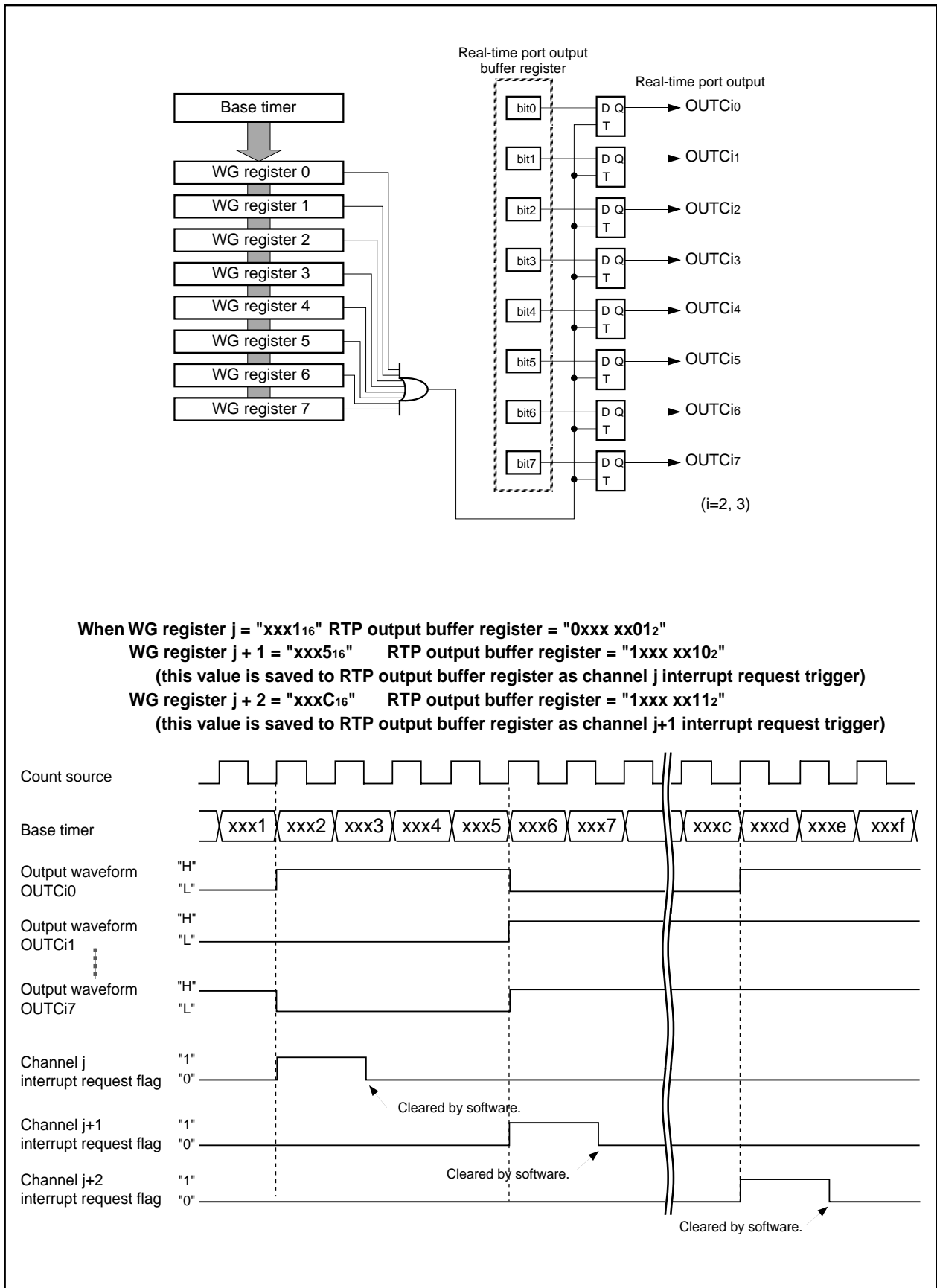


Figure 1. 23. 25. Block diagram and operation timing of parallel real-time port output function

Serial I/O (group 0 to 2)

Intelligent I/O groups 0 to 2 each have two internal 8-bit shift registers. When used in conjunction with the time measurement (TM) function or WG function, these shift registers enable clock synchronous/asynchronous serial communications.

(1) Clock synchronous serial I/O mode (group 0, 1)

Intelligent I/O groups 0 and 1 each have communication block that have two internal 8-bit shift registers. When used in conjunction with the communication block and WG function, these shift registers enable 8-bit clock synchronous and HDLC data process function. When used in conjunction with the communication block, TM function and WG function, these shift registers enable 8-bit clock asynchronous communication.

Table 1.23.11 lists using registers in group 0 and 1, figure 1.23.26 to 1.23.29 shows the related registers.

Table 1.23.11. Using registers in group 0 and 1

	Clock synchronous serial I/O	UART	HDLC
Base timer control register 0	√	√	√
Base timer control register 1	√	√	√
Time measument control register 2	—	√	—
Waveform generate control register 0	√	√	√
Waveform generate control register 1	—	—	√
Waveform generate control register 2	√	√	—
Waveform generate control register 3	√	√	—
Waveform generate register 0	√	√	√
Waveform generate register 1	√	—	√
Time measument /Waveform generate register 2	√	√	—
Waveform generate register 3	√	√	—
Function select register	√	√	√
Function enable register	√	√	√
SI/O communication mode register	√	√	√
SI/O extended mode register	—	—	√
SI/O communication control register	√	√	√
SI/O extended transmit control register	—	—	√
SI/O extended receive control register	—	—	√
SI/O special communication interrupt detect register	—	—	√
SI/O receive buffer register	√	√	√
Transmit buffer	√	√	√
(Receive data register)	—	—	√
Data compare register j (j=0 to 3)	—	—	√
Data mask register j (j=0, 1)	—	—	√
Transmit CRC code register	—	—	√
Receive CRC code register	—	—	√
Transmit output register	—	—	√
Receive input register	—	—	√

√ : Use — : Not use

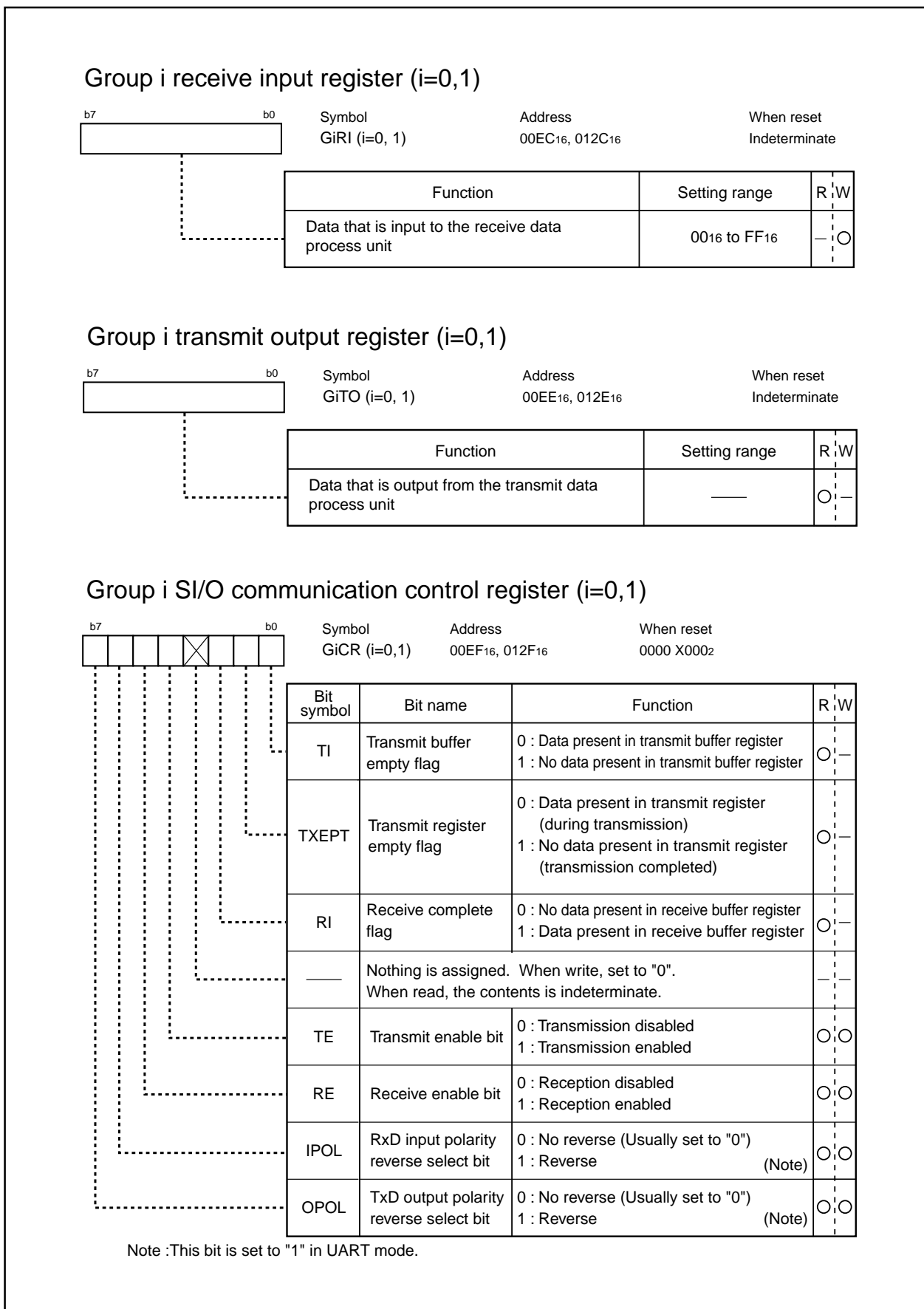
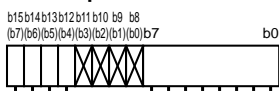


Figure 1. 23. 26. Group 0 and 1 related register (1)

Group i SI/O receive buffer register (i=0,1)

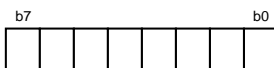


Symbol: GiBF(i=0,1) Address: 00E9₁₆, 00E8₁₆, 0129₁₆, 0128₁₆ When reset: Indeterminate

Bit symbol	Bit name	Function	R W
—	Receive buffer	Receive data	○ ○
—	Nothing is assigned. When read, their value are indeterminate.		— —
OER	Overrun error flag (Note)	0 : No overrun error 1 : Overrun error found	○ —
FER	Framing error flag (Note)	0 : No Framing error 1 : Framing error found	○ —
PER	Parity error flag (Note)	0 : No parity error 1 : Parity error found	○ —
—	Nothing is assigned. When read, its value is indeterminate.		— —

Note: Only effective for receive data.

Group i SI/O communication mode register (i=0,1)



Symbol: GiMR (i=0,1) Address: 00ED₁₆, 012D₁₆ When reset: 00₁₆

Bit symbol	Bit name	Function	R W
GMD0	Communication mode select bit	b1 b0 0 0 : UART mode 0 1 : Serial I/O mode	○ ○
GMD1		1 0 : Special communication mode 1 1 : HDLC data process mode	○ ○
CKDIR	Internal/external clock select bit	0 : Internal clock (Note 2) 1 : External clock (Note 3)	○ ○
STPS	Stop bit length select bit (Note 1)	0 : 1 stop bit 1 : 2 stop bits	○ ○
PRY	Odd/even parity select bit (Note 1)	0 : Odd parity 1 : Even parity	○ ○
PRYE	Parity enable select bit (Note 1)	0 : Parity disabled 1 : Parity enabled	○ ○
UFORM	Transfer direction select bit	0 : LSB first 1 : MSB first	○ ○
IRS	Transmit interrupt cause select bit	0 : Transmit buffer is empty 1 : Transmit is completed	○ ○

Note 1: Can be used only in the UART mode.

Note 2: Select a pin for clock output by setting the waveform generation control register, input function select register, and function select registers A, B and C.
Data transmission pins are the same as clock output pins.

Note 3: Select which pins will input the clock with the input function select register and set those pins to the input port using function select register A.
Data input pins are the same as with clock input pins.

Figure 1. 23. 27. Group 0 and 1 related register (2)

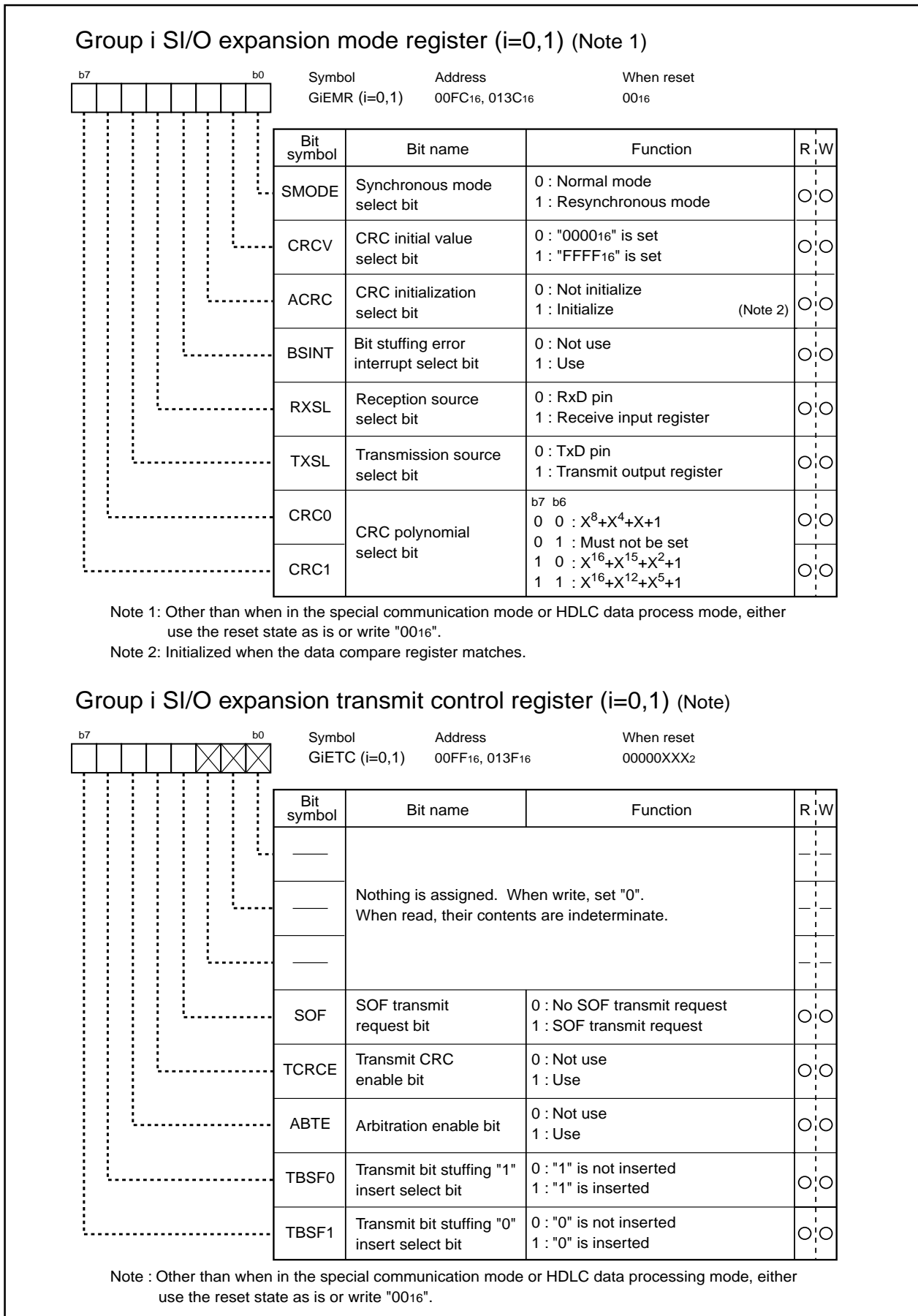


Figure 1. 23. 28. Group 0 and 1 related register (3)

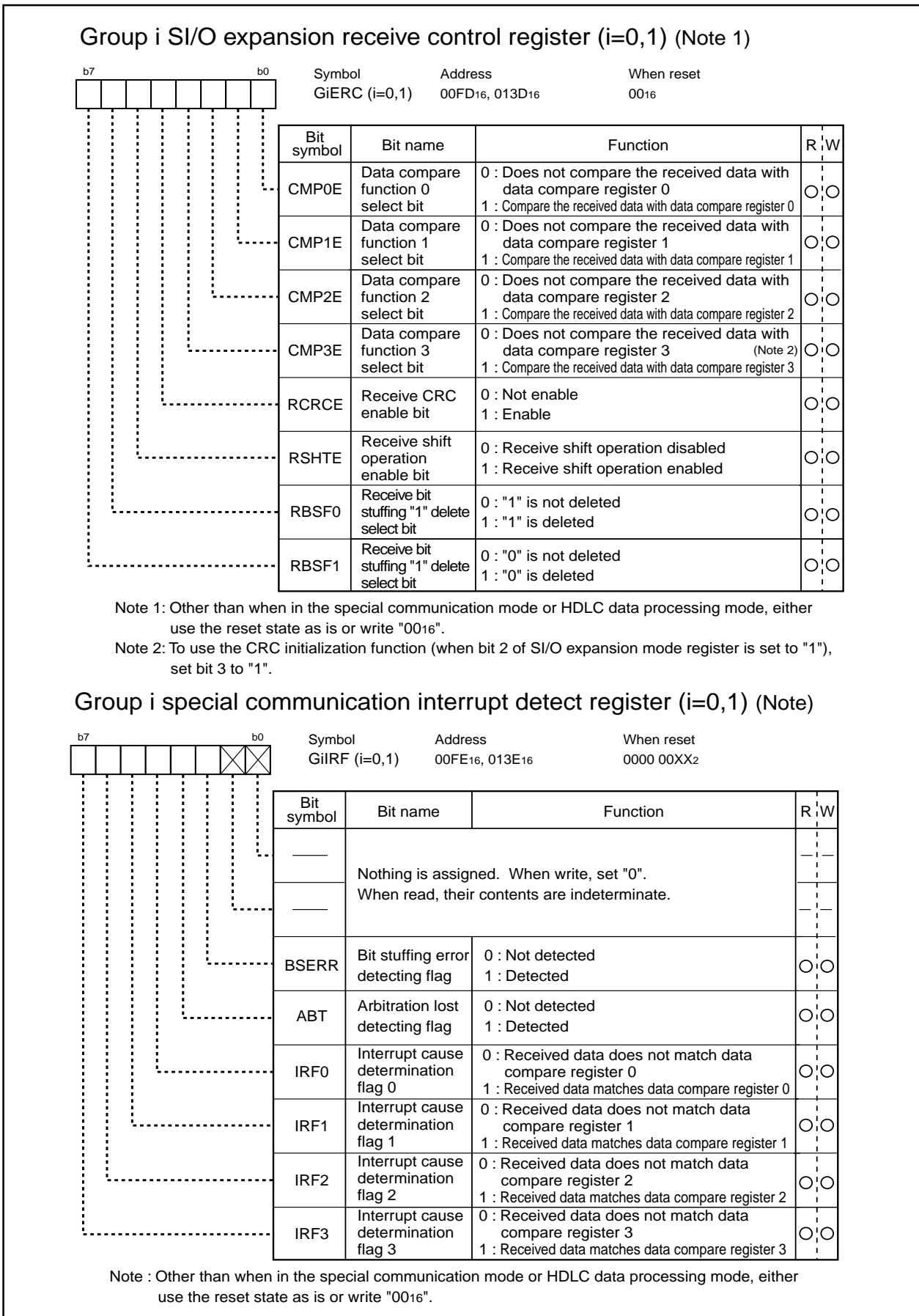


Figure 1. 23. 29. Group 0 and 1 related register (4)

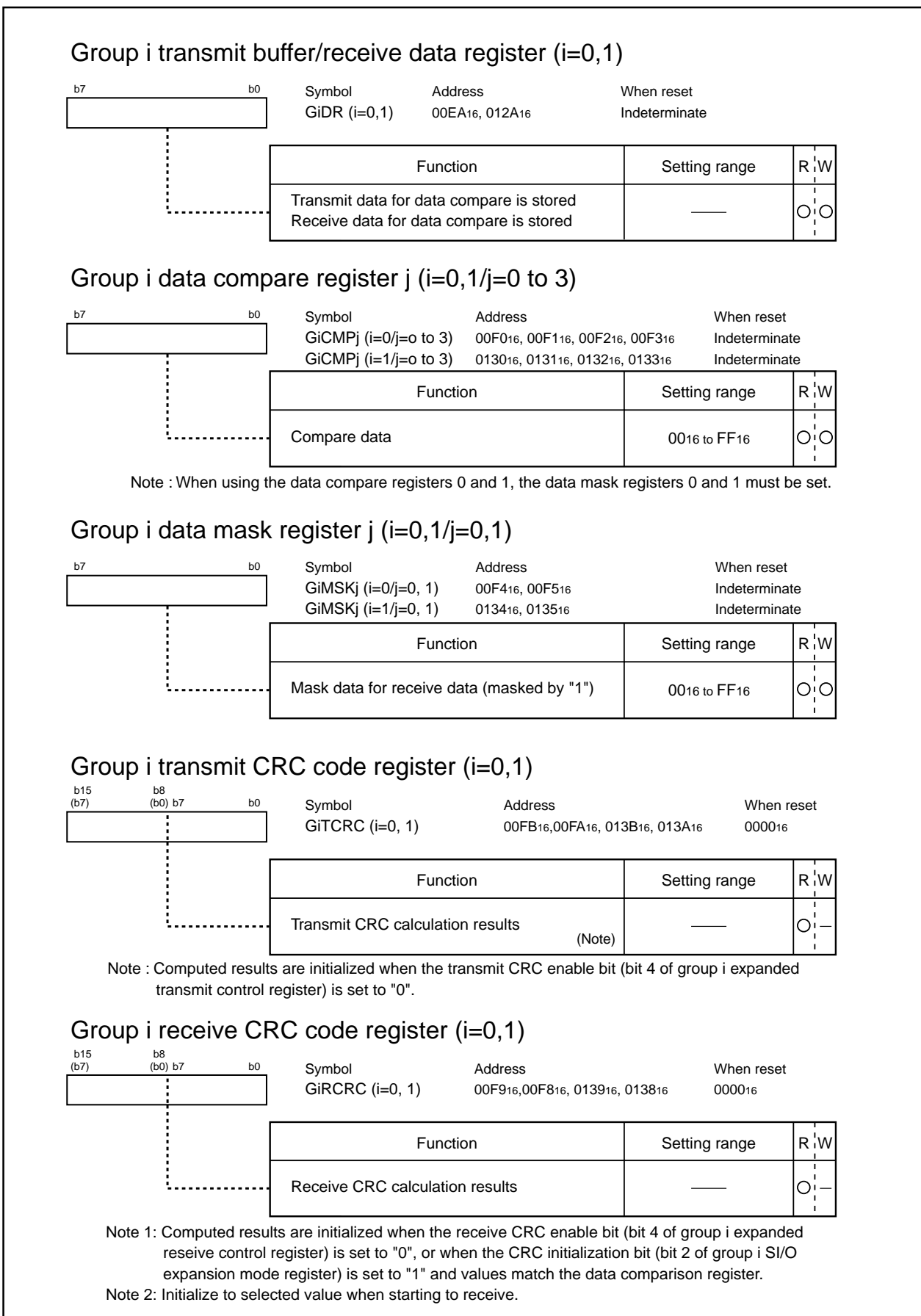


Figure 1. 23. 30. Group 0 and 1 related register (5)

• Clock synchronous serial I/O mode (group 0 and 1)

Table 1.23.12 gives specifications for the clock synchronous serial I/O mode.

Table 1.23.12. Specifications of clock synchronous serial I/O mode (group 0 and 1)

Item	Specification
Transfer data format	<ul style="list-style-type: none"> Transfer data length: 8 bits fixed
Transfer clock	<ul style="list-style-type: none"> When internal clock is selected <ul style="list-style-type: none"> Transfer speed is determined when the base timer is reset by the ch0 WG function Transfer rate (bps) = base timer count source (frequency) / (k+2) / 2 k : values set to WG register 0 Transfer clock is generated when the transfer clock in the phase delayed waveform output mode Transmit clock : ch3 WG function Receive clock : ch2 WG function Sets the same value in the WG registers on ch2 and ch3 When external clock is selected <ul style="list-style-type: none"> Transfer rate (bps) = Clock input to ISCLK pin
Transmission start condition	To start transmission, the following requirements must be met: <ul style="list-style-type: none"> Transmit enable bit = "1" Write data to transmit buffer
Reception start condition	To start reception, the following requirements must be met: <ul style="list-style-type: none"> Receive enable bit = "1"
Interrupt request generation timing	<ul style="list-style-type: none"> When transmitting <ul style="list-style-type: none"> When transmit buffer is empty, transmit interrupt cause select bit = "0" When transmission is completed, transmit interrupt cause select bit = "1" When receiving When data is transferred to SI/O receive buffer register
Error detection	<ul style="list-style-type: none"> Overrun error This error occurs when the next data is ready before the contents of SI/O receive buffer register are read out
Select function	<ul style="list-style-type: none"> LSB first/MSB first selection When transmission/reception begins with bit 0 or bit 7, it can be selected Transmit/receive data polarity switching This function is reversing ISTxD pin output and ISRxD pin input. (All I/O data level is reversed.)

Note: Set the transmission clock to at least 6 divisions of the base timer clock.

Table 1.23.13 lists I/O pin functions for the clock synchronous serial I/O mode of groups 0 and 1. From when the operating mode is selected until transmission starts, the ISTxDi pin is "H" level. Figure 1.23.31 shows typical transmit/receive timings in clock synchronous serial I/O mode in group 0 and 1.

Table 1.23.13. I/O pin functions in clock synchronous serial I/O mode of group 0, group 1

Pin name	Function	Selected method
ISTxD (P76, P150, P73, P110)	Serial data output	<ul style="list-style-type: none"> • Use the ch0 WG function • Sets "111" for the operating mode select bit (bits 2, 1 and 0) in WG control register 0 • Selects ISTxD output for the port using function select registers A, B and C
ISRxD (P80, P152, P75, P112)	Serial data input	<ul style="list-style-type: none"> • Selects a using port with input function select register • Selects I/O with function select register A • Sets a selected port to input using the port direction register
ISCLK (P77, P151, P74, P111)	Transfer clock output	<ul style="list-style-type: none"> • Use the ch1 WG function • Sets "111" for the operating mode select bit (bits 2, 1 and 0) in WG control register 1 • Sets "0" for the internal/external clock select bit (bit 2) of the S/I/O communication mode register • Selects ISCLK output for the port using function select registers A, B and C
	Transfer clock input	<ul style="list-style-type: none"> • Selects a using port with input function select register • Sets "1" for the internal/external clock select bit (bit 2) of the S/I/O communication mode register • Sets a selected port to input using the port direction register • Selects I/O port with function select register A

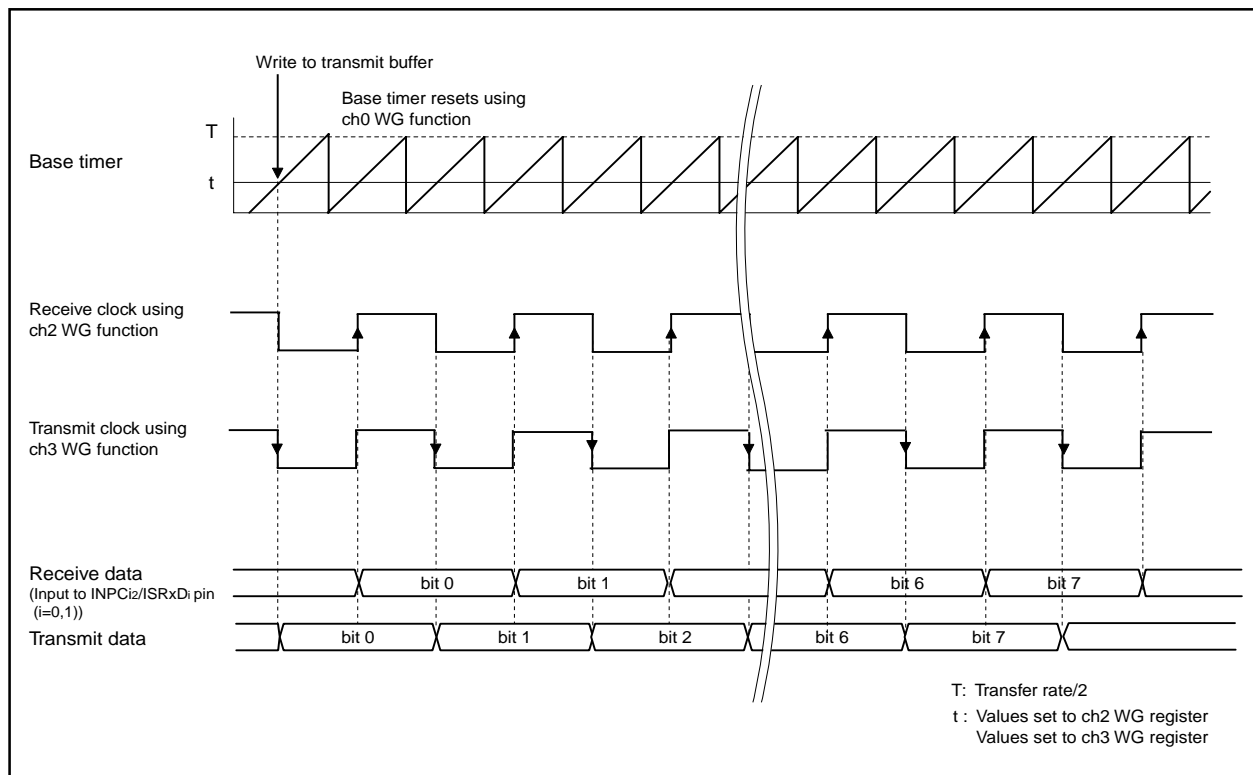


Figure 1.23.31. Typical transmit/receive timings in clock synchronous serial I/O mode in group 0 and 1

(2) Clock asynchronous serial I/O mode (UART) (group 0 and 1)

Table 1.23.14 lists the specifications for the UART mode.

Table 1.23.14. Specifications of UART mode

Item	Specification
Transfer data format	<ul style="list-style-type: none"> • Character bit (transfer data) : 8 bits • Start bit : 1 bit • Parity bit : Odd, even, or nothing selected • Stop bit : 1 bit or 2 bits selected
Transfer clock	<ul style="list-style-type: none"> • When internal clock is selected (Generates the transmit/receive clock in the phase delayed waveform output mode) <ul style="list-style-type: none"> – Transfer speed is determined when the base timer is reset by the ch0 WG function Transfer rate (bps) = base timer count source (frequency) / (k+2) / 2 k : values set to WG register 0 – Transfer clock is generated when the transfer clock in the phase delayed waveform output mode Transmit clock : ch3 WG function Receive clock : Change ch2 TM function to WG function Detects falling edge of start bit Changes to the WG mode when the time measurement interrupt arrives • When external clock is selected <ul style="list-style-type: none"> – Transfer rate (bps) = Clock input to ISCLK pin
Transmission start condition	To start transmission, the following requirements must be met: <ul style="list-style-type: none"> • Transmit enable bit = "1" • Write data to transmit buffer
Reception start condition	To start reception, the following requirements must be met: <ul style="list-style-type: none"> • Receive enable bit = "1"
Interrupt request generation timing	<ul style="list-style-type: none"> • When transmitting <ul style="list-style-type: none"> – When transmit buffer is empty, transmit interrupt cause select bit = "0" – When transmission is completed, transmit interrupt cause select bit = "1" • When receiving <ul style="list-style-type: none"> – When data is transferred to SI/O receive buffer register
Error detection	<ul style="list-style-type: none"> • Overrun error : This error occurs when the next data is ready before contents of SI/O 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
Select function	<ul style="list-style-type: none"> • Stop bit length : Stop bit length can be selected as 1 bit or 2 bits • Parity : Parity can be turned on/off : When parity is on, odd/even parity can be selected • LSB first/MSB first selection : Whether transmit/receive begins with bit 0 or bit 7 can be selected • Transmit/receive data polarity switching : This function is reversing ISTxD port output and ISRxD port input. (All I/O data level are reversed.) • Data transfer bit length : Transmission data length can be set between 1 to 8 bits

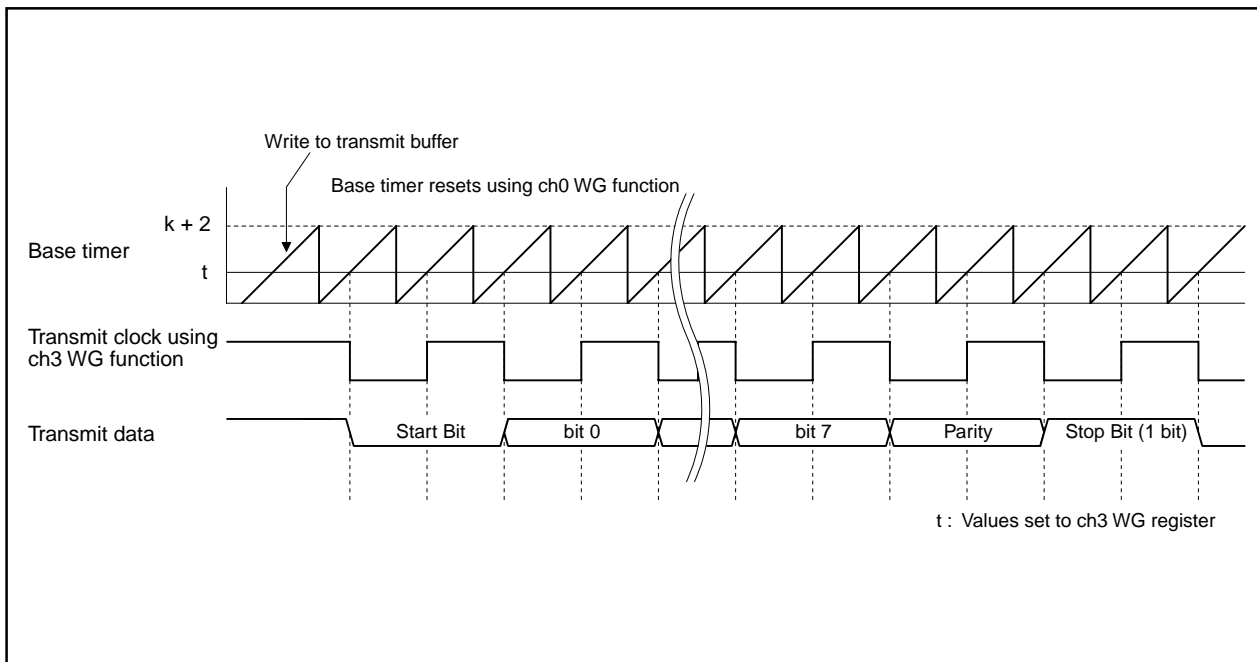


Figure 1.23.32. Typical transmit timings in UART mode

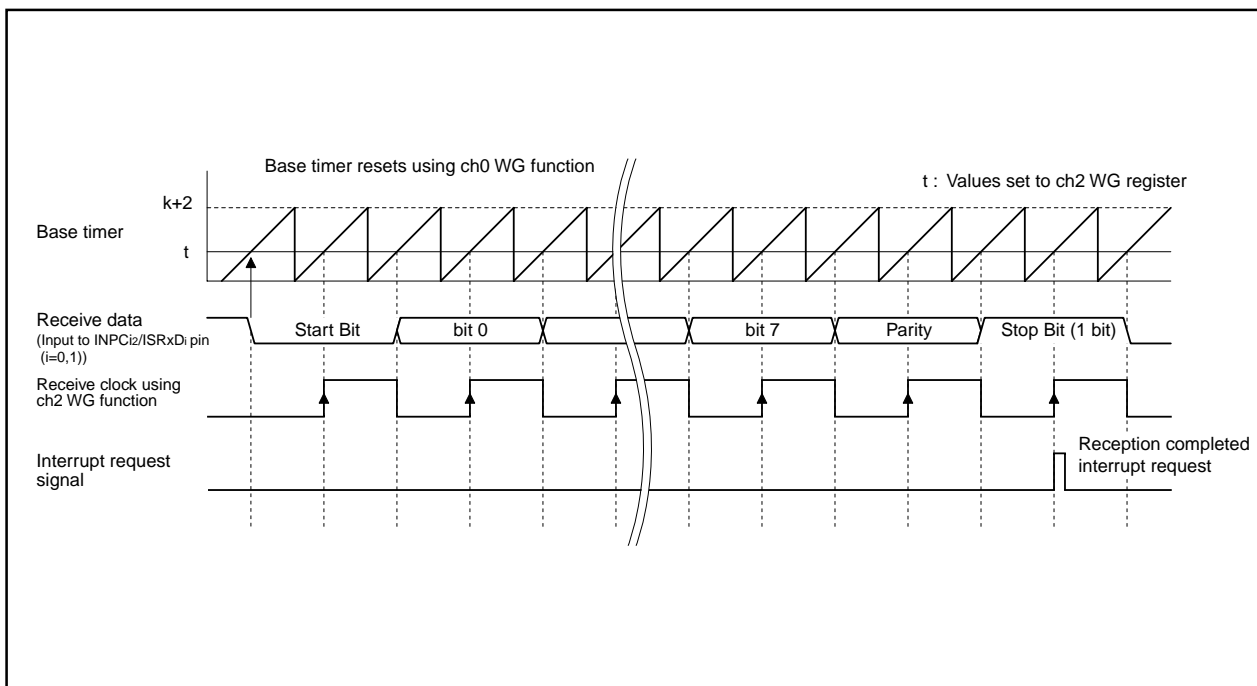


Figure 1.23.33. Typical receive timing in UART mode

TxD, RxD I/O polarity reverse function

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) are reversed. TxD output polarity reverse select bit is set to "0" (not to reverse) for usual use.

(2) Clock synchronous serial I/O mode (group 2)

Intelligent I/O groups 2 has communication block that have two internal 8-bit shift registers. When used in conjunction with the communication block and WG function, these shift registers enable variable clock synchronous and IE Bus ^(Note) communications.

Table 1.23.16 lists using registers in group 2, figure 1.23.34 to 1.23.37 shows the related registers.

Note : IE Bus is a trademark of NEC corporation.

Table 1.23.16. Using registers in group 2

	Clock synchronous serial I/O	IE Bus
Base timer control register 0	√	√
Base timer control register 1	√	√
Waveform generate control register 0	√	√
Waveform generate control register 1	—	√
Waveform generate control register 2	√	√
Waveform generate control register 3	—	√
Waveform generate control register 4	—	√ (Note 1)
Waveform generate control register 5	—	√
Waveform generate control register 6	—	√
Waveform generate control register 7	—	√
Waveform generate register 0	√	√
Waveform generate register 1	—	√
Waveform generate register 2	√	√
Waveform generate register 3	—	√
Waveform generate register 4	—	√
Waveform generate register 5	—	√
Waveform generate register 6	—	√
Waveform generate register 7	—	√
Function enable register	√	√
SI/O communication mode register	√	√
SI/O communication control register	√	√
IE Bus control register	—	√
IE Bus address register	—	√
IE Bus transmit interrupt cause detect register	—	√
IE Bus receive interrupt cause detect register	—	√
SI/O receive buffer register	√	√
SI/O transmit buffer register	√	√

√ : Use — : Not use

Note 1: When receiving slave, set corresponding value with 32.5 μs. Don't set 170 μs.

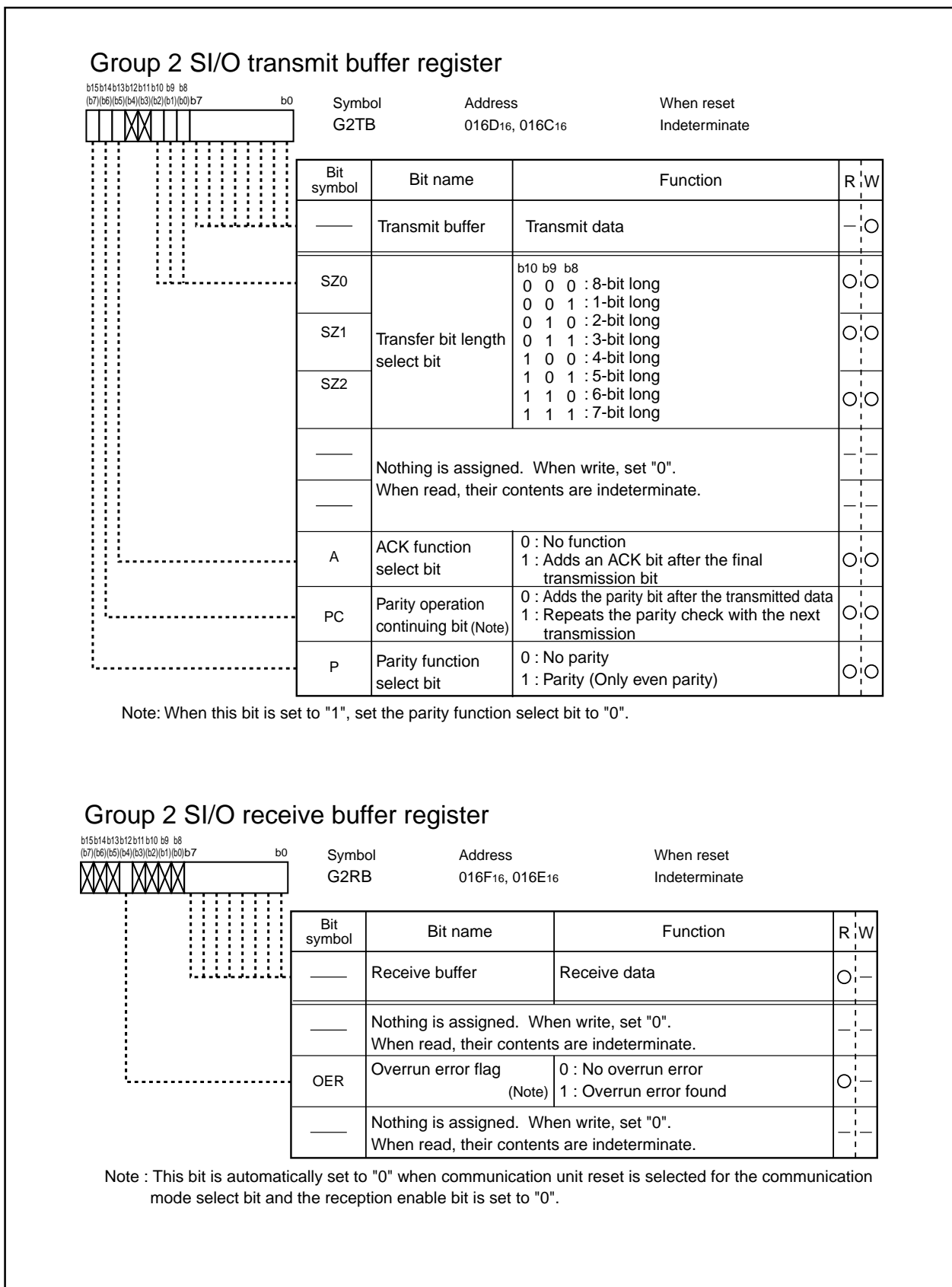


Figure 1. 23. 34. Group 2 Intelligent I/O-related register (1)

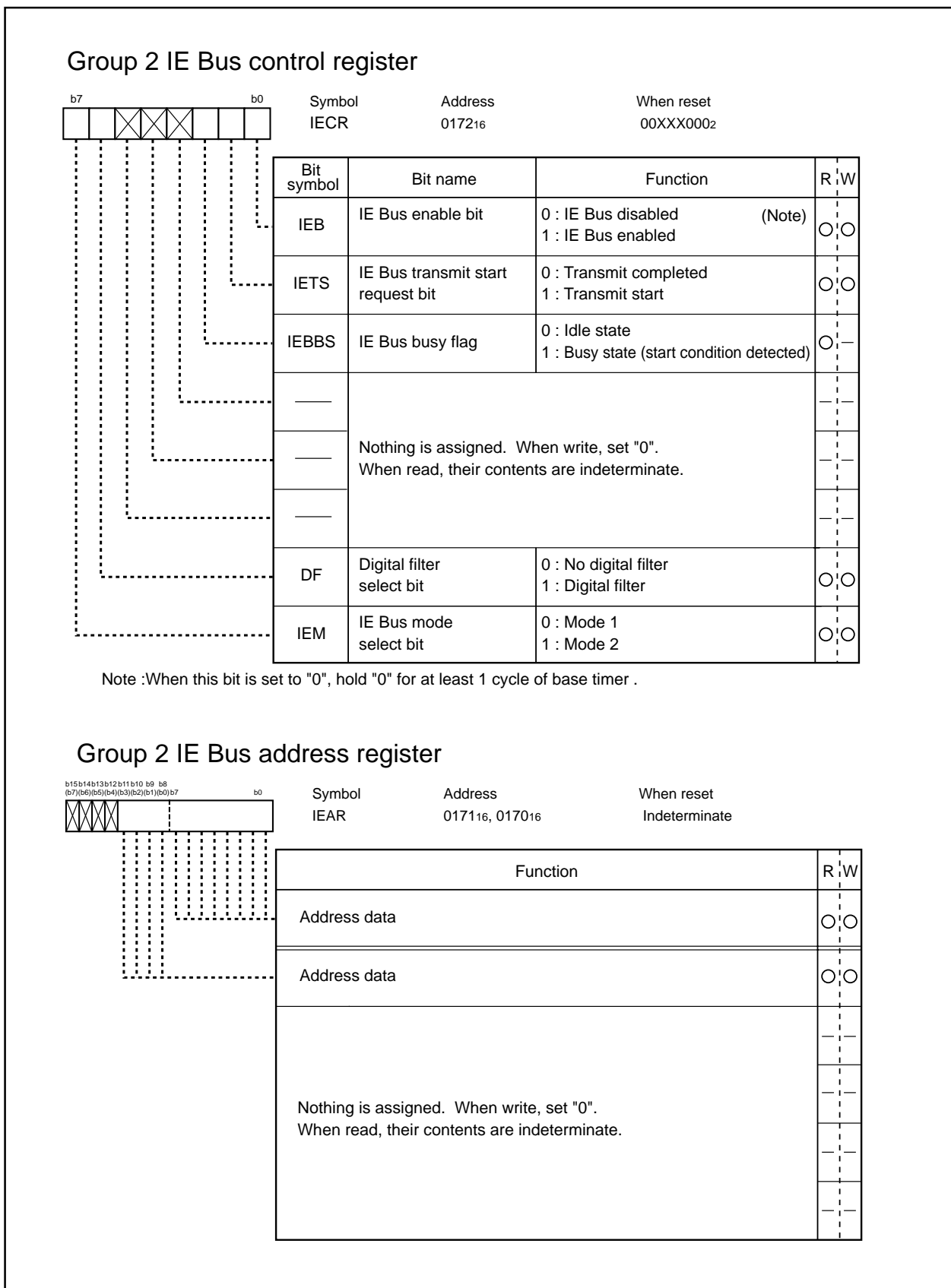


Figure 1. 23. 35. Group 2 Intelligent I/O-related register (2)

Group 2 IE Bus transmit interrupt cause determination register

		Symbol IETIF	Address 0173 ₁₆	When reset XXX00000 ₂
Bit symbol	Bit name	Function	R	W
IETNF	Normal termination flag	0 : Terminated in error 1 : Terminated normally (Note)	○	○
IEACK	ACK error flag	0 : No error 1 : Error found (Note)	○	○
IETMB	Max. transfer byte error flag	0 : No error 1 : Error found (Note)	○	○
IETT	Timing error flag	0 : No error 1 : Error found (Note)	○	○
IEABL	Arbitration lost flag	0 : No error 1 : Error found (Note)	○	○
—	Nothing is assigned. When write, set "0". When read, their contents are indeterminate.		—	—
—			—	—
—			—	—

Note : Only "0" can be written for this bit. Also, it is cleared to "0" when "0" is written for bit 0 of the IE Bus control register. At this time, hold "0" for at least 1 cycle of base timer clock.

Group 2 IE Bus receive interrupt cause determination register

		Symbol IERIF	Address 0174 ₁₆	When reset XXX00000 ₂
Bit symbol	Bit name	Function	R	W
IERNF	Normal termination flag	0 : Terminated in error 1 : Terminated normally (Note)	○	○
IEPAR	Parity error flag	0 : No error 1 : Error found (Note)	○	○
IERMB	Max. transfer byte error flag	0 : No error 1 : Error found (Note)	○	○
IERT	Timing error flag	0 : No error 1 : Error found (Note)	○	○
IERETC	Other cause receive completed flag	0 : No error 1 : Error found (Note)	○	○
—	Nothing is assigned. When write, set "0". When read, their contents are indeterminate.		—	—
—			—	—
—			—	—

Note : Only "0" can be written for this bit. Also, it is cleared to "0" when "0" is written for bit 0 of the IE Bus control register. At this time, hold "0" for at least 1 cycle of base timer clock.

Figure 1. 23. 36. Group 2 Intelligent I/O-related register (3)

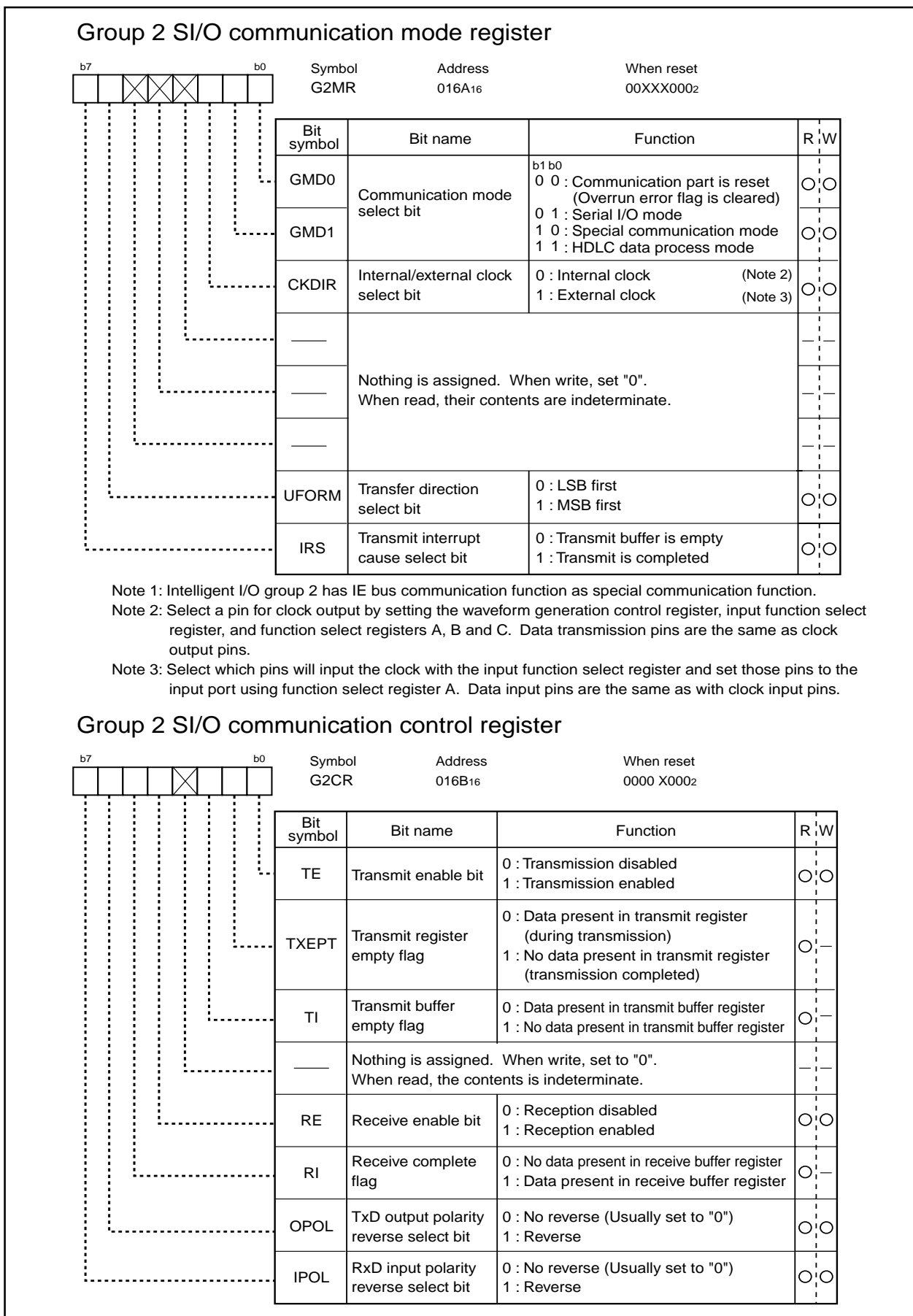


Figure 1. 23. 37. Group 2 Intelligent I/O-related register (4)

• Clock synchronous serial I/O mode (group 2)

Table 1.23.17 gives specifications for the group 2 clock synchronous serial I/O mode.

Table 1.23.17. Specifications of clock synchronous serial I/O mode

Item	Specification
Transfer data format	<ul style="list-style-type: none"> • Transfer data length: Variable length (group2)
Transfer clock	<ul style="list-style-type: none"> • When internal clock is selected, the transfer clock in the single waveform output mode is generated. <ul style="list-style-type: none"> – Transfer speed is determined when the base timer is reset by the ch0 WG function Transfer rate (bps) = base timer count source (frequency) / (k+2) k : values set to WG register 0 – Transfer clock is generated by ch2 single phase WG function Ch3 WG register = (k+2)/2 (Note 1) • When external clock is selected <ul style="list-style-type: none"> – Transfer rate (bps) = Clock input to ISCLK pin (Note 2)
Transmission start condition	<ul style="list-style-type: none"> • To start transmission, the following requirements must be met: <ul style="list-style-type: none"> – Transmit enable bit = "1" – Write data to SI/O transmit buffer register
Reception start condition	<ul style="list-style-type: none"> • To start reception, the following requirements must be met: <ul style="list-style-type: none"> – Receive enable bit = "1" – Transmit enable bit = "1" – Write data to SI/O transmit buffer register
Interrupt request generation timing	<ul style="list-style-type: none"> • When transmitting <ul style="list-style-type: none"> – When SI/O communication buffer register is empty, transmit interrupt cause select bit = "0" – When transmission is completed, transmit interrupt cause select bit = "1" • When receiving <ul style="list-style-type: none"> – When data is transferred to SI/O receive buffer register
Error detection	<ul style="list-style-type: none"> • Overrun error This error occurs when the next data is ready before the contents of SI/O receive buffer register are read out
Select function	<ul style="list-style-type: none"> • LSB first/MSB first selection When transmission/reception begins with bit 0 or bit 7, it can be selected. • Transmit/receive data polarity switching <ul style="list-style-type: none"> – This function is reversing ISTxD pin output and ISRxD pin input. (All I/O data level is reversed.) • Data transfer bit length <ul style="list-style-type: none"> – Transmission data length can be set between 1 to 8 bits

Note 1: When the transfer clock and transfer data are transmission, transfer clock is set to at least 6 divisions of the base timer clock. Except this, transfer clock is set to at least 20 divisions of the base timer clock.

Note 2: Transfer clock is set to at least 20 divisions of the base timer clock.

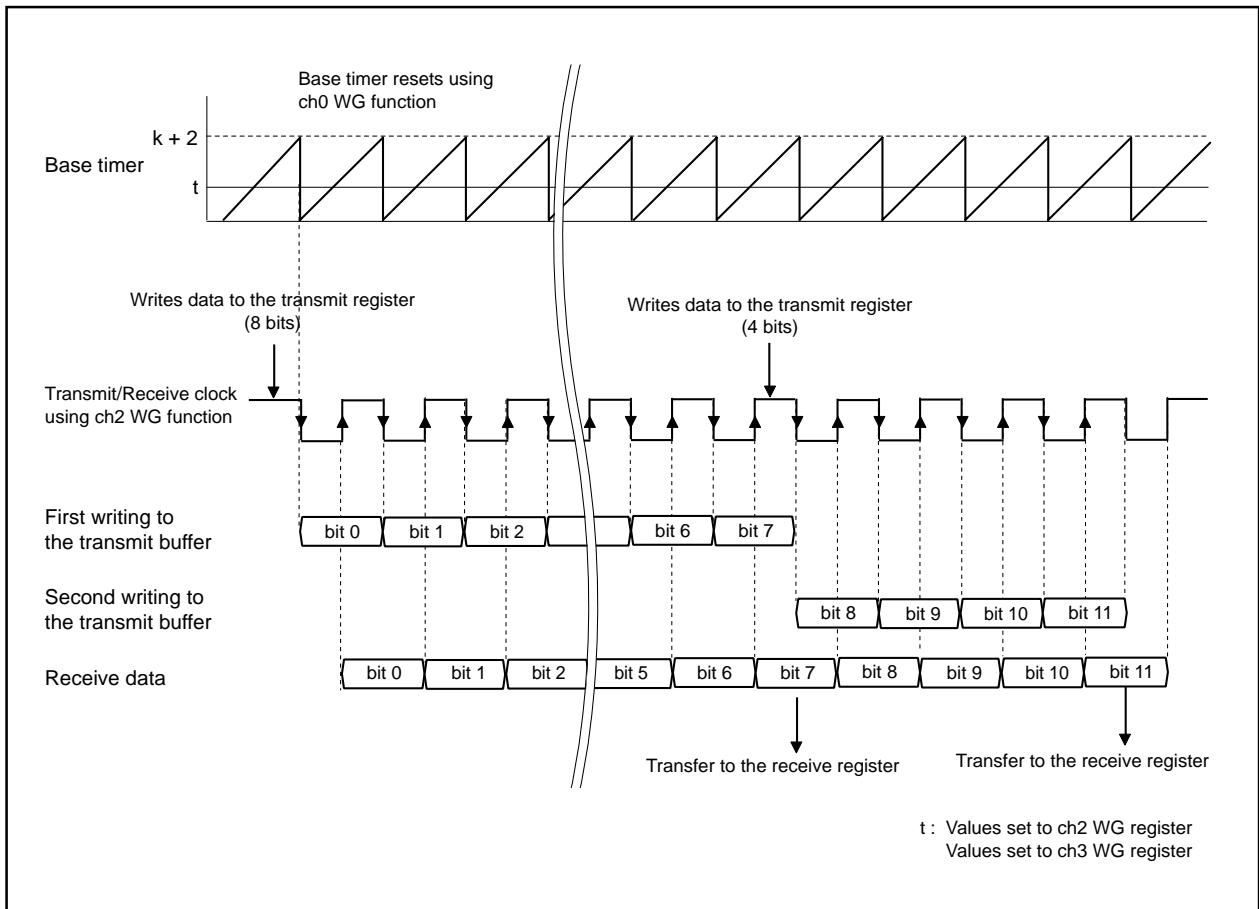


Figure 1. 23. 38. Typical transmit/receive timings in clock synchronous serial I/O mode in group 2

A-D Converter

A-D Converter

The A-D converter consists of two 10-bit successive approximation A-D converter circuit with a capacitive coupling amplifier. Pins P10₀ to P10₇, P15₀ to P15₇, P0₀ to P0₇, P2₀ to P2₇, P9₅, and P9₆ are shared as the analog signal input pins. Pins P15₀ to P15₇, P0₀ to P0₇ and P2₀ to P2₇ can be used as the analog signal input pins and switched by analog input port select bit. However, P0₀ to P0₇ and P2₀ to P2₇ can be used in single chip mode. Set input to direction register corresponding to a pin doing A-D conversion.

The result of A-D conversion is stored in the A-D registers of the selected pins.

Table 1.24.1 shows the performance of the A-D converter. Figure 1.24.1 shows the block diagram of the A-D converter, and Figures 1.24.2 to 1.24.7 show the A-D converter-related registers.

This section is described to 144-pin version as example.

In 100-pin version, AN10 to AN17 cannot be selected because there is no P15.

A-D Converter

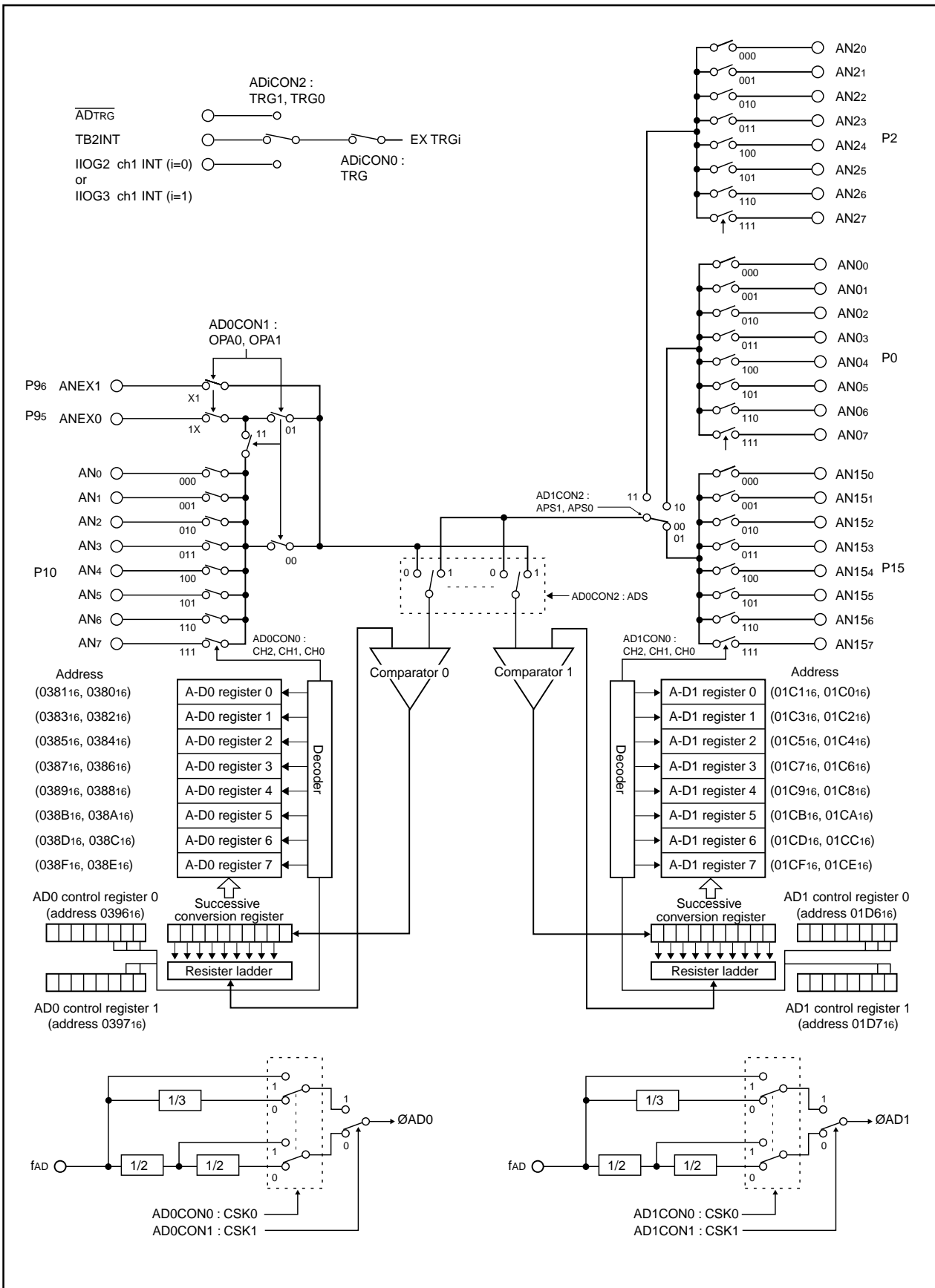


Figure 1.24.1. Block diagram of A-D converter

A-D Converter

Table 1.24.1. Performance of A-D converter

Item	Performance
Method of A-D conversion	Successive approximation (capacitive coupling amplifier)
Analog input voltage (Note 1)	0V to AVCC (VCC)
Operating clock ϕ_{AD} (Note 2)	fAD, fAD/2, fAD/3, fAD/4 fAD=f(XIN)
Resolution	8-bit or 10-bit (selectable)
Operating modes	One-shot mode, repeat mode, single sweep mode, repeat sweep mode 0, and repeat sweep mode 1
Analog input pins	34 pins AN, AN0, AN2, AN15(Note 3) each 8 pins Extended input 2 pins (ANEX0(Note 4) and ANEX1(Note 5))
A-D conversion start condition	<ul style="list-style-type: none"> • 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 by outbreak of the following factors chosen among in three (Note 6) <ul style="list-style-type: none"> · ADTRG/P97 input changes from "H" to "L" · Timer B2 interrupt occurrences frequency counter overflow · Interrupt of Intelligent I/O group 2 or 3 channel 1
Conversion speed per pin	<ul style="list-style-type: none"> • Without sample and hold function 8-bit resolution: 49 ϕ_{AD} cycles 10-bit resolution: 59 ϕ_{AD} cycles • With sample and hold function 8-bit resolution: 28 ϕ_{AD} cycles 10-bit resolution: 33 ϕ_{AD} cycles

Note 1: Does not depend on use of sample and hold function.

Note 2: When f(XIN) is over 10 MHz, the fAD frequency must be under 10 MHz by dividing.

Without sample and hold function, set the fAD frequency to 250kHz or more.

With the sample and hold function, set the fAD frequency to 1MHz or more.

Note 3: When port P15 is used as analog input port, port P15 input peripheral function select bit (bit 2 of address 017816) must set to be "1".

Note 4: When port P95 is used as analog input port, port P95 output peripheral function select bit (bit 5 of address 03B716) must set to be "1".

Note 5: When port P96 is used as analog input port, port P96 output peripheral function select bit (bit 6 of address 03B716) must set to be "1".

Note 6: Set the port direction register to input.

A-D Converter

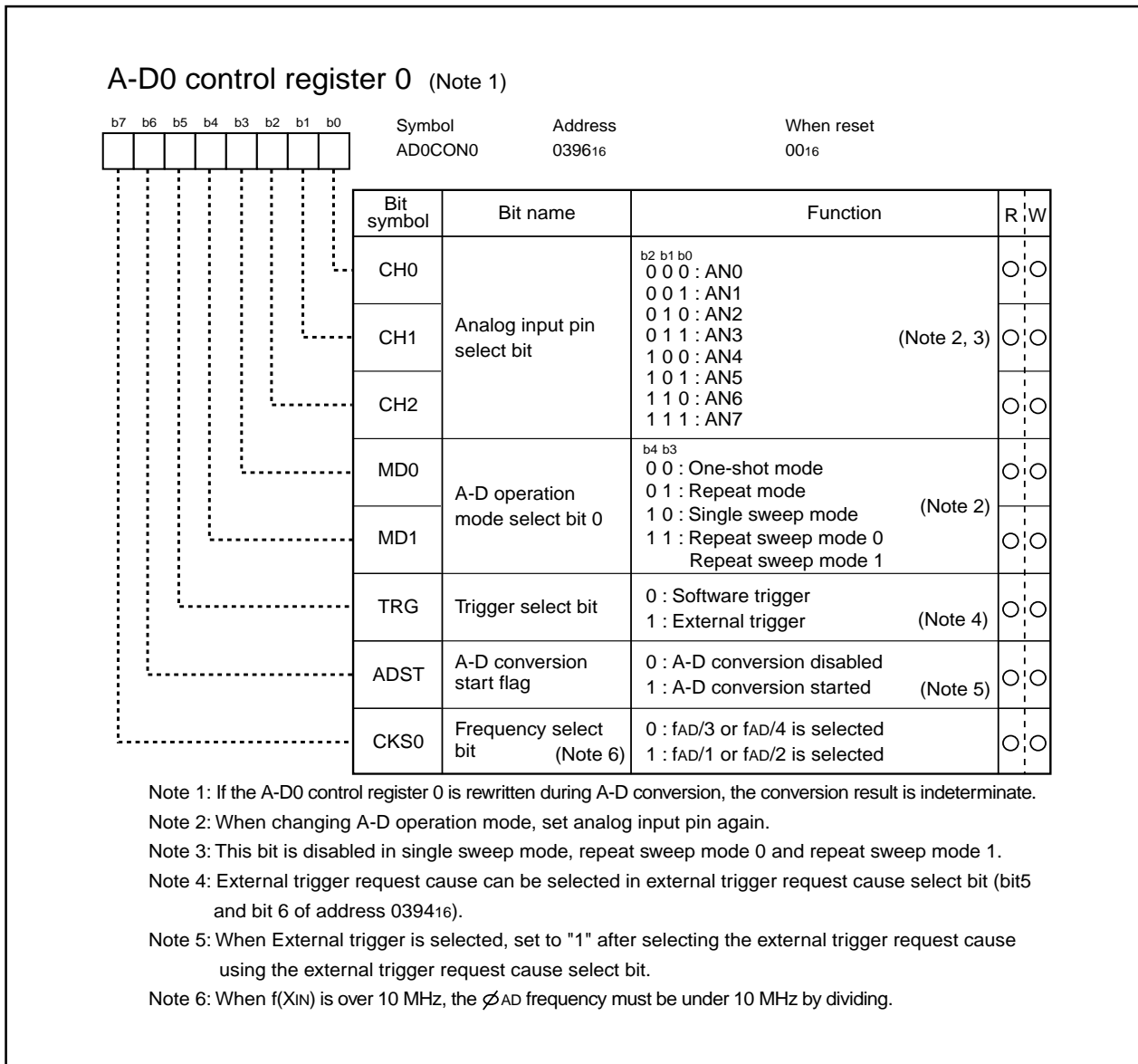


Figure 1.24.2. A-D converter-related registers (1)

A-D Converter

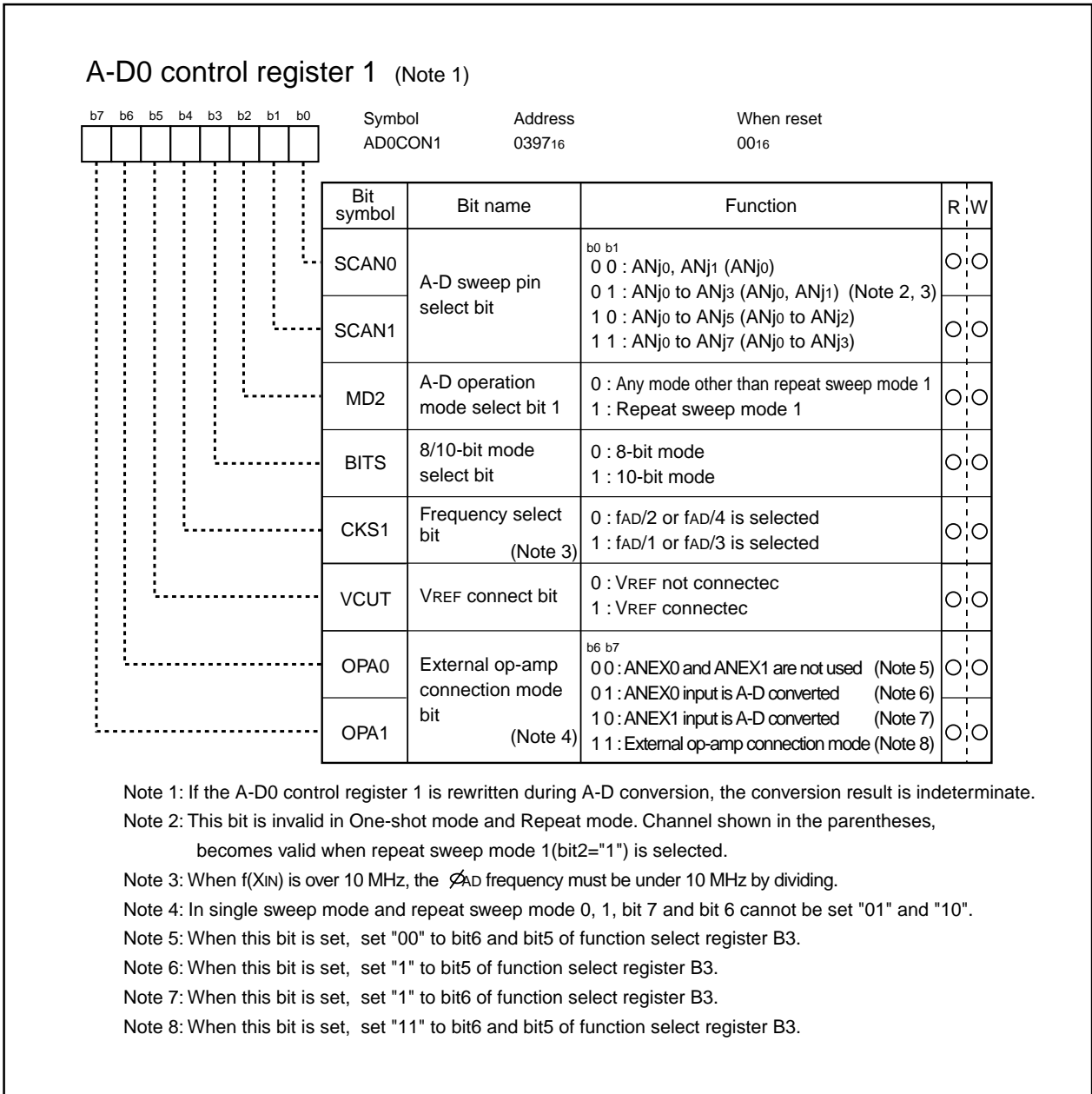


Figure 1.24.3. A-D converter-related registers (2)

A-D Converter

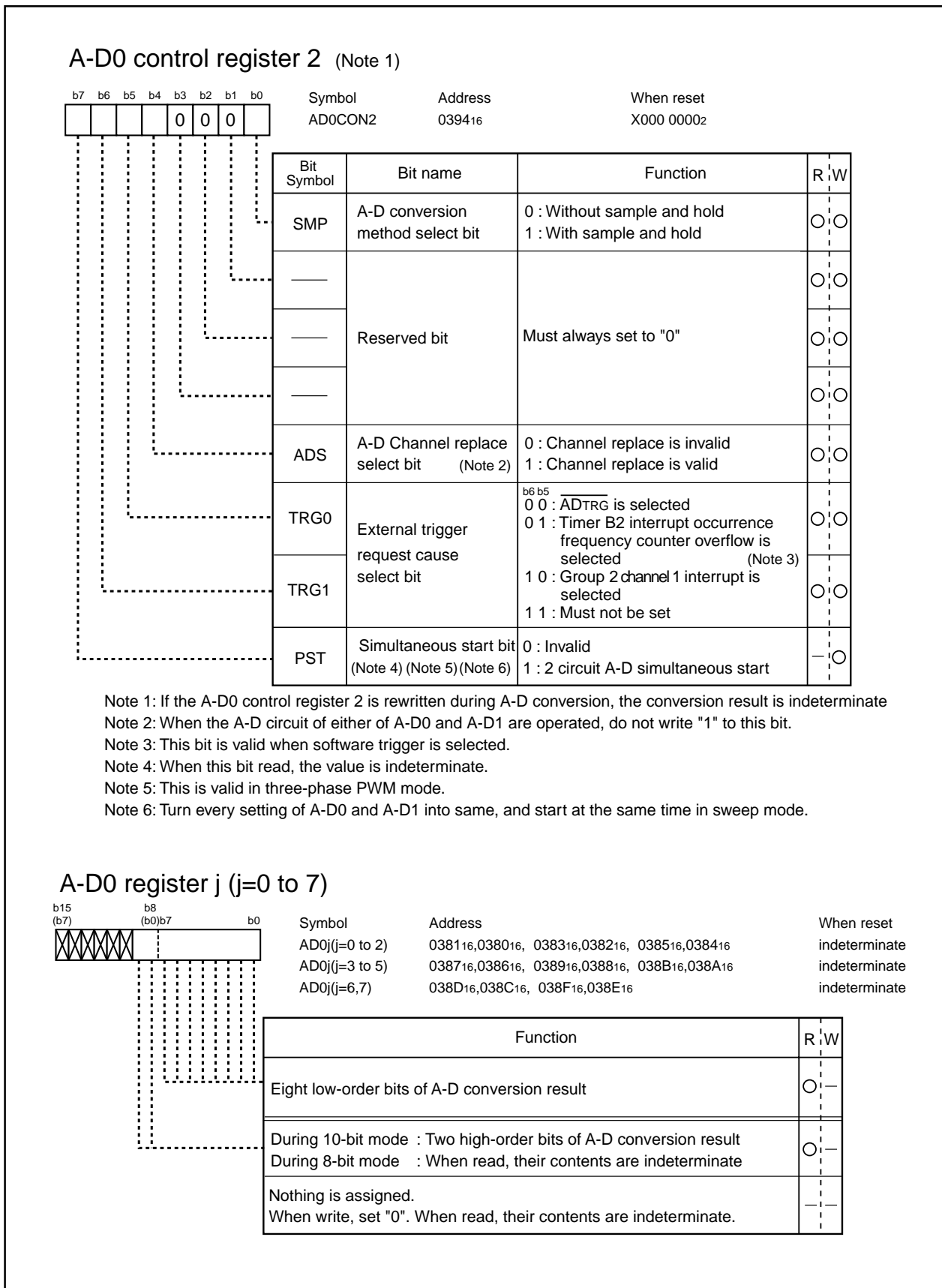


Figure 1.24.4. A-D converter-related registers (3)

A-D Converter

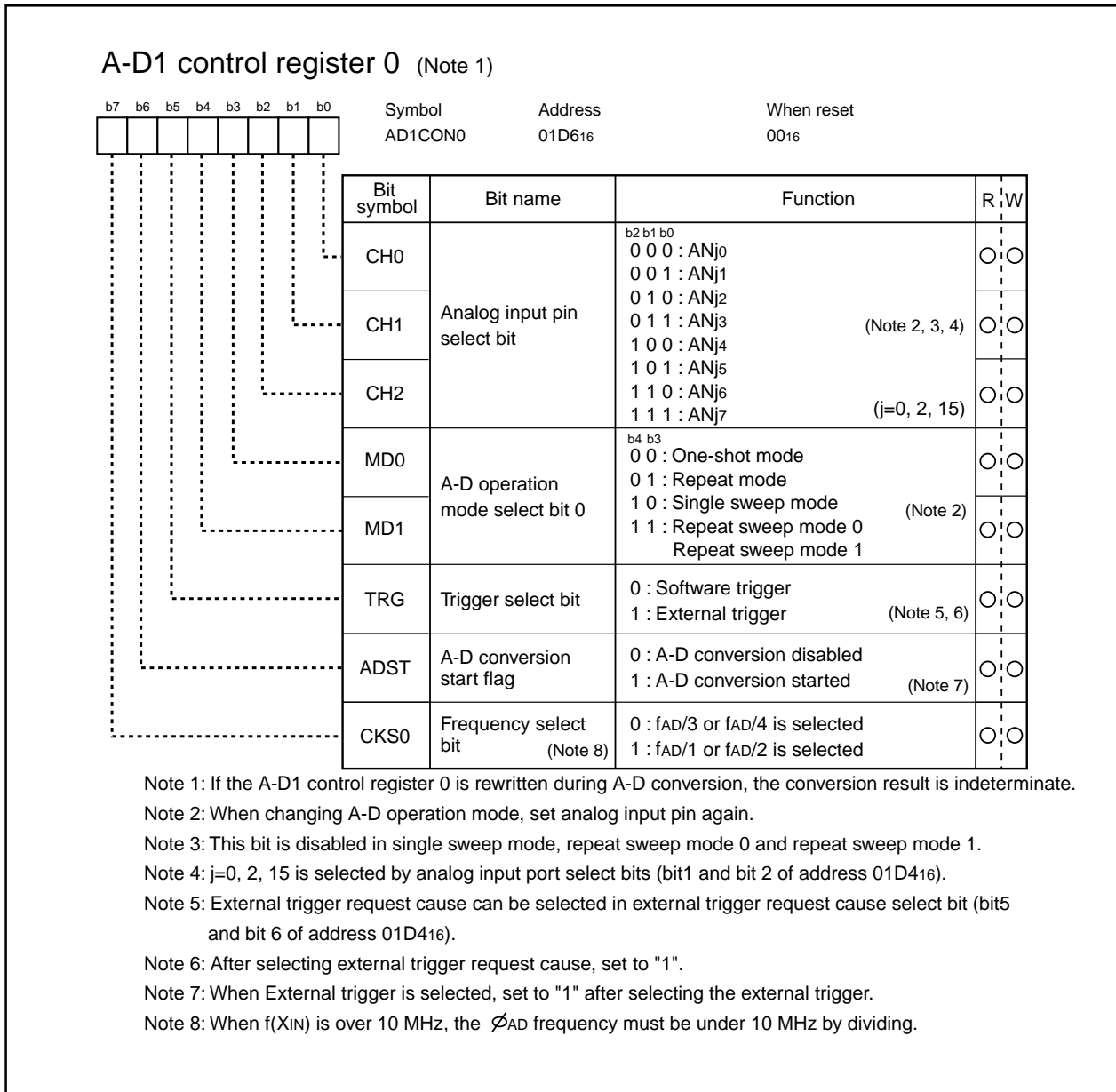


Figure 1.24.5. A-D converter-related registers (4)

A-D Converter

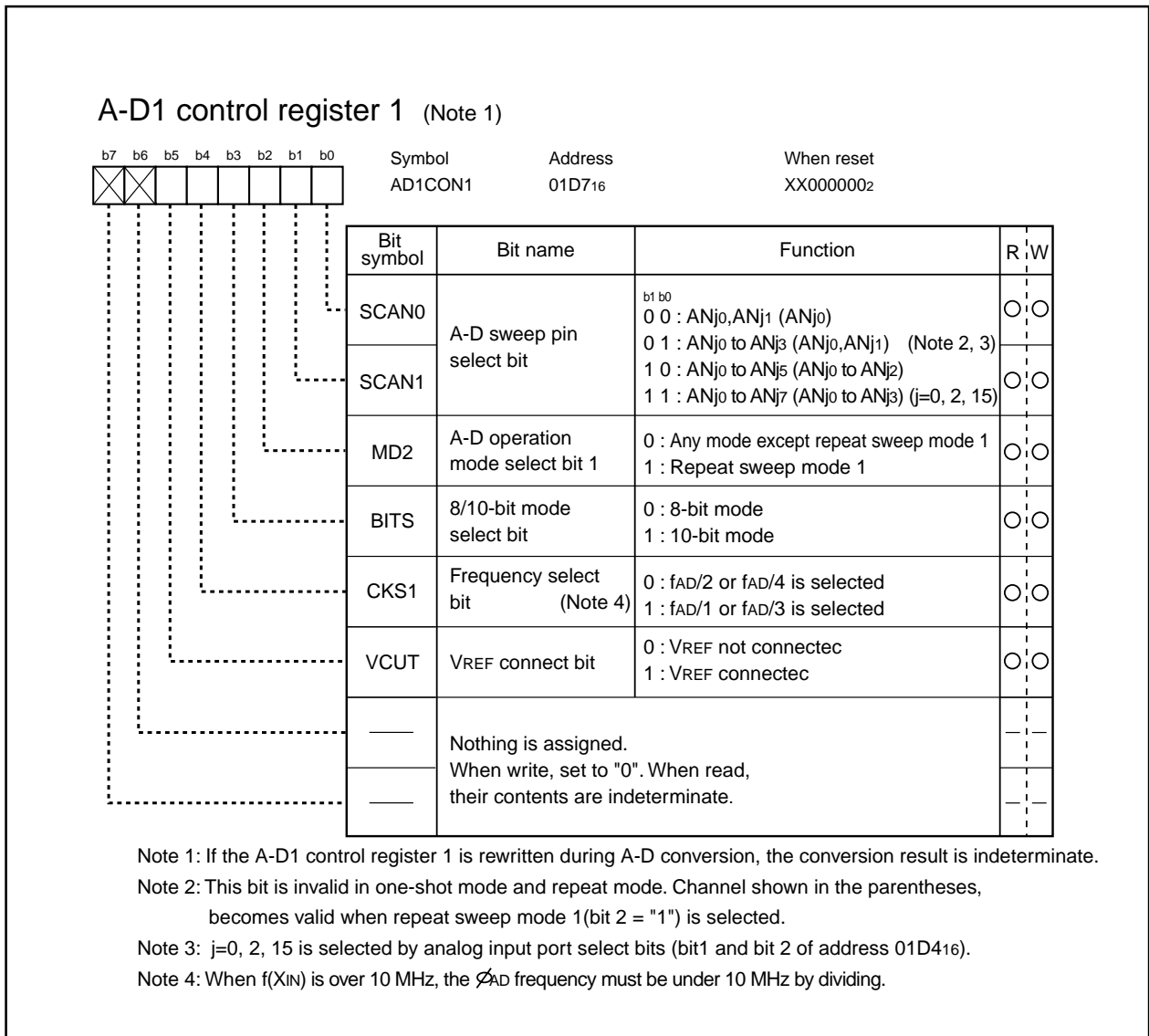


Figure 1.24.6. A-D converter-related registers (5)

A-D Converter

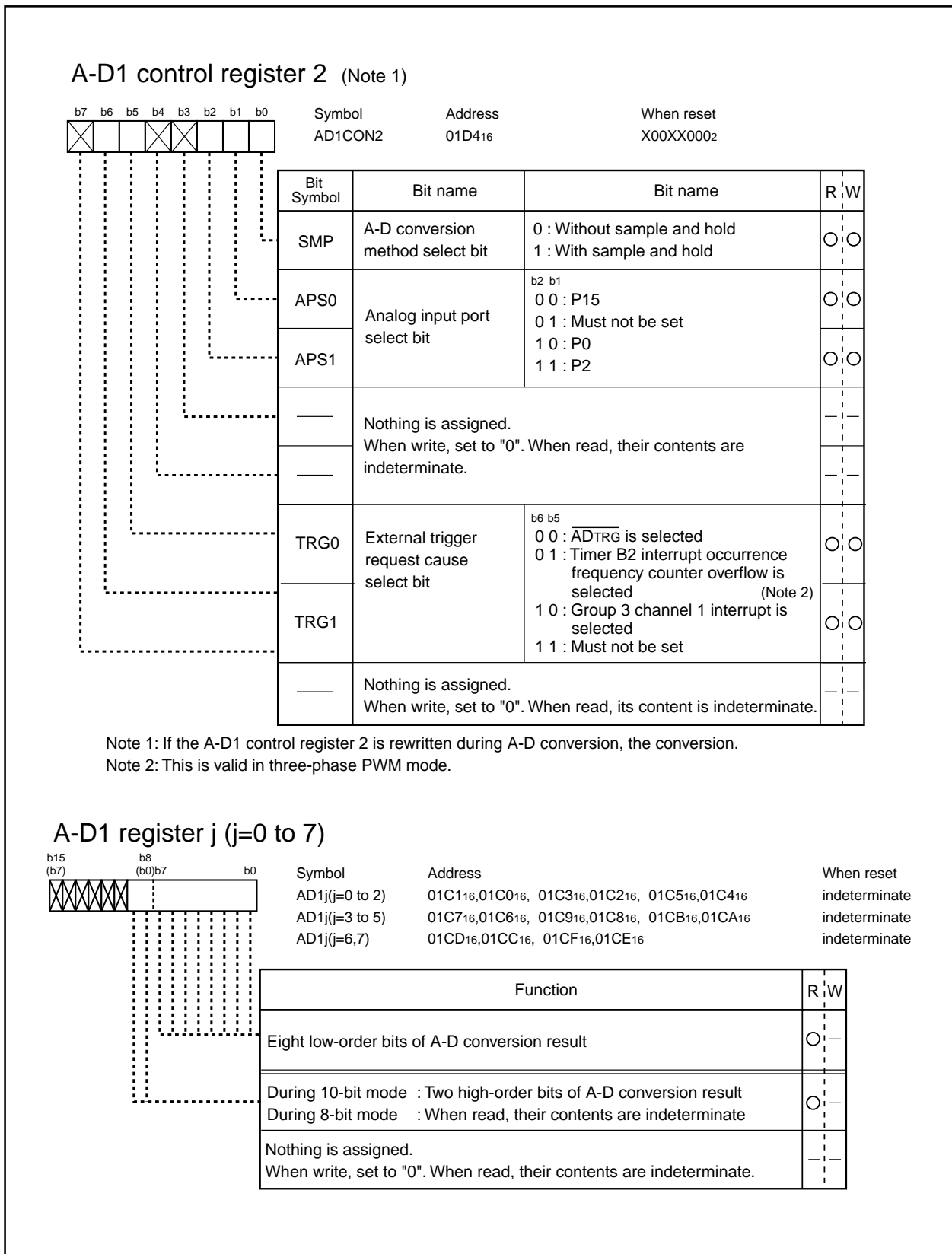


Figure 1.24.7. A-D converter-related registers (6)

A-D Converter

(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 1.24.2 shows the specifications of one-shot mode.

Table 1.24.2. 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-Di conversion start flag, external trigger
Stop condition	<ul style="list-style-type: none"> • End of A-Di conversion (A-Di 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 ANj0 to ANj7 (j =non, 0, 2, 15), ANEX0, ANEX1
Reading of result of A-D converter	Read A-D register corresponding to selected pin

(2) Repeat mode

In repeat mode, the pin selected using the analog input pin select bit is used for repeated A-D conversion. Table 1.24.3 shows the A-D control register in repeat mode.

Table 1.24.3. Repeat mode specifications

Item	Specification
Function	The pin selected by the analog input pin select bit is used for repeated A-D conversion
Start condition	Writing "1" to A-D conversion start flag, external trigger
Stop condition	Writing "0" to A-D conversion start flag
Interrupt request generation timing	None generated
Input pin	One of ANj0 to ANj7 (j =non, 0, 2, 15), ANEX0, ANEX1
Reading of result of A-D converter	Read A-D register corresponding to selected pin

(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 1.24.4 shows the A-D control register in single sweep mode.

Table 1.24.4. Single sweep mode specifications

Item	Specification
Function	The pins selected by the A-Di sweep pin select bit are used for one-by-one A-D conversion
Start condition	Writing "1" to A-D converter start flag, external trigger
Stop condition	<ul style="list-style-type: none"> • End of A-Di conversion (A-D conversion start flag changes to "0", except when external trigger is selected) • Writing "0" to A-Di conversion start flag
Interrupt request generation timing	End of sweep
Input pin	ANj0 and ANj1 (2 pins), ANj0 to ANj3 (4 pins), ANj0 to ANj5 (6 pins), or ANj0 to ANj7 (8 pins) (j =non, 0, 2, 15)
Reading of result of A-D converter	Read A-D register corresponding to selected pin

A-D Converter**(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 1.24.5 shows the specifications of repeat sweep mode 0.

Table 1.24.5. 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	ANj0 and ANj1 (2 pins), ANj0 to ANj3 (4 pins), ANj0 to ANj5 (6 pins), or ANj0 to AN7 (8 pins) (j =non, 0, 2, 15)
Reading of result of A-D converter	Read A-D register corresponding to selected pin

(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 1.26.6 shows the specifications of repeat sweep mode 1.

Table 1.26.6. 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 : AN0 selected ANj0 → ANj1 → ANj0 → ANj2 → ANj0 → ANj3 etc. (j =non, 0, 2, 15)
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	ANj0 to ANj7 (j =non, 0, 2, 15)
With emphasis on the pin	ANj0 (1 pin), ANj0 and ANj1 (2 pins), ANj0 to ANj2 (3 pins), ANj0 to ANj3 (4 pins) (j =non, 0, 2, 15)
Reading of result of A-D converter	Read A-D register corresponding to selected pin

A-D Converter**(a) Resolution select function****8/10-bit mode select bit of A-D control register 1 (bit 3 at address 039716, 01D716)**

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.

(b) Sample and hold

Sample and hold are selected by setting bit 0 of the A-D control register 2 (address 039416, 01D416) to "1". When sample and hold are selected, the rate of conversion of each pin increases. As a result, a 28 \emptyset AD cycle is achieved with 8-bit resolution and 33 \emptyset 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 are to be used.

(c) Trigger select function

Can appoint start of conversion, by a combination of setting of trigger select bit (bit 5 at address 039616, 01D616) and external trigger request cause select bit (bit 5 and bit 6 at address 039416, 01D416), as follows.

Table 1.24.7. Trigger select function setting

	Trigger select bit="0"	Trigger select bit="1"		
		External trigger cause select bits		
		00	01	10
A-D0	Software trigger	ADTRG	Timer B2 OFCOI ^(Note)	Group 2 channel 1 interrupt
A-D1	Software trigger	ADTRG	Timer B2 OFCOI ^(Note)	Group 3 channel 1 interrupt

Timer B2 OFCOI : Timer B2 occurrence frequency counter overflow interrupt

Note :Valid in three-phase PWM mode.

(d) Two circuit same time start (software trigger)

Two A-D converters can start at the same time by setting simultaneous start bit (bit 7 of address 039416) to "1".

During the A-D circuit of either of A-D0 and A-D1 are operated, do not set "1" to the simultaneous start bit. Do not set to "1" when external trigger is selected. When using this bit, do not set A-D conversion start flag (bit 6 of address 039616, 01D616) to "1".

(e) Replace function of input pin

Setting "1" to A-D channel replace select bit of A-D0 control register 2 (ADS:bit 4 at address 039416) can replace channel of A-D0 and A-D1. A-D conversion reliability is confirmed by replacing channels.

When ADS bit is "1", a corresponding pin of A-D0 register i is selected by analog input port select bits of A-D1 control register 2 (bits 2 and 1 at address 01D416). In this case, A-D0 control register 0 and A-D1 control register 0 must be set to same value.

A-D Converter

Table 1.24.8. Setting of analog input port replace of A-D converter

Setting value	A-D channel replace select bit	1		
	Analog output port select bit	00	10	11
A-D conversion stored register				
A-D0 register 0	AN150	AN00	AN20	
A-D0 register 1	AN151	AN01	AN21	
A-D0 register 2	AN152	AN02	AN22	
A-D0 register 3	AN153	AN03	AN23	
A-D0 register 4	AN154	AN04	AN24	
A-D0 register 5	AN155	AN05	AN25	
A-D0 register 6	AN156	AN06	AN26	
A-D0 register 7	AN157	AN07	AN27	
A-D1 register 0		AN0		
A-D1 register 1		AN1		
A-D1 register 2		AN2		
A-D1 register 3		AN3		
A-D1 register 4		AN4		
A-D1 register 5		AN5		
A-D1 register 6		AN6		
A-D1 register 7		AN7		

(f) Extended analog input pins

In one-shot mode and repeat mode, the input via the extended analog input pins ANEX0 and ANEX1 can also be converted from analog to digital as AN0 and AN1 analog input signal respectively.

Set the related input peripheral function of the function select register B3 to disabled.

(g) External operation amp connection mode

In this mode, multiple external analog inputs via the extended analog input pins, ANEX0 and ANEX1, can be amplified together by just one operation amp and used as the input for A-D conversion.

When bit 6 and bit 7 of the A-D control register 1 (address 039716) is "11", input via AN0 to AN7 is output from ANEX0.

The input from ANEX1 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 ANEX0 and ANEX1 pins directly. Figure 1.24.8 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.

A-D Converter

Table 1.24.9. Setting of extended analog input pins

A-D0 control register 1		ANEX0 function	ANEX1 function
Bit 7	Bit 6		
0	0	Not used	Not used
0	1	P95 analog input	Not used
1	0	Not used	P96 analog input
1	1	Output to external ope-amp	Input from external ope-amp

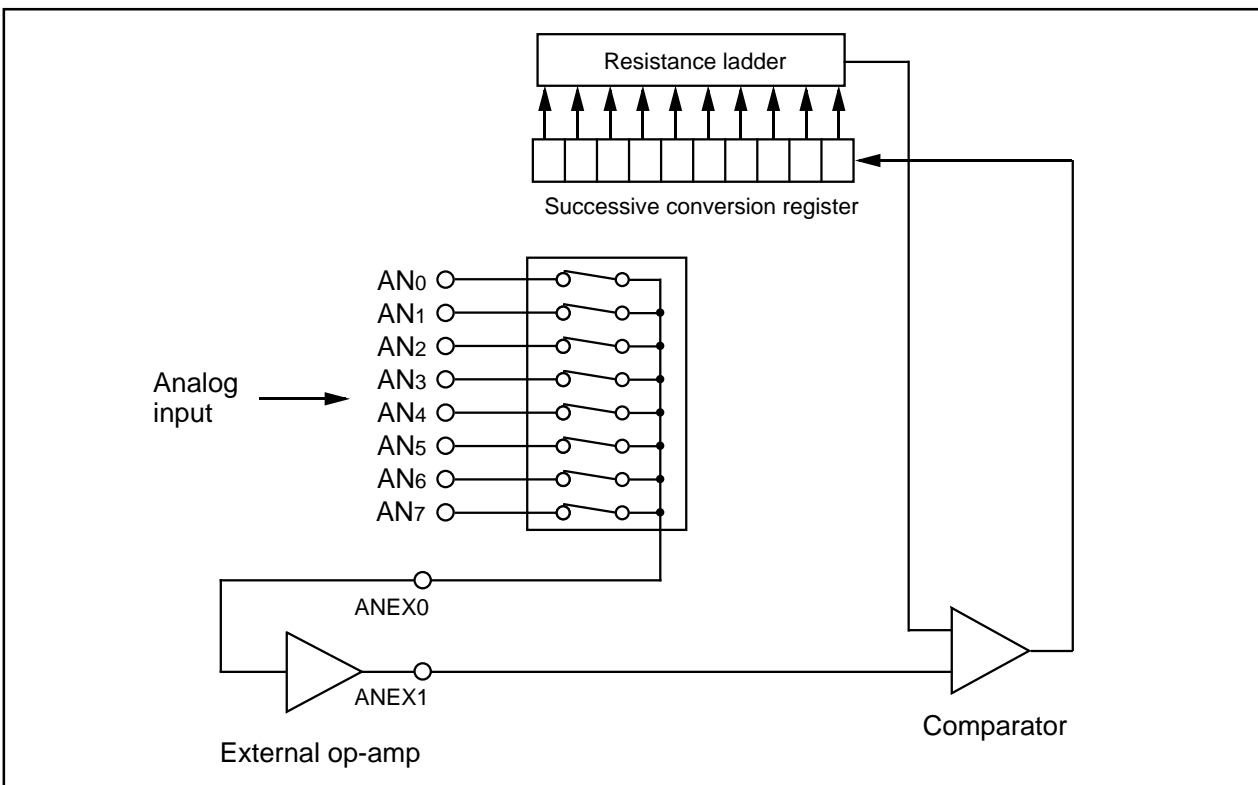


Figure 1.24.8. Example of external op-amp connection mode

(h) Power consumption reduction function**VREF connect bit (bit 5 at addresses 039716, 01D716)**

The VREF connect bit (bit 5 at address 039716, 01D716) 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 connecting VREF.

Do not write A-D conversion start flag and VREF connect bit to "1" at the same time. Do not clear VREF connect bit to "0" during A-D conversion. This VREF is without reference to D-A converter's VREF.

A-D Converter

Precaution

After A-D conversion is complete, if the CPU reads the A-D register at the same time as the A-D conversion result is being saved to A-D register, wrong A-D conversion value is saved into the A-D register. This happens when the internal CPU clock is selected from divided main clock or sub-clock.

• When using the one-shot or single sweep mode

Confirm that A-D conversion is complete before reading the A-D register.

(Note: When A-D conversion interrupt request bit is set, it shows that A-D conversion is completed.)

• When using the repeat mode or repeat sweep mode 0 or 1

Use the undivided main clock as the internal CPU clock.

D-A Conversion

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 A3 to I/O port, the related input peripheral function of the function select register B3 to disabled and the direction register to input mode. Do not set the target port to pulled-up when D-A output is enabled.

Output analog voltage (V) is determined by a set value (n : decimal) in the D-A register.

$$V = V_{REF} \times n / 256 \quad (n = 0 \text{ to } 255)$$

V_{REF} : reference voltage (This is unrelated to bit 5 of A-D control register 1 (addresses 039716, 01D716))

Table 1.25.1 lists the performance of the D-A converter. Figure 1.25.1 shows the block diagram of the D-A converter. Figure 1.25.2 shows the D-A control register. Figure 1.25.3 shows the D-A converter equivalent circuit.

When the D-A converter is not used, set the D-A register to "00" and D-A output enable bit to "0".

Table 1.25.1. Performance of D-A converter

Item	Performance
Conversion method	R-2R method
Resolution	8 bits
Analog output pin	2 channels

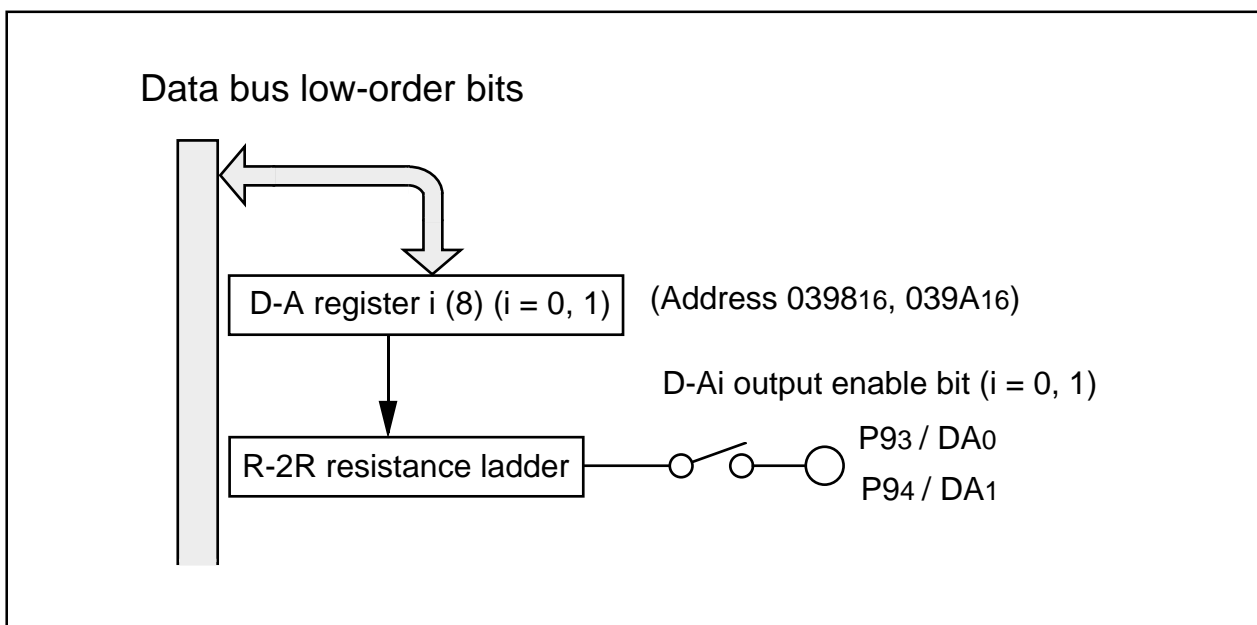


Figure 1.25.1. Block diagram of D-A converter

D-A Conversion

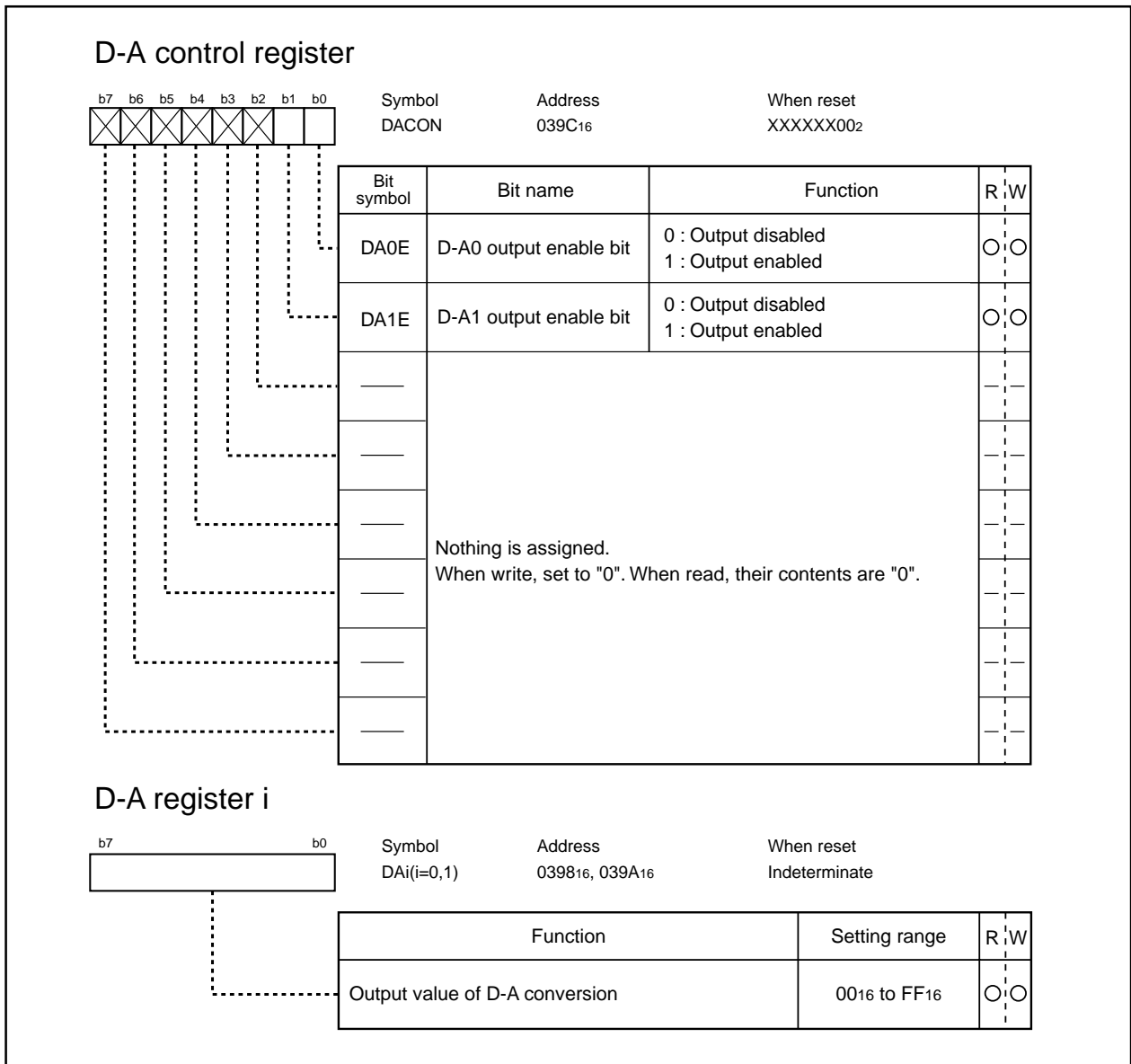


Figure 1.25.2. D-A control register

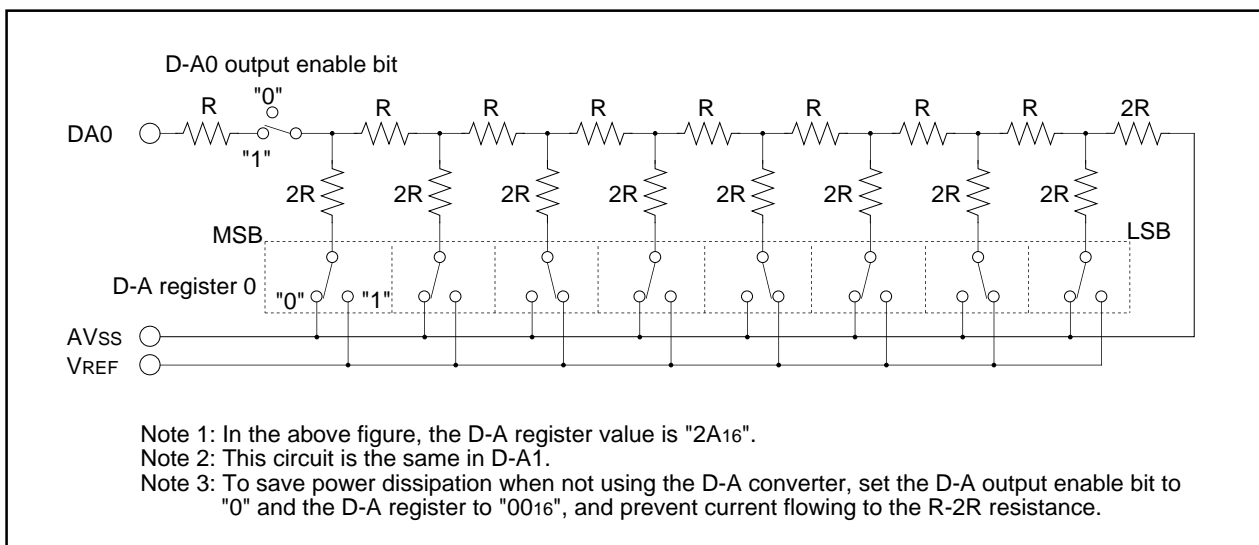


Figure 1.25.3. D-A converter equivalent circuit

CRC

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 1.26.1 shows the block diagram of the CRC circuit. Figure 1.26.2 shows the CRC-related registers. Figure 1.26.3 shows the CRC example.

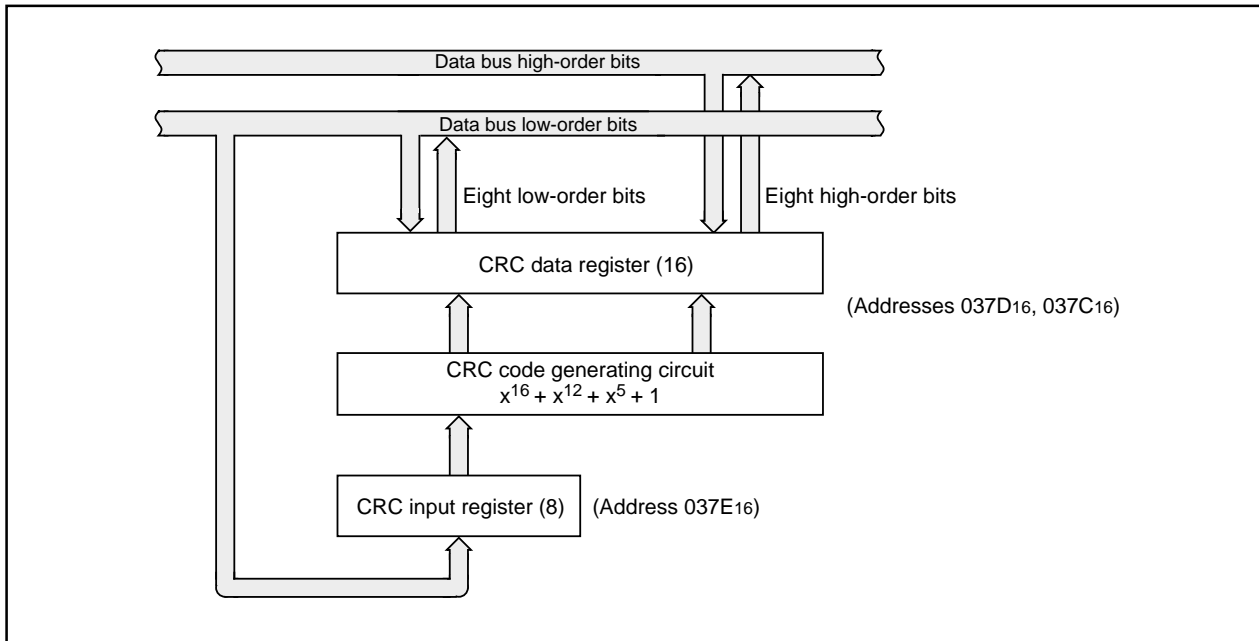


Figure 1.26.1. Block diagram of CRC circuit

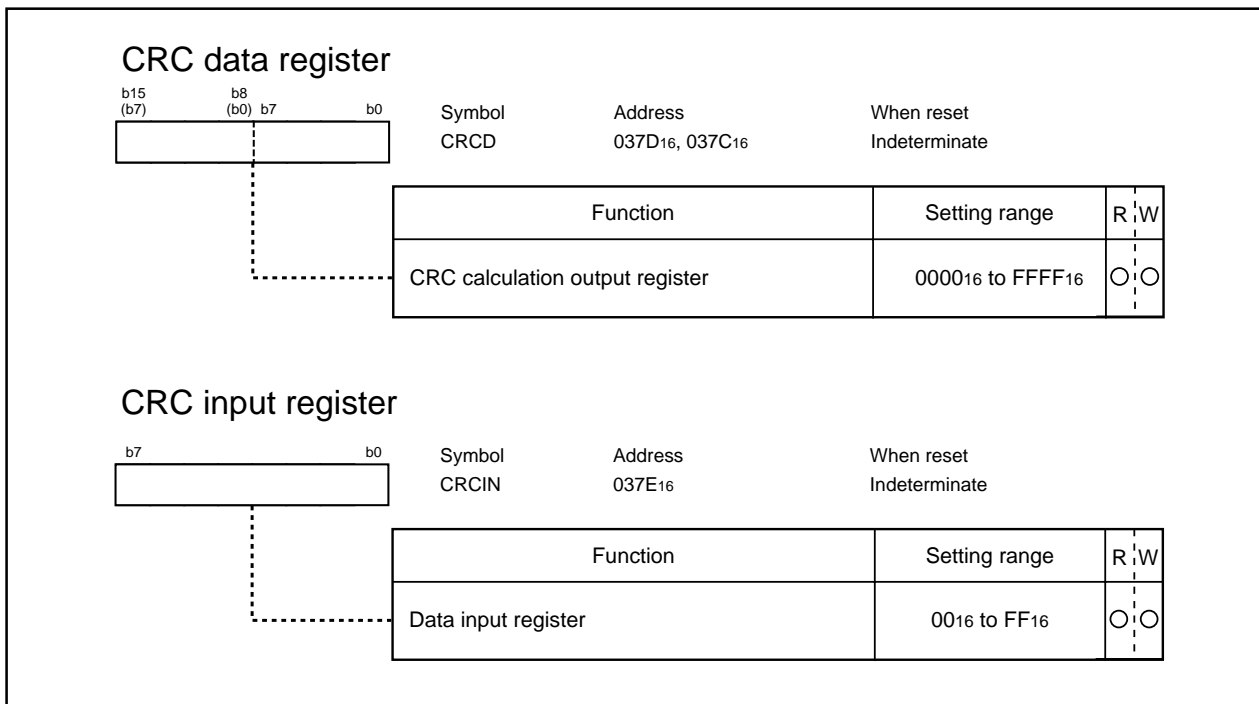


Figure 1.26.2. CRC-related registers

CRC

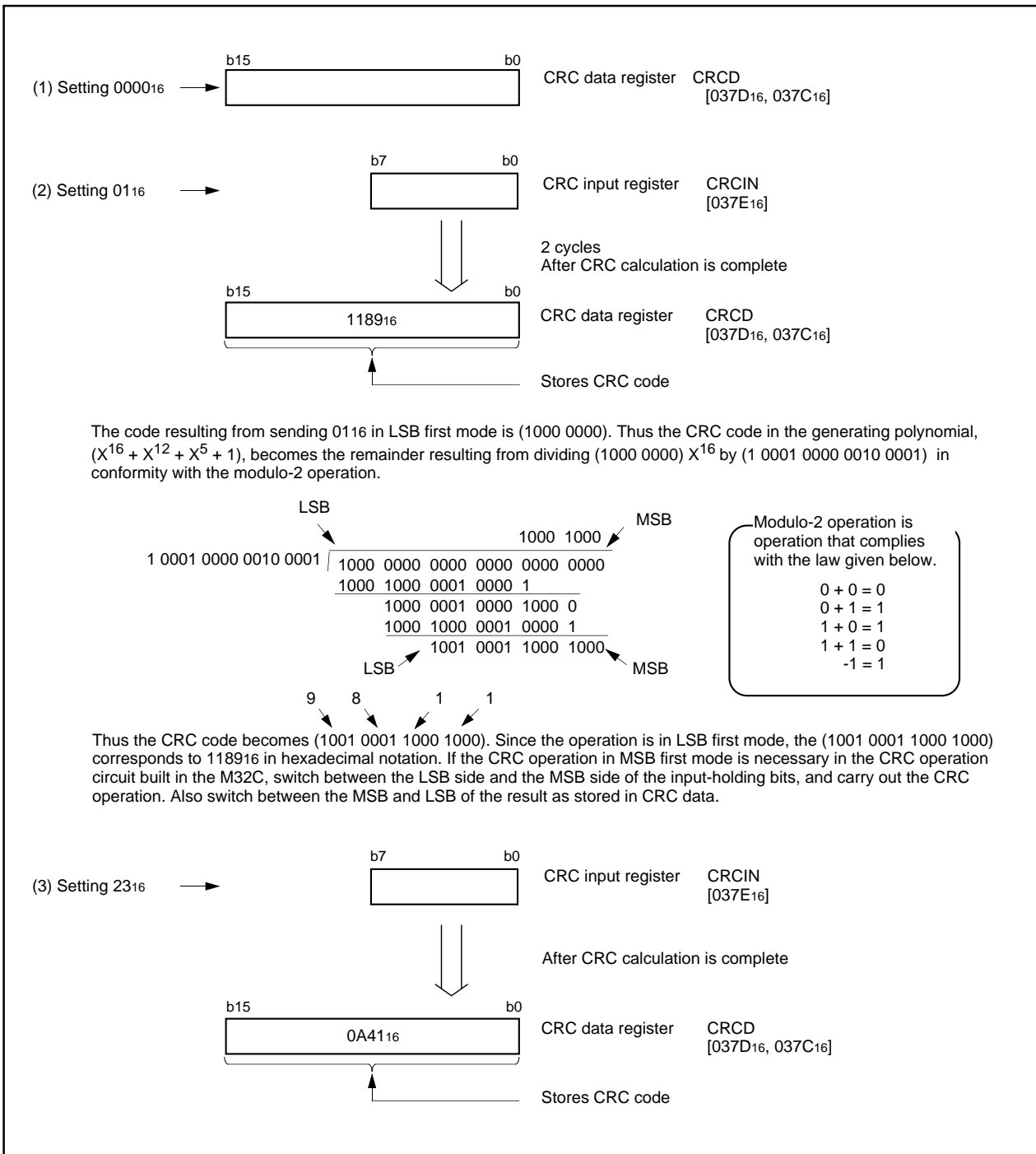


Figure 1.26.3. CRC example

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 1.27.1 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.

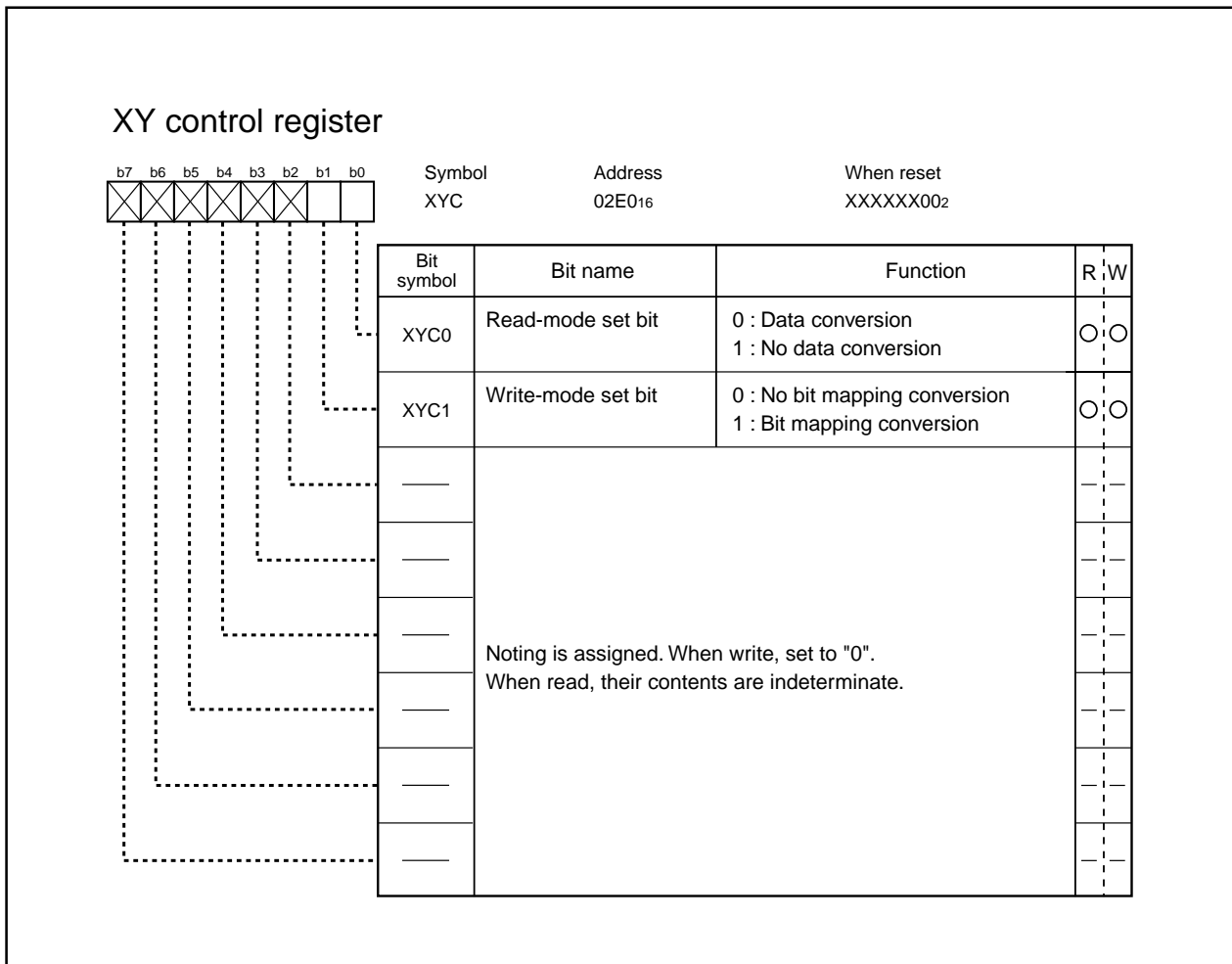


Figure 1.27.1. XY control register

The reading of the Yi register is controlled by the read-mode set bit (bit 0 at address 02E016).

When the read-mode set bit (bit 0 at address 02E016) 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 1.27.2 shows the conversion table when the read mode set bit = "0". Figure 1.27.3 shows the X-Y conversion example.

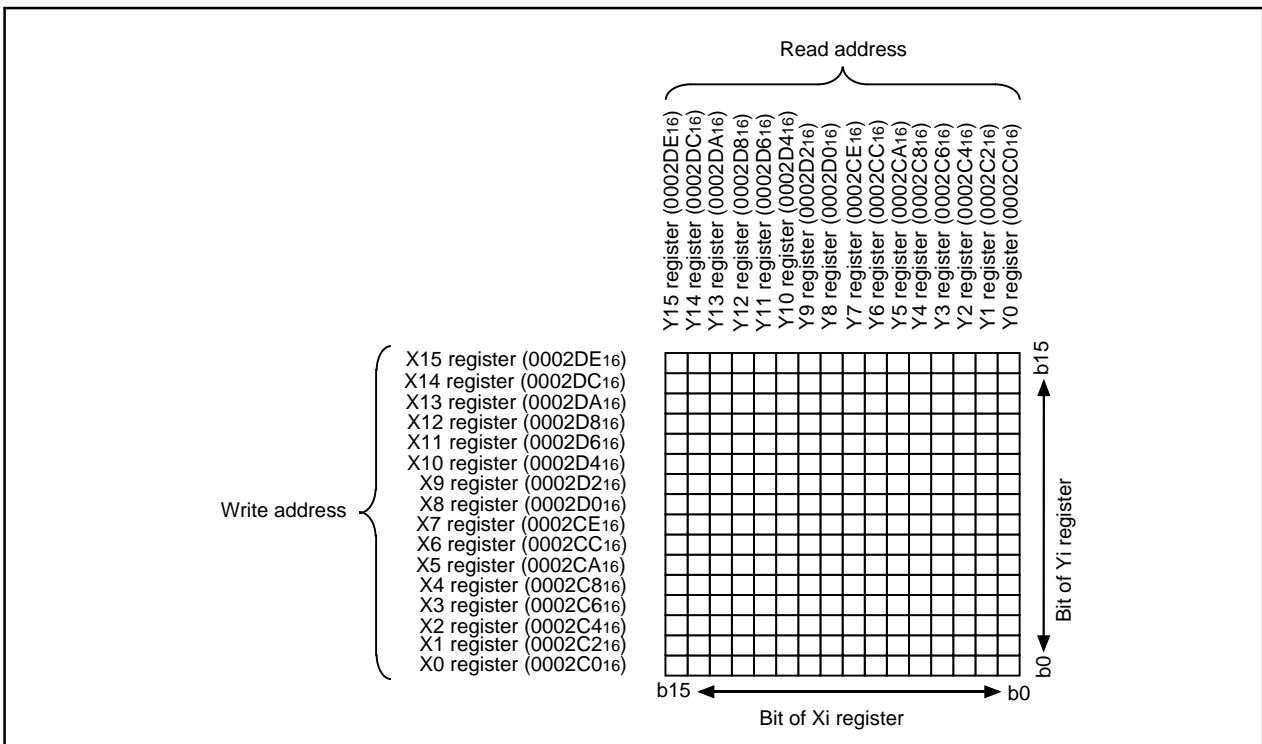


Figure 1.27.2. Conversion table when the read mode set bit = "0"

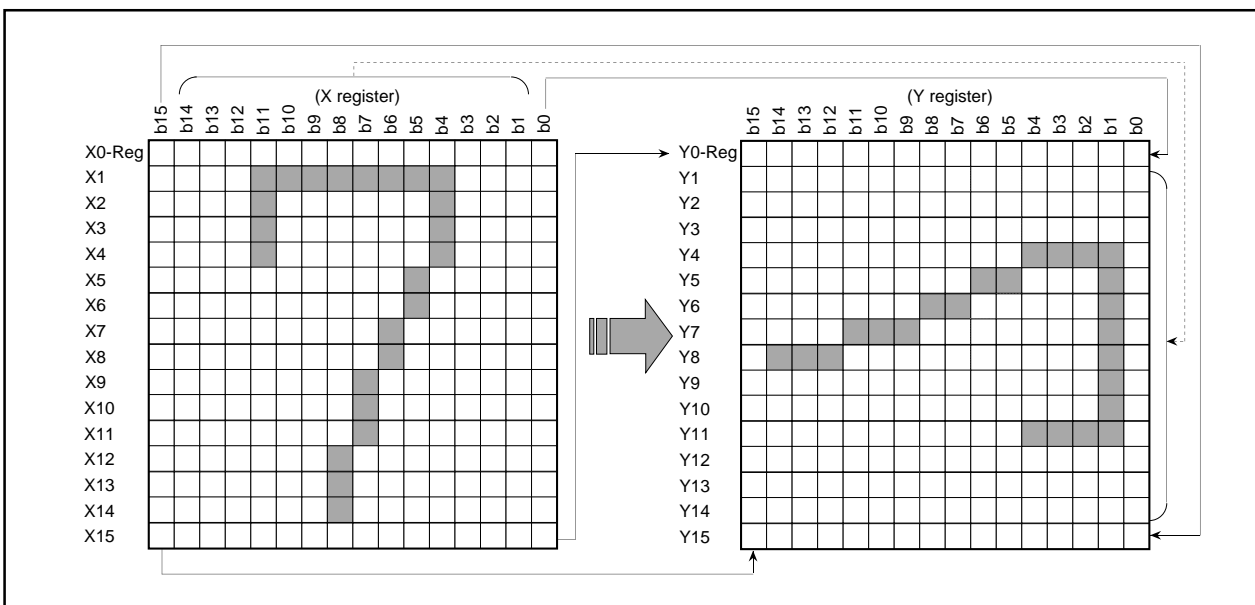


Figure 1.27.3. X-Y conversion example

When the read-mode set bit (bit 0 at address 02E016) is "1", you can read the value written to the Xi register by reading the Yi register. Figure 1.27.4 shows the conversion table when the read mode set bit = "1".

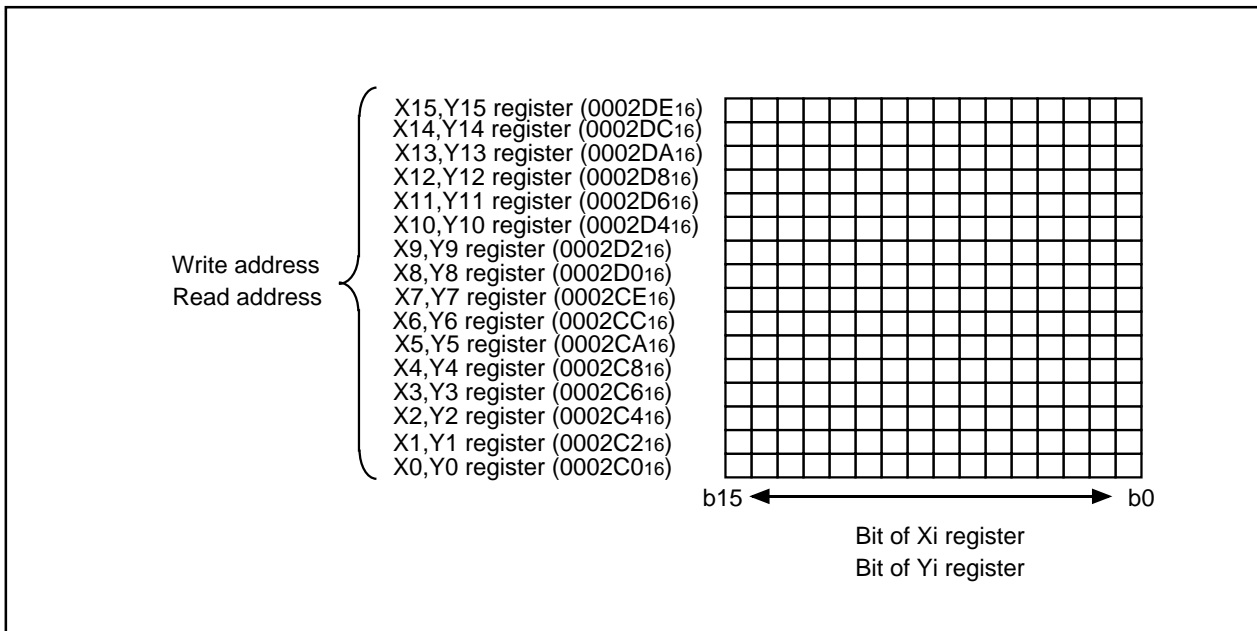


Figure 1.27.4. Conversion table when the read mode set bit = "1"

The value written to the Xi register is controlled by the write mode set bit (bit 1 at address 02E016). When the write mode set bit (bit 1 at address 02E016) 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 02E016) 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 1.27.5 shows the conversion table when the write mode set bit = "1".

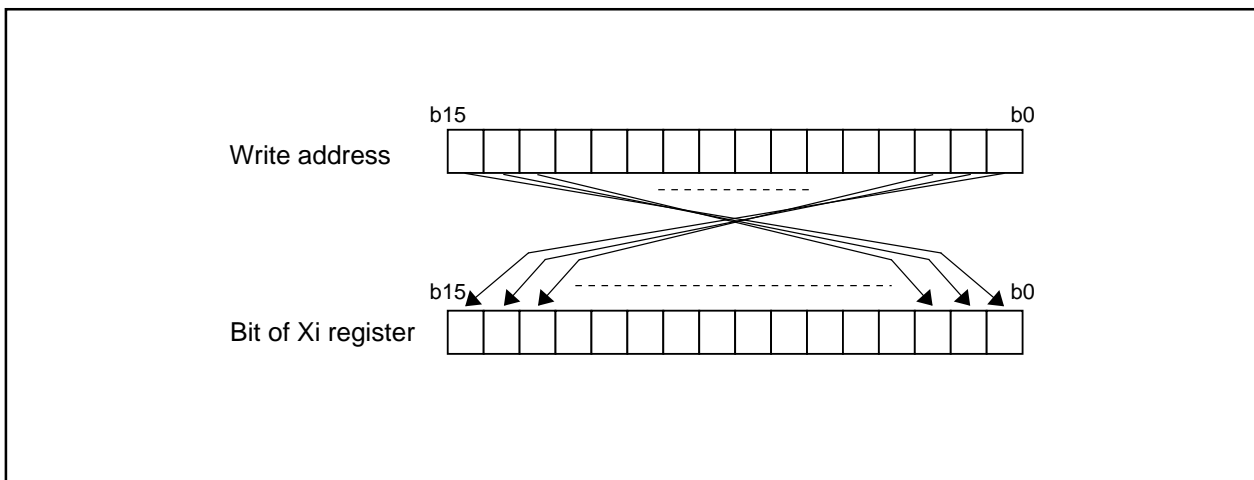


Figure 1.27.5. Conversion table when the write mode set bit = "1"

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 1.28.1 shows the functions of the DRAM controller.

Table 1.28.1. DRAM Controller Functions

DRAM space	512KB, 1MB, 2MB, 4MB, 8MB
Bus control	2CAS/1W
Refresh	CAS before 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 0040₁₆) to specify the DRAM size. Figure 1.28.1 shows the DRAM control register.

The DRAM controller cannot be used in external memory mode 3 (bits 1 and 2 at address 0005₁₆ are "112"). 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 0004₁₆). Set wait time between after DRAM power ON and before memory processing, and processing necessary for dummy cycle to refresh DRAM by software.

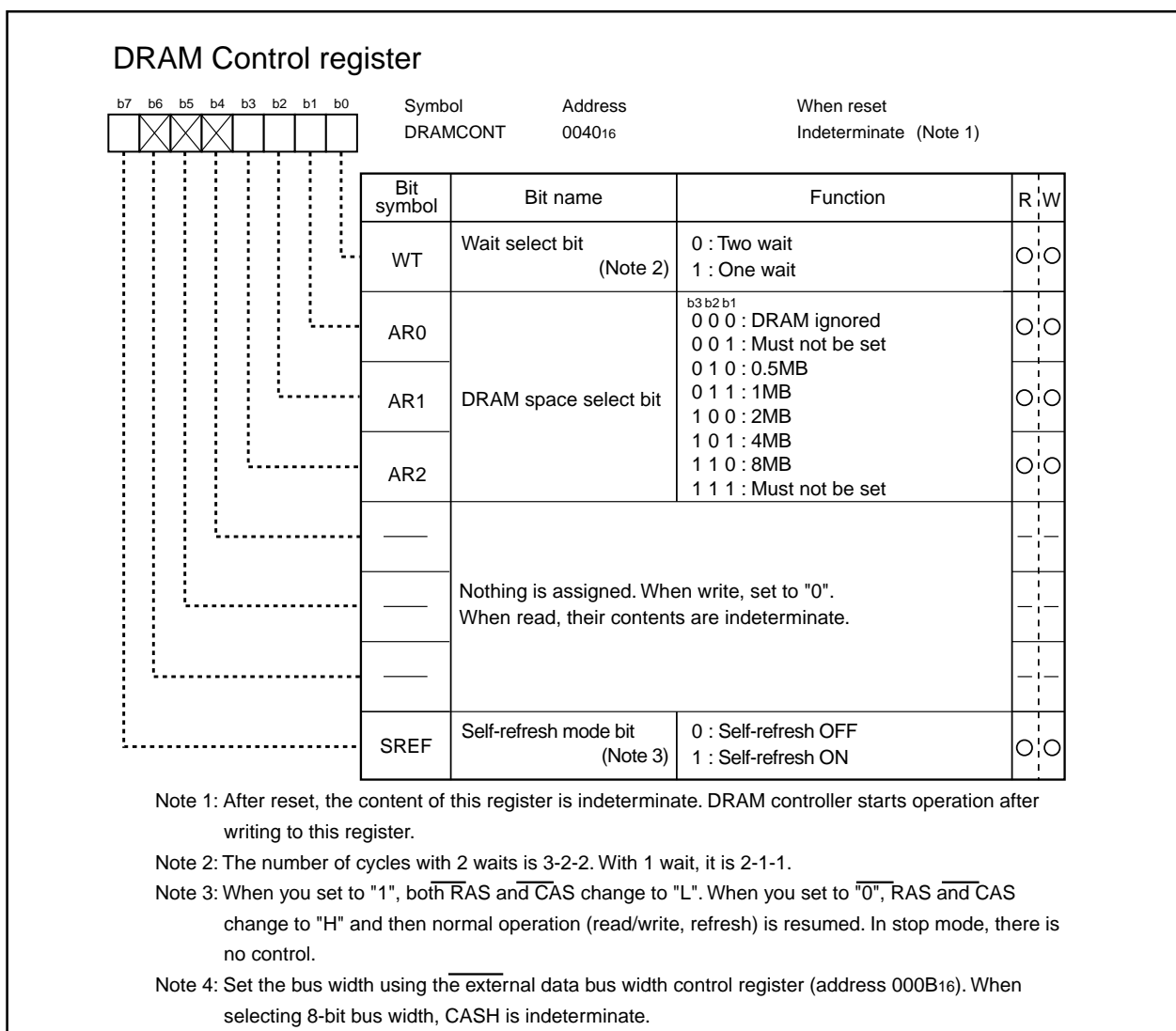


Figure 1.28.1. DRAM control register

• **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 1.28.2 shows the output format for multiplexed addresses.

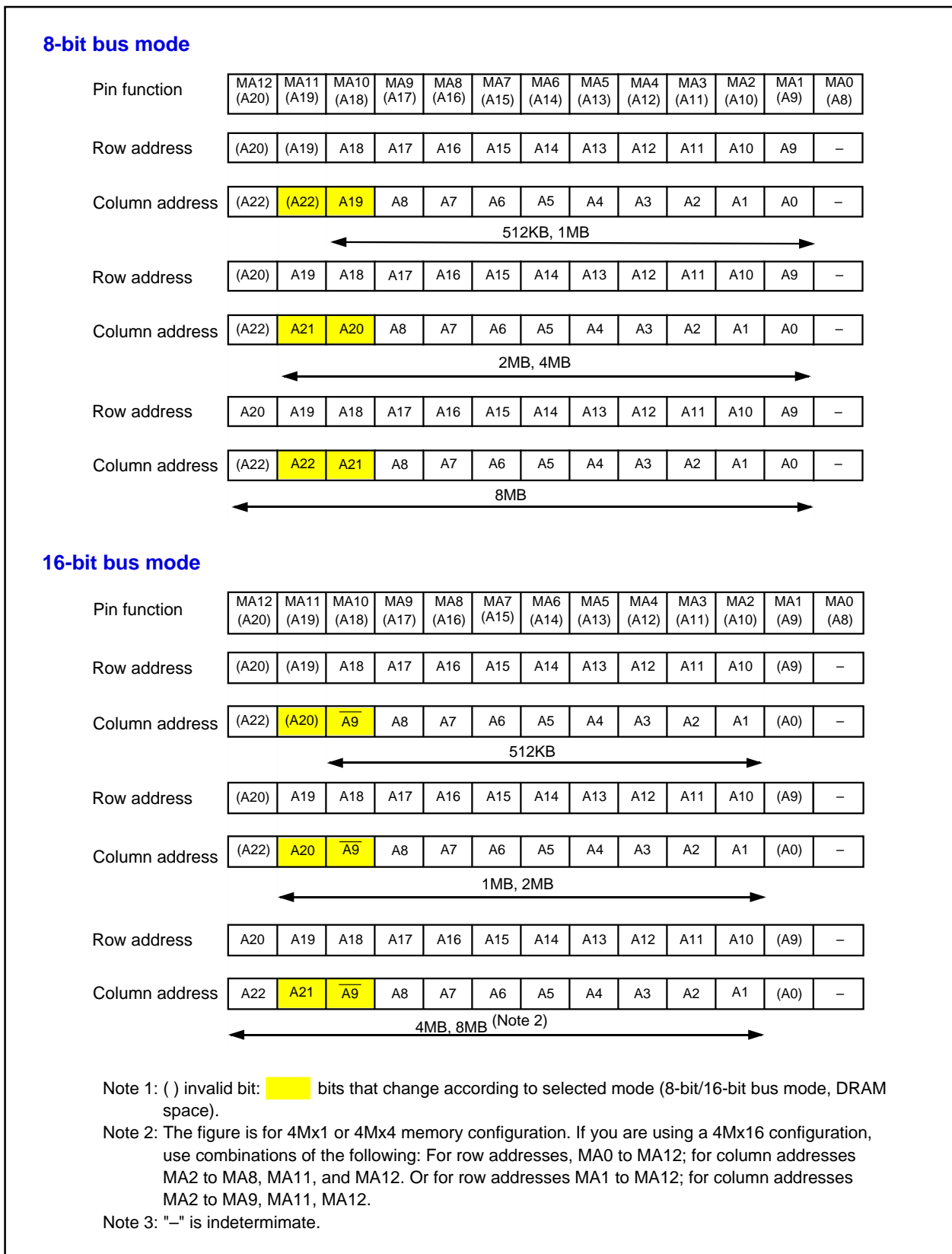


Figure 1.28.2. Output format for multiplexed addresses

• 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 0041₁₆). The refresh signal is not output in HOLD state. Figure 1.28.3 shows the DRAM refresh interval set register.

Use the following formula to determine the value to set in the refresh interval set register.

$$\text{Refresh interval set register value (0 to 255)} = \text{refresh interval time} / (\text{BCLK frequency} \times 32) - 1$$

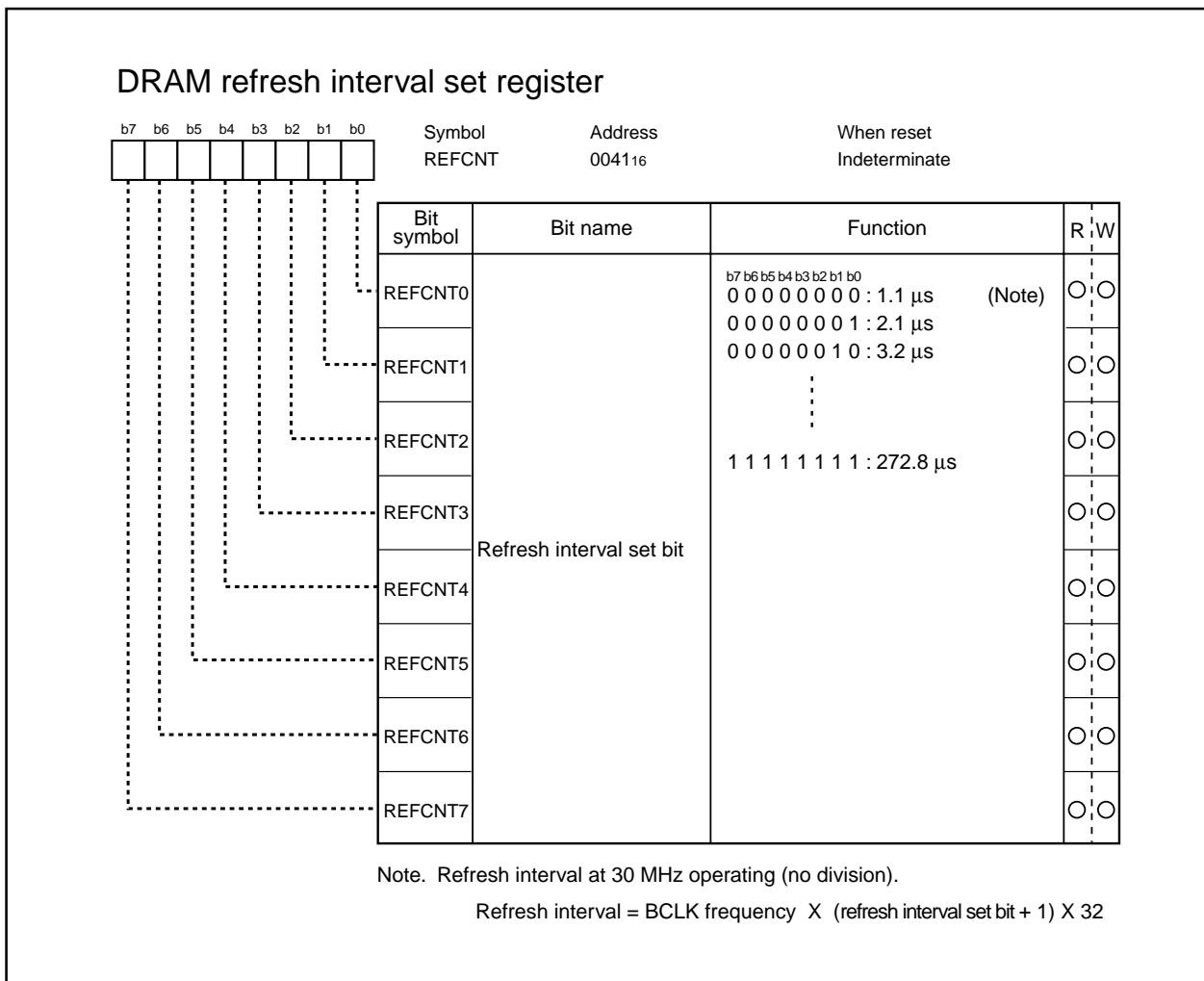


Figure 1.28.3. DRAM refresh interval set register

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.

DRAM Controller

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 1.28.4 to 1.28.6 show the bus timing during DRAM access.

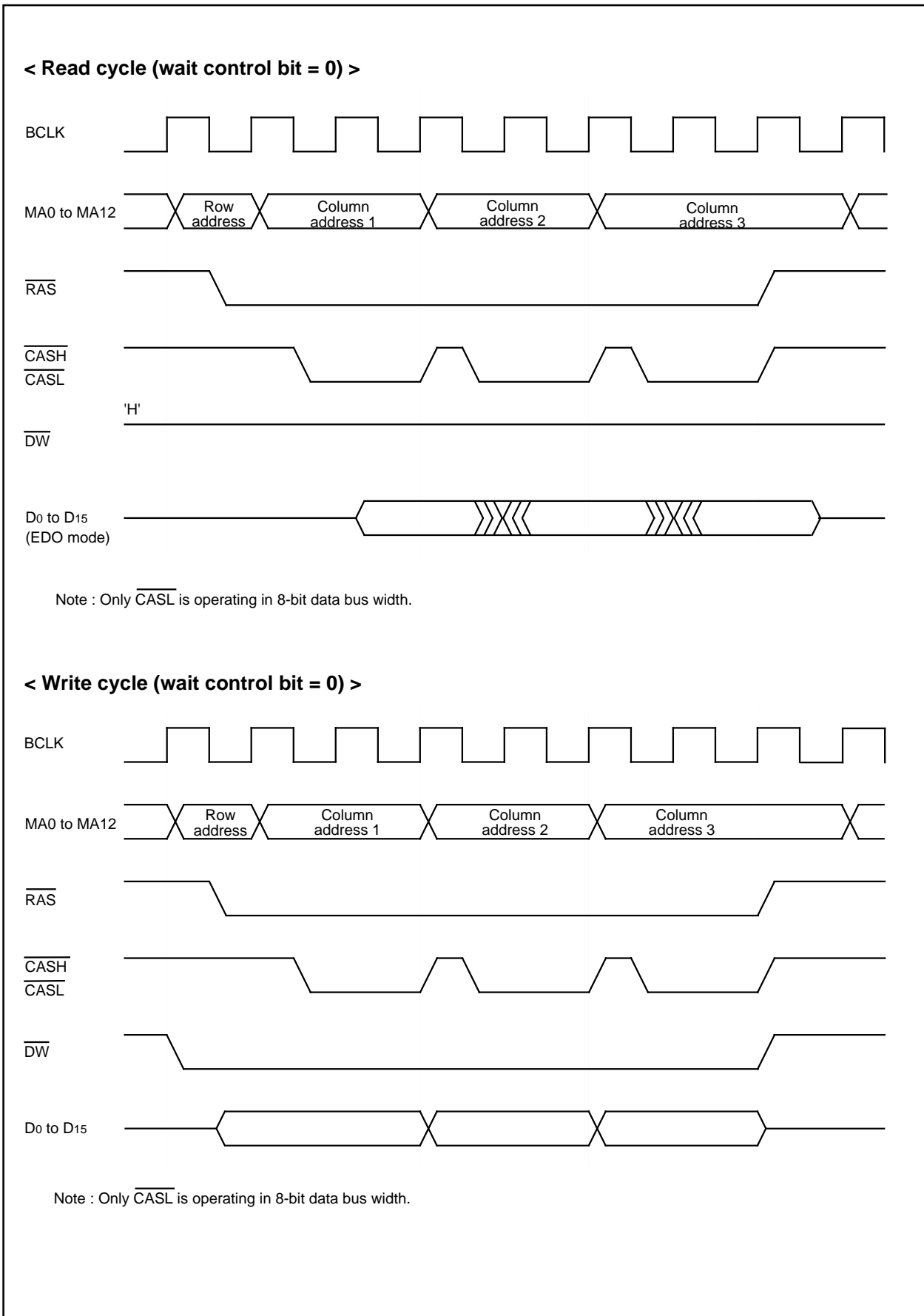


Figure 1.28.4. The bus timing during DRAM access (1)

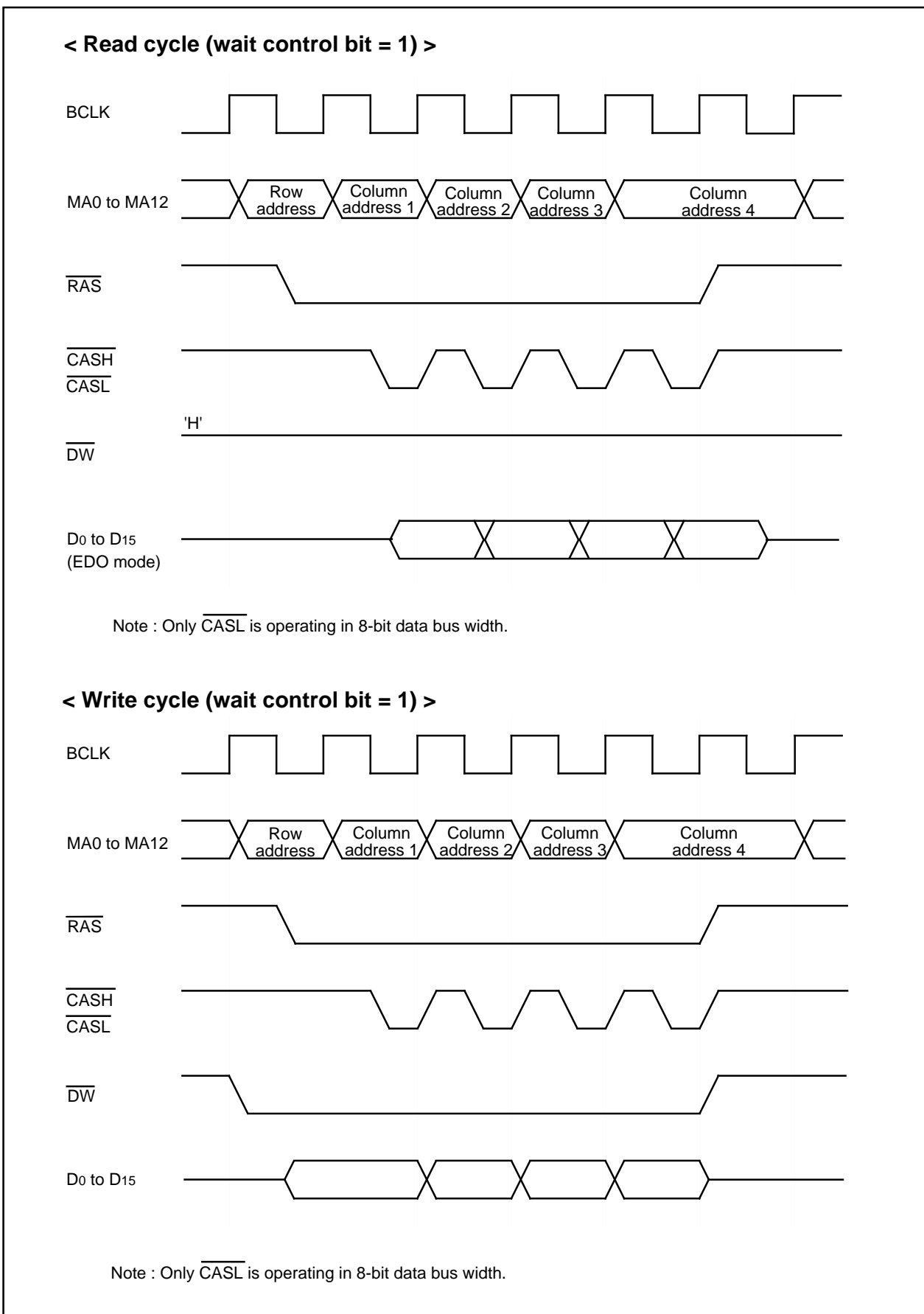


Figure 1.28.5. The bus timing during DRAM access (2)

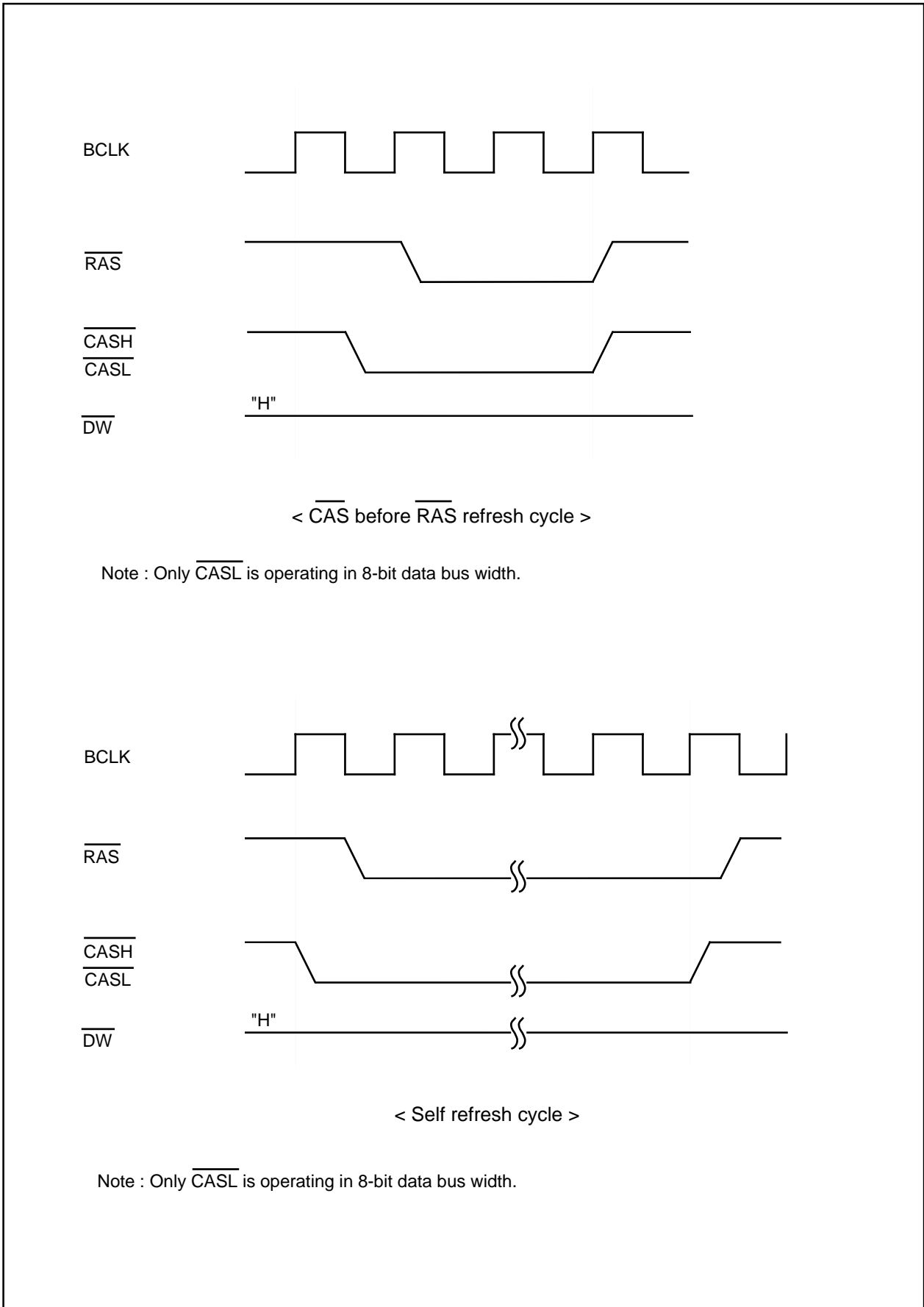


Figure 1.28.6. The bus timing during DRAM access (3)

Programmable I/O Ports

There are 123 programmable I/O ports in 144-pin version: P0 to P15 (excluding P85). There are 87 programmable I/O ports in 100-pin version: P0 to P10 (excluding P85). 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. P85 is an input-only port and has no built-in pull-up resistance.

Figures 1.29.1 to 1.29.4 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 A3 of each pin to I/O port, and set the direction registers to input mode.

See the descriptions of the respective functions for how to set up the built-in peripheral devices.

(1) Direction registers

Figures 1.29.5 shows 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 A0 to A22, $\overline{A23}$, D0 to D15, MA0 to MA12, $\overline{CS0}$ to $\overline{CS3}$, $\overline{WRL/WR/CASL}$, $\overline{WRH/BHE/CASH}$, $\overline{RD/DW}$, BCLK/ALE/CLKOUT, $\overline{HLDA/ALE}$, \overline{HOLD} , ALE/ \overline{RAS} , and \overline{RDY} are not changed.

Note: There is no direction register bit for P85.

(2) Port registers

Figure 1.29.6 shows 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 a port register corresponds one for one to each I/O pin.

In memory expansion and microprocessor mode, the contents of corresponding port register of pins A0 to A22, $\overline{A23}$, D0 to D15, MA0 to MA12, $\overline{CS0}$ to $\overline{CS3}$, $\overline{WRL/WR/CASL}$, $\overline{WRH/BHE/CASH}$, $\overline{RD/DW}$, BCLK/ALE/CLKOUT, $\overline{HLDA/ALE}$, \overline{HOLD} , ALE/ \overline{RAS} , and \overline{RDY} are not changed.

(3) Function select register A

Figures 1.29.7 to 1.29.11 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 1.29.12 and 1.29.13 show the function select registers B.

This register selects the first 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 1.29.14 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 $\overline{KI0}$ to $\overline{KI3}$. 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 1.29.15 to 1.29.17 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.

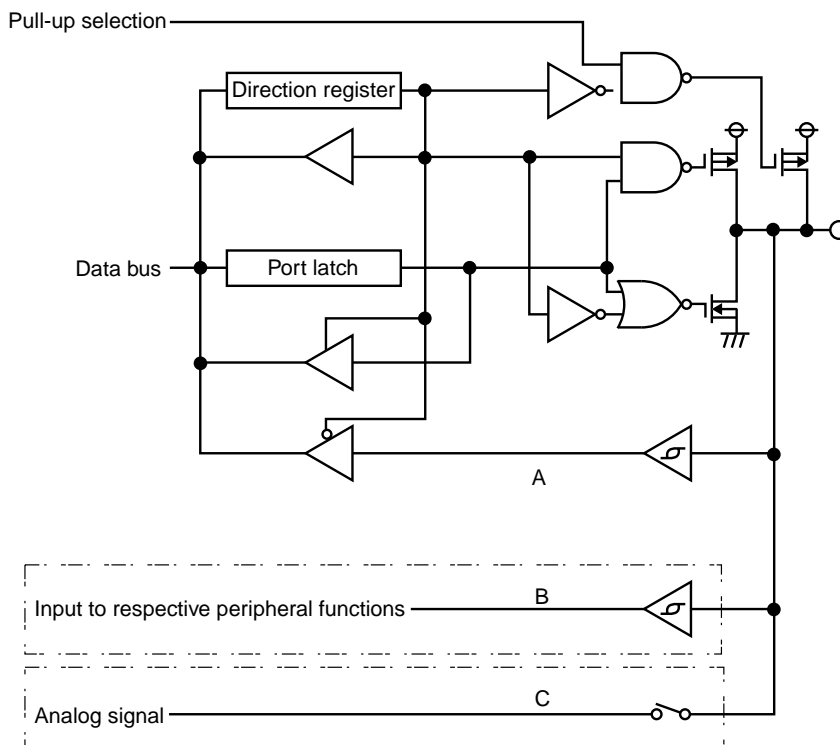
(7) Port control register

Figure 1.29.18 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 CMOS port's Pch is kept always turned off so that the port P1 cannot be a complete open drain. 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.

Programmable I/O ports



Port \ Option	(A) Hysteresis presence	Circuit (B) Input to respective peripheral functions	Circuit (C) Analog I/F
P00 to P07 P20 to P27	—	—	○
P30 to P37 P40 to P47 P50 to P52 P54	—	—	—
P55	—	○	—
P56	—	—	—
P57	—	○	—
P83, P84	○	○	—
P86	—	—	—
P87	—	○	○
P100 to P103	—	—	○
P104 to P107	○	○	○
P114 P144 to P146	—	—	—
P152, P153 P156, P157	—	○	○

(Note)

○ : Present, — : Not present

Note: These ports exist in 144-pin version.

Figure 1.29.1. Programmable I/O ports (1)

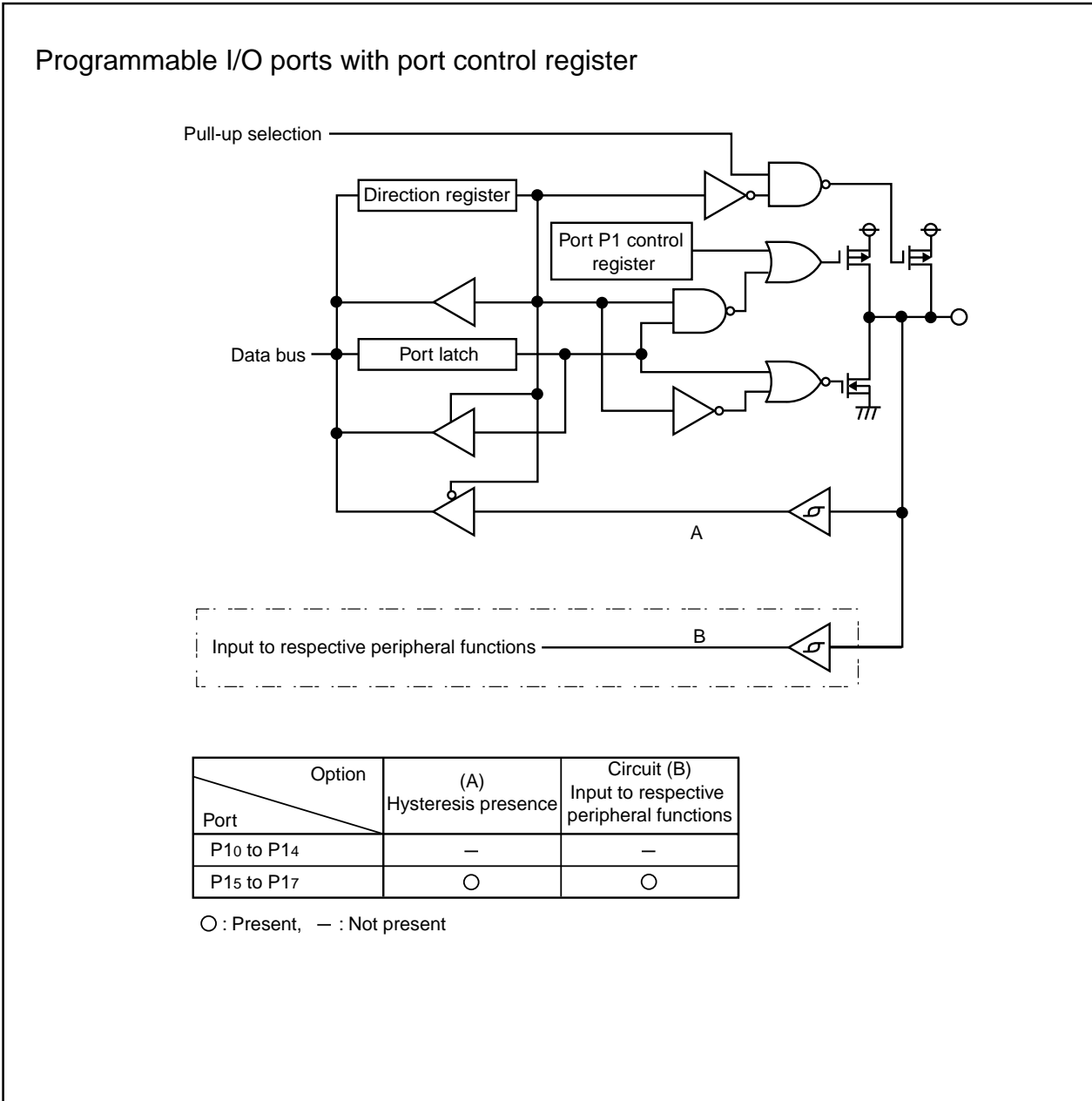
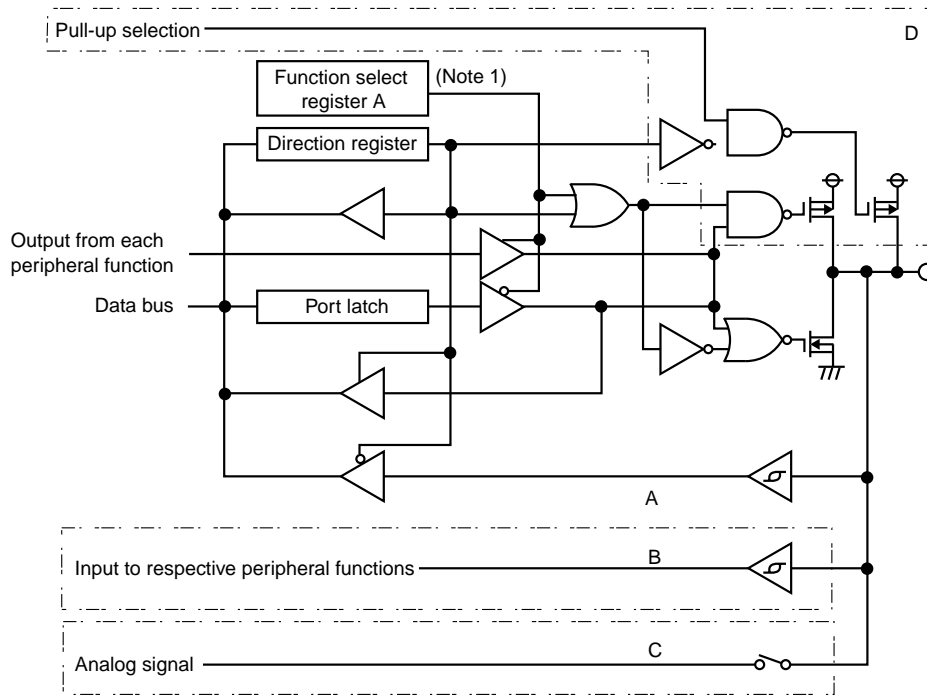


Figure 1.29.2. Programmable I/O ports (2)

Programmable I/O ports with function select register



Option	(A) Hysteresis presence	Circuit (B) Input to respective peripheral functions	Circuit (C) Analog I/F	Circuit (D)
P53 (Note 1)	—	—	—	○
P60, P61 P63 to P65, P67	—	○	—	○
P70, P71 (Note 2)	—	○	—	—
P72 to P77 P80, P81	—	○	—	○
P82	○	○	—	○
P90 to P92	—	○	—	○
P93 to P96	—	○	○	○
P97	—	○	—	○
P110	—	—	—	○
P111, P112	—	○	—	○
P113 P120	—	—	—	○
P121, P122	—	—	—	○
P123 to P127 P130 to P134	—	—	—	○
P135, P136	—	○	—	○
P137 P140, P141	—	—	—	○
P142, P143	—	○	—	○
P150, P151 P154, P155	—	○	○	○

(Note 3)

○ : Present, — : Not present
 Note 1: P53 is clock output select bit for BCLK.
 Note 2: P70 and P71 are N-channel open drain output.
 Note 3: These ports exist in 144-pin version.

Figure 1.29.3. Programmable I/O ports (3)

Programmable I/O Port

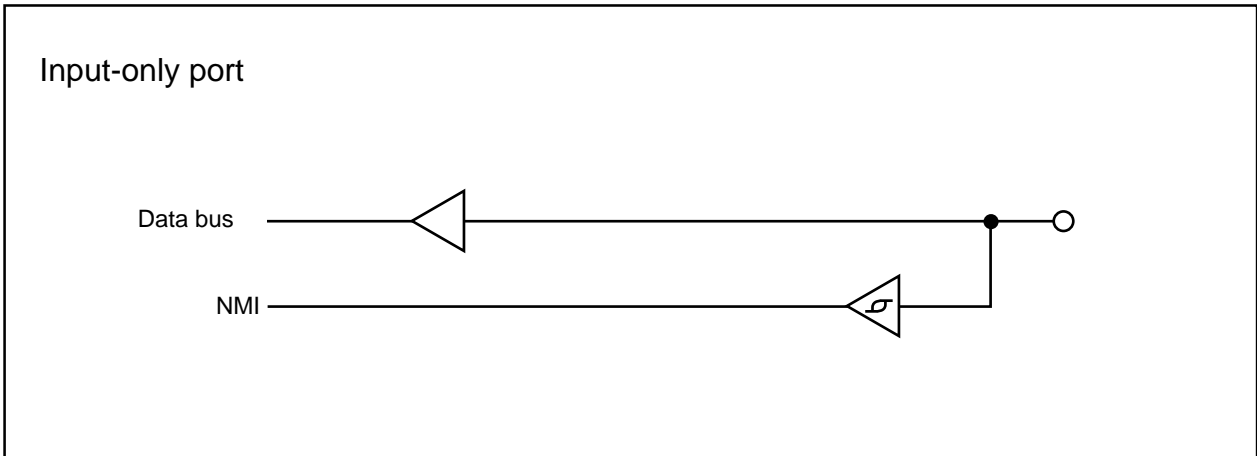


Figure 1.29.4. Programmable I/O ports (4)

Port Pi direction register (Note 1, 2, 3)

Bit	Symbol	Bit name	Function	R	W
b7					
b6					
b5					
b4					
b3					
b2					
b1					
b0					

Symbol	Address	When reset
PDi(i=0 to 5)	03E2 ₁₆ , 03E3 ₁₆ , 03E6 ₁₆ , 03E7 ₁₆ , 03EA ₁₆ , 03EB ₁₆	00 ₁₆
PDi(i=6 to 11)	03C2 ₁₆ , 03C3 ₁₆ , 03C6 ₁₆ , 03C7 ₁₆ , 03CA ₁₆ , 03CB ₁₆	00 ₁₆
PDi(i=12 to 15)	03CE ₁₆ , 03CF ₁₆ , 03D2 ₁₆ , 03D3 ₁₆	00 ₁₆

Bit Symbol	Bit name	Function	R	W
PDi_0	Port Pi0 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	○	○
PDi_1	Port Pi1 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	○	○
PDi_2	Port Pi2 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	○	○
PDi_3	Port Pi3 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	○	○
PDi_4	Port Pi4 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port)	○	○
PDi_5	Port Pi5 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port) (Note 4)	○	○
PDi_6	Port Pi6 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port) (Note 4)	○	○
PDi_7	Port Pi7 direction register	0 : Input mode (Functions as an input port) 1 : Output mode (Functions as an output port) (Note 4)	○	○

Note 1: Set bit 2 of protect register (address 000A₁₆) 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 A₀ to A₂₂, A₂₃, D₀ to D₁₅, MA₀ to MA₁₂, CS₀ to CS₃, WRL/WR/CASL, WRH/BHE/CASH, RD/DW, BCLK/ALE/CLKout, HLDA/ALE, HOLD, ALE/RAS, and RDY are not changed.

Note 3: Port 11 to 15 registers exist in 144-pin version.

Note 4: Nothing is assigned in bit5 of Port P8 direction register, bit7 to bit5 of port P11 direction register and bit7 of port P14 direction register.
When write, set to "0". When read, its content is indeterminate.

Figure 1.29.5. Direction register

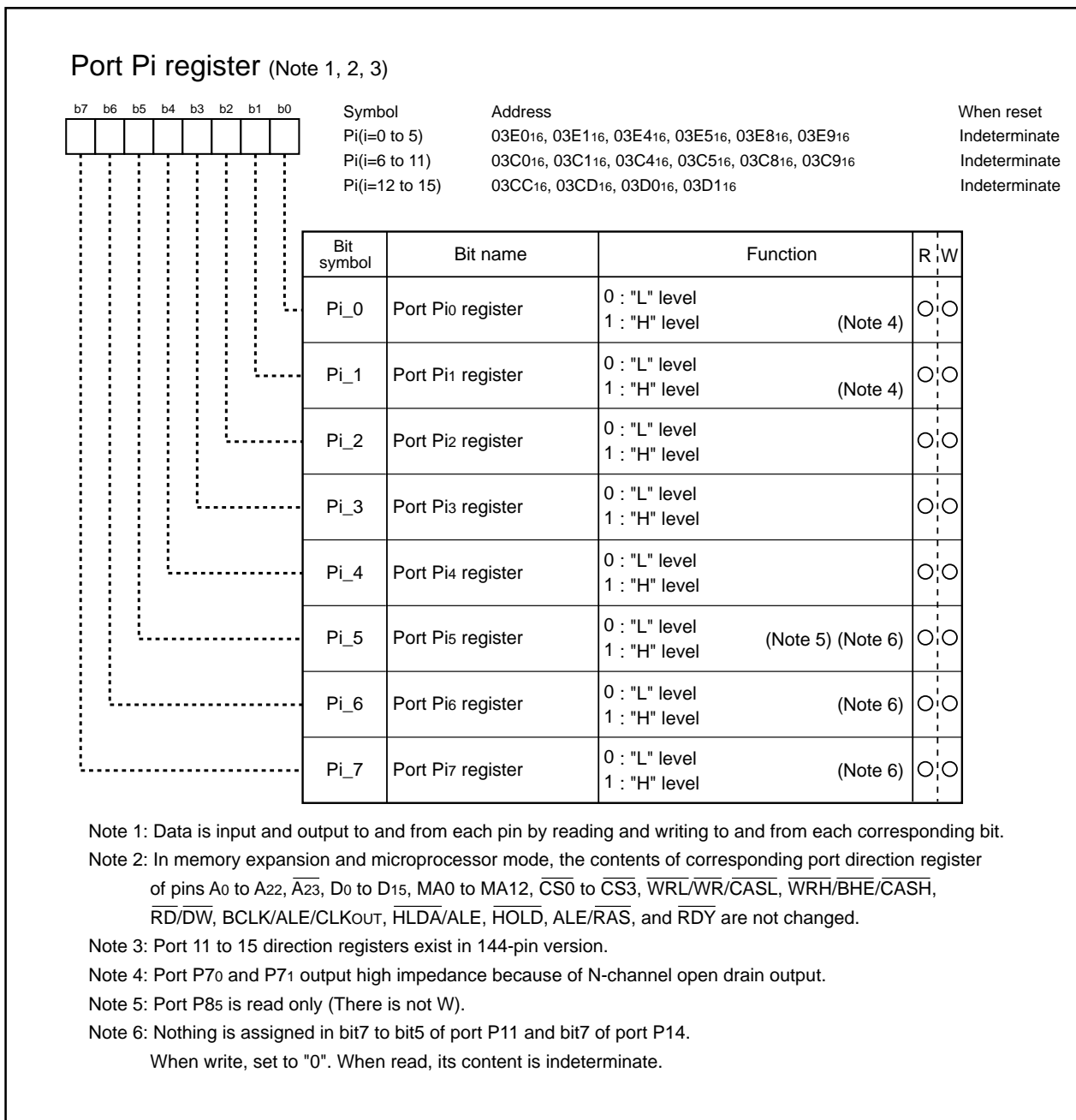


Figure 1.29.6. Port register

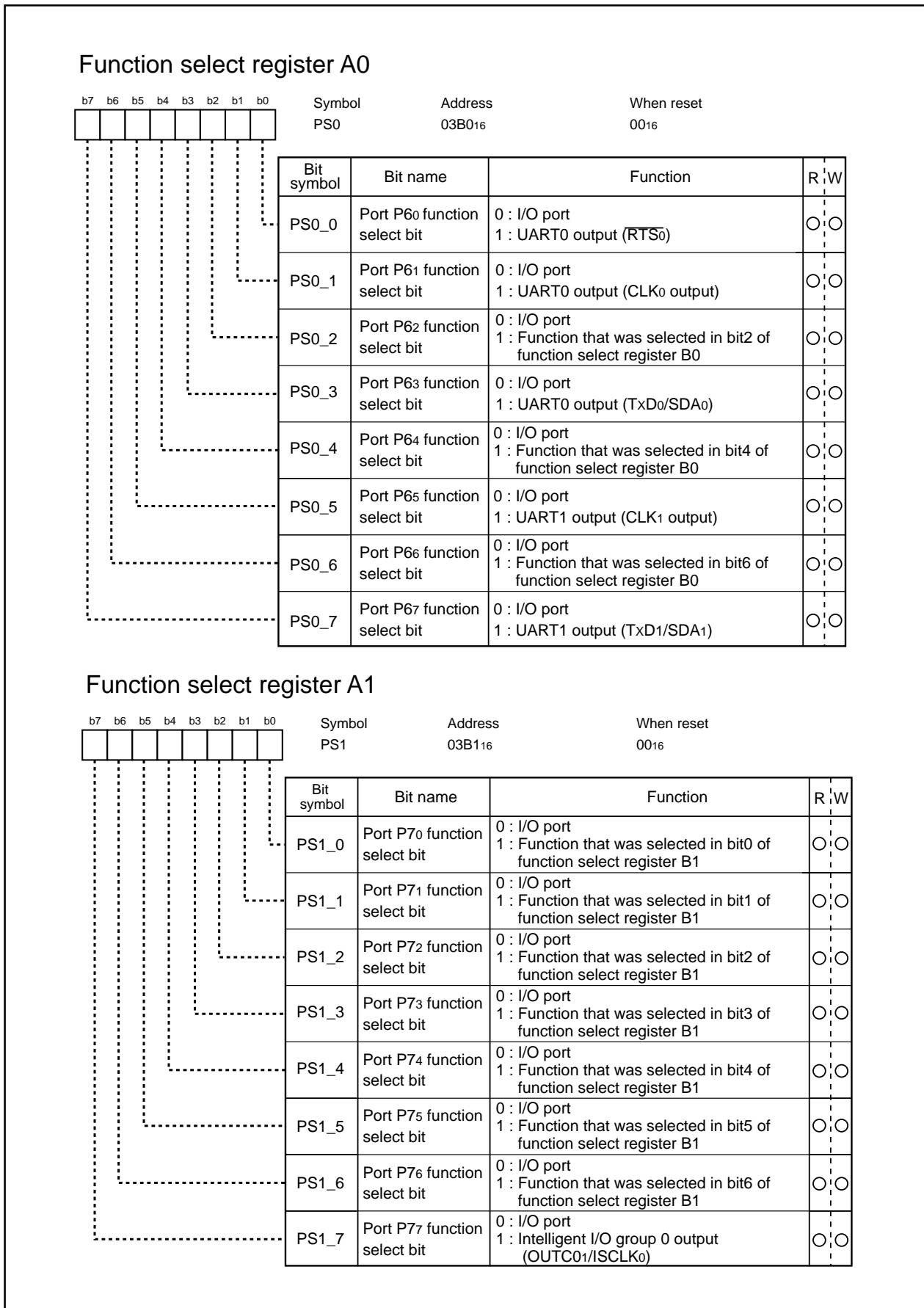


Figure 1.29.7. Function select register A (1)

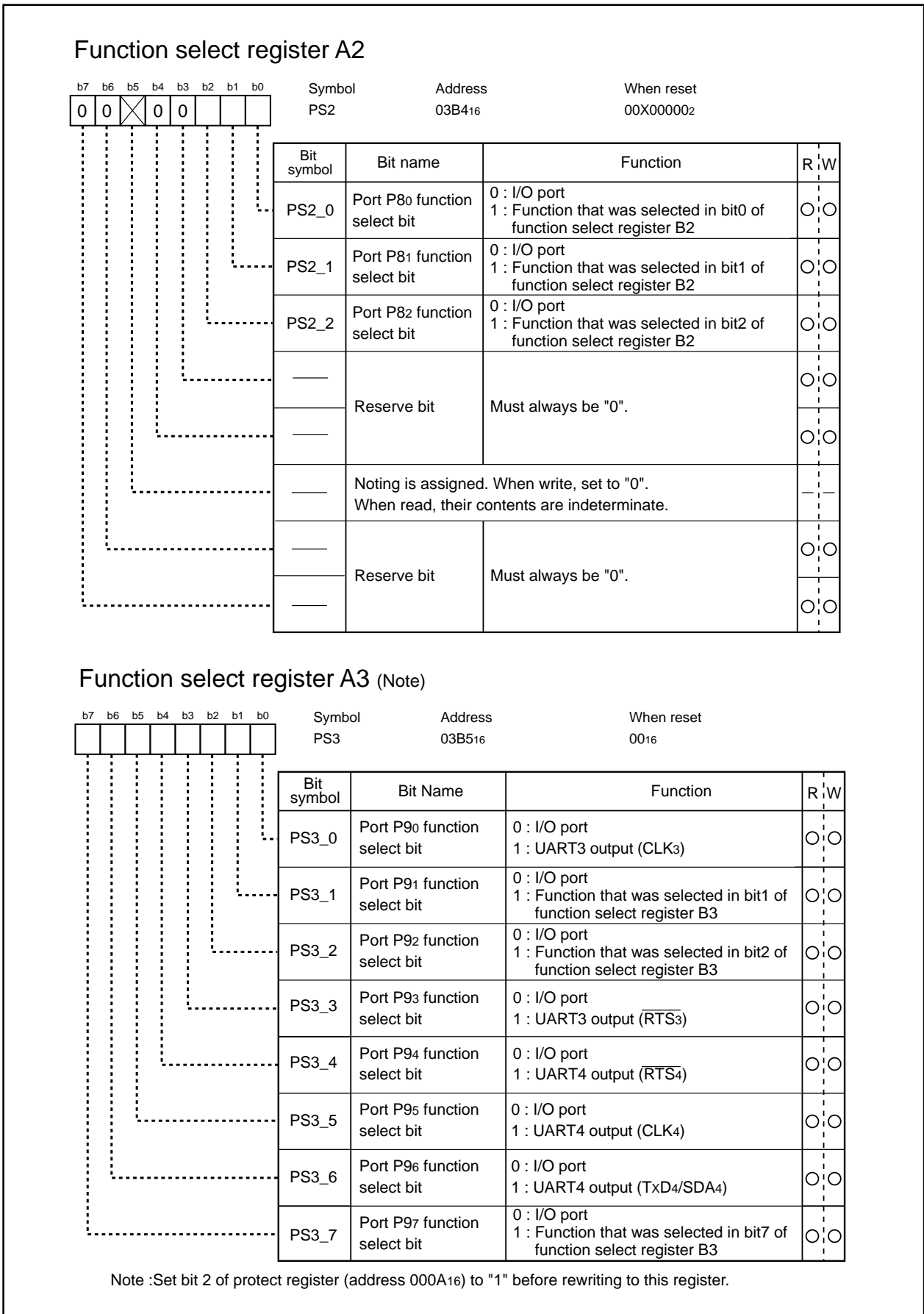


Figure 1.29.8. Function select register A (2)

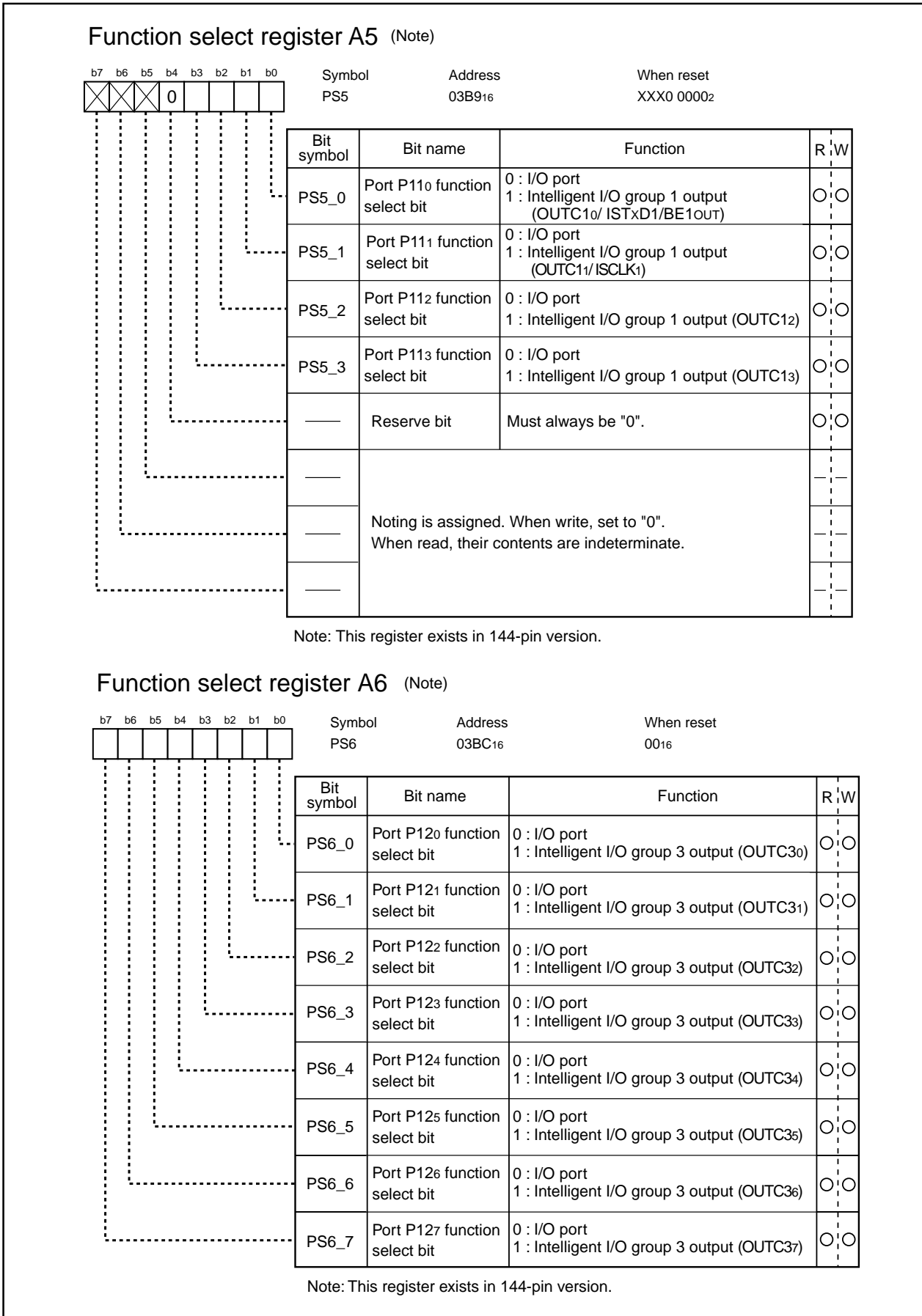


Figure 1.29.9. Function select register A (3)

Function select register A7 (Note)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	When reset
								PS7	03BD16	0016

Bit symbol	Bit name	Function	R/W
PS7_0	Port P130 function select bit	0 : I/O port 1 : Intelligent I/O group 2 output (OUTC24)	○ ○
PS7_1	Port P131 function select bit	0 : I/O port 1 : Intelligent I/O group 2 output (OUTC25)	○ ○
PS7_2	Port P132 function select bit	0 : I/O port 1 : Intelligent I/O group 2 output (OUTC26)	○ ○
PS7_3	Port P133 function select bit	0 : I/O port 1 : Intelligent I/O group 2 output (OUTC23)	○ ○
PS7_4	Port P134 function select bit	0 : I/O port 1 : Intelligent I/O group 2 output (OUTC20/ISTxD2/IEout)	○ ○
PS7_5	Port P135 function select bit	0 : I/O port 1 : Intelligent I/O group 2 output (OUTC22)	○ ○
PS7_6	Port P136 function select bit	0 : I/O port 1 : Intelligent I/O group 2 output (OUTC21/ISCLK2)	○ ○
PS7_7	Port P137 function select bit	0 : I/O port 1 : Intelligent I/O group 2 output (OUTC27)	○ ○

Note: This register exists in 144-pin version.

Function select register A8 (Note)

b7	b6	b5	b4	b3	b2	b1	b0	Symbol	Address	When reset
⊗	0	0	0					PS8	03A016	X0000002

Bit symbol	Bit name	Function	R/W
PS8_0	Port P140 function select bit	0 : I/O port 1 : Intelligent I/O group 1 output (OUTC14)	○ ○
PS8_1	Port P141 function select bit	0 : I/O port 1 : Intelligent I/O group 1 output (OUTC15)	○ ○
PS8_2	Port P142 function select bit	0 : I/O port 1 : Intelligent I/O group 1 output (OUTC16)	○ ○
PS8_3	Port P143 function select bit	0 : I/O port 1 : Intelligent I/O group 1 output (OUTC17)	○ ○
—	Reserve bit	Must always be "0".	○ ○
—			○ ○
—			○ ○
—	Noting is assigned. When write, set to "0". When read, their contents are indeterminate.		— —

Note: This register exists in 144-pin version.

Figure 1.29.10. Function select register A (4)

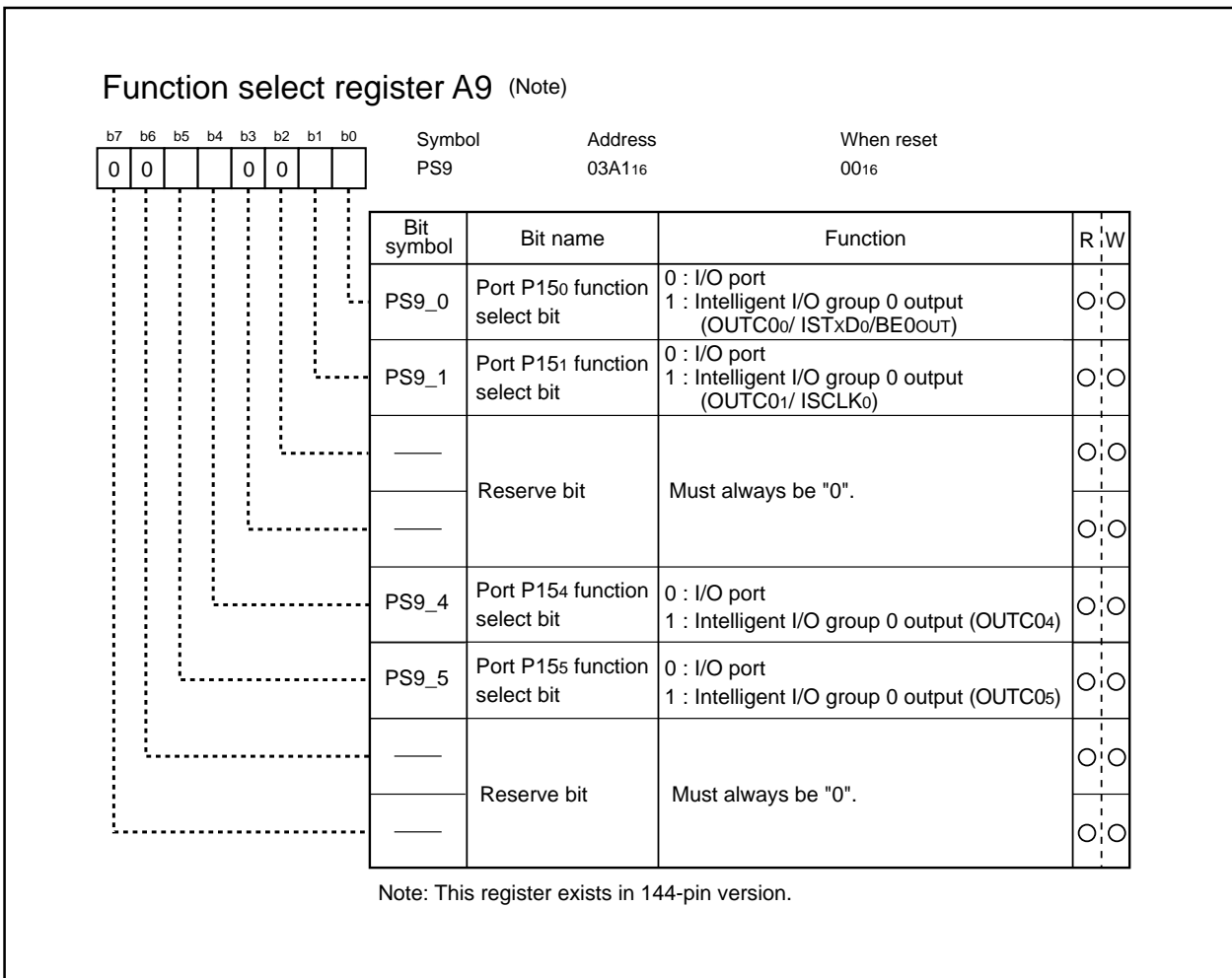


Figure 1.29.11. Function select register A (5)

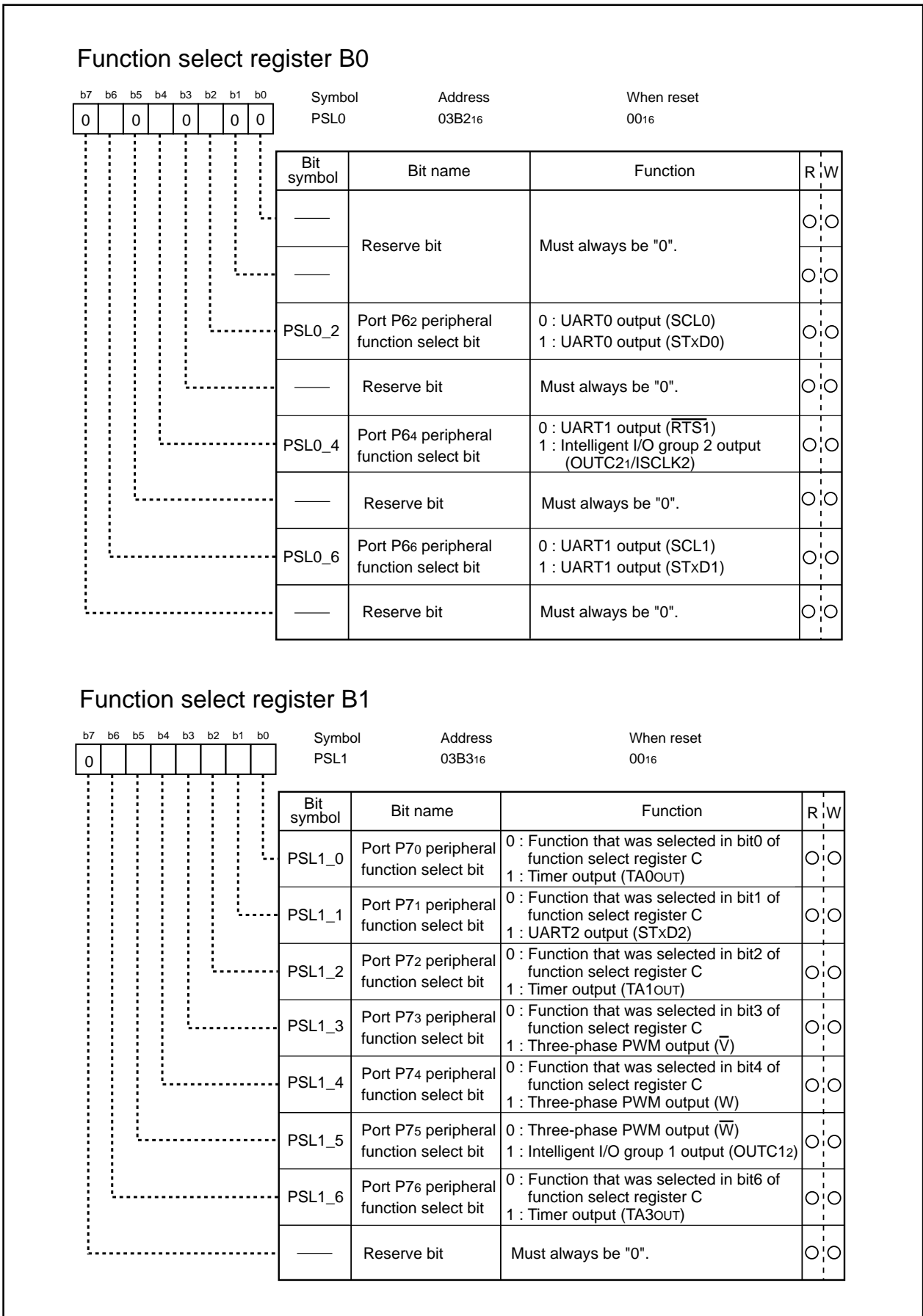


Figure 1.29.12. Function select register B (1)

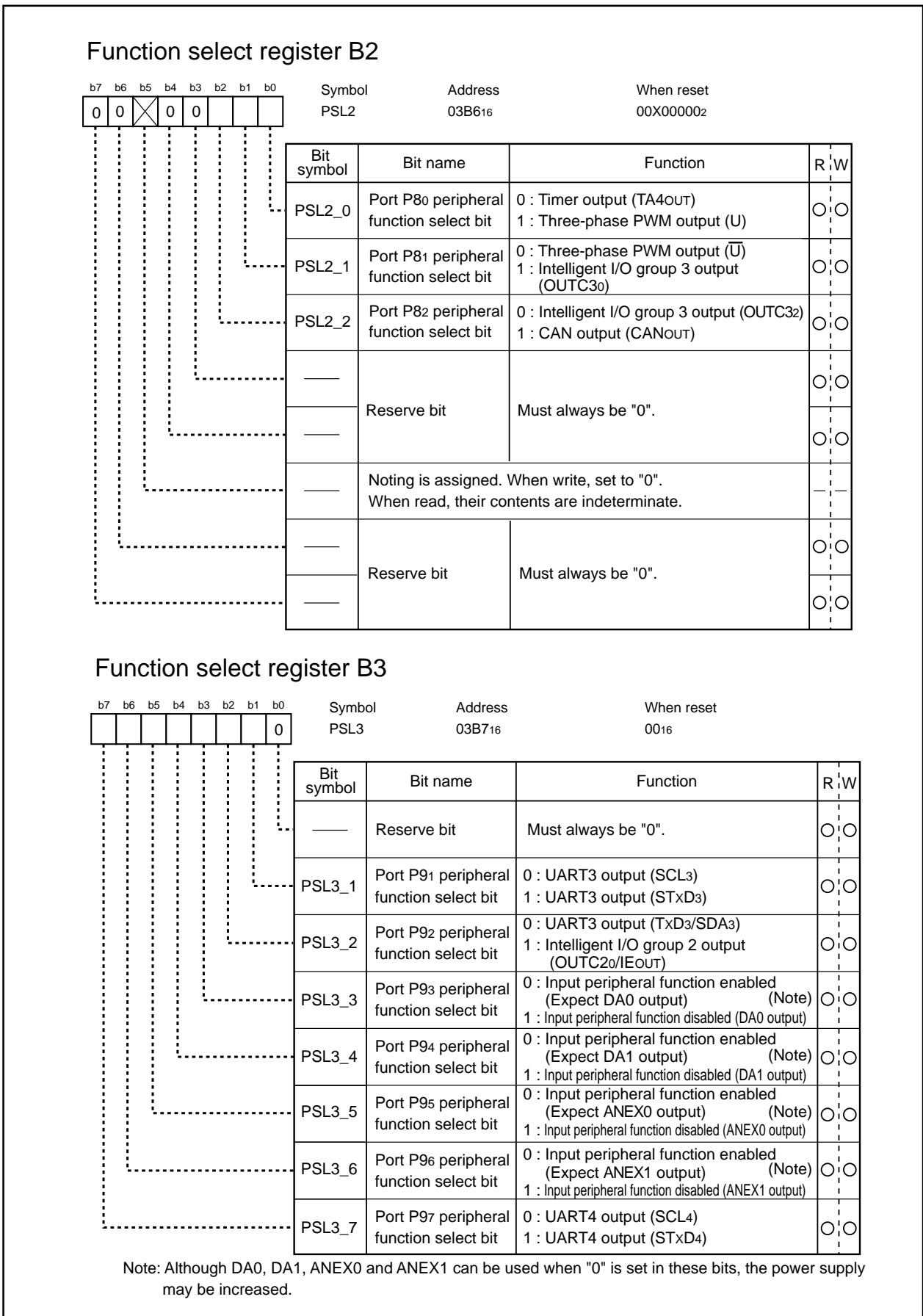


Figure 1.29.13. Function select register B (2)

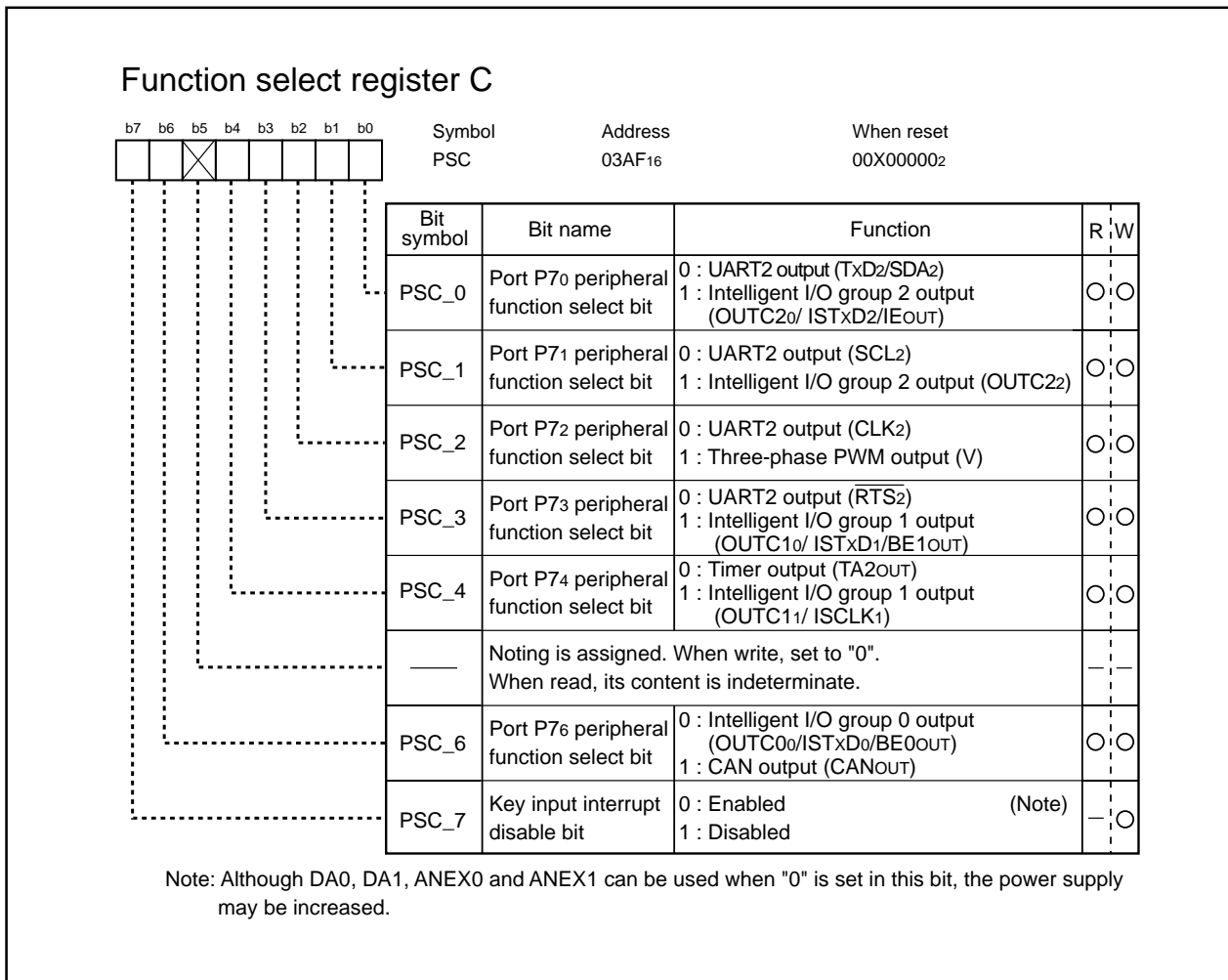


Figure 1.29.14. Function select register C

Programmable I/O Port

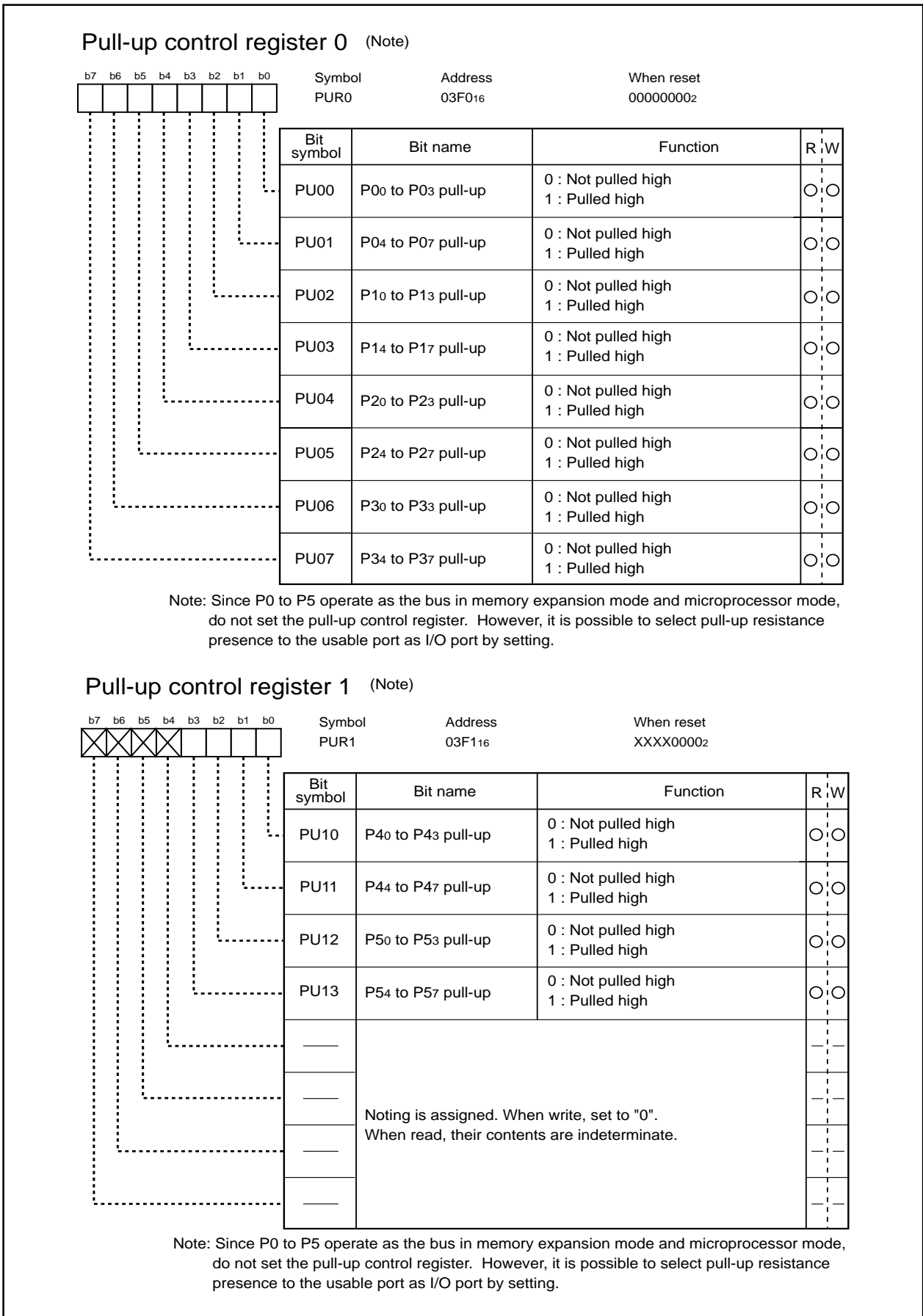


Figure 1.29.15. Pull-up control register (1)

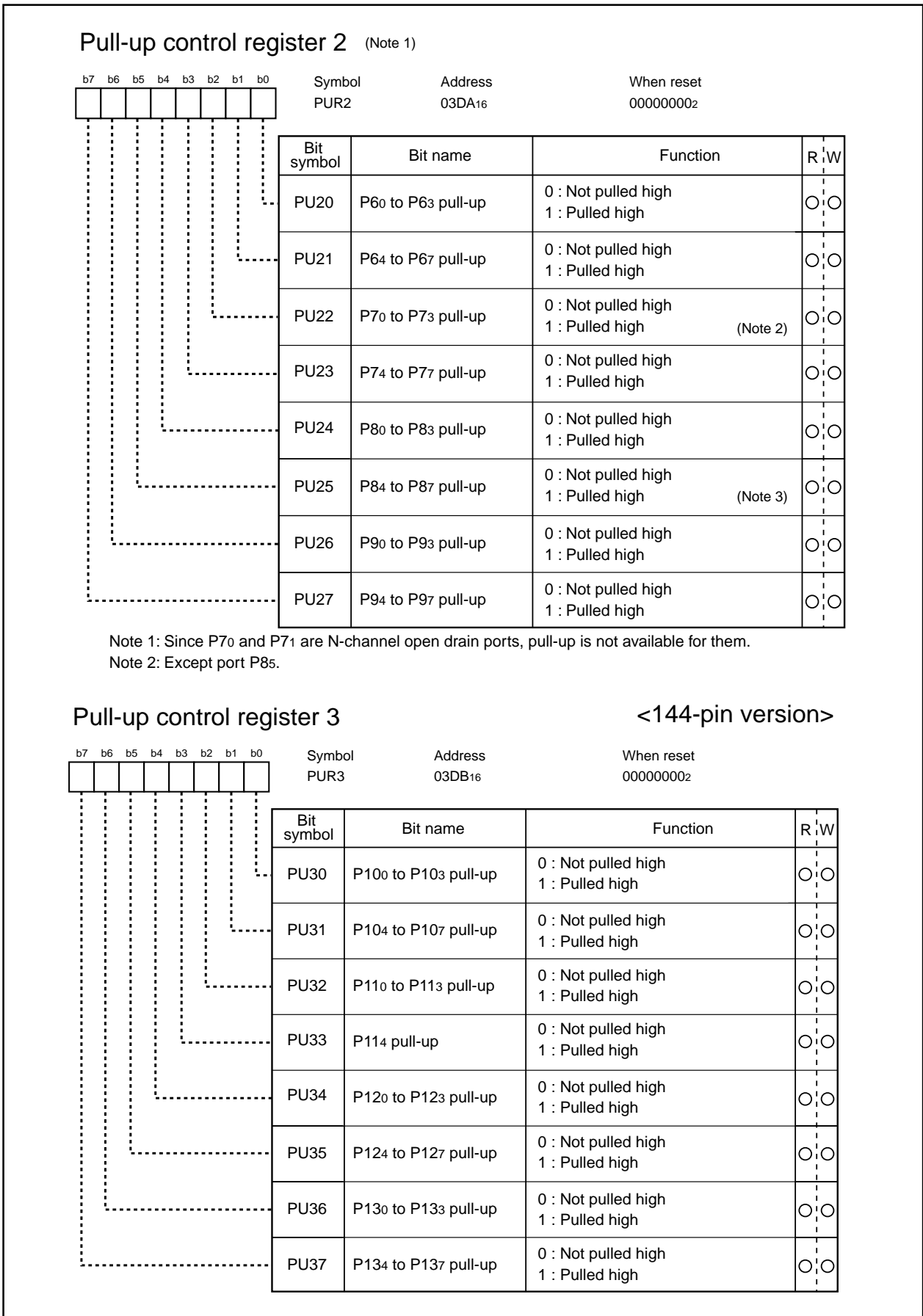


Figure 1.29.16. Pull-up control register (2)

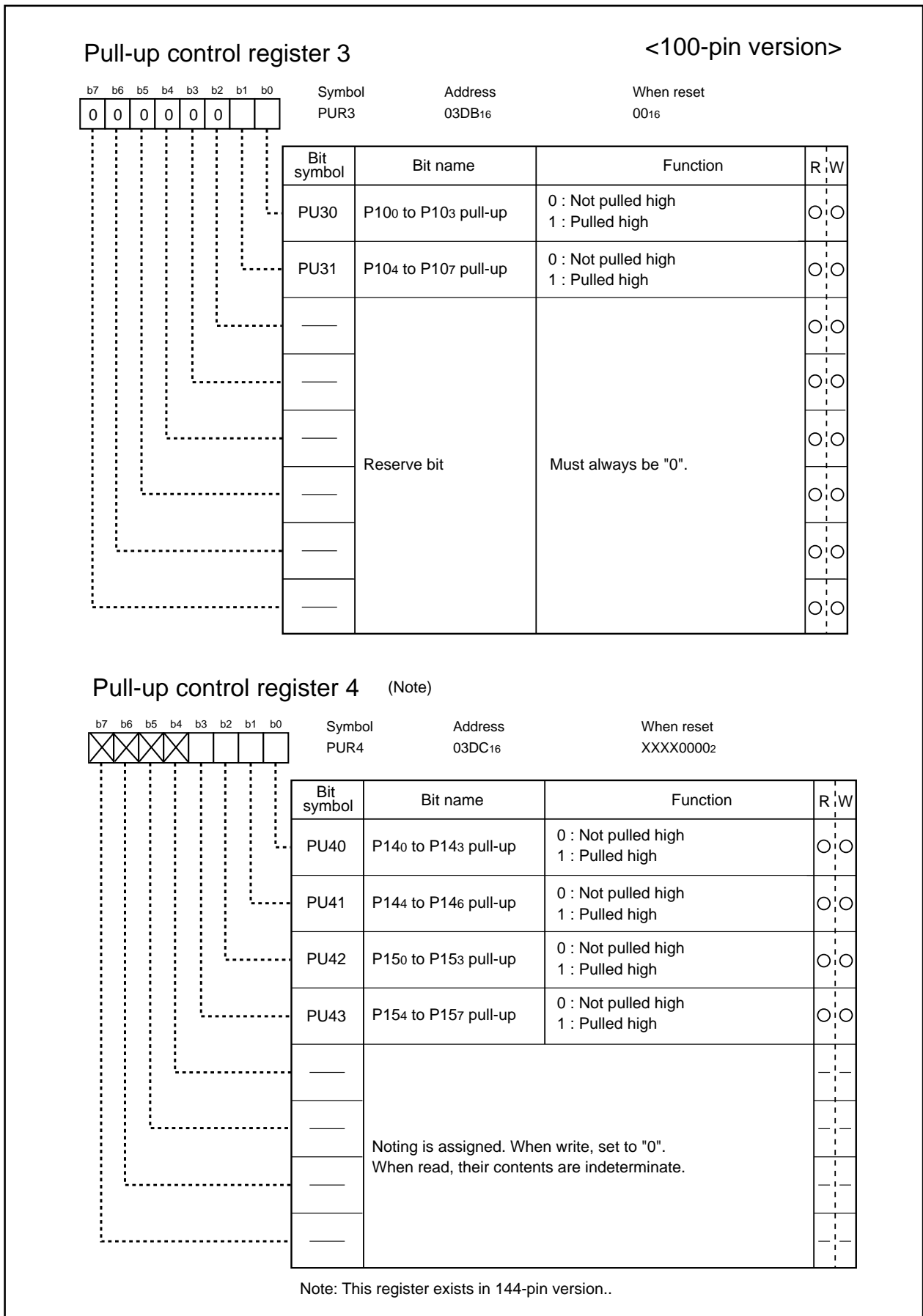


Figure 1.29.17. Pull-up control register (3)

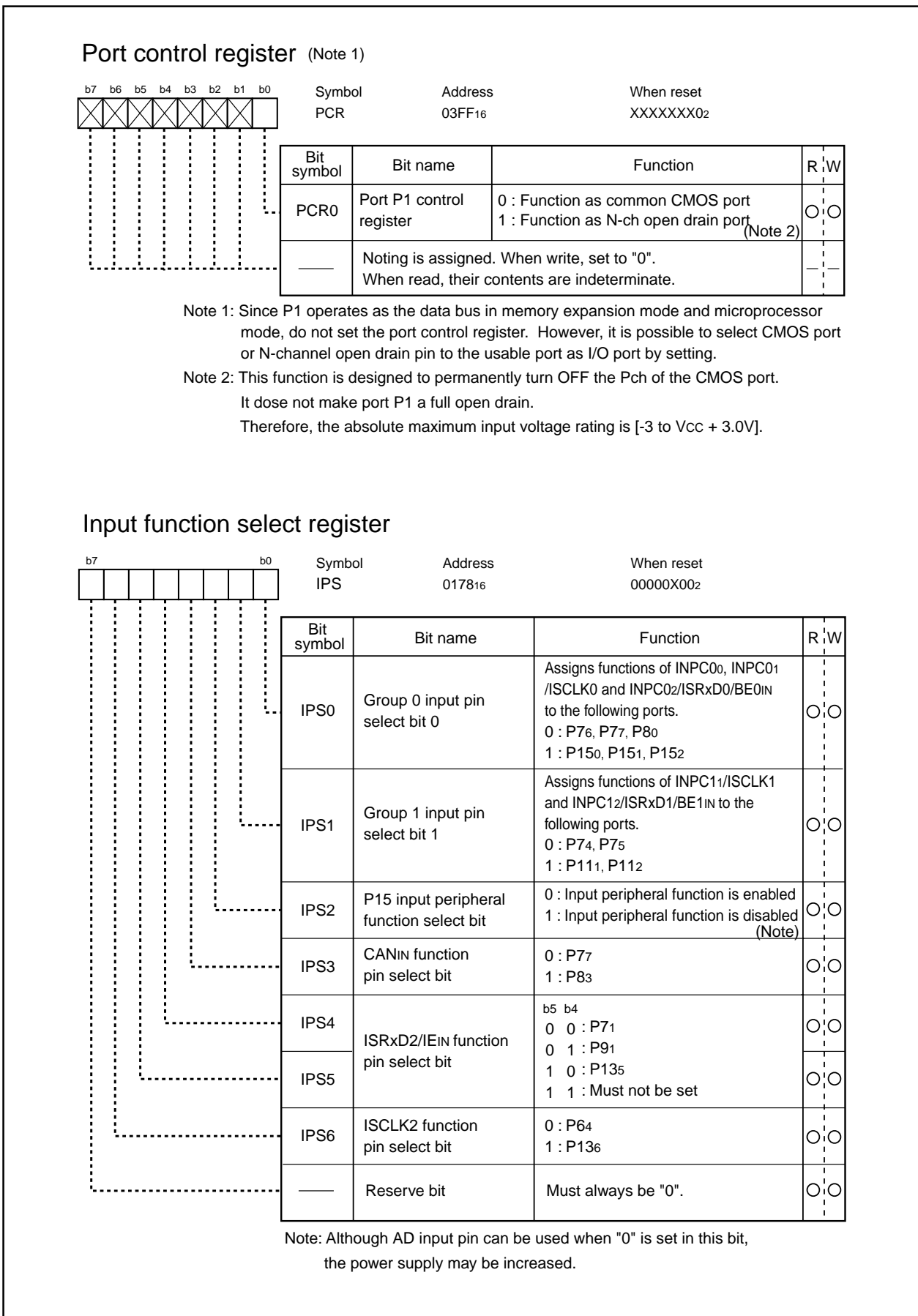


Figure 1.29.18. Port control register and input function select register

Programmable I/O Port

Table 1.29.1. Example connection of unused pins in single-chip mode

Pin name	Connection
Ports P0 to P15 (excluding P85) (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
NMI	Connect via resistance to Vcc (pull-up)
AVcc	Connect to Vcc
AVss, VREF, BYTE	Connect to Vss

Note 1: Ports P11 to P15 exist in 144-pin version.

Note 2: With external clock input to XIN pin.

Table 1.29.2. Example connection of unused pins in memory expansion mode and microprocessor mode

Pin name	Connection
Ports P6 to P15 (excluding P85) (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.
BHE, ALE, HLDA, XOUT(Note 2), BCLK	Open
HOLD, RDY, NMI	Connect via resistance to Vcc (pull-up)
AVcc	Connect to Vcc
AVss, VREF	Connect to Vss

Note 1: Ports P11 to P15 exist in 144-pin version.

Note 2: With external clock input to XIN pin.

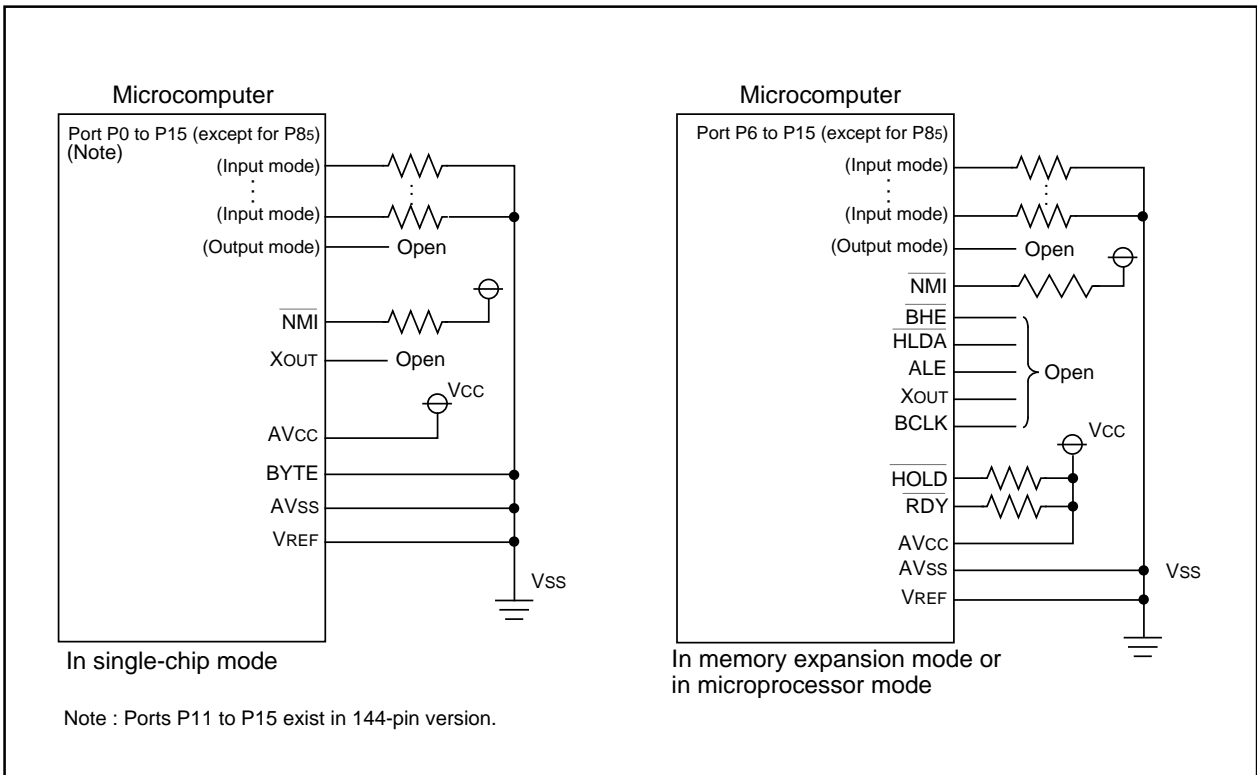


Figure 1.29.19. Example connection of unused pins

Table 1.29.3. Port P6 output control

	PS0 register	PSL0 register
Bit 0	0: P60 1: UART0 output (RTS0) ^(Note)	Must set to "0"
Bit 1	0: P61 1: UART0 output (CLK0) ^(Note)	Must set to "0"
Bit 2	0: P62 1: Selected by PSL0 register	0: UART0 output (SCL0) 1: UART0 output (STxD0)
Bit 3	0: P63 1: UART0 output (TxD0/SDA0) ^(Note)	Must set to "0"
Bit 4	0: P64 1: Selected by PSL0 register	0: UART1 output (RTS1) 1: Intelligent I/O group 2 (OUTC21/ISCLK2)
Bit 5	0: P65 1: UART1 output (CLK1) ^(Note)	Must set to "0"
Bit 6	0: P66 1: Selected by PSL0 register	0: UART1 output (SCL1) 1: UART1 output (STxD1)
Bit 7	0: P67 1: UART1 output (TxD1/SDA1) ^(Note)	Must set to "0"

PS0 register: Function select register A0

PSL0 register: Function select register B0

Note : Select "0" in corresponding bit of PSL0 register.

Table 1.29.4. Port P7 output control

	PS1 register	PSL1 register	PSC register
Bit 0	0: P70 1: Selected by PSL1 register	0: Selected by PSC register 1: Timer output (TA0OUT) ^(Note 1)	0: UART2 output (TxD2/SDA2) 1: Intelligent I/O group 2 (OUTC20/ISTxD2/IEOUT)
Bit 1	0: P71 1: Selected by PSL1 register	0: Selected by PSC register 1: UART2 output (STxD2) ^(Note 1)	0: UART2 output (SCL2) 1: Intelligent I/O group 2 (OUTC22)
Bit 2	0: P72 1: Selected by PSL1 register	0: Selected by PSC register 1: Timer output (TA1OUT) ^(Note 1)	0: UART2 output (CLK2) 1: Three-phase PWM output (V)
Bit 3	0: P73 1: Selected by PSL1 register	0: Selected by PSC register 1: Three-phase PWM output (\bar{V}) ^(Note 1)	0: UART2 output (RTS2) 1: Intelligent I/O group 1 (OUTC10/ISTxD1/BE1OUT)
Bit 4	0: P74 1: Selected by PSL1 register	0: Selected by PSC register 1: Three-phase PWM output (W) ^(Note 1)	0: Timer output (TA2OUT) 1: Intelligent I/O group 1 (OUTC11/ISCLK1)
Bit 5	0: P75 1: Selected by PSL1 register	0: Three-phase PWM output (\bar{W}) ^(Note 1) 1: Intelligent I/O group 1 (OUTC12)	Must set to "0"
Bit 6	0: P76 1: Selected by PSL1 register	0: Selected by PSC register 1: Timer output (TA3OUT)	0: Intelligent I/O group 0 (OUTC00/ISTxD0/BE0OUT) 1: CAN output (CANOUT)
Bit 7	0: P77 1: Intelligent I/O group 0 (OUTC01/ISCLK0)	Must set to "0"	0: Key input interrupt signal enabled 1: Key input interrupt signal disabled

PS1 register: Function select register A1

PSL1 register: Function select register B1

PSC register: Function select register C

Note 1: Select "0" in corresponding bit of PSC register.

Note 2: Select "0" in corresponding bit of PSL1 register.

Programmable I/O Port

Table 1.29.5. Port P8 output control

	PS2 register	PSL2 register
Bit 0	0: P80 1: Selected by PSL2 register	0: Timer output (TA4OUT) 1: Three-phase PWM output (U)
Bit 1	0: P81 1: Selected by PSL2 register	0: Three-phase PWM output (\bar{U}) 1: Intelligent I/O group 3(OUTC30)
Bit 2	0: P82 1: Selected by PSL2 register	0: Intelligent I/O group 3(OUTC32) 1: CAN output (CANOUT)
Bit 3 to 7 Must set to "0"		

PS2 register: Function select register A2

PSL2 register: Function select register B2

Table 1.29.6. Port P9 output control

	PS3 register	PSL3 register
Bit 0	0: P90 1: UART3 output (CLK3) ^(Note)	Must set to "0"
Bit 1	0: P91 1: Selected by PSL3 register	0: UART3 output (SCL3) 1: UART3 output (STxD3)
Bit 2	0: P92 1: Selected by PSL3 register	0: UART3 output (TxD3/SDA3) 1: Intelligent I/O group 2 (OUTC20/IEOUT)
Bit 3	0: P93 1: UART3 output ($\overline{RTS3}$) ^(Note)	0: Except DA0 output 1: DA0 output
Bit 4	0: P94 1: UART4 output ($\overline{RTS4}$) ^(Note)	0: Except DA1 output 1: DA1 output
Bit 5	0: P95 1: UART4 output (CLK4) ^(Note)	0: Except ANEX0 1: ANEX0
Bit 6	0: P96 1: UART4 output (TxD4/SDA4) ^(Note)	0: Except ANEX1 1: ANEX1
Bit 7	0: P97 1: Selected by PSL3 register	0: UART4 output (SCL4) 1: UART4 output (STxD4)

PS3 register: Function select register A3

PSL3 register: Function select register B3

Note : Select "0" in corresponding bit of PSL3 register.

Table 1.29.7. Port P11 output control

	PS5 register
Bit 0	0: P110 1: Intelligent I/O group 1(OUTC10/ISTxD1/BE1OUT)
Bit 1	0: P111 1: Intelligent I/O group 1(OUTC11/ISCLK1)
Bit 2	0: P112 1: Intelligent I/O group 1(OUTC12)
Bit 3	0: P113 1: Intelligent I/O group 1(OUTC13)
Bit 4 to 7 Must set to "0"	

PS5 register: Function select register A5

Table 1.29.8. Port P12 output control

	PS6 register
Bit 0	0: P120 1: Intelligent I/O group 3(OUTC30)
Bit 1	0: P121 1: Intelligent I/O group 3(OUTC31)
Bit 2	0: P122 1: Intelligent I/O group 3(OUTC32)
Bit 3	0: P123 1: Intelligent I/O group 3(OUTC33)
Bit 4	0: P124 1: Intelligent I/O group 3(OUTC34)
Bit 5	0: P125 1: Intelligent I/O group 3(OUTC35)
Bit 6	0: P126 1: Intelligent I/O group 3(OUTC36)
Bit 7	0: P127 1: Intelligent I/O group 3(OUTC37)

PS6 register: Function select register A6

Table 1.29.9. Port P13 output control

	PS7 register
Bit 0	0: P130 1: Intelligent I/O group 2(OUTC24)
Bit 1	0: P131 1: Intelligent I/O group 2(OUTC25)
Bit 2	0: P132 1: Intelligent I/O group 2(OUTC26)
Bit 3	0: P133 1: Intelligent I/O group 2(OUTC23)
Bit 4	0: P134 1: Intelligent I/O group 2(OUTC20/ISTxD2/IEOUT)
Bit 5	0: P135 1: Intelligent I/O group 2(OUTC22)
Bit 6	0: P136 1: Intelligent I/O group 2(OUTC21/ISCLK2)
Bit 7	0: P137 1: Intelligent I/O group 2(OUTC27)

PS7 register: Function select register A7

Table 1.29.10. Port P14 output control

	PS8 register
Bit 0	0: P140 1: Intelligent I/O group 1(OUTC14)
Bit 1	0: P141 1: Intelligent I/O group 1(OUTC15)
Bit 2	0: P142 1: Intelligent I/O group 1(OUTC16)
Bit 3	0: P143 1: Intelligent I/O group 1(OUTC17)
Bit 4 to 7	Must set to "0"

PS8 register: Function select register A8

Programmable I/O Port

Table 1.29.11. Port P15 output control

	PS9 register
Bit 0	0: P150 1: Intelligent I/O group 0 (OUTC00/ISTxD0/BEOUT)
Bit 1	0: P151 1: Intelligent I/O group 0 (OUTC01/ISCLK0)
Bit 2 to 3	Must set to "0"
Bit 4	0: P154 1: Intelligent I/O group 0 (OUTC04)
Bit 5	0: P155 1: Intelligent I/O group 0 (OUTC05)
Bit 6 to 7	Must set to "0"

PS9 register: Function select register A9

VDC

VDC

When power-supply voltage is 3.3V or under, set the internal VDC (Voltage Down Converter) unused.

Follow the steps given below to disable the VDC.

- (1) Set bit 3 of the protect register to "1".
- (2) Set the VDC control register 0 to "0F16".
- (3) Set the VDC control register 0 to "8F16".
- (4) Set bit 3 of the protect register to "0".

These steps must be performed after reset as immediately as possible with divide-by-8 clock. When the VDC select bit has been set to "112" once, do not set any other values.

Figure 1.30.1 shows the VDC control register 0.

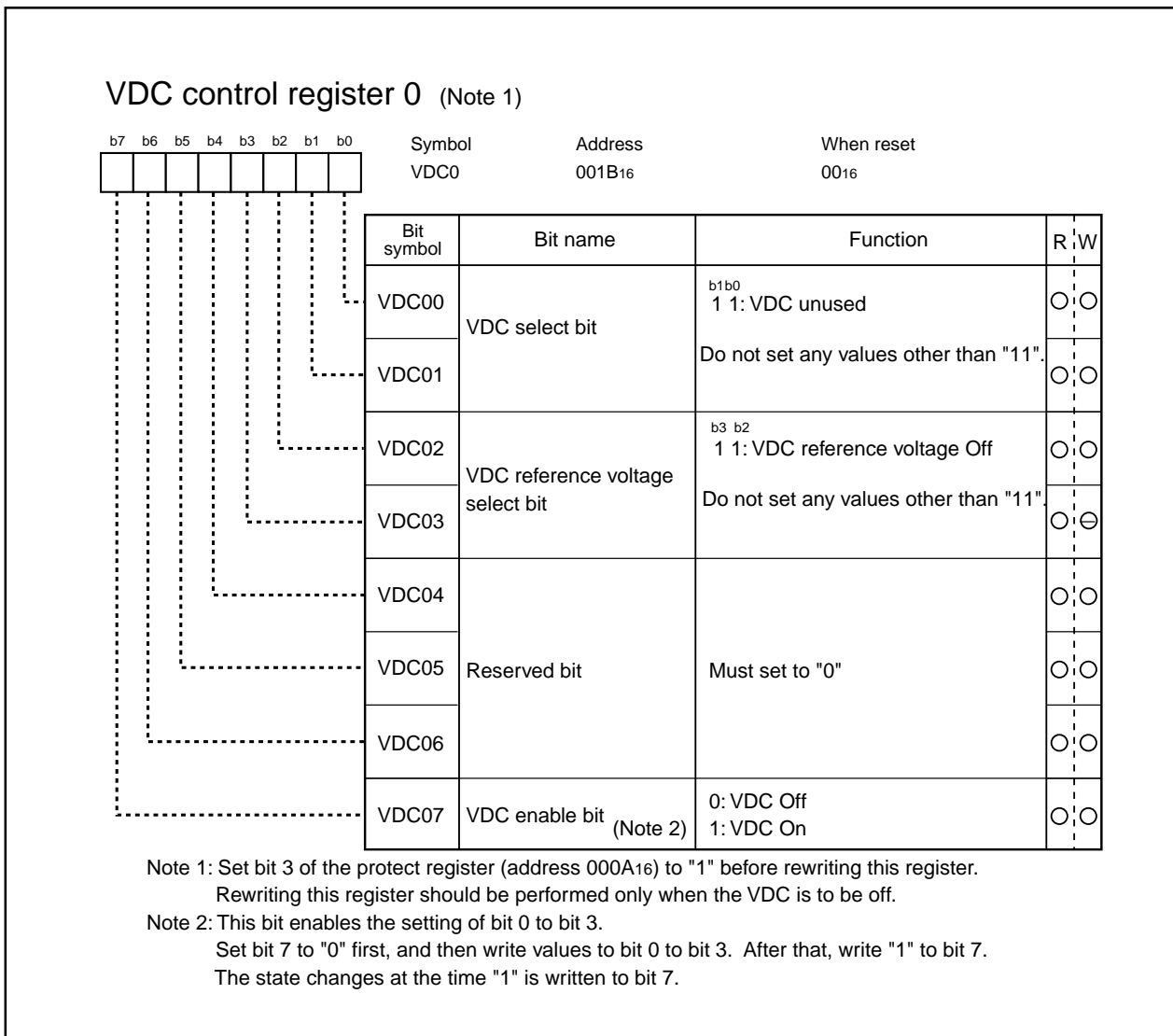


Figure 1.30.1. VDC control register

Usage Precaution

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 while reloading gets "FFFF₁₆". 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 while reloading gets "FFFF₁₆" by underflow or "0000₁₆" 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 "0000₁₆" to the timer register. If counting down, rewrite "FFFF₁₆" 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 TAIOUT 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 TAIIN 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 become “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 while reloading gets “FFFF16”. 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 i (i=0,1) control register 0, to each bit of A-D i control register 1, and to each bit of A-D i 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 μ s 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 1.31.1 has to be completed within a specified period of time T. Let output impedance of sensor equivalent circuit be R_0 , 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).

$$V_c \text{ is generally } V_c = V_{IN} \left\{ 1 - e^{-\frac{t}{C(R_0 + R)}} \right\}$$

$$\text{And when } t = T, \quad V_c = V_{IN} - \frac{X}{Y} V_{IN} = V_{IN} \left(1 - \frac{X}{Y} \right)$$

$$e^{-\frac{T}{C(R_0 + R)}} = \frac{X}{Y}$$

$$-\frac{T}{C(R_0 + R)} = \ln \frac{X}{Y}$$

$$\text{Hence, } R_0 = -\frac{T}{C \cdot \ln \frac{X}{Y}} - R$$

With the model shown in Figure 1.31.1 as an example, when the difference between V_{IN} and V_c becomes 0.1LSB, we find impedance R_0 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, $T = 0.3 \mu$ s in the A-D conversion mode with sample & hold. Output impedance R_0 for sufficiently charging capacitor C within time T is determined as follows.

$$T = 0.3 \mu\text{s}, R = 7.8 \text{ k}\Omega, C = 3 \text{ pF}, X = 0.1, \text{ and } Y = 1024. \text{ Hence,}$$

$$R_0 = -\frac{0.3 \times 10^{-6}}{3.0 \times 10^{-12} \cdot \ln \frac{0.1}{1024}} - 7.8 \times 10^3 \approx 3.0 \times 10^3$$

Usage precaution

Thus, the allowable output impedance of the sensor circuit capable of thoroughly driving the A-D converter turns out to be approximately 3.0 k Ω . Tables 1.31.1 and 1.31.2 show output impedance values based on the LSB values.

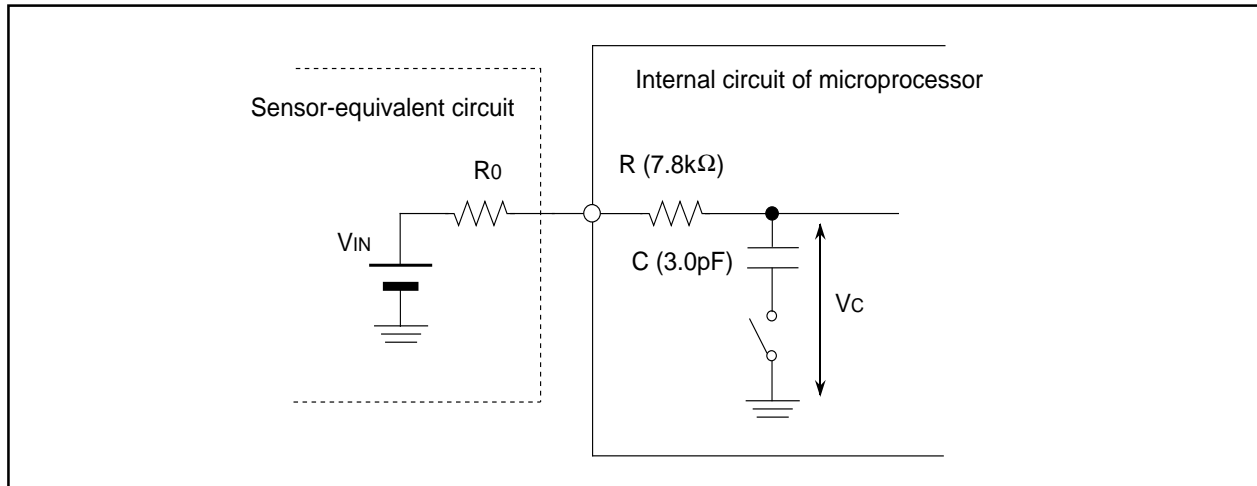


Figure 1.31.1 A circuit equivalent to the A-D conversion terminal

(7) After A-D conversion is complete, if the CPU reads the A-D register at the same time as the A-D conversion result is being saved to A-D register, wrong A-D conversion value is saved into the A-D register. This happens when the internal CPU clock is selected from divided main clock or sub-clock.

- **When using the one-shot or single sweep mode**

Confirm that A-D conversion is complete before reading the A-D register.

(Note: When A-D conversion interrupt request bit is set, it shows that A-D conversion is completed.)

- **When using the repeat mode or repeat sweep mode 0 or 1**

Use the undivided main clock as the internal CPU clock.

Interrupts

(1) Setting the stack pointer

- The value of the stack pointer is initialized to 000000₁₆ 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 VCC pin via a resistance (pulled-up) if unused.
- The $\overline{\text{NMI}}$ pin also serves as P85, 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.

Usage precaution

Tables 1.31.1. Output impedance values based on the LSB values (10-bit mode) Reference value

f(XIN) (MHz)	Cycle (μ s)	Sampling time (μ s)	R (k Ω)	C (pF)	Resolution (LSB)	R0max (k Ω)
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 disabled)	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 1.31.2. Output impedance values based on the LSB values (8-bit mode) Reference value

f(XIN) (MHz)	Cycle (μ s)	Sampling time (μ s)	R (k Ω)	C (pF)	Resolution (LSB)	R0max (k Ω)
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 disabled)	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

Usage precaution

(3) 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 \overline{INT}_0 to \overline{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 \overline{INT}_0 to \overline{INT}_5 regardless of the CPU operation clock. (When $X_{IN}=30\text{MHz}$ and no division mode, at least 233 ns width is necessary.)

- When the polarity of the \overline{INT}_0 to \overline{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 1.31.2 shows the procedure for changing the \overline{INT} interrupt generate factor.

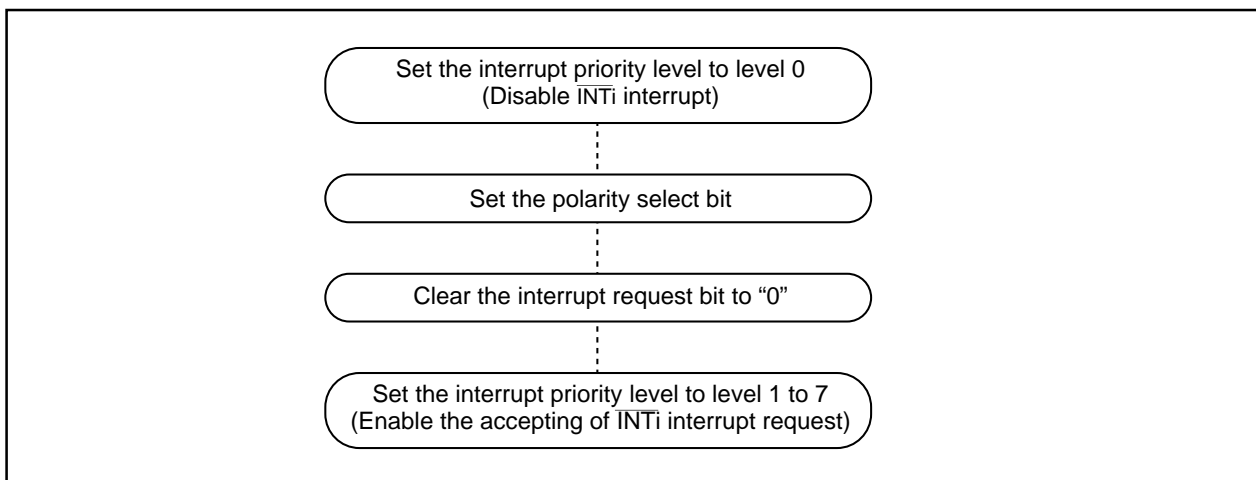


Figure 1.31.2. Switching condition of \overline{INT} interrupt request

(4) Rewrite the interrupt control register

- When an 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 instructions. 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 M32C/83, 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, disable the corresponding DMA channel to disabled before changing the DMAi request cause select bit. To enable DMA at least $8+6 \times N$ cycles (N: enabled channel number) following the instruction to write to the DMAi request cause select register are needed.

Usage precaution

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)

nop
:
ldc       R0, DMD0          ; DMA0 enabled
pop.w    R0                ; Restore R0 register

```

} **At least 8 + 6 x N cycles**
} **(N: enabled channel number)**

Noise

- (1) A bypass capacitor should be inserted between Vcc-Vss line for reducing noise and latch-up. Connect a bypass capacitor (approx. 0.1 μ F) 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 P53/CLKOUT pin (f8, f32 or fc output) in single chip mode, use port P57 as an input only port (port P57 direction register is "0").

Although port P57 may be set as an output port (port P57 direction register is "1"), it will become high impedance and will not output "H" or "L" levels.

HOLD signal

When using the $\overline{\text{HOLD}}$ input while P40 to P47 and P50 to P52 are set as output ports in single-chip mode, you must first set all pins for P40 to P47 and P50 to P52 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 μ s or longer after connecting Vref.
- (2) When using AN4 (P104) to AN7 (P107), 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 P104 to P107 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 P95 and P96 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.

Usage precaution

- (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 "0016".
- (5) When D-A conversion is used, select the input peripheral function disabled with port P93 and P94 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

The DRAM self-refresh operates in stop mode, etc.

When shifting to self-refresh, select DRAM is 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 is 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 is 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 1.31.3 include indeterminate bit when read. Set immediate to these registers.

Store the content of the frequently used register to RAM, change the content of RAM, then transfer to the register.

Table 1.31.3 The object registers

Register name	Symbol	Address
Watchdog timer start register	WDTS	000E ₁₆
Group0 receive input register	G0RI	00EC ₁₆
Group1 receive input register	G1RI	012C ₁₆
Group2 S/O transmit buffer register	G2TB	016D ₁₆ , 016C ₁₆
UART4 bit rate generator	U4BRG	02F9 ₁₆
UART4 transfer buffer register	U4TB	02FB ₁₆ , 02FA ₁₆
Timer A1-1 register	TA11	0303 ₁₆ , 0302 ₁₆
Timer A2-1 register	TA21	0305 ₁₆ , 0304 ₁₆
Timer A4-1 register	TA41	0307 ₁₆ , 0306 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	0369 ₁₆
UART0 transfer buffer register	U0TB	036B ₁₆ , 036A ₁₆
UART1 bit rate generator	U1BRG	02E9 ₁₆
UART1 transfer buffer register	U1TB	02EB ₁₆ , 02EA ₁₆
A-D0 control register 2	ADCON2	0394 ₁₆

Note: In one-shot timer mode and pulse width modulation mode.

Notes on the microprocessor mode and transition after shifting from the microprocessor mode to the memory expansion mode / single-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.

Notes on CNVss pin reset at "H" level

When the CNVss pin is reset at "H" level, the contents of internal ROM cannot be read out.

Electrical characteristics

Table 1.32.1. Absolute maximum ratings

Symbol	Parameter	Condition	Rated value	Unit
Vcc	Supply voltage	Vcc=AVcc	-0.3 to 6.0	V
AVcc	Analog supply voltage	Vcc=AVcc	-0.3 to 6.0	V
Vi	Input voltage	RESET, CNVss, BYTE, P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 ^(Note1) , VREF, XIN	-0.3 to Vcc+0.3	V
		P70, P71	-0.3 to 6.0	V
Vo	Output voltage	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 ^(Note1) , VREF, XIN	-0.3 to Vcc+0.3	V
		P70, P71	-0.3 to 6.0	V
Pd	Power dissipation	Topr=25°C	500	mW
Topr	Operating ambient temperature		-20 to 85/-40 to 85 ^(Note 2)	°C
Tstg	Storage temperature		-65 to 150	°C

Note 1: Ports P11 to P15 exist in 144-pin version.

Note 2: Specify a product of -40 to 85°C to use it.

Table 1.32.2. Recommended operating conditions (referenced to VCC = 3.0V 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(When VDC-ON)		3.0	5.0	5.5	V
	Supply voltage(When VDC-pass through)		3.0	3.3	3.6	V
AVcc	Analog supply voltage			Vcc		V
Vss	Supply voltage			0		V
AVss	Analog supply voltage			0		V
VIH	"H" input voltage	P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 ^(Note5) , XIN, RESET, CNVss, BYTE	0.8Vcc		Vcc	V
		P70, P71	0.8Vcc		6.0	V
		P00-P07, P10-P17 (during single-chip mode)	0.8Vcc		Vcc	V
		P00-P07, P10-P17 (during memory-expansion and microprocessor modes)	0.5Vcc		Vcc	V
VIL	"L" input voltage	P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 ^(Note5) , XIN, RESET, CNVss, BYTE	0		0.2Vcc	V
		P00-P07, P10-P17 (during single-chip mode)	0		0.2Vcc	V
		P00-P07, P10-P17 (during memory-expansion and microprocessor modes)	0		0.16Vcc	V
IOH(peak)	"H" peak output current	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 ^(Note5)			-10.0	mA
IOH(avg)	"H" average output current	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 ^(Note5)			-5.0	mA
IOL(peak)	"L" peak output current	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 ^(Note5)			10.0	mA
IOL(avg)	"L" average output current	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 ^(Note5)			5.0	mA
f(XIN)	Main clock input frequency	VDC-ON	Vcc=4.2 to 5.5V	0	30	MHz
			Vcc=3.0 to 4.2V	0	20	MHz
		VDC-pass through	Vcc=3.0 to 3.6V	0	20	MHz
f(XCIN)	Sub-clock oscillation frequency			32.768		kHz

Note 1: The mean output current is the mean value within 100ms.

Note 2: The total IOL (peak) for ports P0, P1, P2, P86, P87, P9, P10, P11, P14 and P15 must be 80mA max. The total IOH (peak) for ports P0, P1, P2, P86, P87, P9, P10, P11, P14 and P15 must be -80mA max. The total IOL (peak) for ports P3, P4, P5, P6, P7, P80 to P84, P12 and P13 must be 80mA max. The total IOH (peak) for ports P3, P4, P5, P6, P72 to P77, P80 to P84, 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 VIH and VIL of P87 is not when using as XCIN but when using programmable input port.

Note 5: Port P11 to P15 exist in 144-pin version.

Table 1.32.3. Electrical characteristics (referenced to V_{CC}=5V, V_{SS}=0V at Topr=25°C, f(X_{IN})=30MHz unless otherwise specified)

V_{CC} = 5V

Symbol	Parameter		Condition	Standard			Unit
				Min.	Typ.	Max.	
VOH	"H" output voltage	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157(Note1)	IOH=-5mA	3.0			V
VOH	"H" output voltage	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157(Note1)	IOH=-200μA	4.7			V
VOH	"H" output voltage	XOUT	HIGH POWER	IOH=-1mA	3.0		V
			LOW POWER	IOH=-0.5mA	3.0		V
	"H" output voltage	XCOUT	No load applied		3.0		V
VOL	"L" output voltage	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157(Note1)	IOH=5mA			2.0	V
VOL	"L" output voltage	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157(Note1)	IOH=200μA			0.45	V
VOL	"L" output voltage	XOUT	HIGH POWER	IOL=1mA		2.0	V
			LOW POWER	IOL=0.5mA		2.0	V
	"L" output voltage	XCOUT	No load applied		0		V
VT+-VT-	Hysteresis	HOLD, RDY, TA0IN-TA4IN, TB0IN-TB5IN, INT0-INT5, ADTRG, CTS0-CTS4, CLK0-CLK4, TA0OUT-TA4OUT, NM̄I, K̄I0-K̄I3, RxD0-RxD4, SCL0-SCL4, SDA0-SDA4		0.2		1.0	V
VT+-VT-	Hysteresis	RESET		0.2		1.8	V
I _{IH}	"H" input current	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157(Note1), X _{IN} , RESET, CNV _{SS} , BYTE	V _I =5V			5.0	μA
I _{IL}	"L" input current	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157(Note1), X _{IN} , RESET, CNV _{SS} , BYTE	V _I =0V			-5.0	μA
RPULLUP	Pull-up resistance	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157(Note1)	V _I =0V	30	50	167	kΩ
R _{FXIN}	Feedback resistance	X _{IN}			1.5		MΩ
R _{FXCIN}	Feedback resistance	X _{CIN}			10		MΩ
V _{RAM}	RAM retention voltage	V _{DC-ON}		2.5			V
I _{CC}	Power supply current	Measuring condition: In single-chip mode, the output pins are open and other pins are V _{SS} .	f(X _{IN})=30MHz, square wave, no division		38	54	mA
			f(X _{CIN})=32kHz, with WAIT instruction executed		470		μA
			when clock is stopped Topr=25°C		0.4	20	μA

Note 1: Port P11 to P15 exist in 144-pin version.

V_{CC} = 5V

Table 1.32.4. A-D conversion characteristics (referenced to V_{CC} = AV_{CC} = V_{REF} = 5V, V_{SS} = AV_{SS} = 0V at T_{opr} = 25°C, f(X_{IN}) = 30MHz unless otherwise specified)

Symbol	Parameter	Measuring condition	Standard			Unit
			Min.	Typ.	Max.	
-	Resolution	V _{REF} = V _{CC}			10	Bits
INL	Integral nonlinearity error	V _{REF} = V _{CC} = 5V AN ₀ to AN ₇ ANEX ₀ , ANEX ₁ External op-amp connection mode			±3	LSB
					±7	LSB
DNL	Differential nonlinearity error				±1	LSB
-	Offset error				±3	LSB
-	Gain error				±3	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 1.32.5. D-A conversion characteristics (referenced to V_{CC} = V_{REF} = 5V, V_{SS} = AV_{SS} = 0V at T_{opr} = 25°C, f(X_{IN}) = 30MHz 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 "0016".

The A-D converter's ladder resistance is not included.

Also, when the V_{ref} is unconnected at the A-D control register 1, I_{VREF} is sent.

V_{CC} = 5VTiming requirements (referenced to V_{CC} = 5V, V_{SS} = 0V at Topr = 25°C unless otherwise specified)

Table 1.32.6. External clock input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _c	External clock input cycle time	33		ns
t _{w(H)}	External clock input HIGH pulse width	13		ns
t _{w(L)}	External clock input LOW pulse width	13		ns
t _r	External clock rise time		5	ns
t _f	External clock fall time		5	ns

Table 1.32.7. 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)	RDY input setup time	26		ns
t _{su} (HOLD-BCLK)	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)	RDY input hold time	0		ns
t _h (BCLK-HOLD)	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.

$$t_{ac1}(\text{RD-DB}) = \frac{10^9}{f(\text{BCLK}) \times 2} - 35 \quad [\text{ns}]$$

$$t_{ac1}(\text{AD-DB}) = \frac{10^9}{f(\text{BCLK})} - 35 \quad [\text{ns}]$$

$$t_{ac2}(\text{RD-DB}) = \frac{10^9 \times m}{f(\text{BCLK}) \times 2} - 35 \quad [\text{ns}] \quad (m=3, 5 \text{ and } 7 \text{ when } 1 \text{ wait, } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$t_{ac2}(\text{AD-DB}) = \frac{10^9 \times n}{f(\text{BCLK})} - 35 \quad [\text{ns}] \quad (n=2, 3 \text{ and } 4 \text{ when } 1 \text{ wait, } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$t_{ac3}(\text{RD-DB}) = \frac{10^9 \times m}{f(\text{BCLK}) \times 2} - 35 \quad [\text{ns}] \quad (m=3 \text{ and } 5 \text{ when } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$t_{ac3}(\text{AD-DB}) = \frac{10^9 \times n}{f(\text{BCLK}) \times 2} - 35 \quad [\text{ns}] \quad (n=5 \text{ and } 7 \text{ when } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$t_{ac4}(\text{RAS-DB}) = \frac{10^9 \times m}{f(\text{BCLK}) \times 2} - 35 \quad [\text{ns}] \quad (m=3 \text{ and } 5 \text{ when } 1 \text{ wait and } 2 \text{ wait, respectively})$$

$$t_{ac4}(\text{CAS-DB}) = \frac{10^9 \times n}{f(\text{BCLK}) \times 2} - 35 \quad [\text{ns}] \quad (n=1 \text{ and } 3 \text{ when } 1 \text{ wait and } 2 \text{ wait, respectively})$$

$$t_{ac4}(\text{CAD-DB}) = \frac{10^9 \times l}{f(\text{BCLK})} - 35 \quad [\text{ns}] \quad (l=1 \text{ and } 2 \text{ when } 1 \text{ wait and } 2 \text{ wait, respectively})$$

V_{CC} = 5VTiming requirements (referenced to V_{CC} = 5V, V_{SS} = 0V at Topr = 25°C unless otherwise specified)

Table 1.32.8. Timer A input (count input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _c (TA)	TAiIN input cycle time	100		ns
t _w (TAH)	TAiIN input HIGH pulse width	40		ns
t _w (TAL)	TAiIN input LOW pulse width	40		ns

Table 1.32.9. Timer A input (gating input in timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _c (TA)	TAiIN input cycle time	400		ns
t _w (TAH)	TAiIN input HIGH pulse width	200		ns
t _w (TAL)	TAiIN input LOW pulse width	200		ns

Table 1.32.10. Timer A input (external trigger input in one-shot timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _c (TA)	TAiIN input cycle time	200		ns
t _w (TAH)	TAiIN input HIGH pulse width	100		ns
t _w (TAL)	TAiIN input LOW pulse width	100		ns

Table 1.32.11. 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	100		ns
t _w (TAL)	TAiIN input LOW pulse width	100		ns

Table 1.32.12. Timer A input (up/down input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _c (UP)	TAiOUT input cycle time	2000		ns
t _w (UPH)	TAiOUT input HIGH pulse width	1000		ns
t _w (UPL)	TAiOUT input LOW pulse width	1000		ns
t _{su} (UP-TIN)	TAiOUT input setup time	400		ns
t _h (TIN-UP)	TAiOUT input hold time	400		ns

V_{CC} = 5V

Timing requirements (referenced to V_{CC} = 5V, V_{SS} = 0V at Topr = 25°C unless otherwise specified)

Table 1.32.13. Timer B input (count 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 1.32.15. 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 1.32.16. A-D trigger input

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _c (AD)	ADTRG input cycle time (trigger able minimum)	1000		ns
t _w (ADL)	ADTRG input LOW pulse width	125		ns

Table 1.32.17. 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 1.32.18. External interrupt INTi inputs

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _w (INH)	INTi input HIGH pulse width	250		ns
t _w (INL)	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 1.32.19. Memory expansion mode and microprocessor mode (no wait)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address output delay time	Figure 1.32.1		18	ns
th(BCLK-AD)	Address output hold time (BCLK standard)		-3		ns
th(RD-AD)	Address output hold time (RD standard)		0		ns
th(WR-AD)	Address output hold time (WR standard)		(Note)		ns
td(BCLK-CS)	Chip select output delay time			18	ns
th(BCLK-CS)	Chip select output hold time (BCLK standard)		-3		ns
th(RD-CS)	Chip select output hold time (RD standard)		0		ns
th(WR-CS)	Chip select output hold time (WR standard)		(Note)		ns
td(BCLK-ALE)	ALE signal output delay time			18	ns
th(BCLK-ALE)	ALE signal output hold time		-2		ns
td(BCLK-RD)	RD signal output delay time			18	ns
th(BCLK-RD)	RD signal output hold time		-5		ns
td(BCLK-WR)	WR signal output delay time			18	ns
th(BCLK-WR)	WR signal output hold time		-3		ns
td(DB-WR)	Data output delay time (WR standard)		(Note)		ns
th(WR-DB)	Data output hold time (WR standard)		(Note)		ns
tw(WR)	WR signal width		(Note)		ns

Note: Calculated according to the BCLK frequency as follows:

$$td(DB - WR) = \frac{10^9}{f(BCLK)} - 20 \quad [ns]$$

$$th(WR - DB) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$th(WR - AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$th(WR - CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$tw(WR) = \frac{10^9}{f(BCLK) \times 2} - 15 \quad [ns]$$

Vcc = 5V

Switching characteristics (referenced to Vcc = 5V, Vss = 0V at Topr = 25°C unless otherwise specified)

Table 1.32.20. Memory expansion mode and microprocessor mode (with wait, accessing external memory)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address output delay time	Figure 1.32.1		18	ns
th(BCLK-AD)	Address output hold time (BCLK standard)		- 3		ns
th(RD-AD)	Address output hold time (RD standard)		0		ns
th(WR-AD)	Address output hold time (WR standard)		(Note)		ns
td(BCLK-CS)	Chip select output delay time			18	ns
th(BCLK-CS)	Chip select output hold time (BCLK standard)		- 3		ns
th(RD-CS)	Chip select output hold time (RD standard)		0		ns
th(WR-CS)	Chip select output hold time (WR standard)		(Note)		ns
td(BCLK-ALE)	ALE signal output delay time			18	ns
th(BCLK-ALE)	ALE signal output hold time		- 2		ns
td(BCLK-RD)	RD signal output delay time			18	ns
th(BCLK-RD)	RD signal output hold time		- 5		ns
td(BCLK-WR)	WR signal output delay time			18	ns
th(BCLK-WR)	WR signal output hold time		- 3		ns
td(DB-WR)	Data output delay time (WR standard)		(Note)		ns
th(WR-DB)	Data output hold time (WR standard)		(Note)		ns
tw(WR)	WR signal width		(Note)		ns

Note: Calculated according to the BCLK frequency as follows:

$$td(DB - WR) = \frac{10^9 \times n}{f_{(BCLK)}} - 20 \quad [ns] \quad (n=1, 2 \text{ and } 3 \text{ when } 1 \text{ wait, } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$th(WR - DB) = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [ns]$$

$$th(WR - AD) = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [ns]$$

$$th(WR - CS) = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [ns]$$

$$tw(WR) = \frac{10^9 \times n}{f_{(BCLK)} \times 2} - 15 \quad [ns] \quad (n=1, 3 \text{ and } 5 \text{ when } 1 \text{ wait, } 2 \text{ wait and } 3 \text{ wait, respectively})$$

V_{CC} = 5V

Switching characteristics (referenced to V_{CC} = 5V, V_{SS} = 0V at Topr = 25°C unless otherwise specified)

Table 1.32.21. Memory expansion mode and microprocessor mode
(with wait, accessing external memory, multiplex bus area selected)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address output delay time	Figure 1.32.1		18	ns
th(BCLK-AD)	Address output hold time (BCLK standard)		-3		ns
th(RD-AD)	Address output hold time (RD standard)		(Note)		ns
th(WR-AD)	Address output hold time (WR standard)		(Note)		ns
td(BCLK-CS)	Chip select output delay time			18	ns
th(BCLK-CS)	Chip select output hold time (BCLK standard)		-3		ns
th(RD-CS)	Chip select output hold time (RD standard)		(Note)		ns
th(WR-CS)	Chip select output hold time (WR standard)		(Note)		ns
td(BCLK-RD)	RD signal output delay time			18	ns
th(BCLK-RD)	RD signal output hold time		-5		ns
td(BCLK-WR)	WR signal output delay time			18	ns
th(BCLK-WR)	WR signal output hold time		-3		ns
td(DB-WR)	Data output delay time (WR standard)		(Note)		ns
th(WR-DB)	Data output hold time (WR standard)		(Note)		ns
td(BCLK-ALE)	ALE signal output delay time (BCLK standard)			18	ns
th(BCLK-ALE)	ALE signal output hold time (BCLK standard)		-2		ns
td(AD-ALE)	ALE signal output delay time (address standard)		(Note)		ns
th(ALE-AD)	ALE signal output hold time (address standard)		(Note)		ns
tdz(RD-AD)	Address output flowing start time			8	ns

Note: Calculated according to the BCLK frequency as follows:

$$th(RD - AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$th(WR - AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$th(RD - CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$th(WR - CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$td(DB - WR) = \frac{10^9 \times m}{f(BCLK) \times 2} - 25 \quad [ns] \quad (m=3 \text{ and } 5 \text{ when } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$th(WR - DB) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$td(AD - ALE) = \frac{10^9}{f(BCLK) \times 2} - 20 \quad [ns]$$

$$th(ALE - AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

VCC = 5V

Switching characteristics (referenced to Vcc = 5V, Vss = 0V at Topr = 25°C unless otherwise specified)

Table 1.32.22. Memory expansion mode and microprocessor mode
(with wait, accessing external memory, DRAM area selected)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
td(BCLK-RAD)	Row address output delay time	Figure 1.32.1		18	ns
th(BCLK-RAD)	Row address output hold time (BCLK standard)		-3		ns
td(BCLK-CAD)	String address output delay time			18	ns
th(BCLK-CAD)	String address output hold time (BCLK standard)		-3		ns
th(RAS-RAD)	Row address output hold time after RAS output		(Note)		ns
td(BCLK-RAS)	RAS output delay time (BCLK standard)			18	ns
th(BCLK-RAS)	RAS output hold time (BCLK standard)		-3		ns
trp	RAS "H" hold time		(Note)		ns
td(BCLK-CAS)	CAS output delay time (BCLK standard)			18	ns
th(BCLK-CAS)	CAS output hold time (BCLK standard)		-3		ns
td(BCLK-DW)	DW output delay time (BCLK standard)			18	ns
th(BCLK-DW)	DW output hold time (BCLK standard)		-5		ns
tsu(DB-CAS)	CAS output setup time after DB output		(Note)		ns
th(BCLK-DB)	DB signal output hold time (BCLK standard)		-7		ns
tsu(CAS-RAS)	CAS output setup time before RAS output (refresh)		(Note)		ns

Note: Calculated according to the BCLK frequency as follows:

$$th(RAS - RAD) = \frac{10^9}{f(BCLK) \times 2} - 13 \quad [ns]$$

$$trp = \frac{10^9}{f(BCLK) \times 2} \times 3 - 20 \quad [ns]$$

$$tsu(DB - CAS) = \frac{10^9}{f(BCLK)} - 20 \quad [ns]$$

$$tsu(CAS - RAS) = \frac{10^9}{f(BCLK) \times 2} - 13 \quad [ns]$$

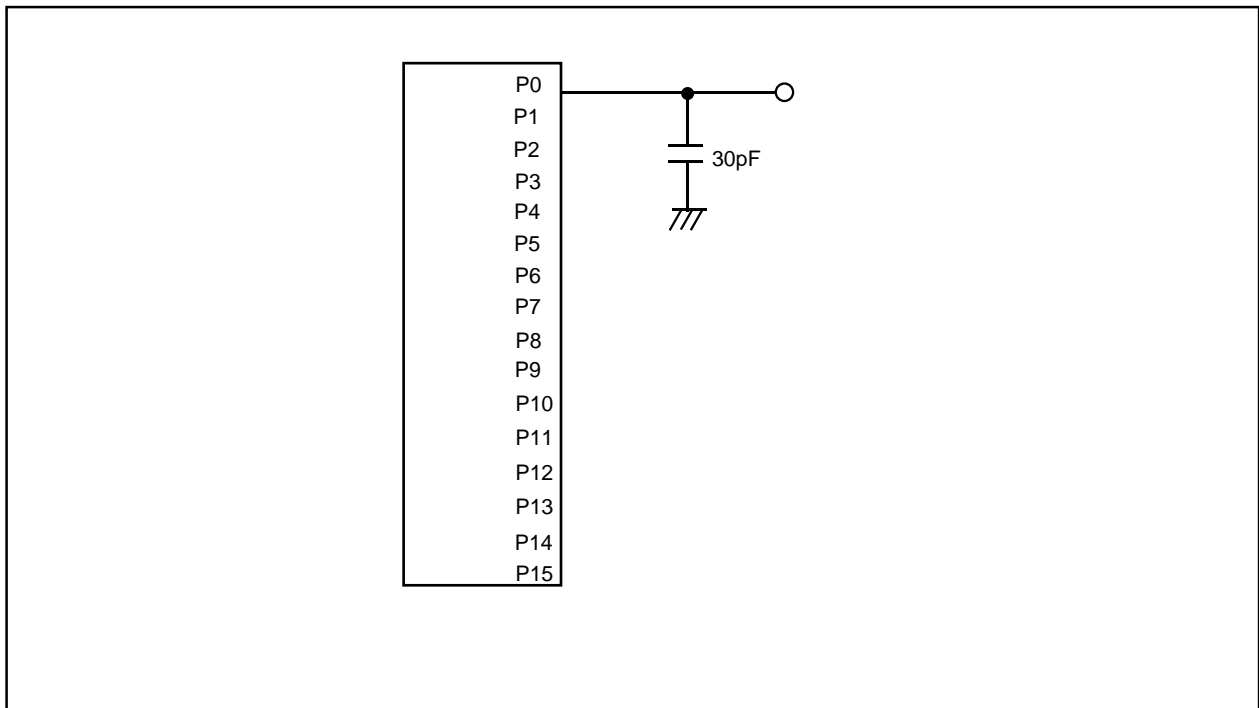


Figure 1.32.1. Port P0 to P15 measurement circuit

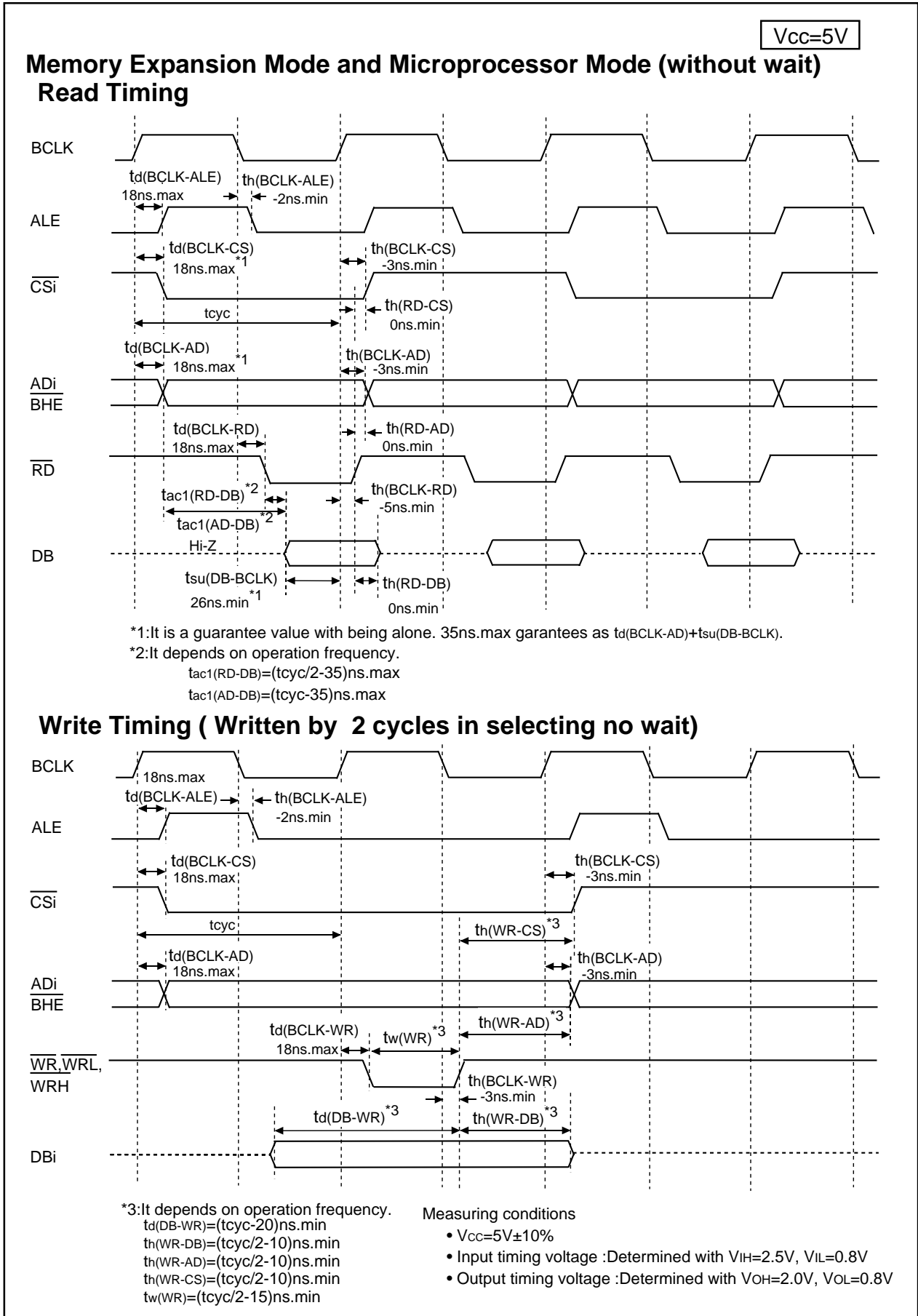


Figure 1.32.2. Vcc=5V timing diagram (1)

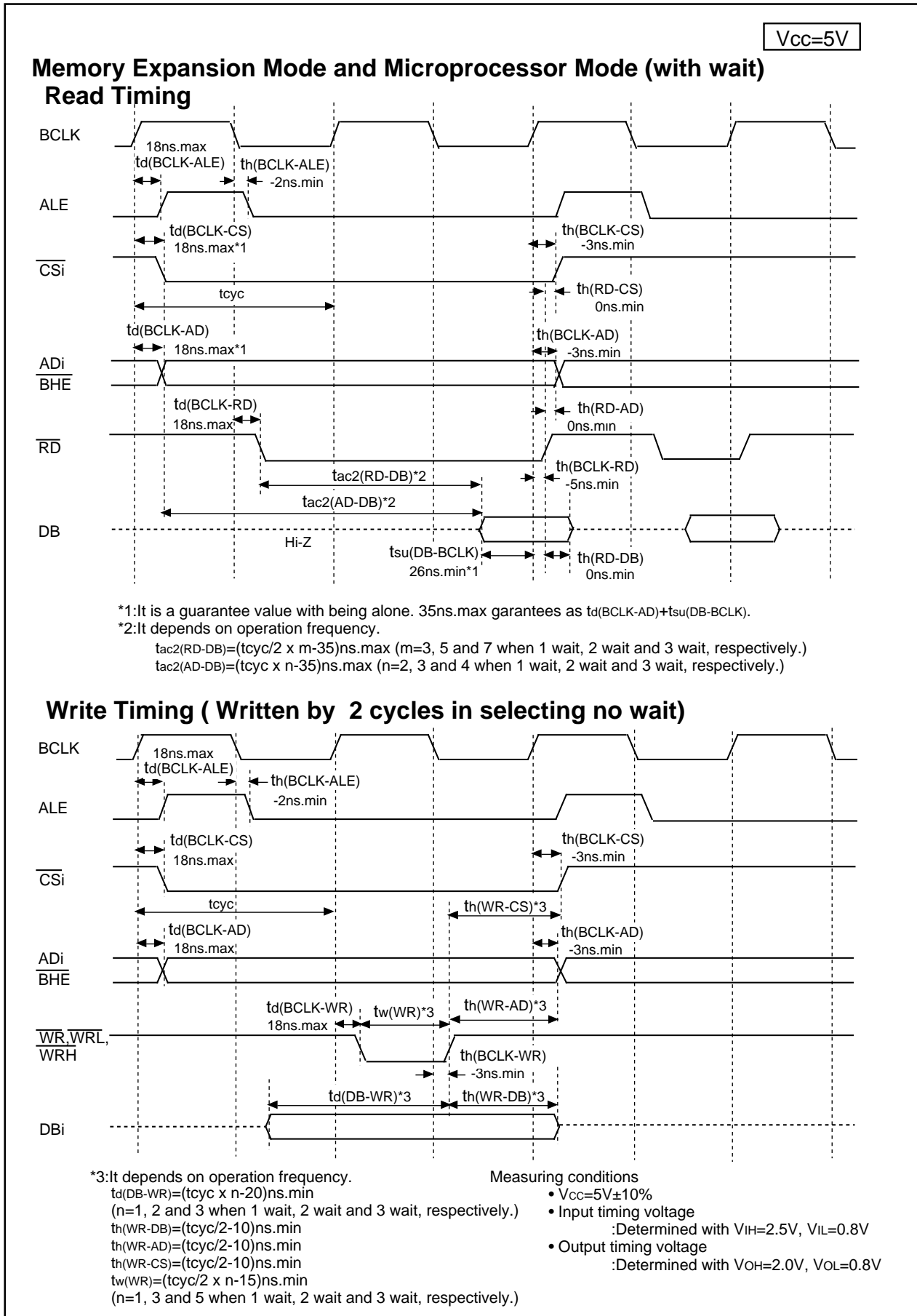


Figure 1.32.3. Vcc=5V timing diagram (2)

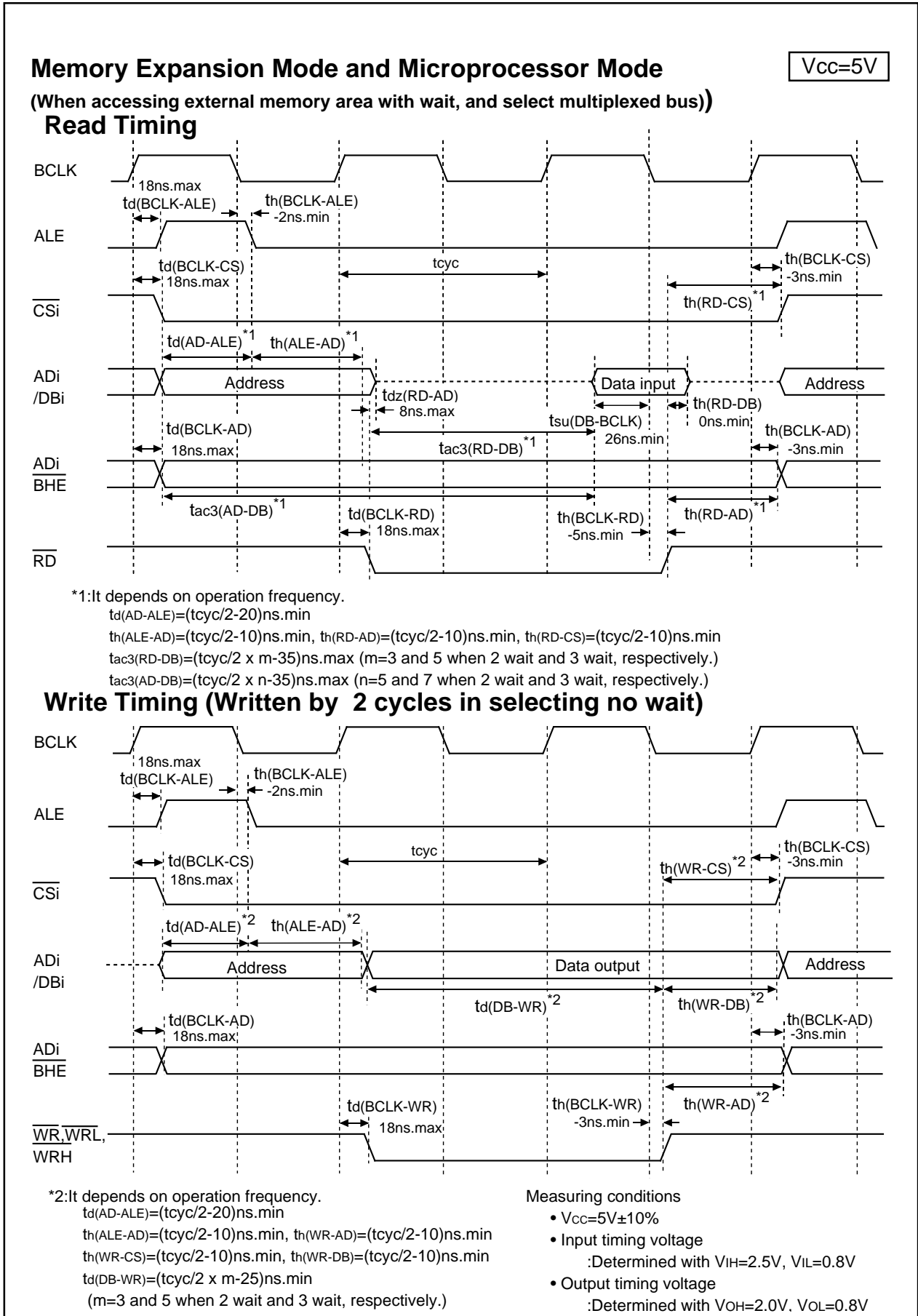


Figure 1.32.4. Vcc=5V timing diagram (3)

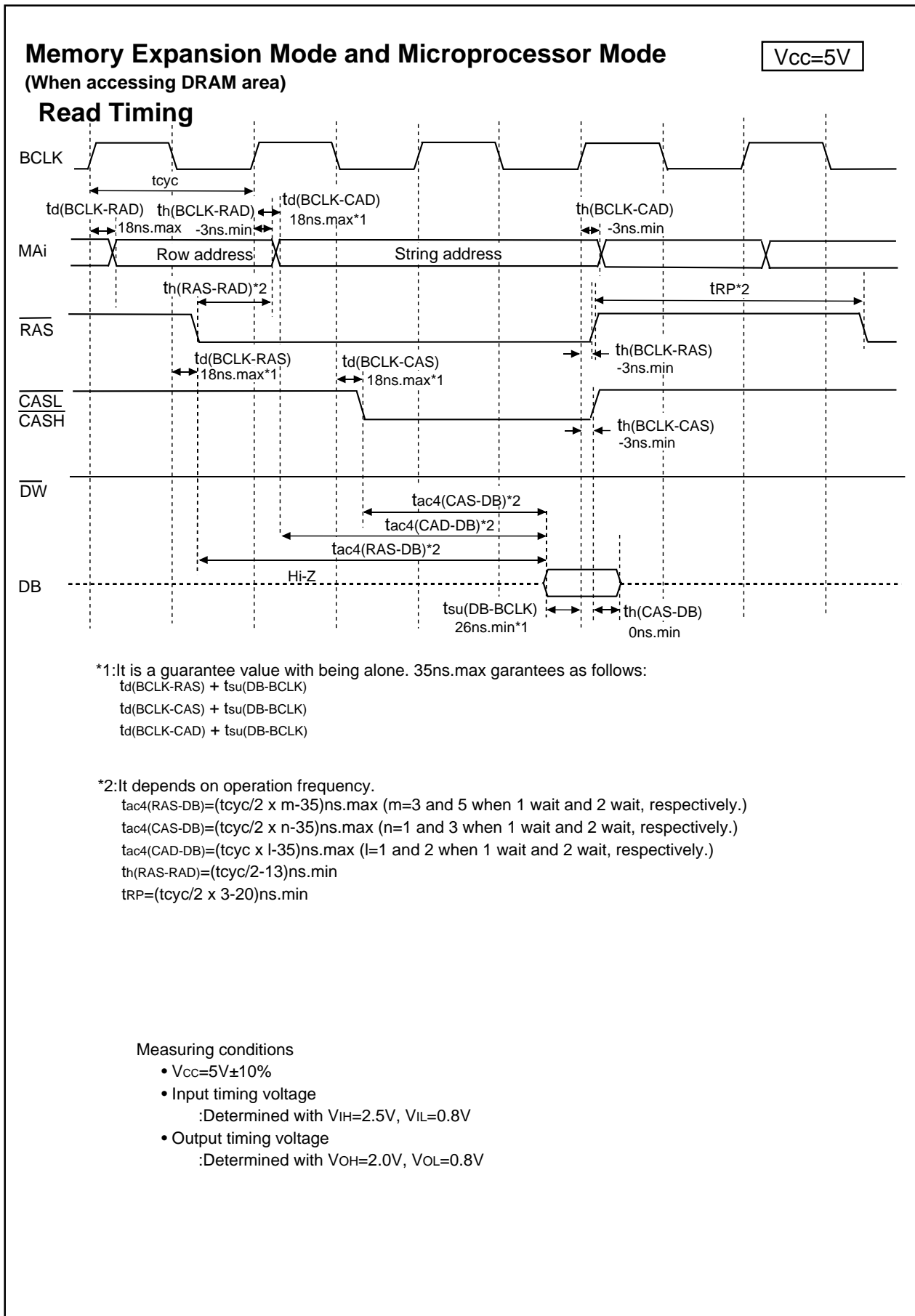


Figure 1.32.5. Vcc=5V timing diagram (4)

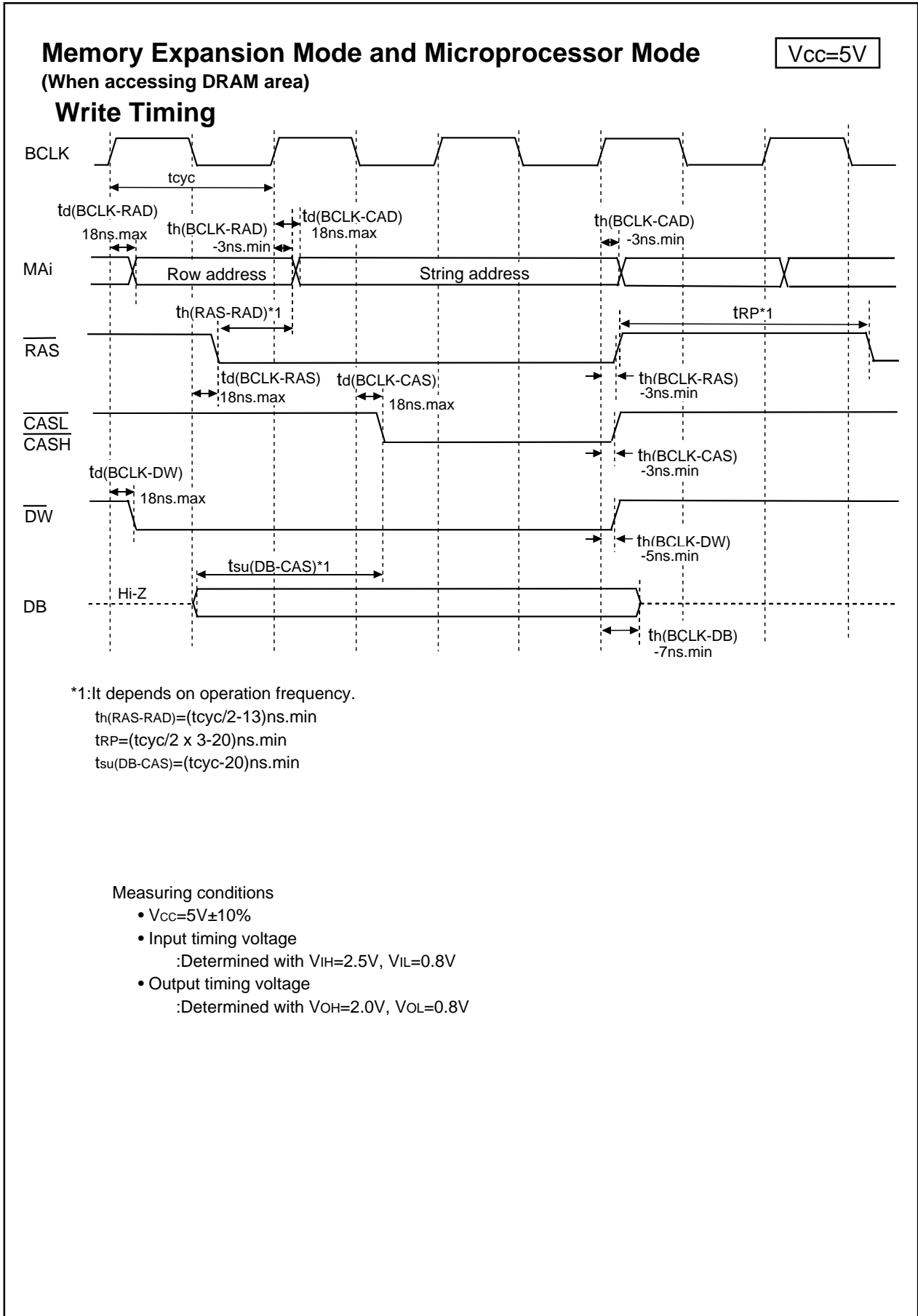


Figure 1.32.6. V_{CC}=5V timing diagram (5)

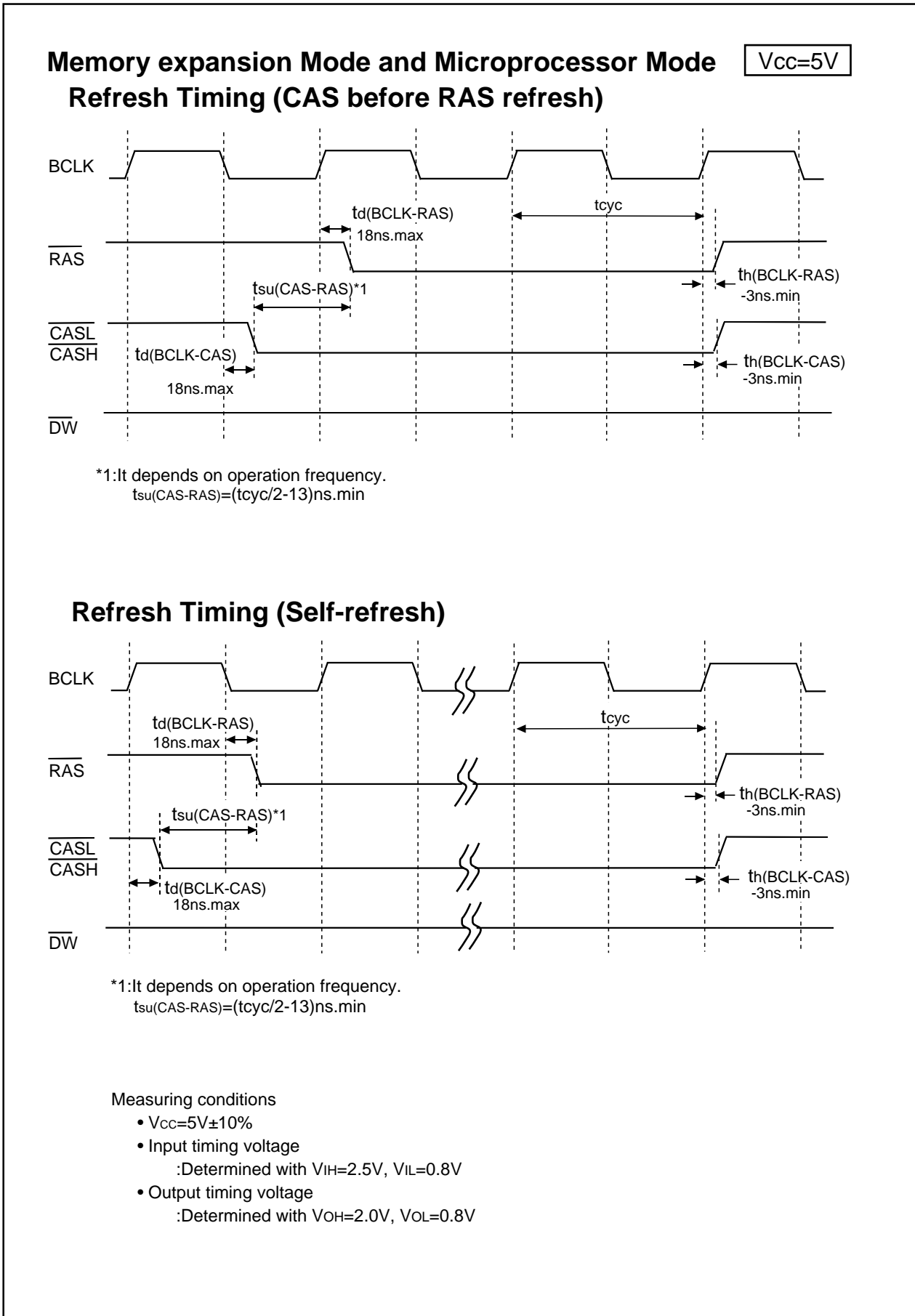


Figure 1.32.7. V_{CC}=5V timing diagram (6)

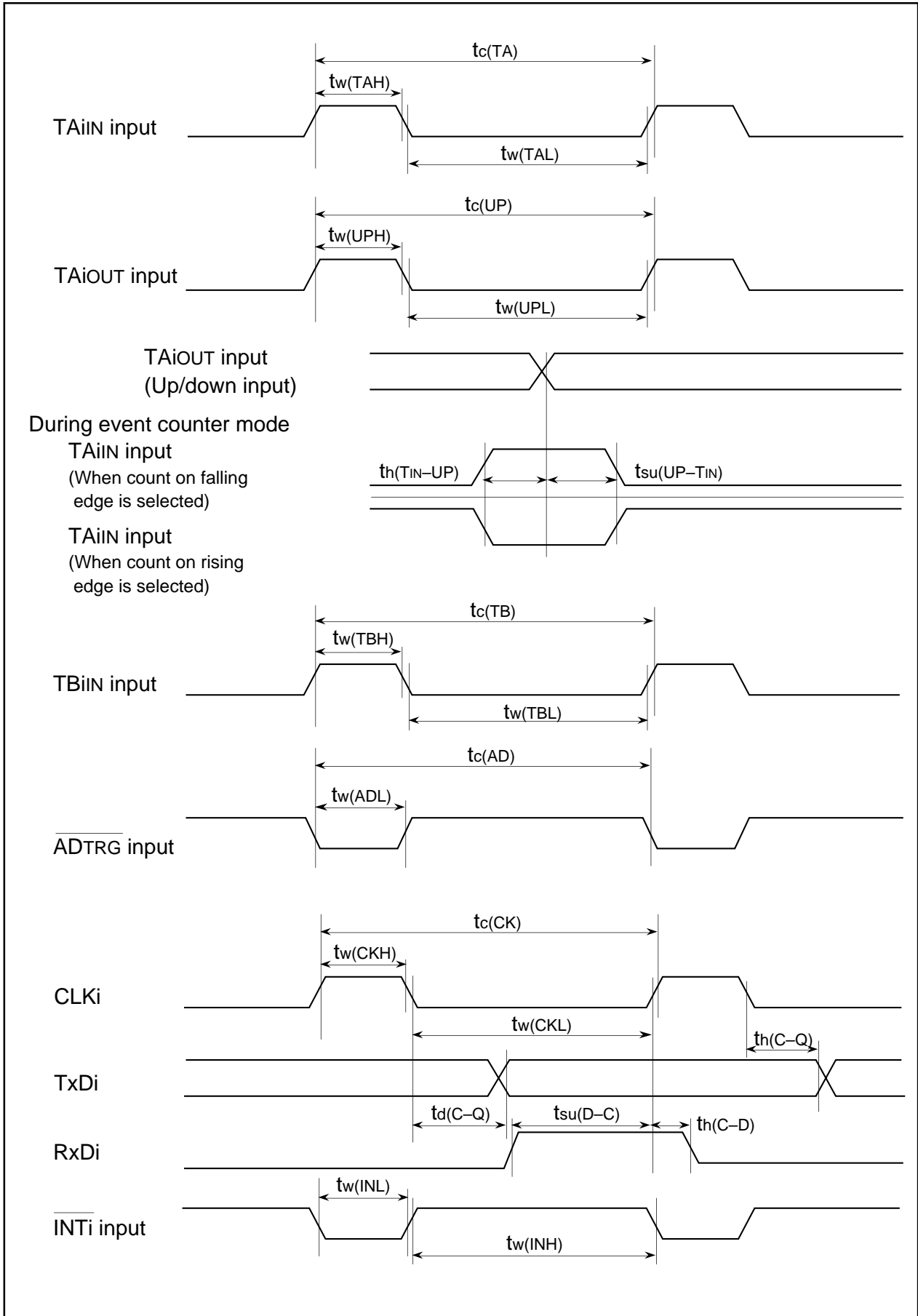
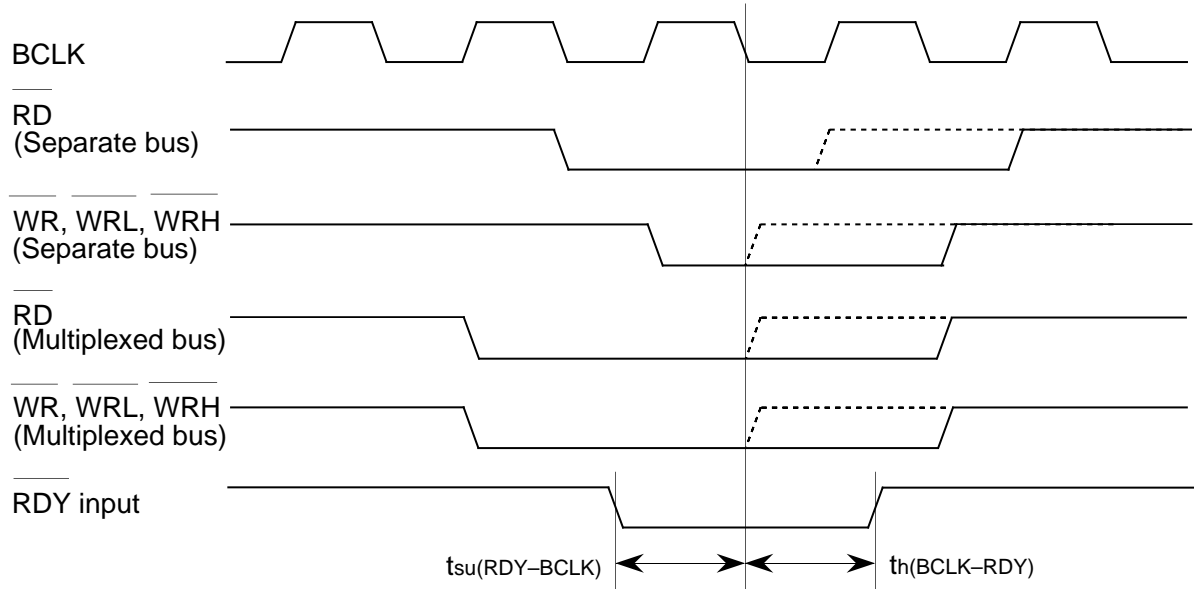
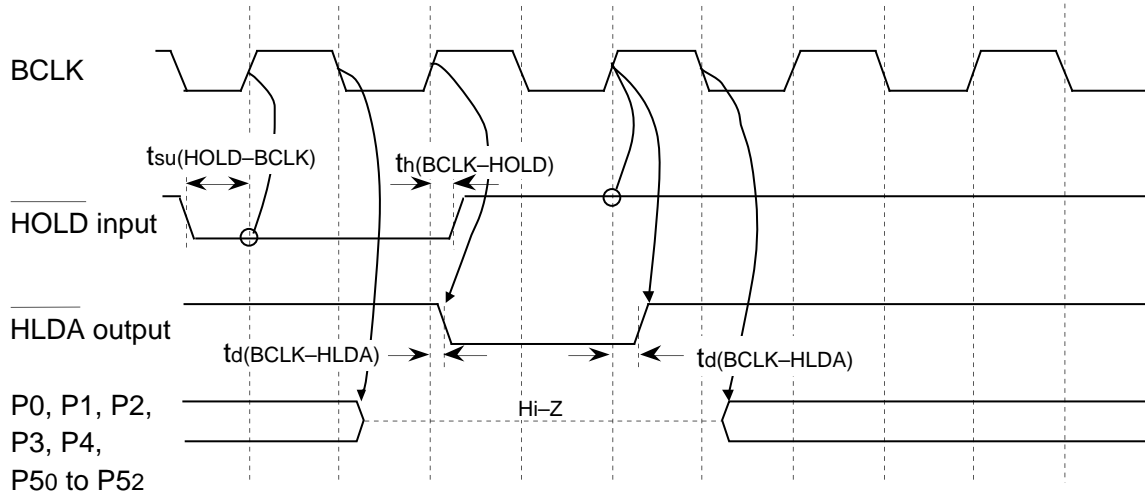


Figure 1.32.8. Vcc=5V timing diagram (7)

Memory Expansion Mode and Microprocessor Mode
(Valid only with wait)



(Valid with or without wait)



Note: Regardless of the level of the BYTE pin input and the setting of the port P40 to P43 function select bit (PM06) of the processor mode register 0, all ports above become the high-impedance state.

Measuring conditions :

- VCC=5V±10%
- Input timing voltage : Determined with $V_{IH}=4.0V$, $V_{IL}=1.0V$
- Output timing voltage : Determined with $V_{OH}=2.5V$, $V_{OL}=2.5V$

Figure 1.32.9. Vcc=5V timing diagram (8)

Electrical characteristics (V_{CC} = 3V)Table 1.32.23. Electrical characteristics (referenced to V_{CC}=3.3V, V_{SS}=0V at
T_{opr}=25°C, f(X_{IN})=20MHz unless otherwise specified)V_{CC} = 3V

Symbol	Parameter		Condition	Standard			Unit
				Min.	Typ.	Max.	
V _{OH}	"H" output voltage	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157(Note1)	I _{OH} =-1mA	2.7			V
V _{OH}	"H" output voltage	XOUT	HIGH POWER	I _{OH} =-0.1mA	2.7		V
			LOW POWER	I _{OH} =-50μA	2.7		V
	"H" output voltage	XCOUT	No load applied		3.0		V
V _{OL}	"L" output voltage	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157(Note1)	I _{OL} =1mA			0.5	
V _{OL}	"L" output voltage	XOUT	HIGH POWER	I _{OL} =0.1mA		0.5	V
			LOW POWER	I _{OL} =50μA		0.5	V
	"L" output voltage	XCOUT	No load applied		0		V
V _{T+} -V _{T-}	Hysteresis	HOLD, RDY, TA0IN-TA4IN, TB0IN-TB5IN, INT0-INT5, ADTRG, CTS0-CTS4, CLK0-CLK4, TA0OUT-TA4OUT, NMI, KI0-KI3, RxD0-RxD4, SCL0-SCL4, SDA0-SDA4		0.2		1.0	V
V _{T+} -V _{T-}	Hysteresis	RESET		0.2		1.8	V
I _{IH}	"H" input current	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157(Note1), XIN, RESET, CNV _{SS} , BYTE	V _I =3V			4.0	μA
I _{IL}	"L" input current	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157(Note1), XIN, RESET, CNV _{SS} , BYTE	V _I =0V			-4.0	μA
R _{PULLUP}	Pull-up resistance	P00-P07, P10-P17, P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P84, P86, P87, P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157(Note1)	V _I =0V	66	120	500	kΩ
R _{fXIN}	Feedback resistance	XIN			3.0		MΩ
R _{fXCIN}	Feedback resistance	XCIN			20.0		MΩ
V _{RAM}	RAM retention voltage	VDC-ON		2.5			V
		VDC-pass through		2.0			V
I _{CC}	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		26	38	mA
			f(X _{CIN})=32kHz, with WAIT, VDC-pass through		5.0		μA
			f(X _{CIN})=32kHz, with WAIT, VDC-ON		340		μA
			when clock is stopped T _{opr} =25°C		0.4	20	μA

Note 1: Port P11 to P15 exist in 144-pin version.

$V_{CC} = 3V$

Table 1.32.24. A-D conversion characteristics (referenced to $V_{CC} = AV_{CC} = V_{REF} = 3V$, $V_{SS} = AV_{SS} = 0V$ at $T_{opr} = 25^{\circ}C$, $f(X_{IN}) = 20MHz$ unless otherwise specified)

Symbol	Parameter		Measuring condition	Standard			Unit
				Min.	Typ.	Max	
-	Resolution		$V_{REF} = V_{CC}$			10	Bits
ISL	Integral nonlinearity error	No S&H function(8-bit)				± 2	LSB
DSL	Differential nonlinearity error	No S&H function(8-bit)				± 1	LSB
-	Offset error	No S&H function(8-bit)				± 2	LSB
-	Gain error	No S&H function(8-bit)				± 2	LSB
RLADDER	Ladder resistance		$V_{REF} = V_{CC}$	10		40	k Ω
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

S&H: Sample and hold

Note: Divide the frequency if $f(X_{IN})$ exceeds 10 MHz, and make $\emptyset AD$ equal to or lower than 10 MHz.

Table 1.32.25. D-A conversion characteristics (referenced to $V_{CC} = V_{REF} = 3V$, $V_{SS} = AV_{SS} = 0V$, at $T_{opr} = 25^{\circ}C$, $f(X_{IN}) = 20MHz$ 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	k Ω
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 "0016". The A-D converter's ladder resistance is not included.

Also, the Vref is unconnected at the A-D control register 1, IVREF is sent.

V_{CC} = 3VTiming requirements (referenced to V_{CC} = 3V, V_{SS} = 0V at Topr = 25°C unless otherwise specified)

Table 1.32.26. 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 1.32.27. 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	30		ns
t _{su} (RDY-BCLK)	RDY input setup time	40		ns
t _{su} (HOLD-BCLK)	HOLD input setup time	60		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)	RDY input hold time	0		ns
t _h (BCLK-HOLD)	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.

$$t_{ac1}(\text{RD} - \text{DB}) = \frac{10^9}{f(\text{BCLK}) \times 2} - 35 \quad [\text{ns}]$$

$$t_{ac1}(\text{AD} - \text{DB}) = \frac{10^9}{f(\text{BCLK})} - 35 \quad [\text{ns}]$$

$$t_{ac2}(\text{RD} - \text{DB}) = \frac{10^9 \times m}{f(\text{BCLK}) \times 2} - 35 \quad [\text{ns}] \quad (m=3, 5 \text{ and } 7 \text{ when } 1 \text{ wait, } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$t_{ac2}(\text{AD} - \text{DB}) = \frac{10^9 \times n}{f(\text{BCLK})} - 35 \quad [\text{ns}] \quad (n=2, 3 \text{ and } 4 \text{ when } 1 \text{ wait, } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$t_{ac3}(\text{RD} - \text{DB}) = \frac{10^9 \times m}{f(\text{BCLK}) \times 2} - 35 \quad [\text{ns}] \quad (m=3 \text{ and } 5 \text{ when } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$t_{ac3}(\text{AD} - \text{DB}) = \frac{10^9 \times n}{f(\text{BCLK}) \times 2} - 35 \quad [\text{ns}] \quad (n=5 \text{ and } 7 \text{ when } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$t_{ac4}(\text{RAS} - \text{DB}) = \frac{10^9 \times m}{f(\text{BCLK}) \times 2} - 35 \quad [\text{ns}] \quad (m=3 \text{ and } 5 \text{ when } 1 \text{ wait and } 2 \text{ wait, respectively})$$

$$t_{ac4}(\text{CAS} - \text{DB}) = \frac{10^9 \times n}{f(\text{BCLK}) \times 2} - 35 \quad [\text{ns}] \quad (n=1 \text{ and } 3 \text{ when } 1 \text{ wait and } 2 \text{ wait, respectively})$$

$$t_{ac4}(\text{CAD} - \text{DB}) = \frac{10^9 \times l}{f(\text{BCLK})} - 35 \quad [\text{ns}] \quad (l=1 \text{ and } 2 \text{ when } 1 \text{ wait and } 2 \text{ wait, respectively})$$

V_{CC} = 3V

Timing requirements (referenced to V_{CC} = 3V, V_{SS} = 0V at Topr = 25°C unless otherwise specified)

Table 1.32.28. Timer A input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _c (TA)	TAiIN input cycle time	100		ns
t _w (TAH)	TAiIN input HIGH pulse width	40		ns
t _w (TAL)	TAiIN input LOW pulse width	40		ns

Table 1.32.29. Timer A input (gating input in timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _c (TA)	TAiIN input cycle time	400		ns
t _w (TAH)	TAiIN input HIGH pulse width	200		ns
t _w (TAL)	TAiIN input LOW pulse width	200		ns

Table 1.32.30. Timer A input (external trigger input in one-shot timer mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _c (TA)	TAiIN input cycle time	200		ns
t _w (TAH)	TAiIN input HIGH pulse width	100		ns
t _w (TAL)	TAiIN input LOW pulse width	100		ns

Table 1.32.31. 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	100		ns
t _w (TAL)	TAiIN input LOW pulse width	100		ns

Table 1.32.32. Timer A input (up/down input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _c (UP)	TAiOUT input cycle time	2000		ns
t _w (UPH)	TAiOUT input HIGH pulse width	1000		ns
t _w (UPL)	TAiOUT input LOW pulse width	1000		ns
t _{su} (UP-TiN)	TAiOUT input setup time	400		ns
t _h (TiN-UP)	TAiOUT input hold time	400		ns

V_{CC} = 3VTiming requirements (referenced to V_{CC} = 3V, V_{SS} = 0V at Topr = 25°C unless otherwise specified)

Table 1.32.33. Timer B input (counter input in event counter mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _c (TB)	TB _{iIN} input cycle time (counted on one edge)	100		ns
t _w (TBH)	TB _{iIN} input HIGH pulse width (counted on one edge)	40		ns
t _w (TBL)	TB _{iIN} input LOW pulse width (counted on one edge)	40		ns
t _c (TB)	TB _{iIN} input cycle time (counted on both edges)	200		ns
t _w (TBH)	TB _{iIN} input HIGH pulse width (counted on both edges)	80		ns
t _w (TBL)	TB _{iIN} input LOW pulse width (counted on both edges)	80		ns

Table 1.32.34. Timer B input (pulse period measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _c (TB)	TB _{iIN} input cycle time	400		ns
t _w (TBH)	TB _{iIN} input HIGH pulse width	200		ns
t _w (TBL)	TB _{iIN} input LOW pulse width	200		ns

Table 1.32.35. Timer B input (pulse width measurement mode)

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _c (TB)	TB _{iIN} input cycle time	400		ns
t _w (TBH)	TB _{iIN} input HIGH pulse width	200		ns
t _w (TBL)	TB _{iIN} input LOW pulse width	200		ns

Table 1.32.36. 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 1.32.37. Serial I/O

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _c (CK)	CLK _i input cycle time	200		ns
t _w (CKH)	CLK _i input HIGH pulse width	100		ns
t _w (CKL)	CLK _i input LOW pulse width	100		ns
t _d (C-Q)	TxD _i output delay time		80	ns
t _h (C-Q)	TxD _i hold time	0		ns
t _{su} (D-C)	RxD _i input setup time	30		ns
t _h (C-D)	RxD _i input hold time	90		ns

Table 1.32.38. External interrupt $\overline{\text{INT}}_i$ inputs

Symbol	Parameter	Standard		Unit
		Min.	Max.	
t _w (INH)	$\overline{\text{INT}}_i$ input HIGH pulse width	250		ns
t _w (INL)	$\overline{\text{INT}}_i$ input LOW pulse width	250		ns

Vcc = 3V

Switching characteristics (referenced to Vcc = 3V, Vss = 0V at Topr = 25°C, CM15="1" unless otherwise specified)

Table 1.32.39. Memory expansion and microprocessor modes (with no wait)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address output delay time	Figure 1.32.1		18	ns
th(BCLK-AD)	Address output hold time (BCLK standard)		0		ns
th(RD-AD)	Address output hold time (RD standard)		0		ns
th(WR-AD)	Address output hold time (WR standard)		(Note)		ns
td(BCLK-CS)	Chip select output delay time			18	ns
th(BCLK-CS)	Chip select output hold time (BCLK standard)		0		ns
th(RD-CS)	Chip select output hold time (RD standard)		0		ns
th(WR-CS)	Chip select output hold time (WR standard)		(Note)		ns
td(BCLK-ALE)	ALE signal output delay time			18	ns
th(BCLK-ALE)	ALE signal output hold time		- 2		ns
td(BCLK-RD)	RD signal output delay time			18	ns
th(BCLK-RD)	RD signal output hold time		- 3		ns
td(BCLK-WR)	WR signal output delay time			18	ns
th(BCLK-WR)	WR signal output hold time		0		ns
td(DB-WR)	Data output delay time (WR standard)		(Note)		ns
th(WR-DB)	Data output hold time (WR standard)		(Note)		ns
tw(WR)	Write pulse width		(Note)		ns

Note: Calculated according to the BCLK frequency as follows:

$$td(DB - WR) = \frac{10^9}{f(BCLK)} - 20 \quad [ns]$$

$$th(WR - DB) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$th(WR - AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$th(WR - CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$tw(WR) = \frac{10^9}{f(BCLK) \times 2} - 15 \quad [ns]$$

V_{CC} = 3V

Switching characteristics (referenced to V_{cc} = 3V, V_{ss} = 0V at Topr = 25°C unless otherwise specified)

**Table 1.32.40. Memory expansion and microprocessor modes
(with wait, accessing external memory)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address output delay time	Figure 1.32.1		18	ns
th(BCLK-AD)	Address output hold time (BCLK standard)		0		ns
th(RD-AD)	Address output hold time (RD standard)		0		ns
th(WR-AD)	Address output hold time (WR standard)		(Note)		ns
td(BCLK-CS)	Chip select output delay time			18	ns
th(BCLK-CS)	Chip select output hold time (BCLK standard)		0		ns
th(RD-CS)	Chip select output hold time (RD standard)		0		ns
th(WR-CS)	Chip select output hold time (WR standard)		(Note)		ns
td(BCLK-ALE)	ALE signal output delay time			18	ns
th(BCLK-ALE)	ALE signal output hold time		- 2		ns
td(BCLK-RD)	RD signal output delay time			18	ns
th(BCLK-RD)	RD signal output hold time		- 3		ns
td(BCLK-WR)	WR signal output delay time			18	ns
th(BCLK-WR)	WR signal output hold time		0		ns
td(DB-WR)	Data output delay time (WR standard)		(Note)		ns
th(WR-DB)	Data output hold time (WR standard)		(Note)		ns
tw(WR)	Write pulse width		(Note)		ns

Note: Calculated according to the BCLK frequency as follows:

$$td(DB - WR) = \frac{10^9 \times n}{f_{(BCLK)}} - 20 \quad [\text{ns}] \quad (n=1, 2 \text{ and } 3 \text{ when } 1 \text{ wait, } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$th(WR - DB) = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$th(WR - AD) = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$th(WR - CS) = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$tw(WR) = \frac{10^9 \times n}{f_{(BCLK)} \times 2} - 15 \quad [\text{ns}] \quad (n=1, 3 \text{ and } 5 \text{ when } 1 \text{ wait, } 2 \text{ wait and } 3 \text{ wait, respectively})$$

V_{CC} = 3V

Switching characteristics (referenced to V_{CC} = 3V, V_{SS} = 0V at Topr = 25°C unless otherwise specified)

Table 1.32.41. Memory expansion and microprocessor modes
 (with wait, accessing external memory, multiplex bus area selected)

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
td(BCLK-AD)	Address output delay time	Figure 1.32.1		18	ns
th(BCLK-AD)	Address output hold time (BCLK standard)		0		ns
th(RD-AD)	Address output hold time (RD standard)		(Note)		ns
th(WR-AD)	Address output hold time (WR standard)		(Note)		ns
td(BCLK-CS)	Chip select output delay time			18	ns
th(BCLK-CS)	Chip select output hold time (BCLK standard)		0		ns
th(RD-CS)	Chip select output hold time (RD standard)		(Note)		ns
th(WR-CS)	Chip select output hold time (WR standard)		(Note)		ns
td(BCLK-RD)	RD signal output delay time			18	ns
th(BCLK-RD)	RD signal output hold time		- 3		ns
td(BCLK-WR)	WR signal output delay time			18	ns
th(BCLK-WR)	WR signal output hold time		0		ns
td(DB-WR)	Data output delay time (WR standard)		(Note)		ns
th(WR-DB)	Data output hold time (WR standard)		(Note)		ns
td(BCLK-ALE)	ALE signal output delay time (BCLK standard)			18	ns
th(BCLK-ALE)	ALE signal output hold time (BCLK standard)		- 2		ns
td(AD-ALE)	ALE signal output delay time (address standard)		(Note)		ns
th(ALE-AD)	ALE signal output hold time (address standard)		(Note)		ns
tdz(RD-AD)	Address output flowing start time			8	ns

Note: Calculated according to the BCLK frequency as follows:

$$th(RD - AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$th(WR - AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$th(RD - CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$th(WR - CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$td(DB - WR) = \frac{10^9 \times m}{f(BCLK) \times 2} - 25 \quad [ns] \quad (m=3 \text{ and } 5 \text{ when } 2 \text{ wait and } 3 \text{ wait, respectively})$$

$$th(WR - DB) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$td(AD - ALE) = \frac{10^9}{f(BCLK) \times 2} - 20 \quad [ns]$$

$$th(ALE - AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

VCC = 3V

Switching characteristics (referenced to Vcc = 3V, Vss = 0V at Topr = 25°C unless otherwise specified)

**Table 1.32.42. Memory expansion and microprocessor modes
(with wait, accessing external memory, DRAM area selected)**

Symbol	Parameter	Measuring condition	Standard		Unit
			Min.	Max.	
td(BCLK-RAD)	Row address output delay time	Figure 1.32.1		18	ns
th(BCLK-RAD)	Row address output hold time (BCLK standard)		0		ns
td(BCLK-CAD)	String address output delay time			18	ns
th(BCLK-CAD)	String address output hold time (BCLK standard)		0		ns
th(RAS-RAD)	Row address output hold time after RAS output		(Note)		ns
td(BCLK-RAS)	RAS output delay time (BCLK standard)			18	ns
th(BCLK-RAS)	RAS output hold time (BCLK standard)		0		ns
tRP	RAS "H" hold time		(Note)		ns
td(BCLK-CAS)	CAS output delay time (BCLK standard)			18	ns
th(BCLK-CAS)	CAS output hold time (BCLK standard)		0		ns
td(BCLK-DW)	Data output delay time (BCLK standard)			18	ns
th(BCLK-DW)	Data output hold time (BCLK standard)		- 3		ns
tsu(DB-CAS)	CAS after DB output setup time		(Note)		ns
th(BCLK-DB)	DB signal output hold time (BCLK standard)		- 7		ns
tsu(CAS-RAS)	CAS output setup time before RAS output (refresh)		(Note)		ns

Note: Calculated according to the BCLK frequency as follows:

$$th(RAS - RAD) = \frac{10^9}{f(BCLK) \times 2} - 13 \quad [ns]$$

$$tRP = \frac{10^9 \times 3}{f(BCLK) \times 2} - 20 \quad [ns]$$

$$tsu(DB - CAS) = \frac{10^9}{f(BCLK)} - 20 \quad [ns]$$

$$tsu(CAS - RAS) = \frac{10^9}{f(BCLK) \times 2} - 13 \quad [ns]$$

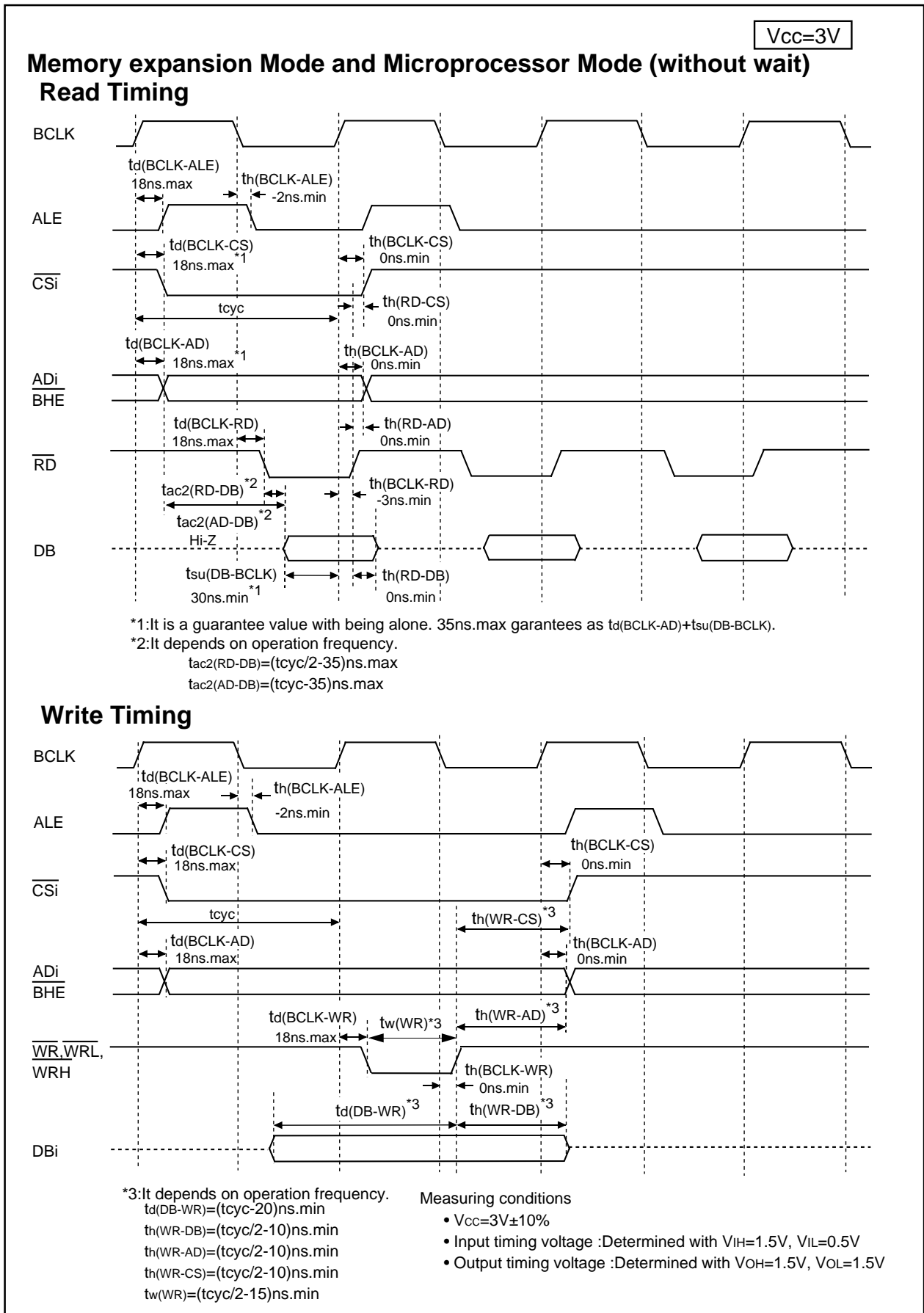


Figure 1.32.10. V_{CC}=3V timing diagram (1)

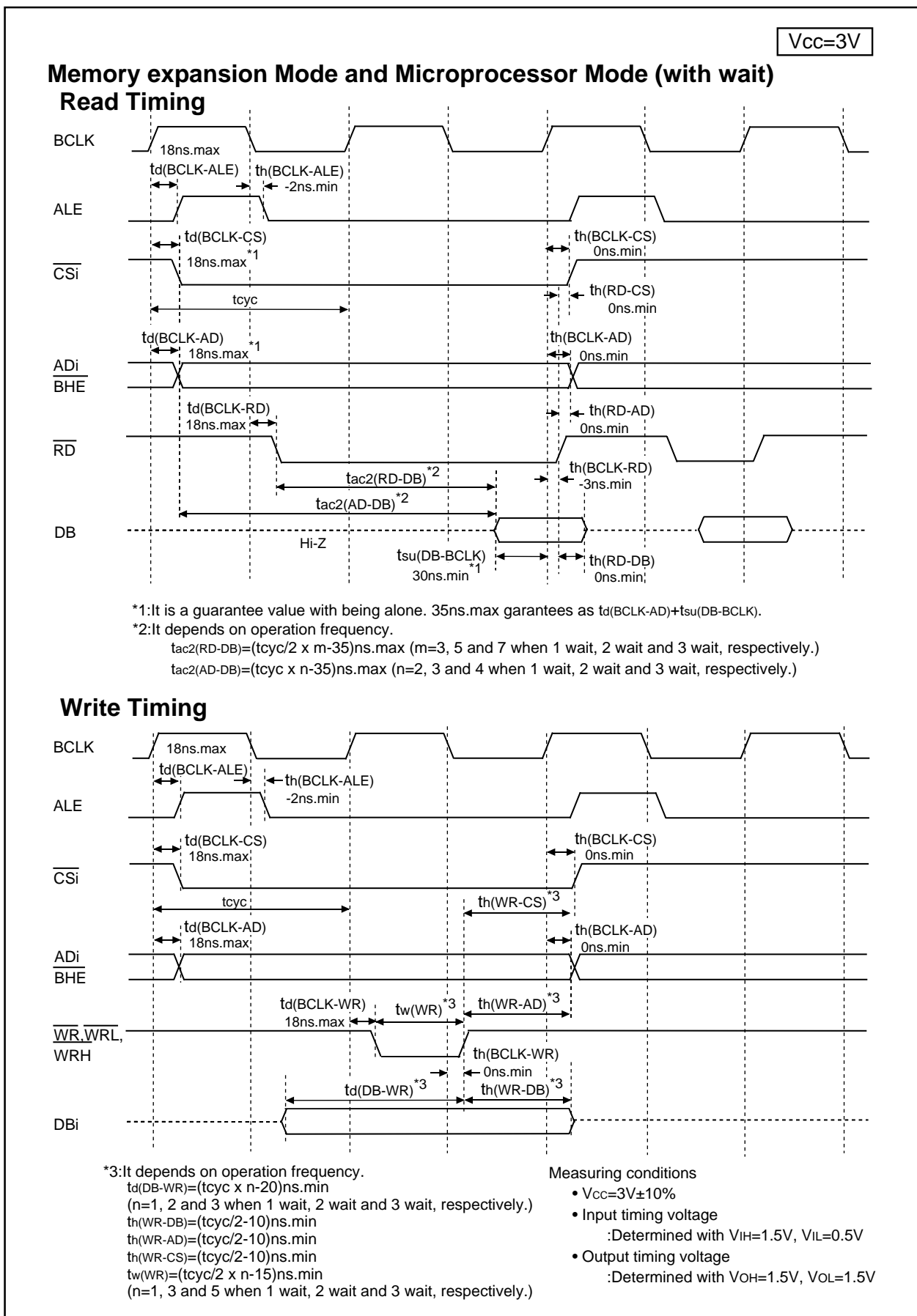


Figure 1.32.11. Vcc=3V timing diagram (2)

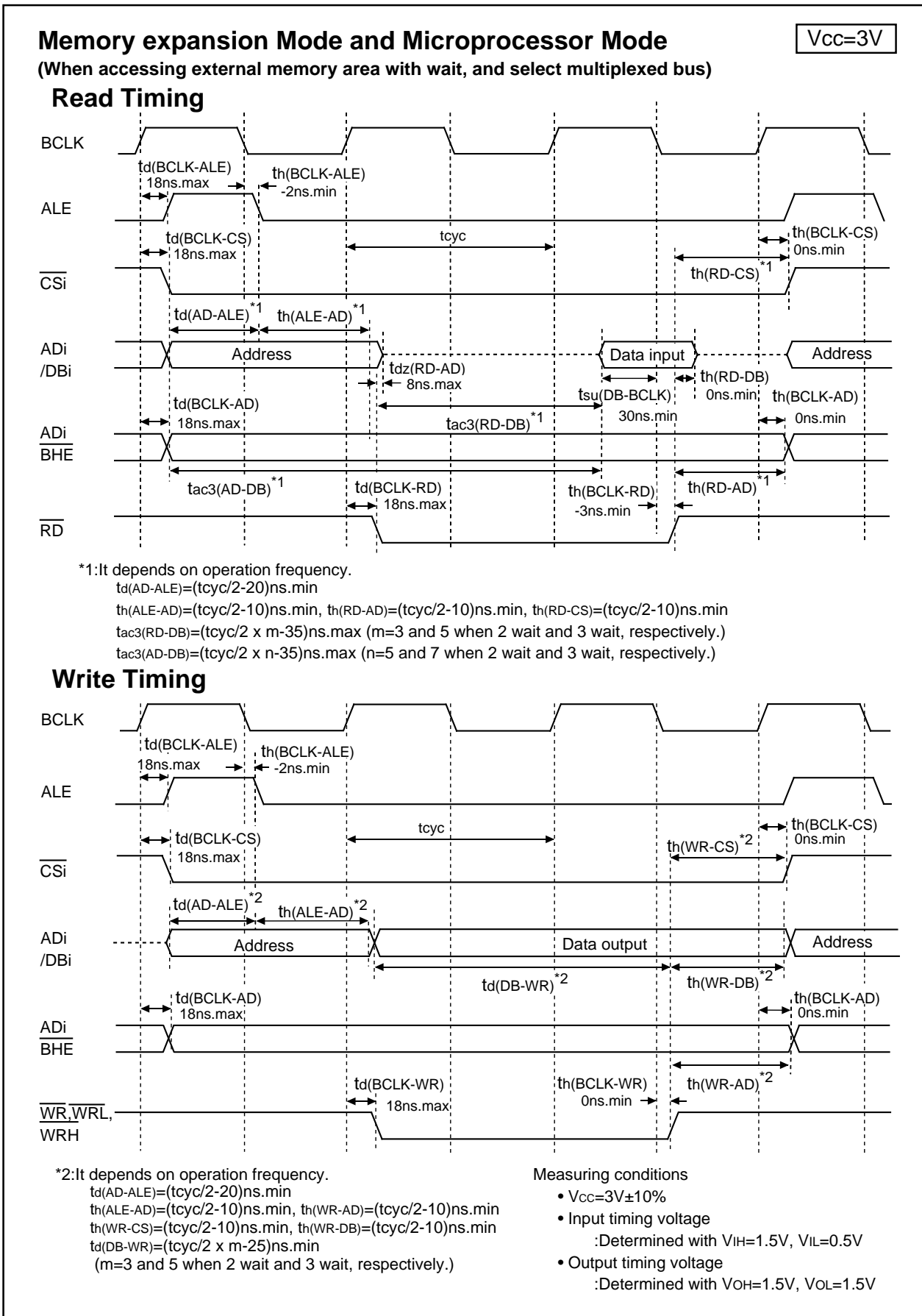


Figure 1.32.12. Vcc=3V timing diagram (3)

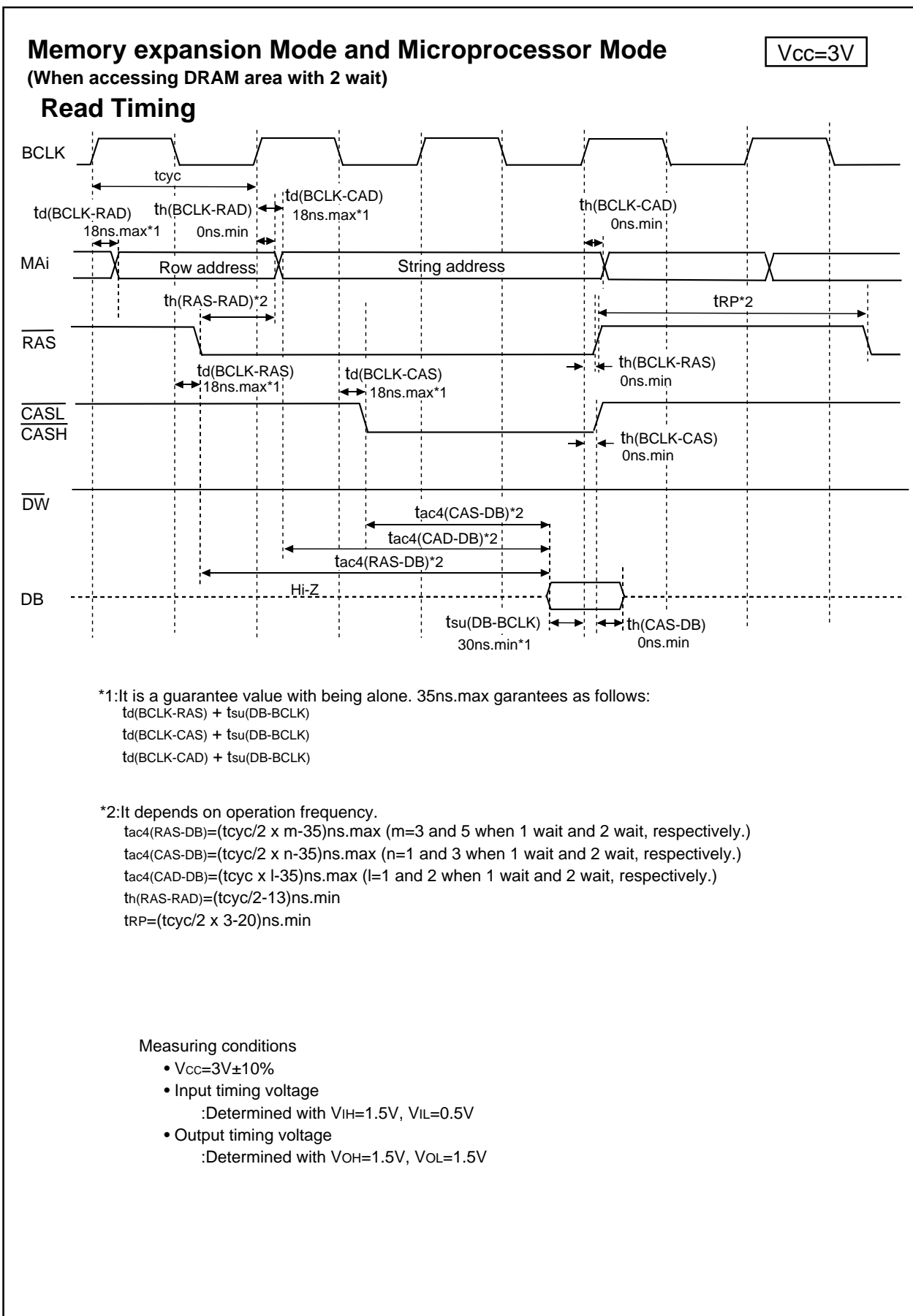


Figure 1.32.13. Vcc=3V timing diagram (4)

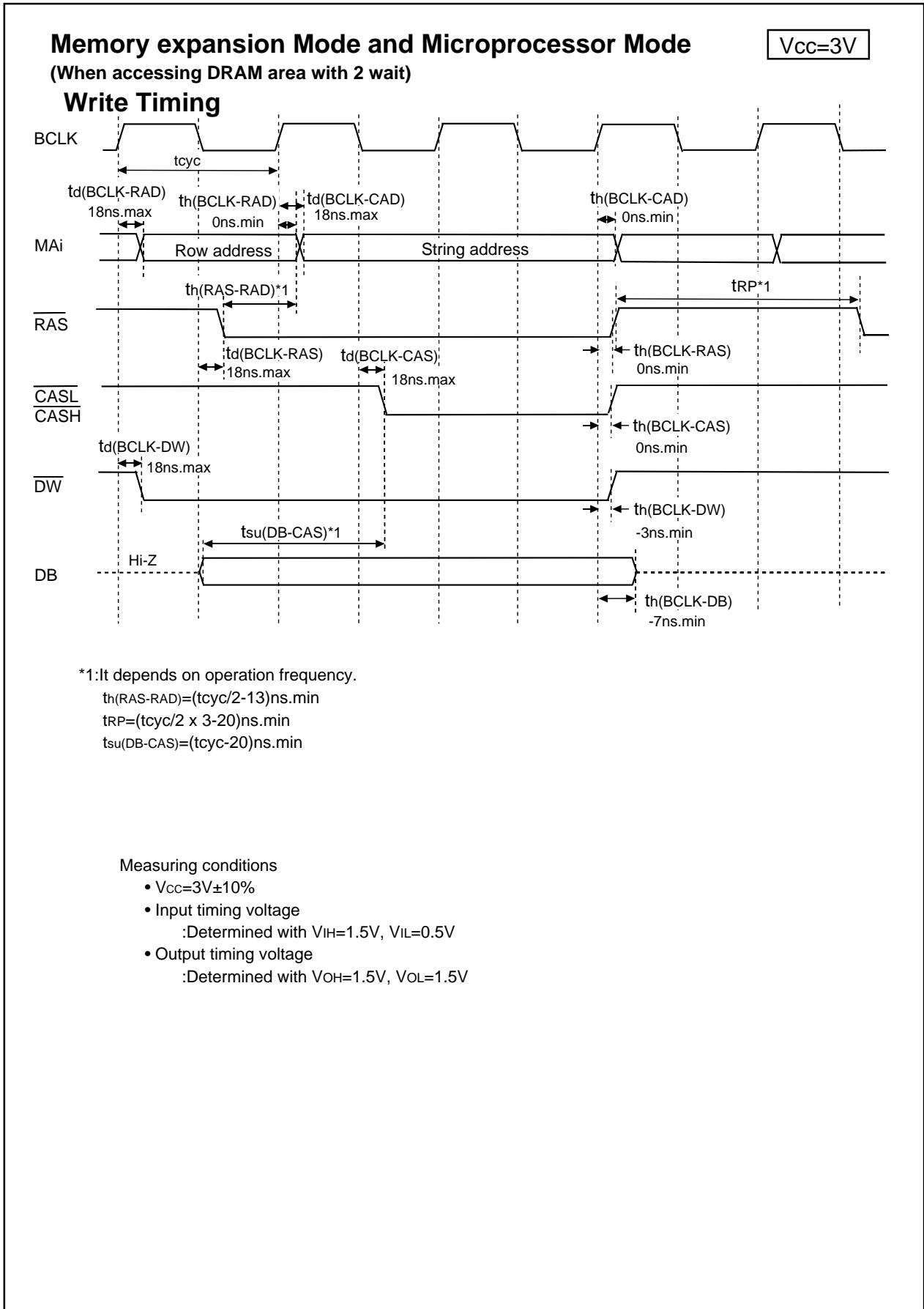


Figure 1.32.14. V_{CC}=3V timing diagram (5)

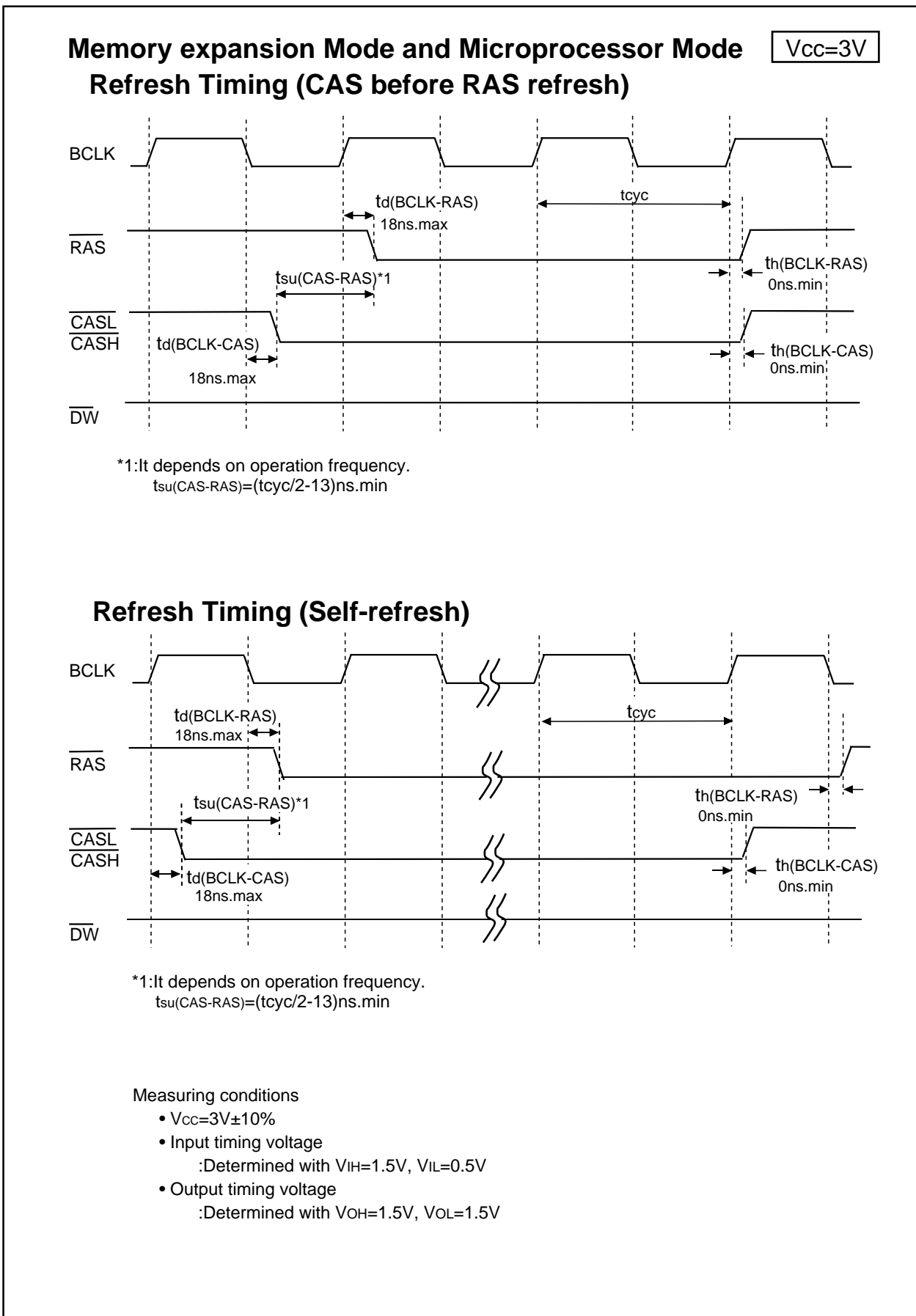


Figure 1.32.15. V_{CC}=3V timing diagram (6)

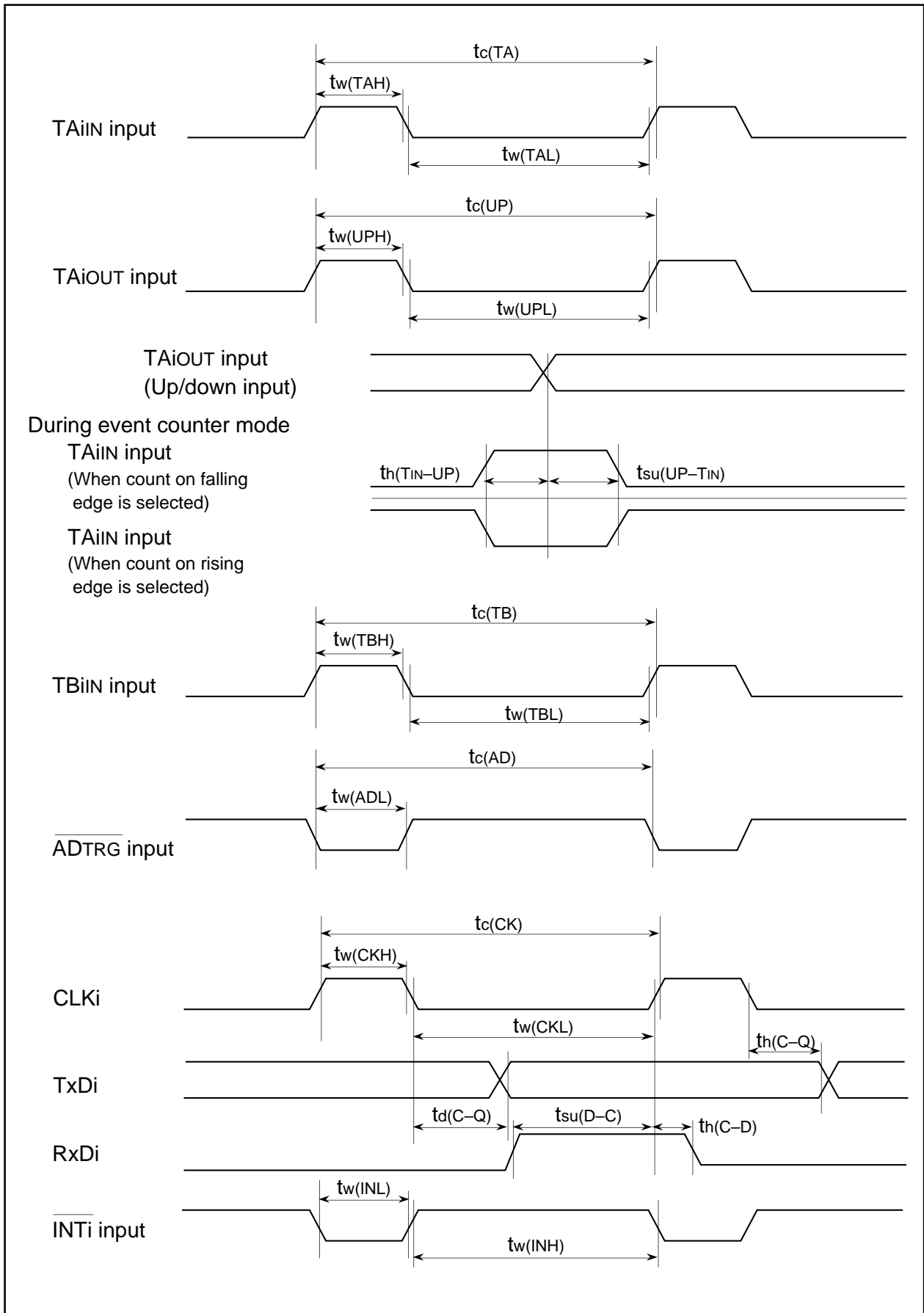


Figure 1.32.16. $V_{CC}=3V$ timing diagram (7)

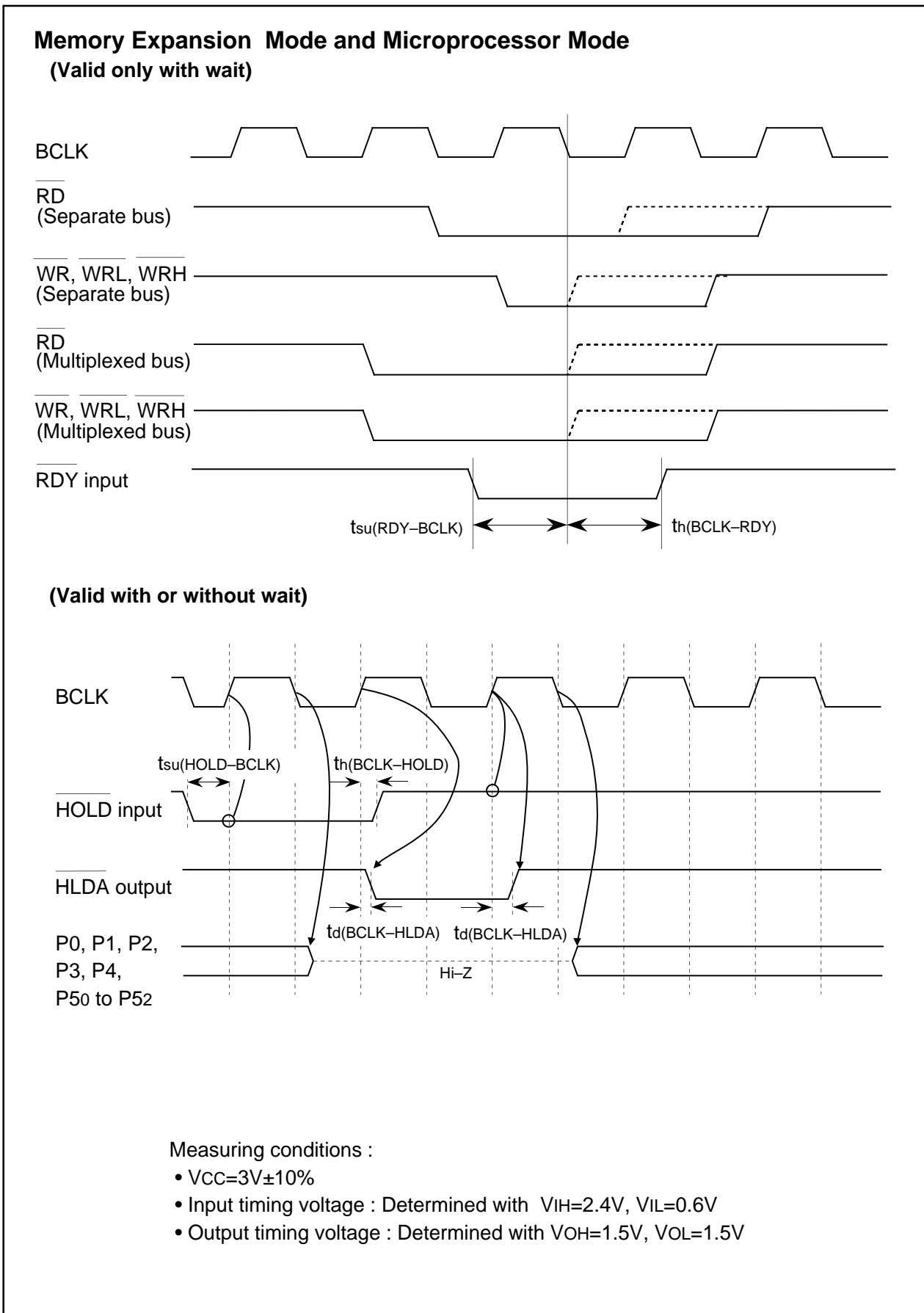


Figure 1.32.17. $V_{CC}=3V$ timing diagram (8)

Description (Flash Memory Version)

Outline Performance

Table 1.33.1 shows the outline performance of the M32C/83 (flash memory version).

Table 1.33.1. Outline Performance of the M32C/83 (flash memory version)

Item		Performance
Power supply voltage		f(XIN)=30MHz, without wait, 4.2V to 5.5V f(XIN)=20MHz, without wait, 3.0V to 3.6V
Program/erase voltage		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 1.33.3
	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 M32C/83 products (flash memory version).

- (1) ROM capacity
- (2) Package
 - 100P6S-A ... Plastic molded QFP
 - 100P6Q-A ... Plastic molded QFP
 - 144P6Q-A ... Plastic molded QFP

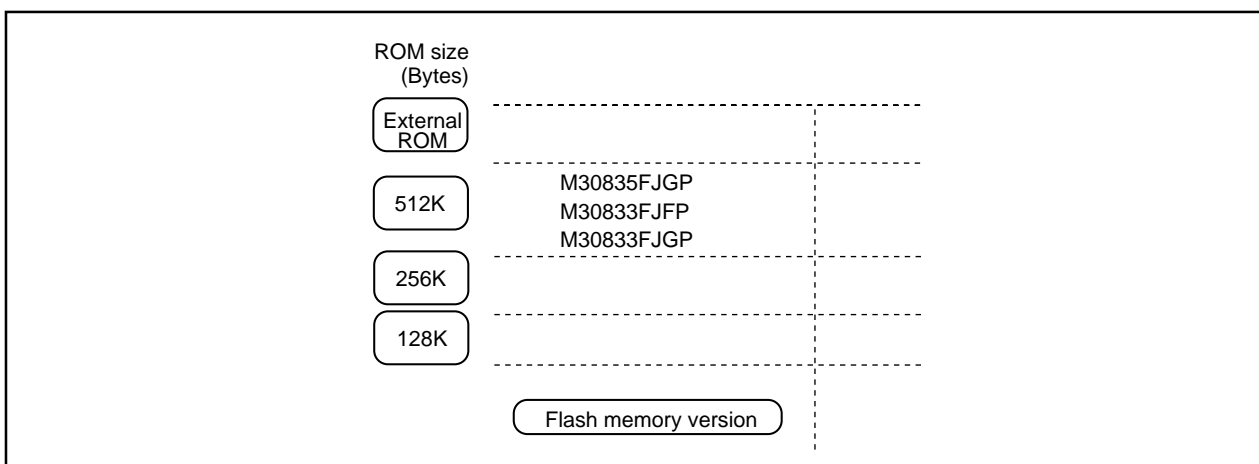


Figure 1.33.1. ROM Expansion

Description (Flash Memory Version)

The following lists the M32C/83 products to be supported in the future.

Table 1.33.2. Product List

As of Nov., 2001

Type No	ROM capacity	RAM capacity	Package type	Remarks
M30835FJGP **	512 Kbytes	31 Kbytes	144P6Q-A	
M30833FJGP **			100P6Q-A	
M30833FJFP **			100P6S-A	

** : Under development

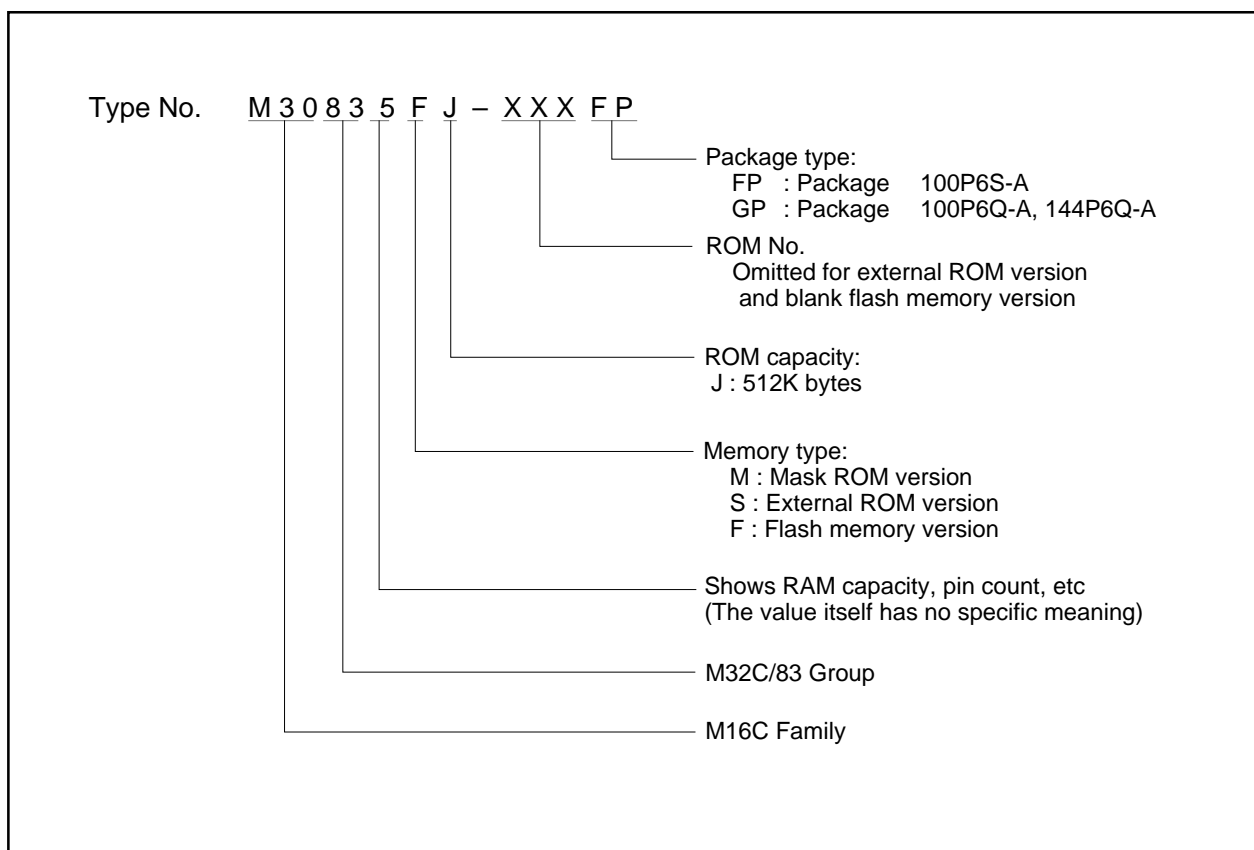


Figure 1.33.2. Type No., memory size, and package

Description (Flash Memory Version)

Flash Memory

The M32C/83 (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 1.33.3, 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.

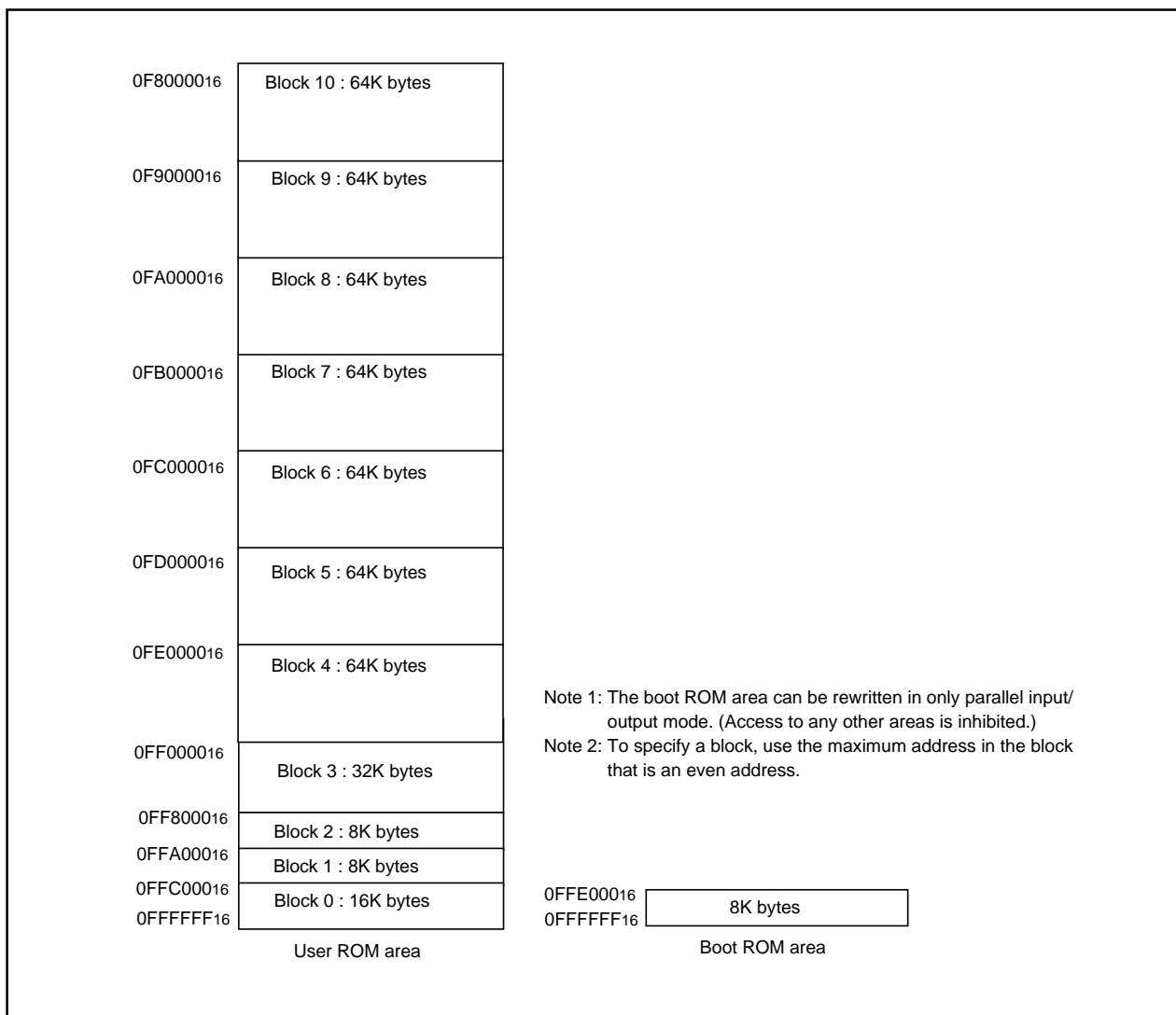


Figure 1.33.3. Block diagram of flash memory version

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 1.33.3 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 1.33.3 for details about the boot ROM area.

Normal microcomputer mode is entered when the microcomputer is reset with pulling CNVss 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 P55 pin low, the CNVss pin high, and the P50 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 of 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 037716) 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 1.34.1 shows the flash memory control register 0.

CPU Rewrite Mode (Flash Memory Version)

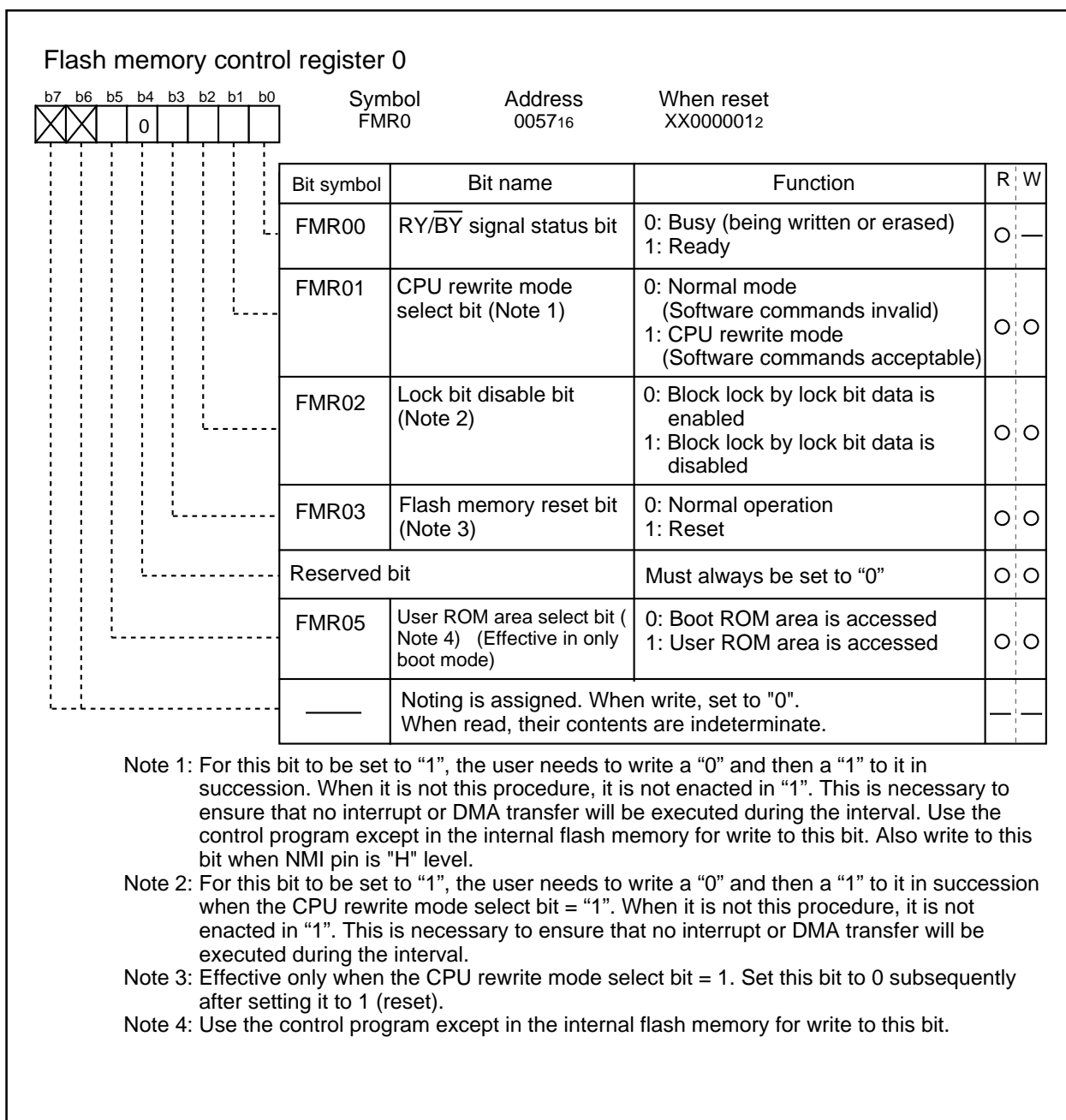


Figure 1.34.1. Flash memory control register

Flash memory control register (address 0057₁₆)

Bit 0 of the flash memory control register 0 is the RY/BY signal status bit 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".

CPU Rewrite Mode (Flash Memory Version)

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.

Figure 1.34.2 shows a flowchart for setting/releasing the CPU rewrite mode. Always perform operation as indicated in these flowcharts.

CPU Rewrite Mode (Flash Memory Version)

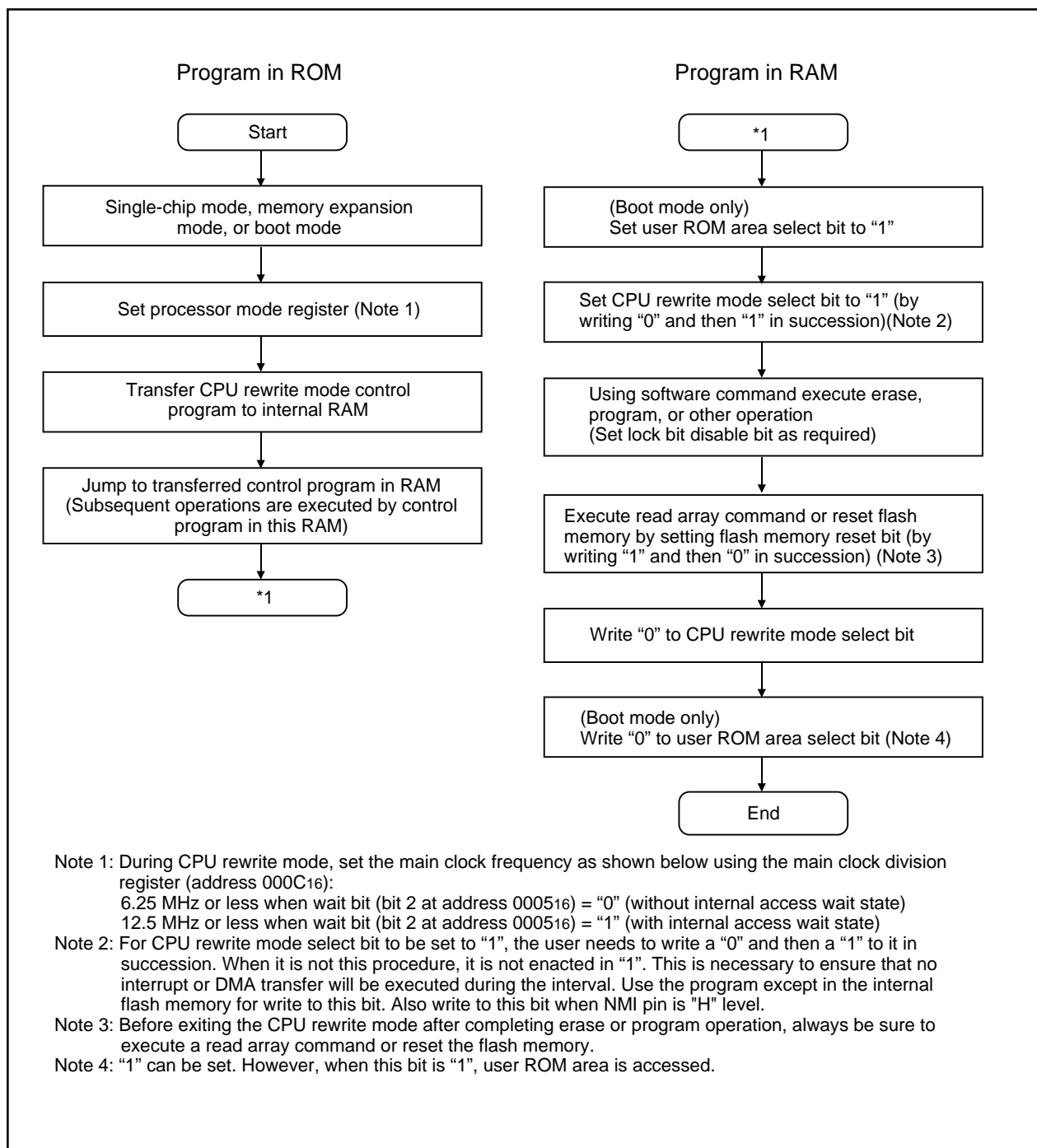


Figure 1.34.2. CPU rewrite mode set/reset flowchart

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 000C16):

6.25 MHz or less when wait bit (bit 2 at address 000516) = 0 (without internal access wait state)

12.5 MHz or less when wait bit (bit 2 at address 000516) = 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 rewrite 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 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 and lock bit disable 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 1.34.1 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 (D8 to D15) is ignored.

The content of each software command is explained below.

Table 1.34.1. 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 (D₈ to D₁₅) is ignored.

Note 2: SRD = Status Register Data

Note 3: WA = Write Address, WD = Write Data

WA and WD must be set sequentially from 00₁₆ to FE₁₆ (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: D₆ corresponds to the block lock status. Block not locked when D₆ = 1, block locked when D₆ = 0.

Note 6: X denotes a given address in the user ROM area (that is an even address).

Read Array Command (FF₁₆)

The read array mode is entered by writing the command code "FF₁₆" 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 (D₀–D₁₅), 16 bits at a time.

The read array mode is retained intact until another command is written.

Read Status Register Command (70₁₆)

When the command code "70₁₆" is written in the first bus cycle, the content of the status register is read out at the data bus (D₀–D₇) by a read in the second bus cycle.

The status register is explained in the next section.

Clear Status Register Command (50₁₆)

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 "50₁₆" in the first bus cycle.

Page Program Command (41₁₆)

Page program allows for high-speed programming in units of 256 bytes. Page program operation starts when the command code "41₁₆" 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 A0-A7 need to be incremented by 2 from "00₁₆" to "FE₁₆." 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 (FF₁₆) or Read Lock Bit Status command (71₁₆) is written or the flash memory is reset using its reset bit.

The RY/ $\overline{\text{BY}}$ signal status bit 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 1.34.3 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.

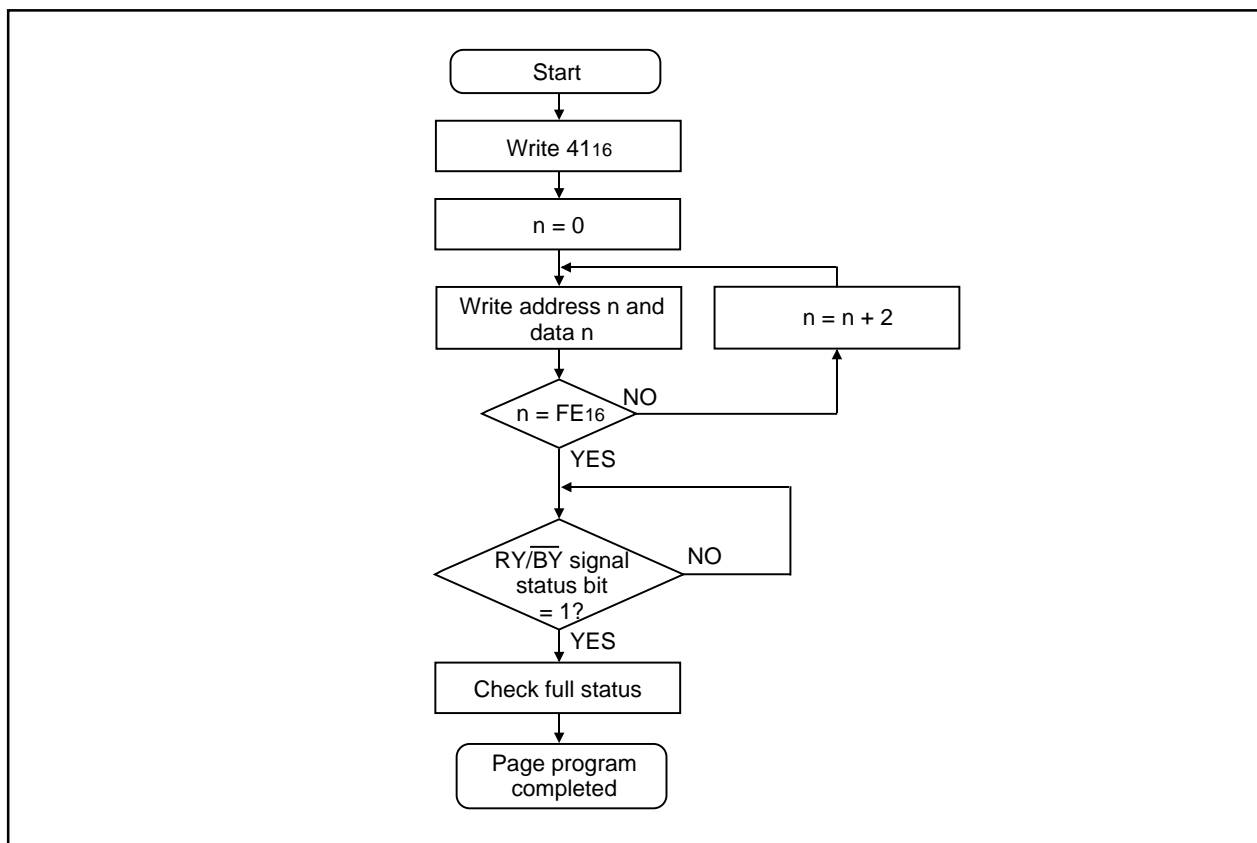


Figure 1.34.3. Page program flowchart

Block Erase Command (20₁₆/D0₁₆)

By writing the command code "20₁₆" in the first bus cycle and the confirmation command code "D0₁₆" 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 (FF₁₆) or Read Lock Bit Status command (71₁₆) is written or the flash memory is reset using its reset bit.

The RY/ $\overline{\text{BY}}$ signal status bit 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 1.34.4 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.

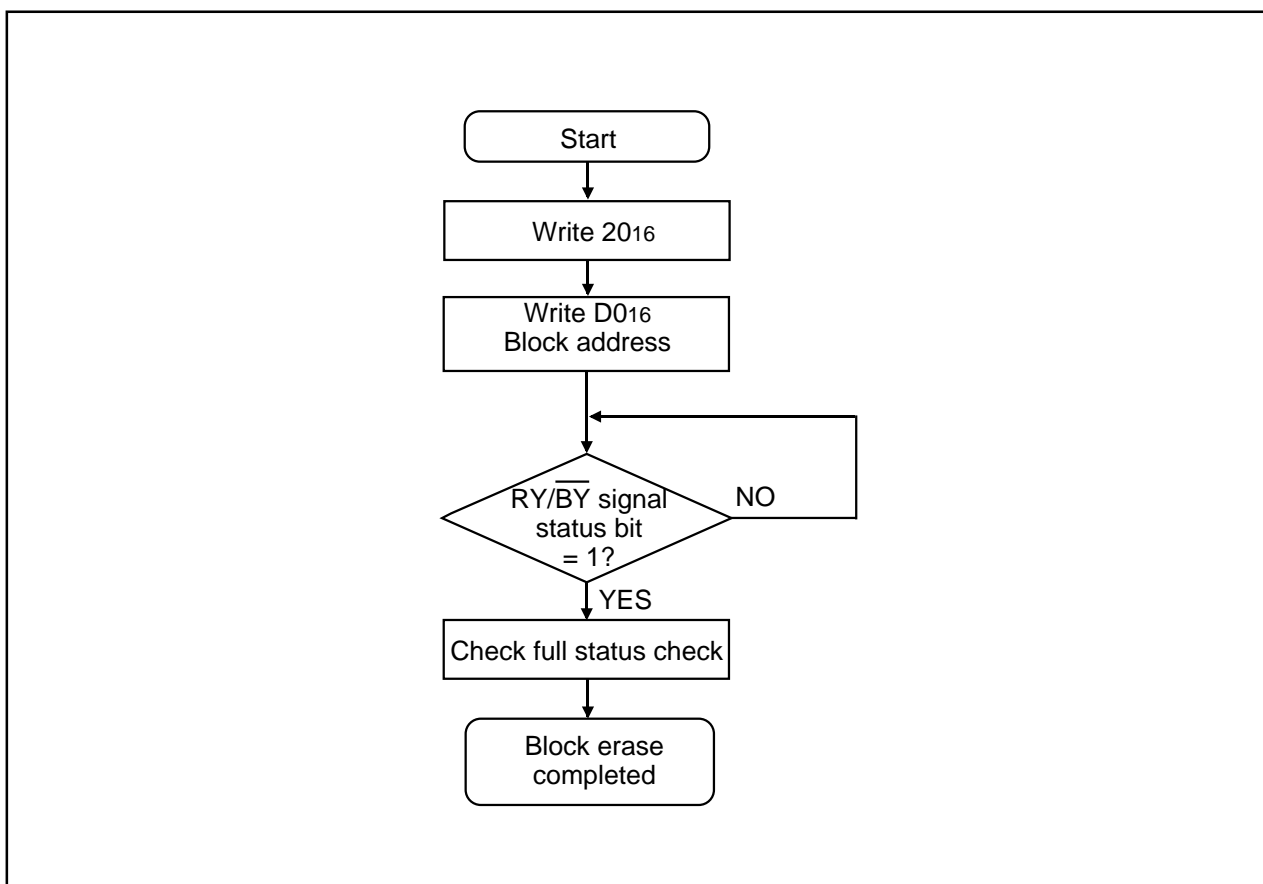


Figure 1.34.4. Block erase flowchart

Erase All Unlock Blocks Command (A716/D016)

By writing the command code "A716" in the first bus cycle and the confirmation command code "D016" 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 (7716/D016)

By writing the command code "7716" in the first bus cycle and the confirmation command code "D016" 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 1.34.5 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.

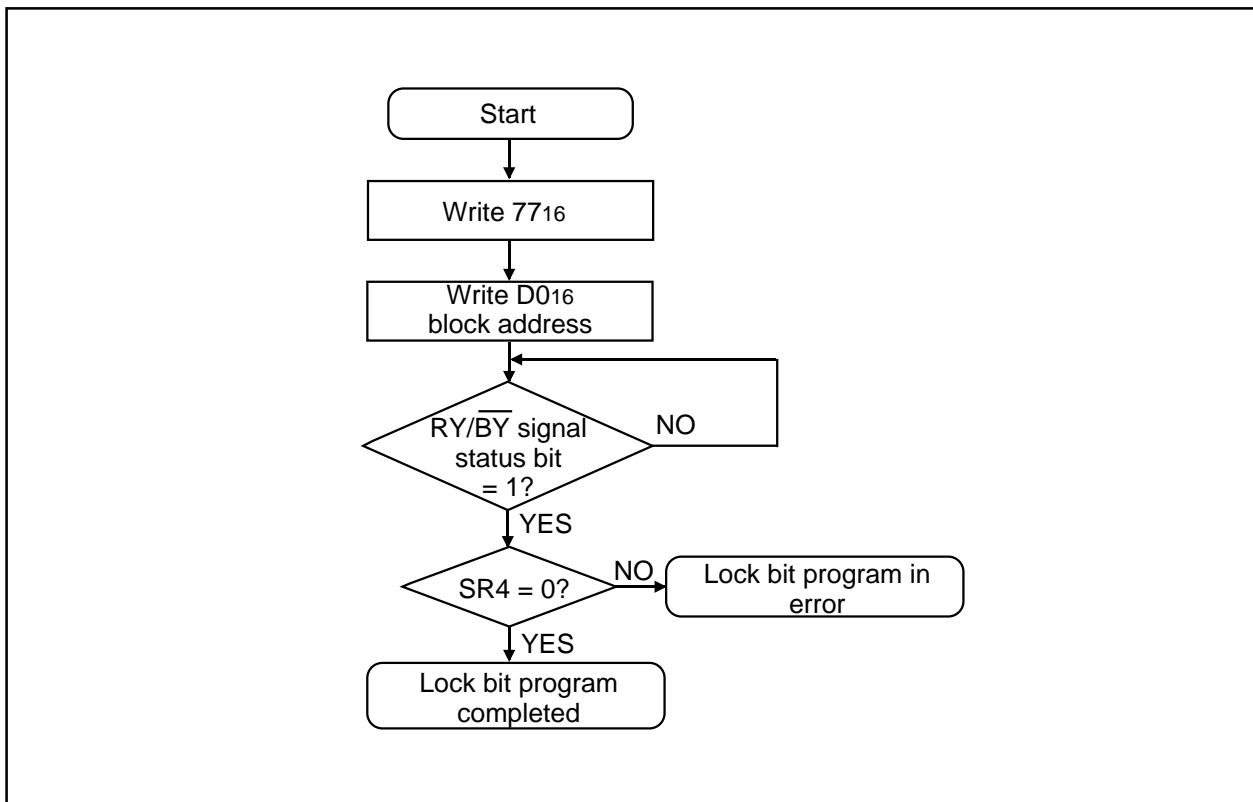


Figure 1.34.5. Lock bit program flowchart

Read Lock Bit Status Command (7116)

By writing the command code "7116" 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 1.34.6 shows an example of a read lock bit program flowchart.

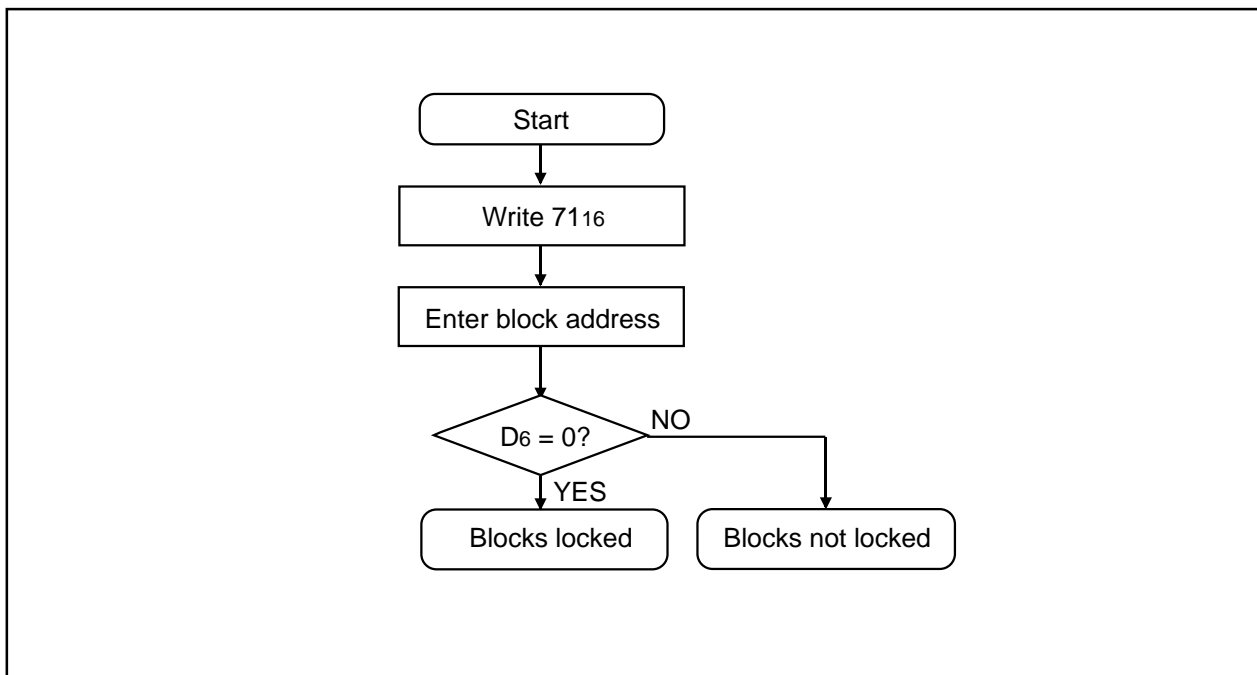


Figure 1.34.6. Read lock bit status flowchart

Data Protect Function (Block Lock)

Each block in Figure 1.33.3 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 (70₁₆). Table 1.34.2 details the status register.

The status register is cleared by writing the Clear Status Register command (50₁₆).

After a reset, the status register is set to "80₁₆."

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 (20₁₆) is not the confirmation command (D0₁₆), 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 (77₁₆/D0₁₆), block erase (20₁₆/D0₁₆), or erase all unlock blocks (A7₁₆/D0₁₆) is not the D0₁₆ or FF₁₆. However, if FF₁₆ 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, "80₁₆" is output; when writing fails, "90₁₆" is output; and when excessive data is written, "88₁₆" is output.

Table 1.34.2. 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 1.34.7 shows a full status check flowchart and the action to be taken when each error occurs.

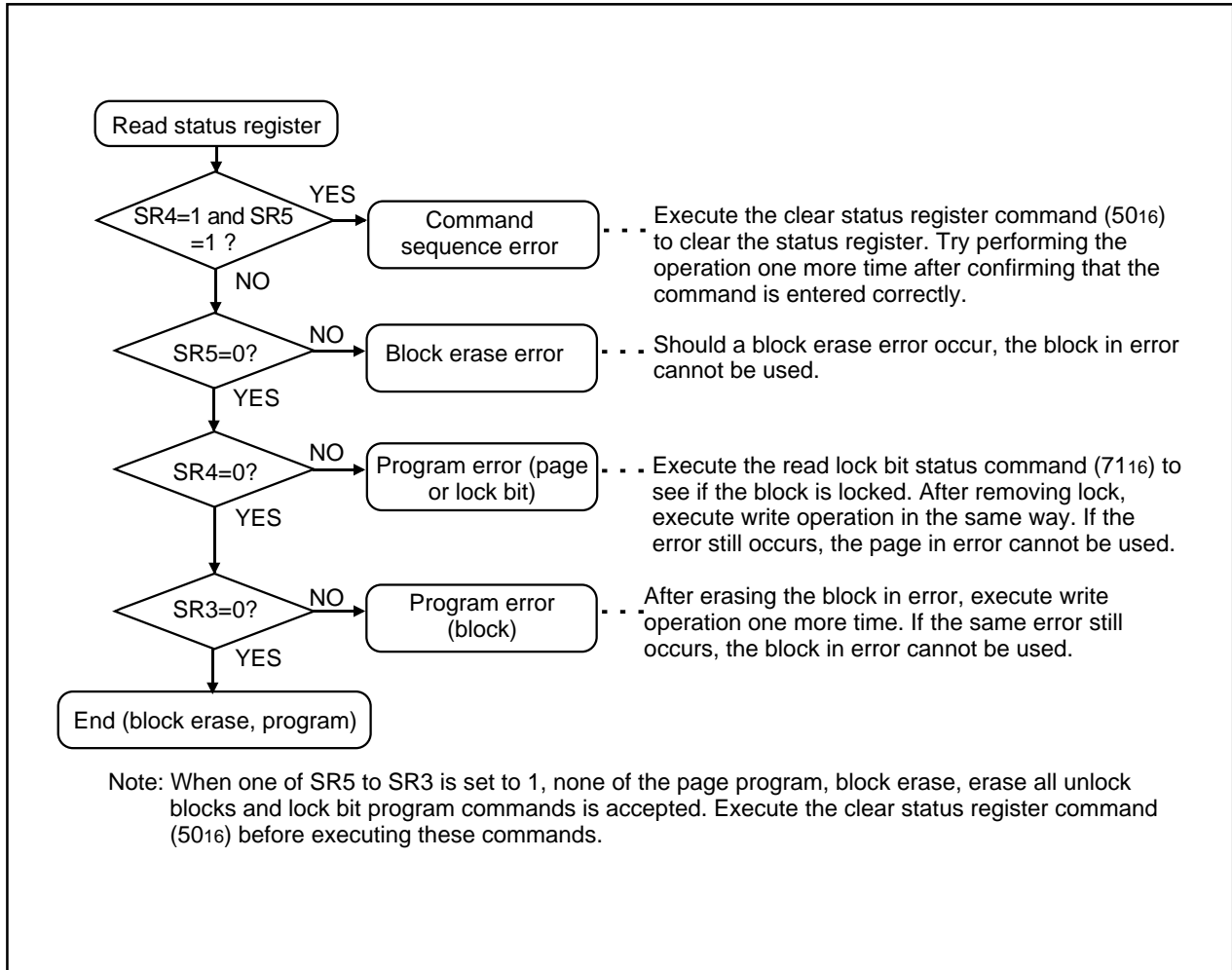


Figure 1.34.7. Full status check flowchart and remedial procedure for errors

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 (0FFFFFFF₁₆) during parallel I/O mode. Figure 1.34.8 shows the ROM code protect control address (0FFFFFFF₁₆). (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.

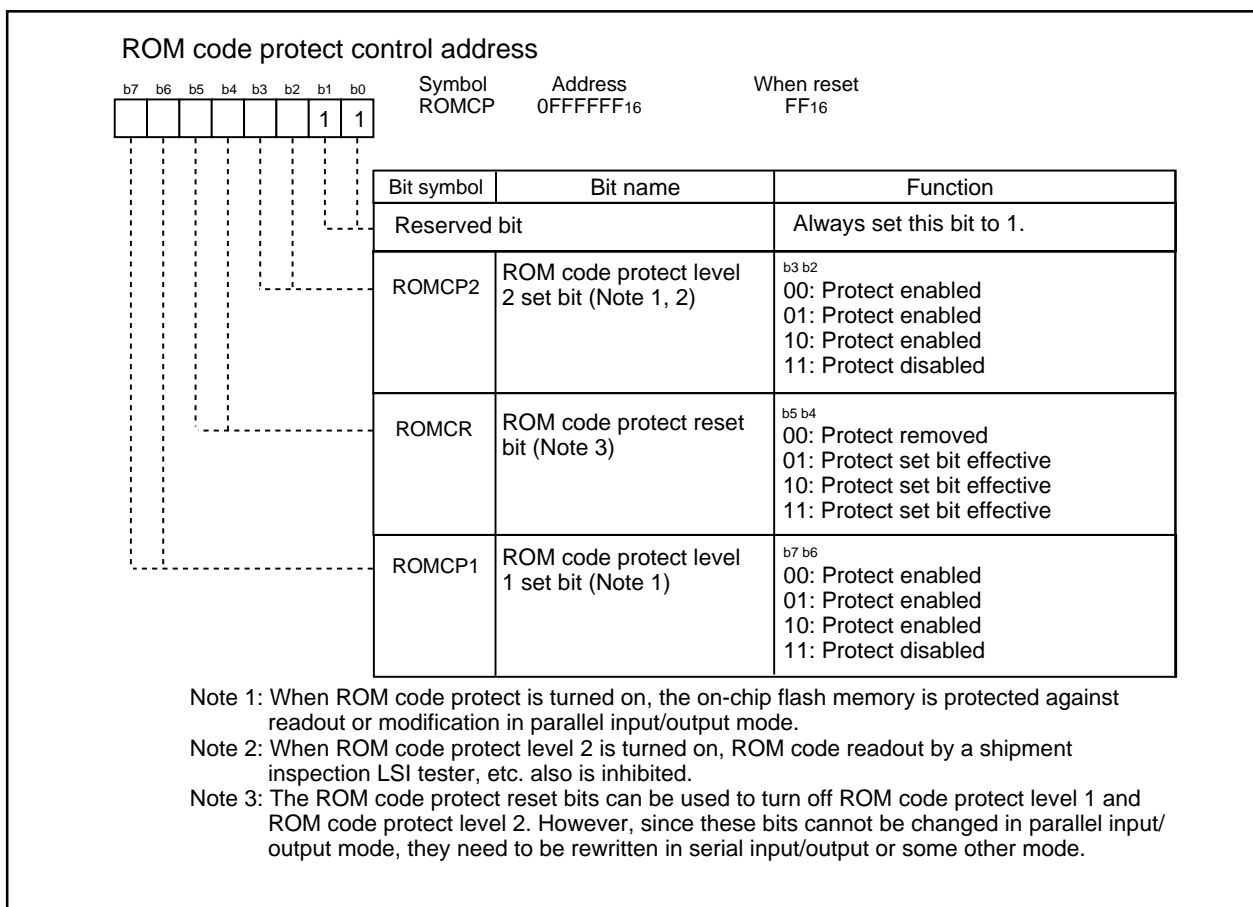


Figure 1.34.8. ROM code protect control address

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 0FFFFDF₁₆, 0FFFFE3₁₆, 0FFFFE8₁₆, 0FFFFE7₁₆, 0FFFFE3₁₆, 0FFFFF7₁₆, and 0FFFFFB₁₆. Write a program which has had the ID code preset at these addresses to the flash memory.

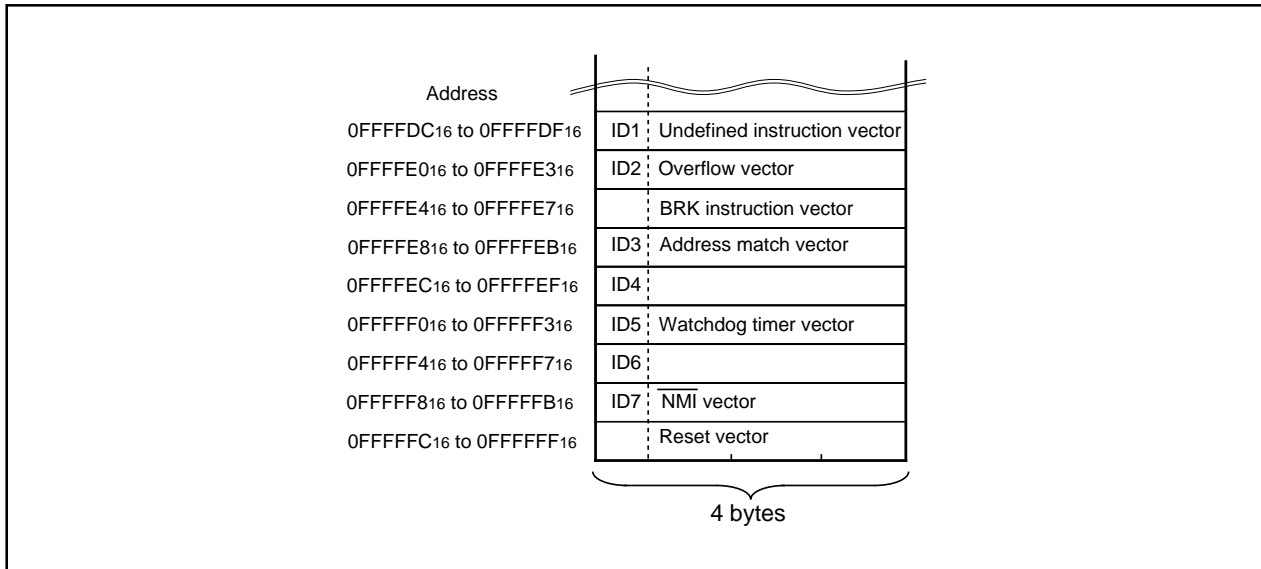


Figure 1.34.9. ID code store addresses

Parallel I/O Mode

In this mode, the M32C/83 (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 M32C/83 cannot be programmed by a programmer for the flash memory.

Use an exclusive programmer supporting M32C/83 (flash memory version).

Refer to the instruction manual of each programmer 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 1.33.3 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 1.33.3.

The boot ROM area is 8 Kbytes in size. In parallel I/O mode, it is located at addresses 0FFE000₁₆ through 0FFFFFF₁₆. 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.

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 asynchronous. 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 P50 (\overline{CE}) pin is "H" level, the P55 (\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 1.35.1 to 1.35.3 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 asynchronous) according to the level of CLK1 pin when the reset is released.

To use standard serial I/O mode 1 (clock synchronized), set the CLK1 pin to "H" level and the TxD1 pin to "L" level, and release the reset. The CLK1 pin is connected to Vcc via pull-up resistance and the TxD1 is connected to Vss via pull-down resistance. The operation uses the four UART1 pins CLK1, RxD1, TxD1 and RTS1 (BUSY). The CLK1 pin is the transfer clock input pin through which an external transfer clock is input. The TxD1 pin is for CMOS output. The RTS1 (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 asynchronous), set the CLK1 pin to "L" level and release the reset. The operation uses the two UART1 pins RxD1 and TxD1.

In the standard serial I/O mode, only the user ROM area indicated in Figure 1.35.20 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.

Pin functions (Flash memory standard serial I/O mode)

Pin	Name	I/O	Description
Vcc,Vss	Power input		Apply 4.2V to 5.5V to Vcc pin and 0 V to Vss pin.
CNVss	CNVss	I	Connect to Vcc pin.
RESET	Reset input	I	Reset input pin. While reset is "L" level, a 20 cycle or longer clock must be input to XIN pin.
XIN	Clock input	I	Connect a ceramic resonator or crystal oscillator between XIN and XOUT pins. To input an externally generated clock, input it to XIN pin and open XOUT pin.
XOUT	Clock output	O	
BYTE	BYTE	I	Connect this pin to Vcc or Vss.
AVcc, AVss	Analog power supply input	I	Connect AVSS to Vss and AVcc to Vcc, respectively.
VREF	Reference voltage input	I	Enter the reference voltage for A-D converter from this pin.
P00 to P07	Input port P0	I	Input "H" or "L" level signal or open.
P10 to P17	Input port P1	I	Input "H" or "L" level signal or open.
P20 to P27	Input port P2	I	Input "H" or "L" level signal or open.
P30 to P37	Input port P3	I	Input "H" or "L" level signal or open.
P40 to P47	Input port P4	I	Input "H" or "L" level signal or open.
P51 to P54, P56, P57	Input port P5	I	Input "H" or "L" level signal or open.
P50	CE input	I	Input "H" level signal.
P55	EPM input	I	Input "L" level signal.
P60 to P63	Input port P6	I	Input "H" or "L" level signal or open.
P64	BUSY output	O	Standard serial mode 1: BUSY signal output pin Standard serial mode 2: Monitors the program operation check
P65	SCLK input	I	Standard serial mode 1: Serial clock input pin Standard serial mode 2: Input "L" level signal.
P66	RxD input	I	Serial data input pin
P67	TxD output	O	Serial data output pin. When using standard serial mode 1, an "L" level must be input to TxD pin while the RESET pin is "L". For this reason, this pin should be pulled down. After being reset, this pin functions as a data output pin. Thus adjust pull-down resistance value with the system not to affect data transfer.
P70 to P77	Input port P7	I	Input "H" or "L" level signal or open.
P80 to P84, P86, P87	Input port P8	I	Input "H" or "L" level signal or open.
P85	NMI input	I	Connect this pin to Vcc.
P90 to P97	Input port P9	I	Input "H" or "L" level signal or open.
P100 to P107	Input port P10	I	Input "H" or "L" level signal or open.
P110 to P114	Input port P11	I	Input "H" or "L" level signal or open. (Note)
P120 to P127	Input port P12	I	Input "H" or "L" level signal or open. (Note)
P130 to P137	Input port P13	I	Input "H" or "L" level signal or open. (Note)
P140 to P146	Input port P14	I	Input "H" or "L" level signal or open. (Note)
P150 to P157	Input port P15	I	Input "H" or "L" level signal or open. (Note)

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

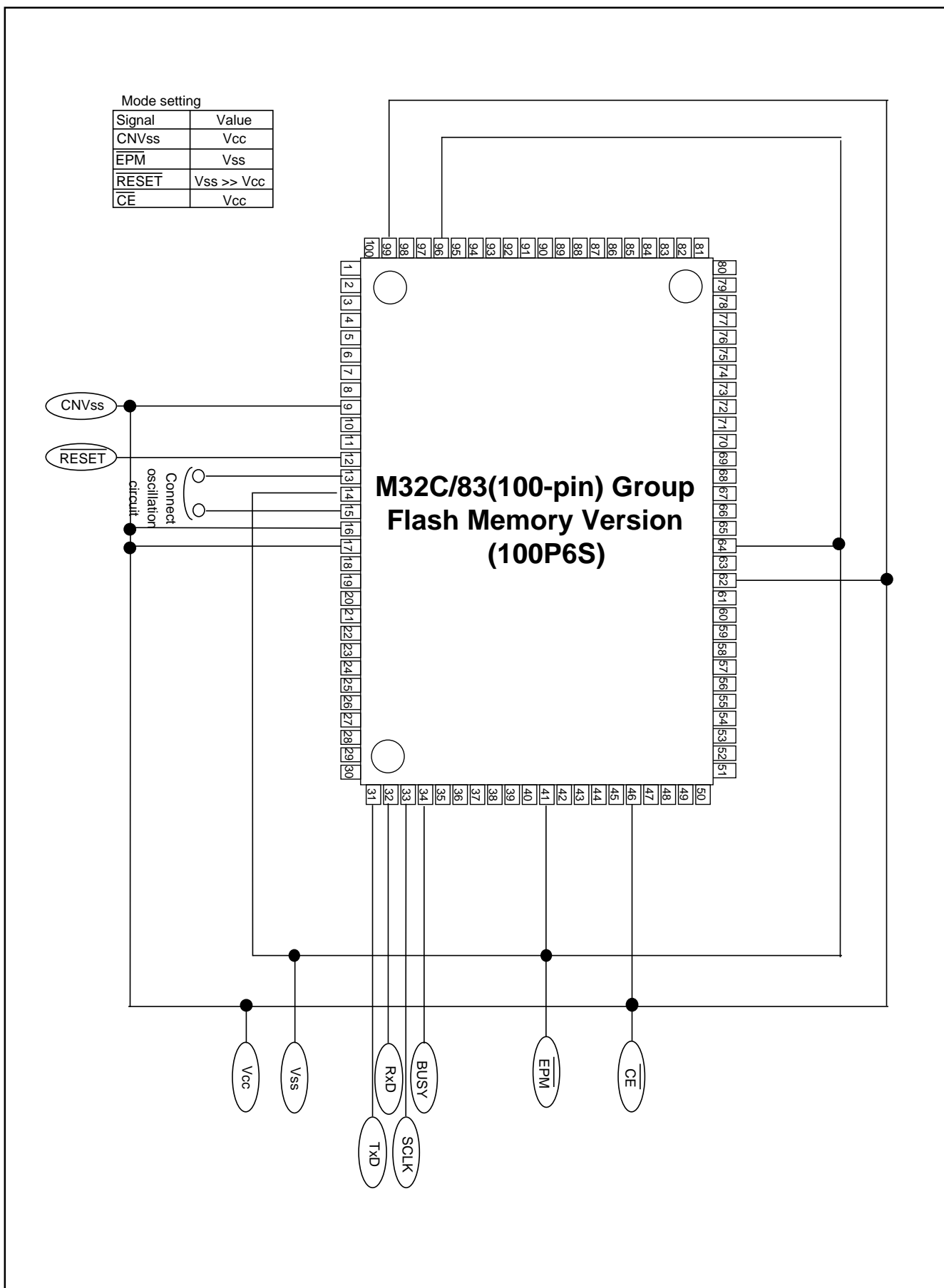


Figure 1.35.1. Pin connections for standard serial I/O mode (1)

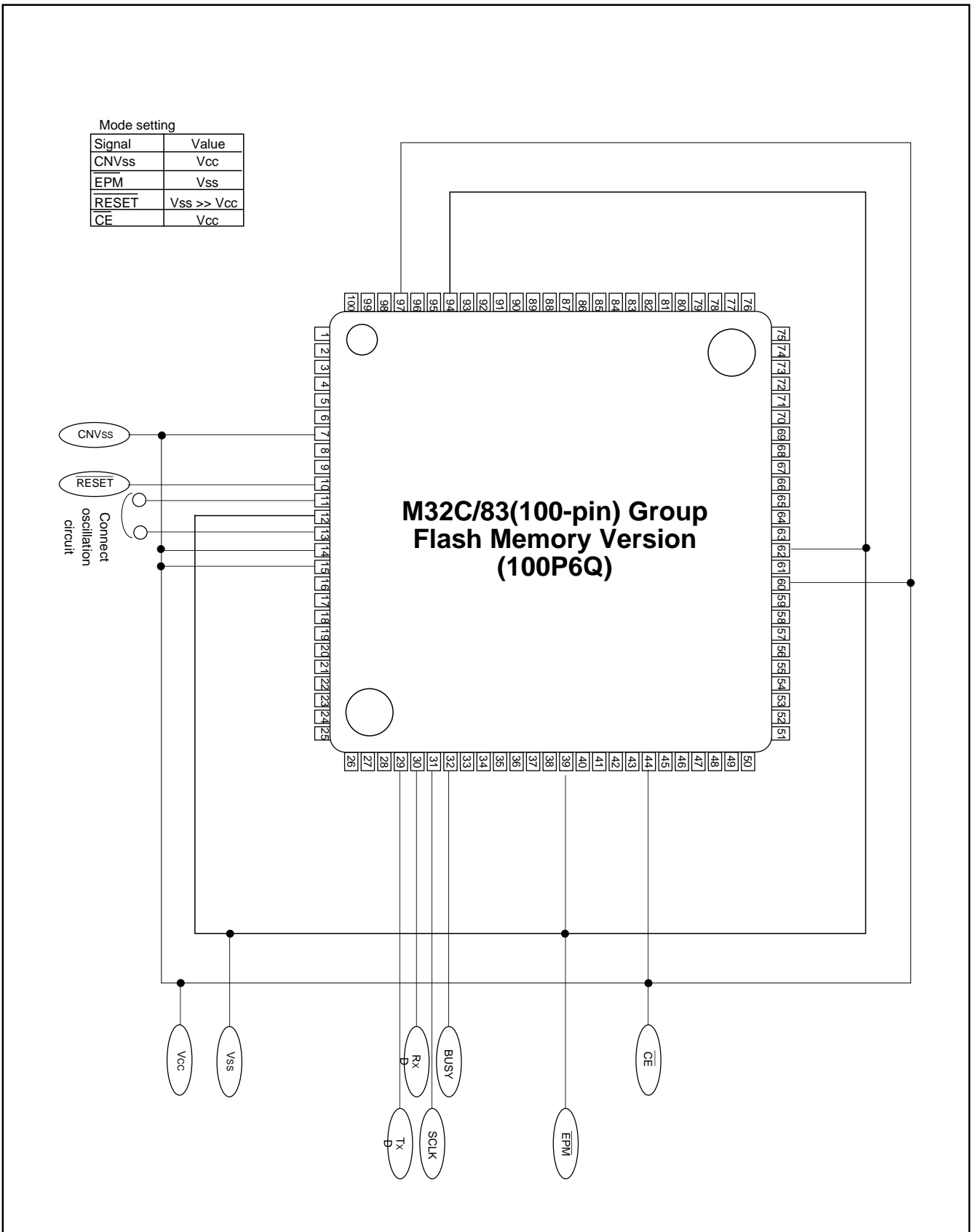


Figure 1.35.2. Pin connections for standard serial I/O mode (2)

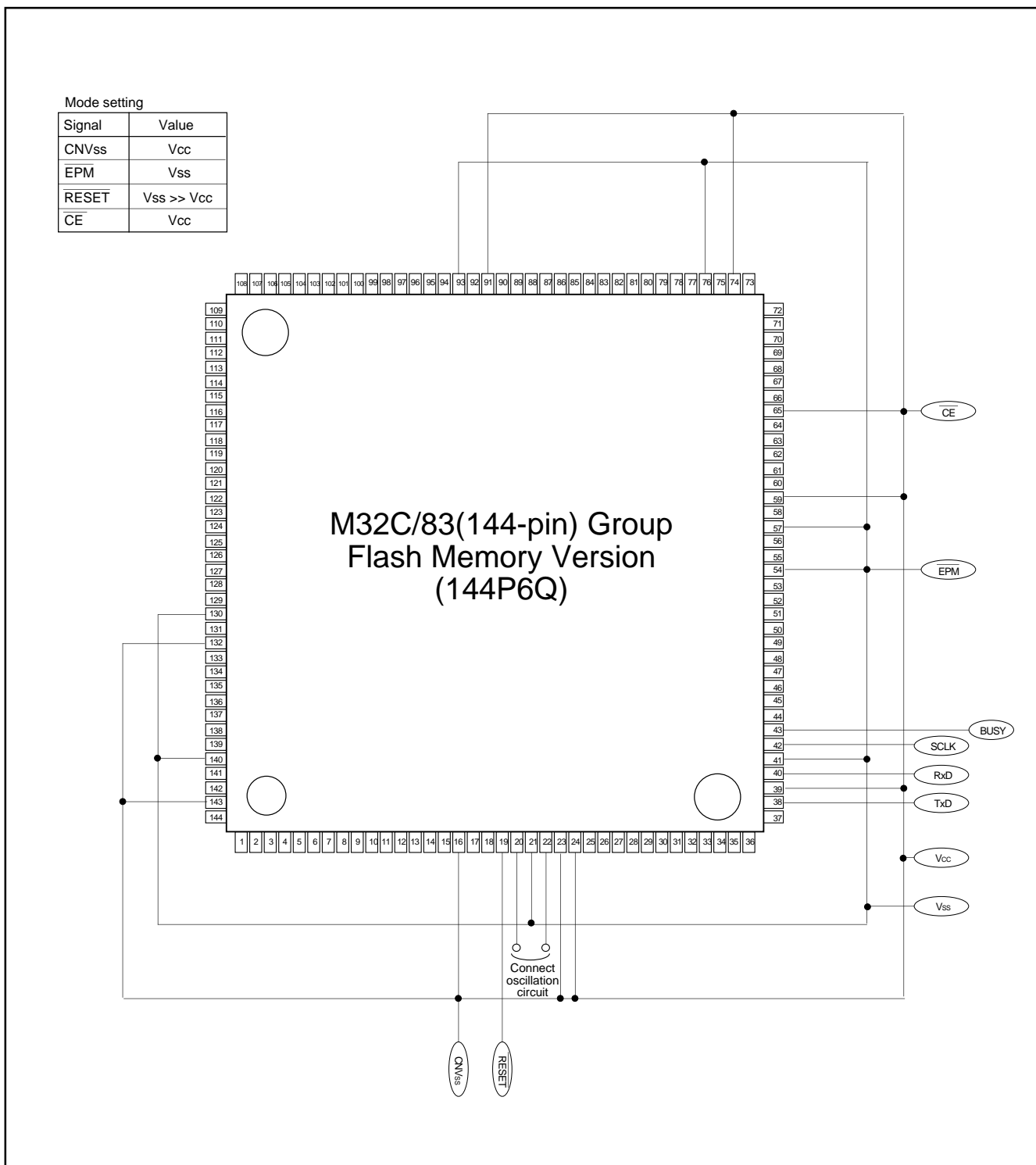


Figure 1.35.3. Pin connections for standard serial I/O mode (3)

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 programmer, etc.) using 4-wire clock-synchronized serial I/O (UART1). Standard serial I/O mode 1 is engaged by releasing the reset with the P65 (CLK1) 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 CLK1 pin, and are then input to the MCU via the RxD1 pin. In transmission, the read data and status are synchronized with the fall of the transfer clock, and output from the TxD1 pin.

The TxD1 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 RTS1 (BUSY) pin is "H" level. Accordingly, always start the next transfer after the RST1 (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 1.35.1 lists software commands. In the standard serial I/O mode 1, erase operations, programs and reading are controlled by transferring software commands via the RxD1 pin. Software commands are explained here below.

Table 1.35.1. 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 A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) From the 4th byte onward, data (D0–D7) for the page (256 bytes) specified with addresses A8 to A23 will be output sequentially from the smallest address first in sync with the rise of the clock.

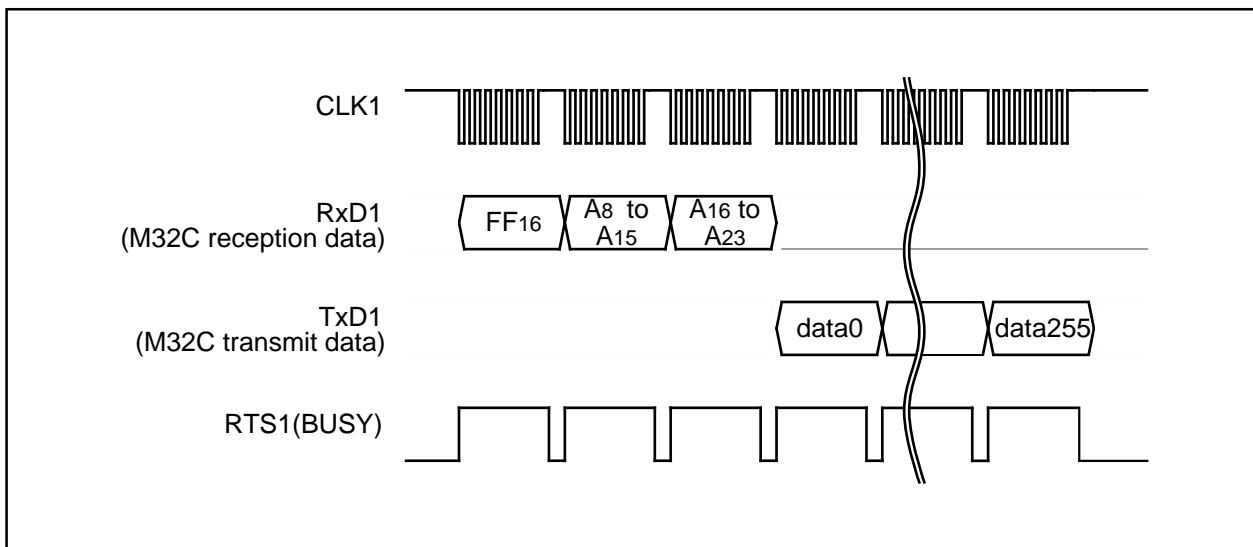


Figure 1.35.4. Timing for page read

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 "4116" command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) From the 4th byte onward, as write data (D0–D7) for the page (256 bytes) specified with addresses A8 to A23 is input sequentially from the smallest address first, that page is automatically written.

When reception setup for the next 256 bytes ends, the RTS1 (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.

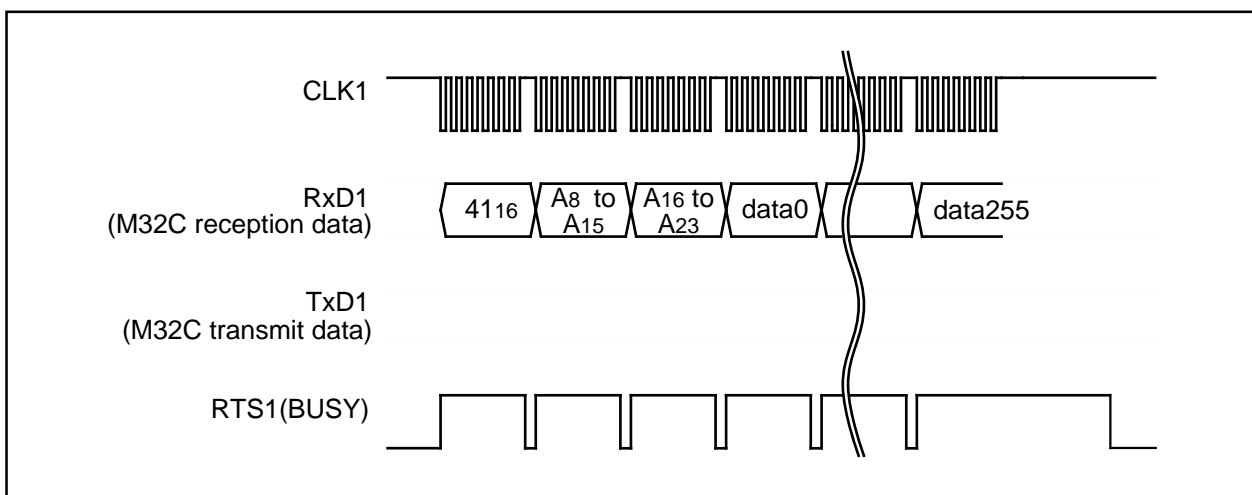


Figure 1.35.5. Timing for the page program

Block Erase Command

This command erases the data in the specified block. Execute the block erase command as explained here following.

- (1) Transfer the "2016" command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) Transfer the verify command code "D016" 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 A16 to A23.

When block erasing ends, the RTS1 (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.

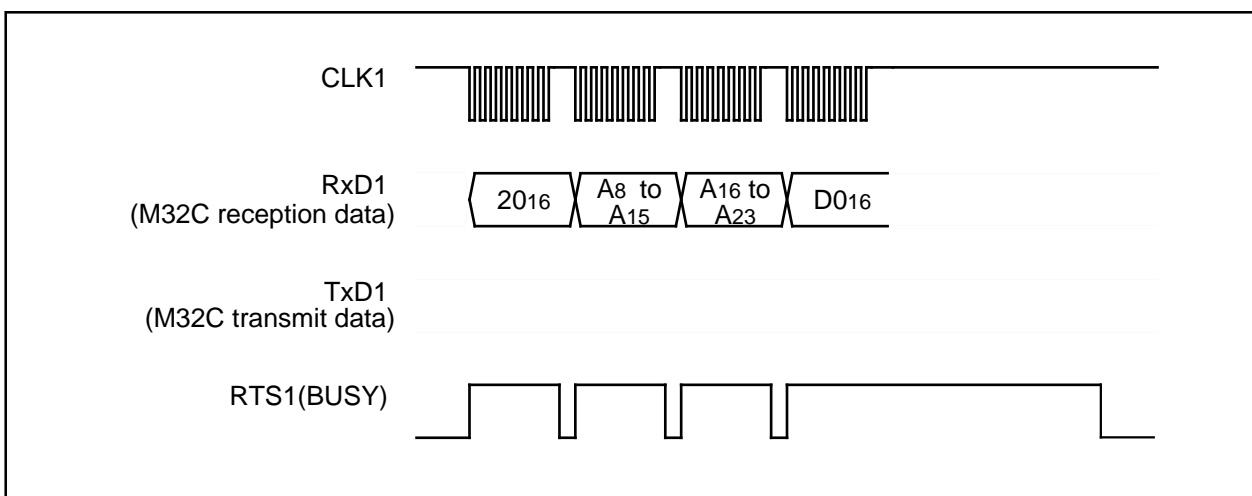


Figure 1.35.6. Timing for block erasing

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 “A7₁₆” command code with the 1st byte.
- (2) Transfer the verify command code “D0₁₆” 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 RTS1 (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.

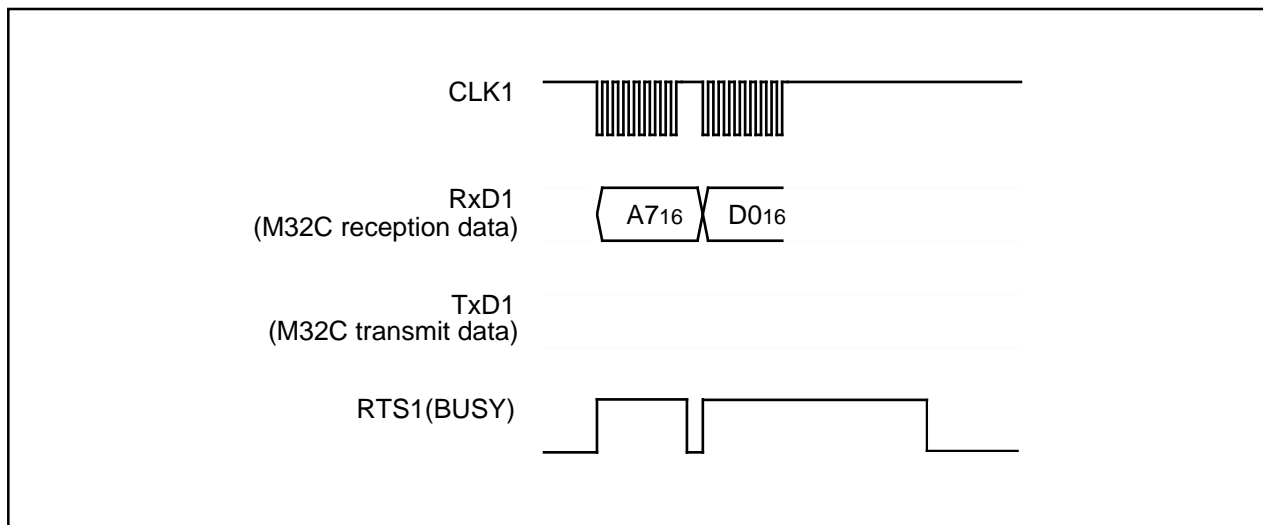


Figure 1.35.7. Timing for erasing all unlocked blocks

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.

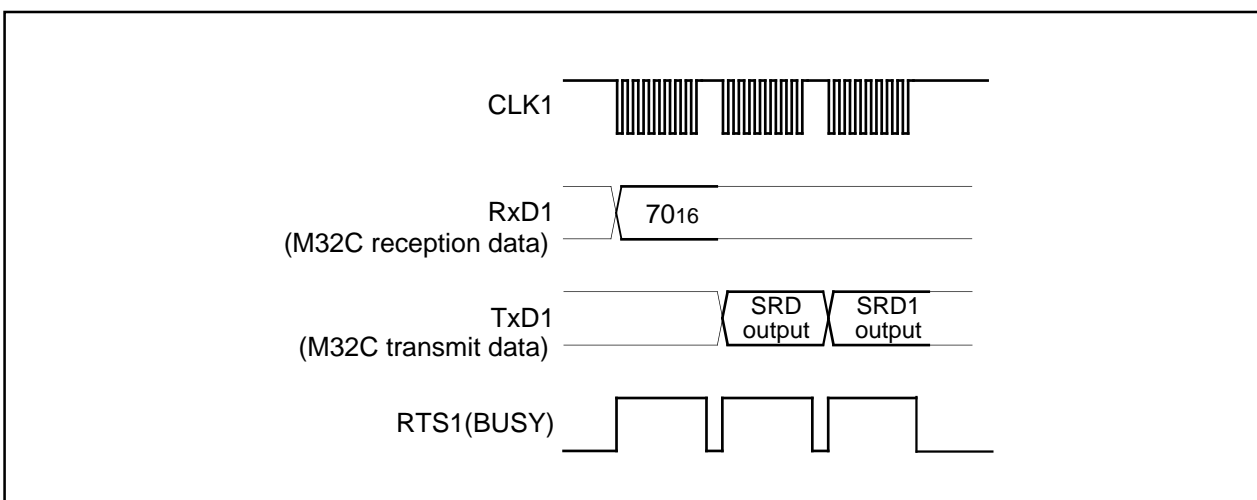


Figure 1.35.8. Timing for reading the status register

Clear Status Register Command

This command clears the bits (SR3–SR5) which are set when the status register operation ends in error. When the “50₁₆” command code is sent with the 1st byte, the aforementioned bits are cleared. When the clear status register operation ends, the RTS₁ (BUSY) signal changes from the “H” to the “L” level.

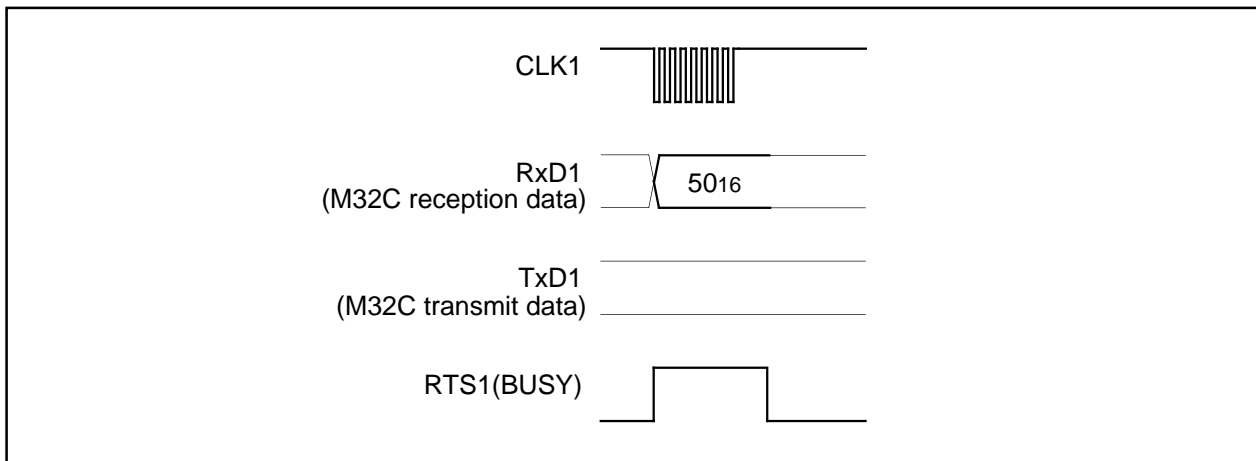


Figure 1.35.9. Timing for clearing the status register

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 “71₁₆” command code with the 1st byte.
- (2) Transfer addresses A₈ to A₁₅ and A₁₆ to A₂₃ 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 A₈ to A₂₃.

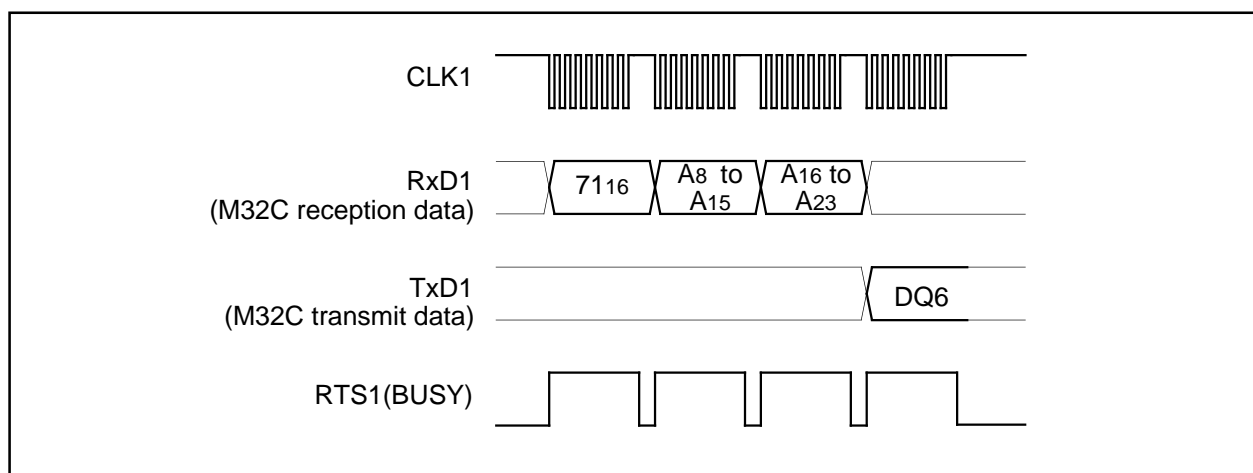


Figure 1.35.10. Timing for reading lock bit status

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 "7716" command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) Transfer the verify command code "D016" 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 A8 to A23.

When writing ends, the RTS1 (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.

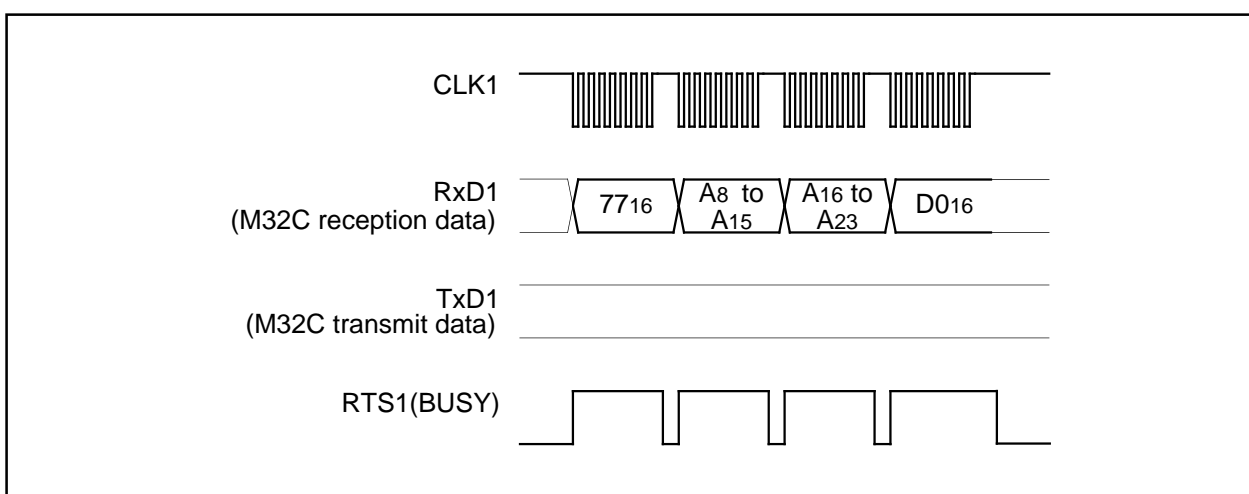


Figure 1.35.11. Timing for the lock bit program

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 "7A16" 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.

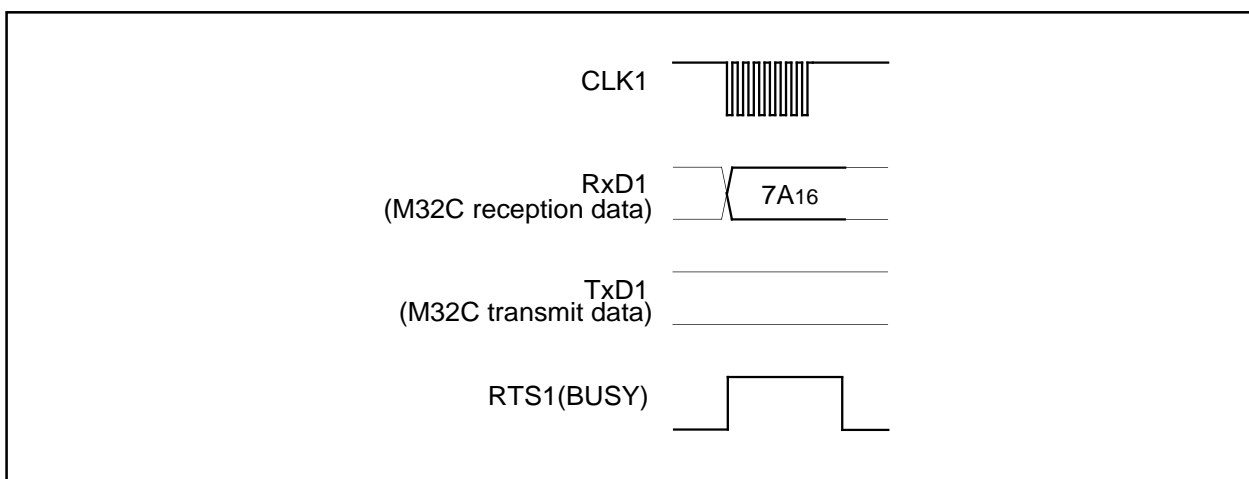


Figure 1.35.12. Timing for enabling the lock bit

Lock Bit Disable Command

This command disables the lock bit. The command code “7516” 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.

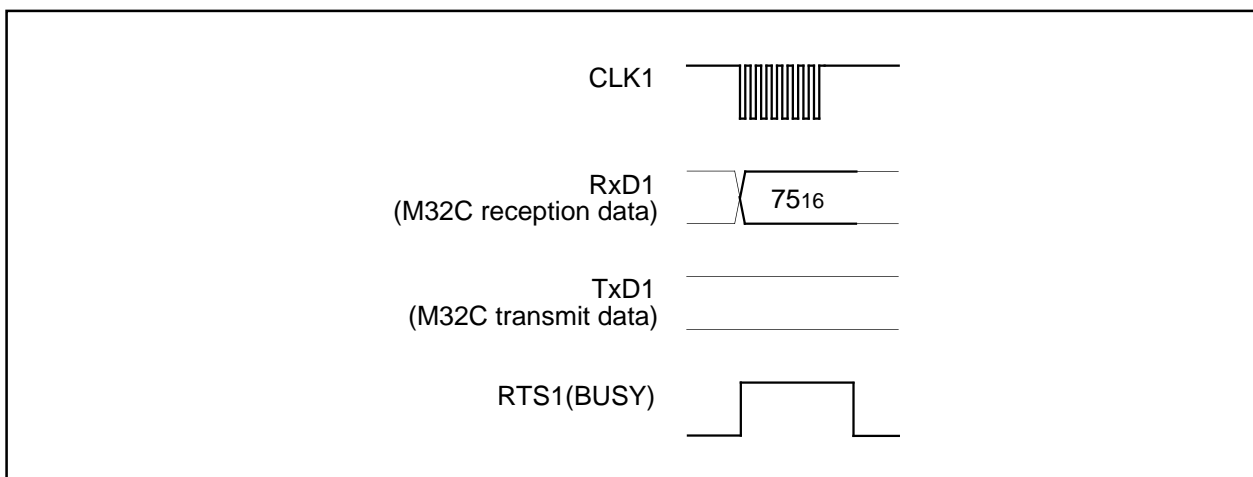


Figure 1.35.13. Timing for disabling the lock bit

ID Check

This command checks the ID code. Execute the boot ID check command as explained here following.

- (1) Transfer the “F516” command code with the 1st byte.
- (2) Transfer addresses A0 to A7, A8 to A15 and A16 to A23 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.

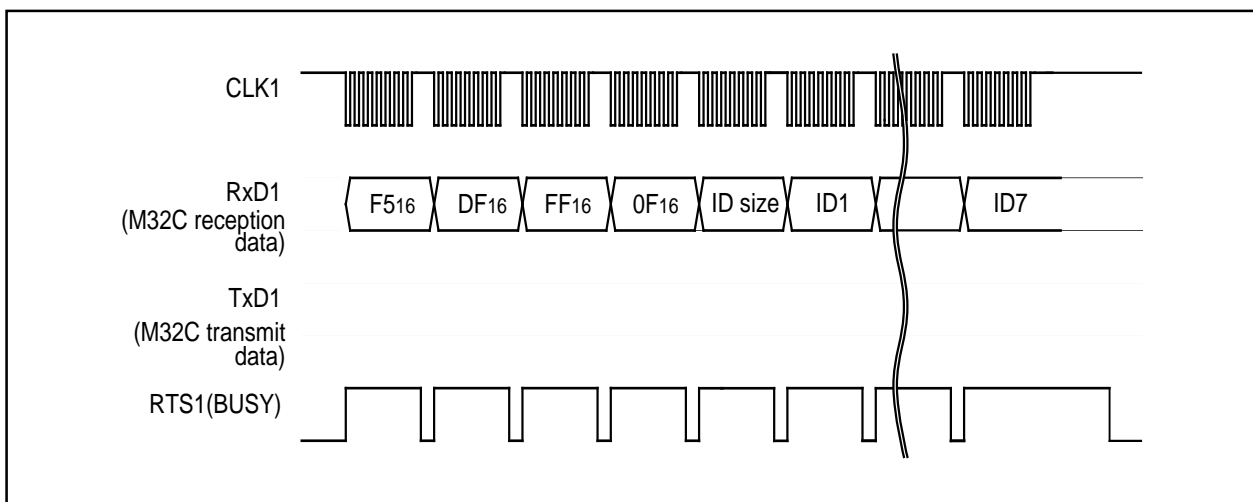


Figure 1.35.14. Timing for the ID check

Download Command

This command downloads a program to the RAM for execution. Execute the download command as explained here following.

- (1) Transfer the "FA16" 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.

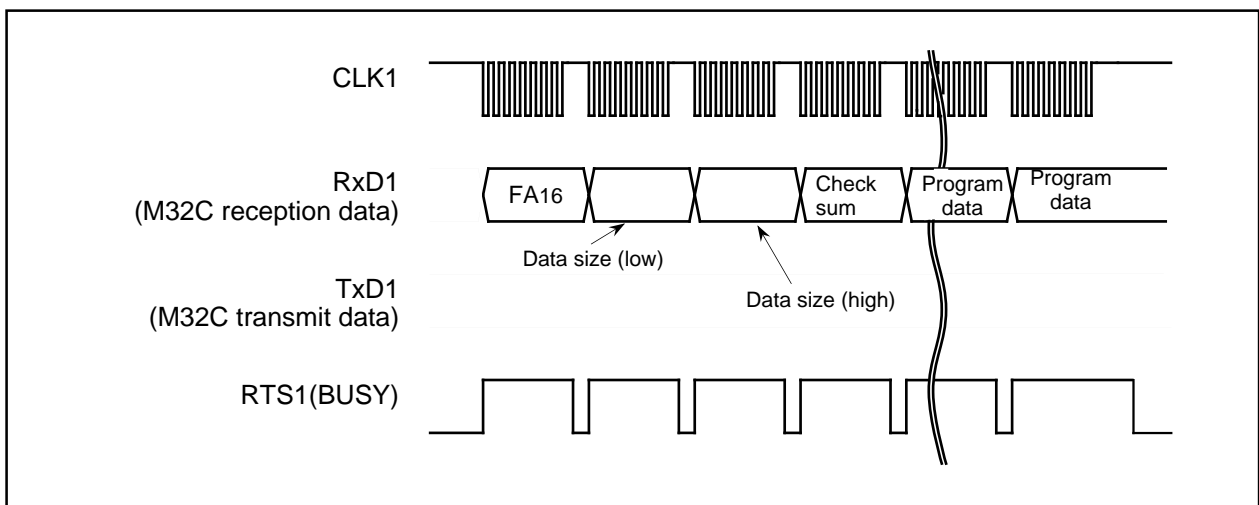


Figure 1.35.15. Timing for download

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.

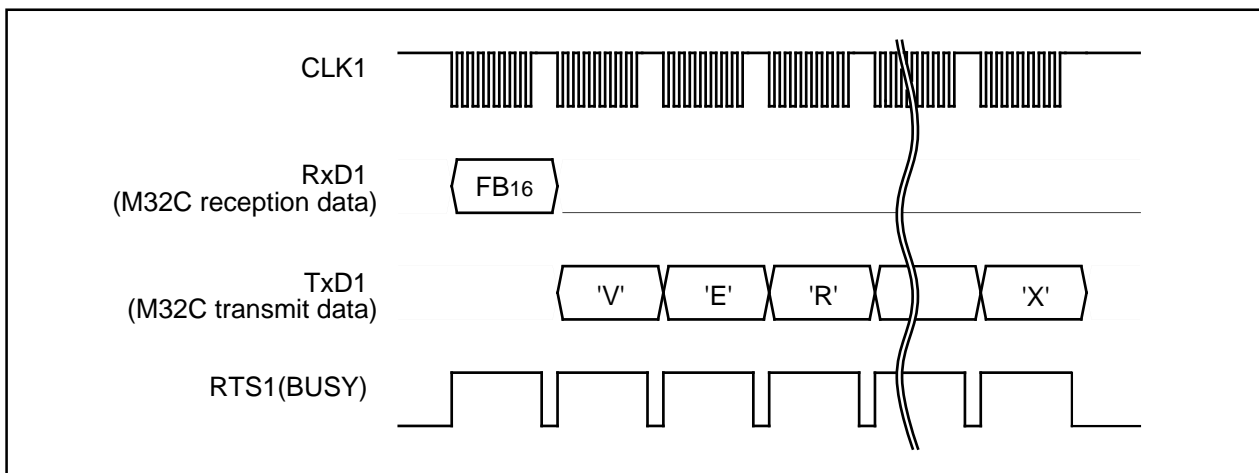


Figure 1.35.16. Timing for version information output

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 A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) From the 4th byte onward, data (D0–D7) for the page (256 bytes) specified with addresses A8 to A23 will be output sequentially from the smallest address first, in sync with the rise of the clock.

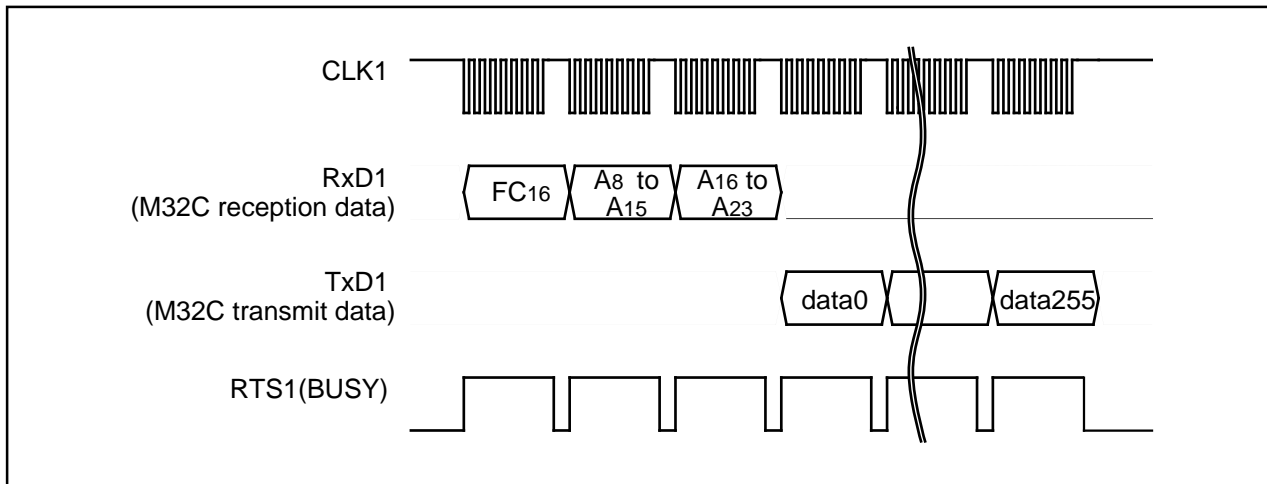


Figure 1.35.17. Timing for boot ROM area output

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 "FD16" 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.

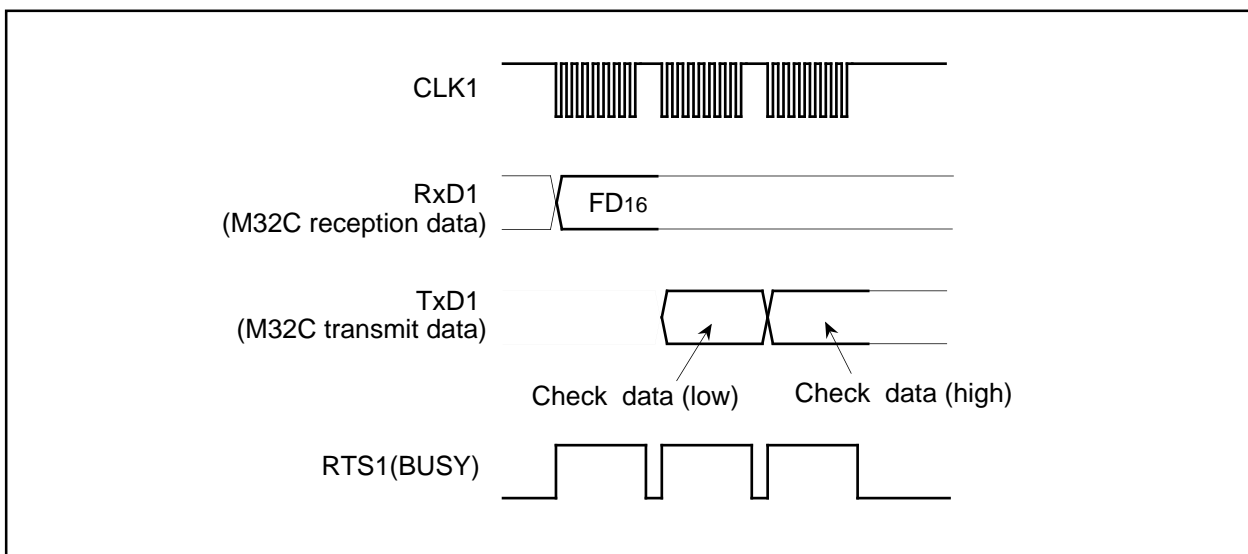


Figure 1.35.18. Timing for the read check data

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 0FFFFDF₁₆, 0FFFFE3₁₆, 0FFFFEB₁₆, 0FFFFEF₁₆, 0FFFFF3₁₆, 0FFFFF7₁₆ and 0FFFFFB₁₆. Write a program into the flash memory, which already has the ID code set for these addresses.

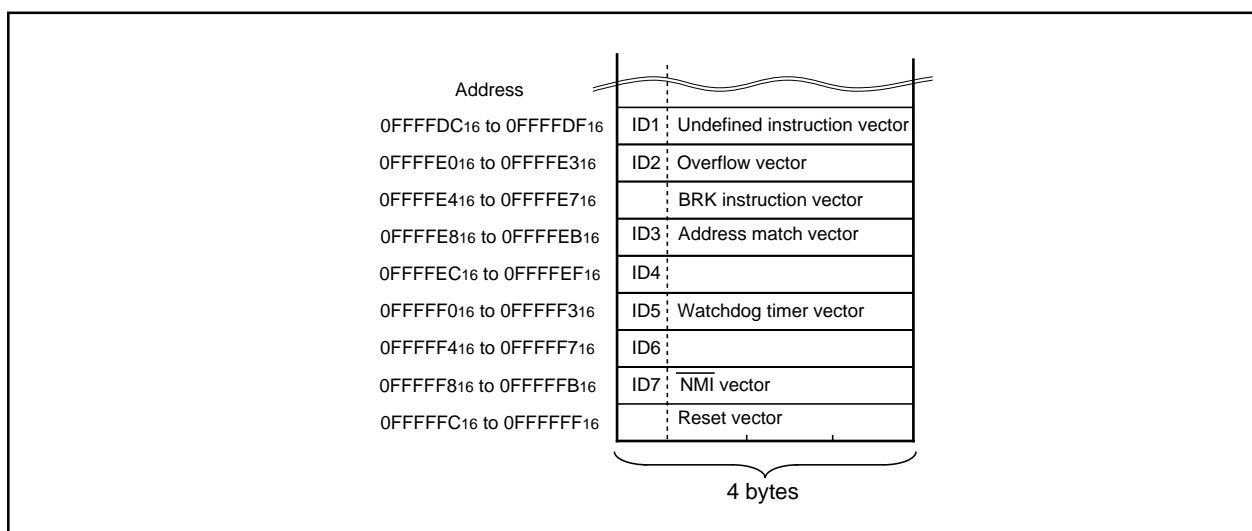


Figure 1.35.19. ID code storage addresses

Data Protection (Block Lock)

Each of the blocks in Figure 1.35.20 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.

0F80000 ₁₆	Block 10 : 64K bytes
0F90000 ₁₆	Block 9 : 64K bytes
0FA0000 ₁₆	Block 8 : 64K bytes
0FB0000 ₁₆	Block 7 : 64K bytes
0FC0000 ₁₆	Block 6 : 64K bytes
0FD0000 ₁₆	Block 5 : 64K bytes
0FE0000 ₁₆	Block 4 : 64K bytes
0FF0000 ₁₆	Block 3 : 32K bytes
0FF8000 ₁₆	Block 2 : 8K bytes
0FFA000 ₁₆	Block 1 : 8K bytes
0FFC000 ₁₆	Block 0 : 16K bytes
0FFFFFF ₁₆	

User ROM area

Figure 1.35.20. 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 (70₁₆). Also, the status register is cleared by writing the clear status register command (50₁₆). Table 1.35.2 gives the definition of each status register bit. After clearing the reset, the status register outputs "80₁₆".

Table 1.35.2. Status register (SRD)

SRD bits	Status name	Definition	
		"1"	"0"
SR0 (bit0)	Reserved	-	-
SR1 (bit1)	Reserved	-	-
SR2 (bit2)	Reserved	-	-
SR3 (bit3)	Block status after program	Terminated in error	Terminated normally
SR4 (bit4)	Program status	Terminated in error	Terminated normally
SR5 (bit5)	Erase status	Terminated in error	Terminated normally
SR6 (bit6)	Reserved	-	Busy
SR7 (bit7)	Write state machine (WSM) status	Ready	-

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, "80₁₆" is output; when writing fails, "90₁₆" is output; and when excessive data is written, "88₁₆" 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 (50₁₆) and clear the status register.

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".

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".

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.

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 (7016). Also, status register 1 is cleared by writing the clear status register command (5016).

Table 1.35.3 gives the definition of each status register 1 bit. "0016" is output when power is turned ON and the flag status is maintained even after the reset.

Table 1.35.3. Status register 1 (SRD1)

SRD1 bits	Status name	Definition	
		"1"	"0"
SR8 (bit0)	Reserved	-	-
SR9 (bit1)	Data receive time out	Time out	Normal operation
SR10 (bit2)	ID check completed bits	00	Not verified
SR11 (bit3)		01	Verification mismatch
		10	Reserved
		11	Verified
SR12 (bit4)	Checksum match bit	Match	Mismatch
SR13 (bit5)	Reserved	-	-
SR14 (bit6)	Reserved	-	-
SR15 (bit7)	Boot update completed bit	Update completed	Not update

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.

ID Check Completed Bits (SR11 and SR10)

These flags indicate the result of ID checks. Some commands cannot be accepted without an ID check.

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.

Boot Update Completed Bit (SR15)

This flag indicates whether the control program was downloaded to the RAM or not, using the download function.

Full Status Check

Results from executed erase and program operations can be known by running a full status check. Figure 1.35.21 shows a flowchart of the full status check and explains how to remedy errors which occur.

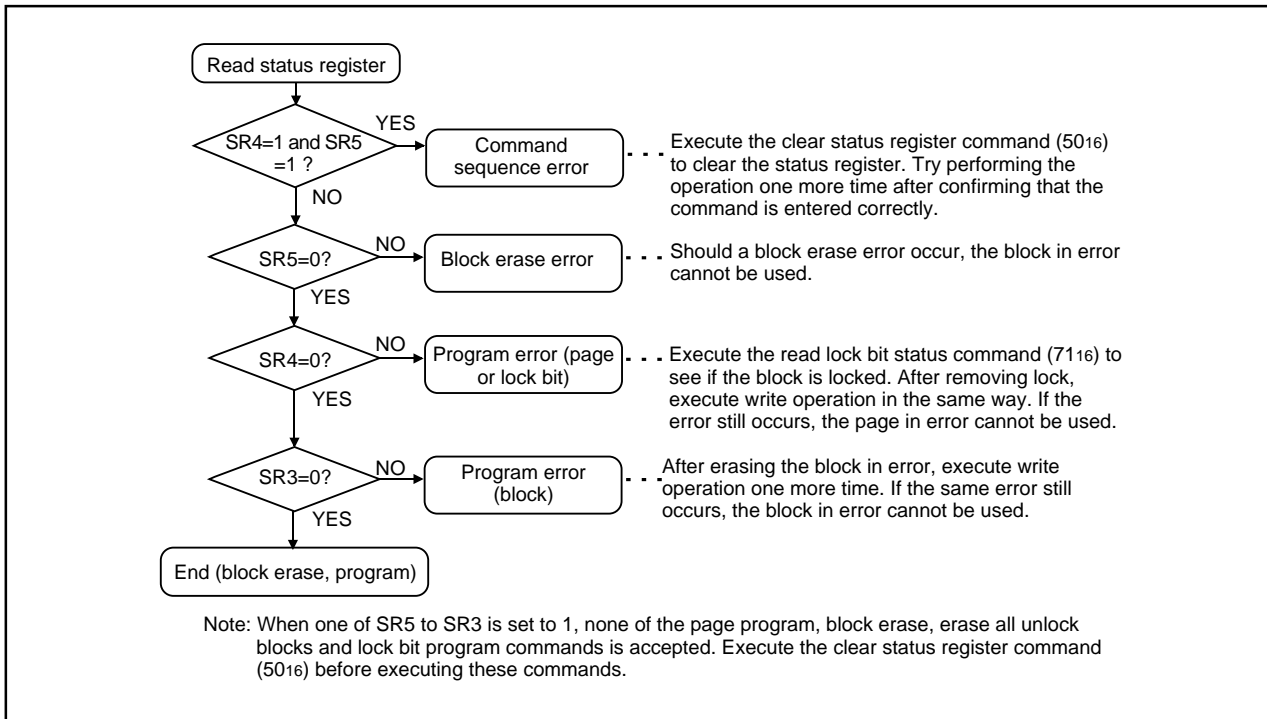


Figure 1.35.21. Full status check flowchart and remedial procedure for errors

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.

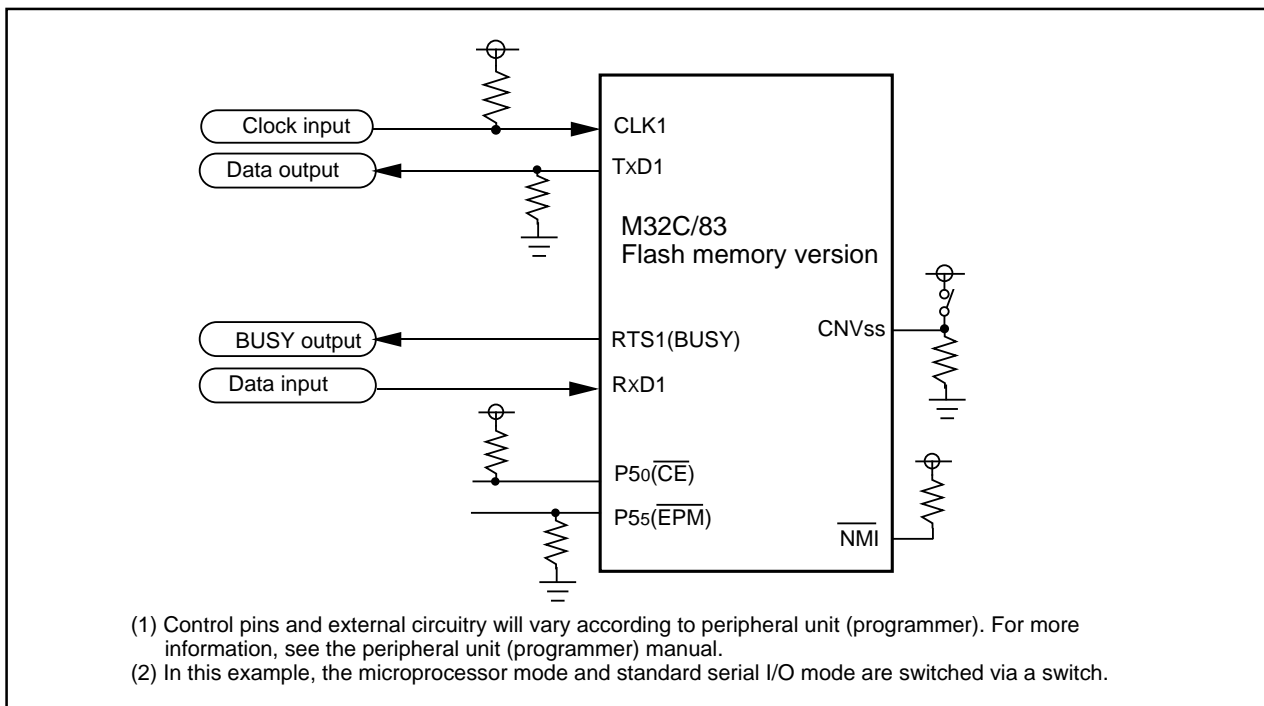


Figure 1.35.22. Example circuit application for the standard serial I/O mode 1

Overview of standard serial I/O mode 2 (clock asynchronous)

In standard serial I/O mode 2, software commands, addresses and data are input and output between the MCU and peripheral units (serial programmer, etc.) using 2-wire clock-asynchronized serial I/O (UART1). Standard serial I/O mode 2 is engaged by releasing the reset with the P65 (CLK1) pin "L" level.

The TxD1 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 1.35.23) are made with a peripheral unit. However, this requires a main clock with a minimum 2 MHz input oscillation frequency. Baud rate can 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 1.35.23).

- (1) Transmit "00₁₆" from a peripheral unit 16 times. (The MCU with internal flash memory sets the bit rate generator so that "00₁₆" can be successfully received.)
- (2) The MCU with internal flash memory outputs the "B0₁₆" 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 "B0₁₆" successfully, change the oscillation frequency of the main clock.

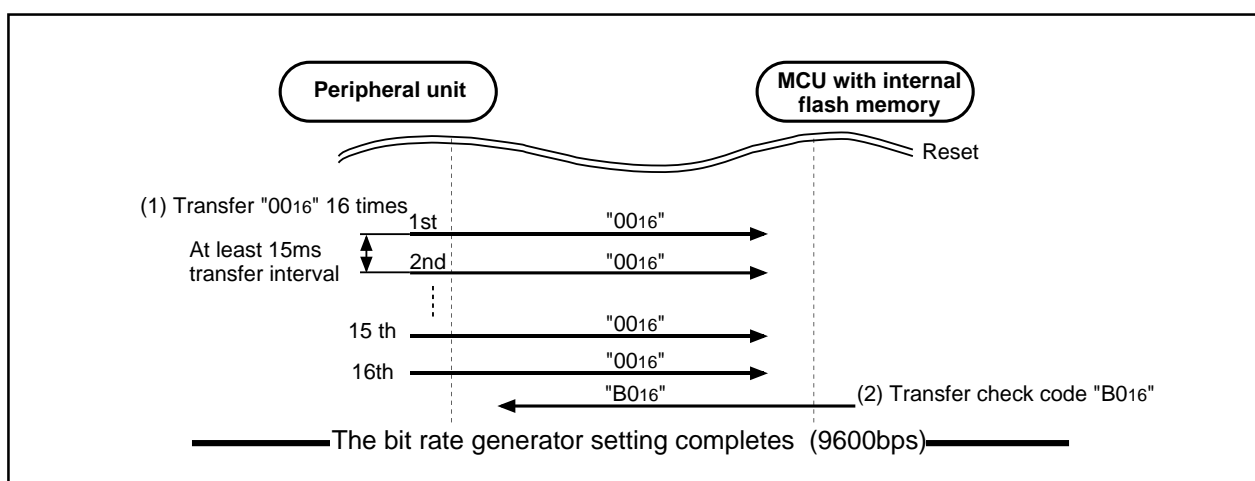


Figure 1.35.23. Peripheral unit and initial communication

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 - 30 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 1.35.4 gives the operation frequency and the baud rate that can be attained for.

Table 1.35.4 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
30MHz	√	√	√	√	—
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 1.35.5 lists software commands. In the standard serial I/O mode 2, erase operations, programs and reading are controlled by transferring software commands via the RxD1 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 1.35.5. 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 "FF16" command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) From the 4th byte onward, data (D0–D7) for the page (256 bytes) specified with addresses A8 to A23 will be output sequentially from the smallest address first in sync with the rise of the clock.

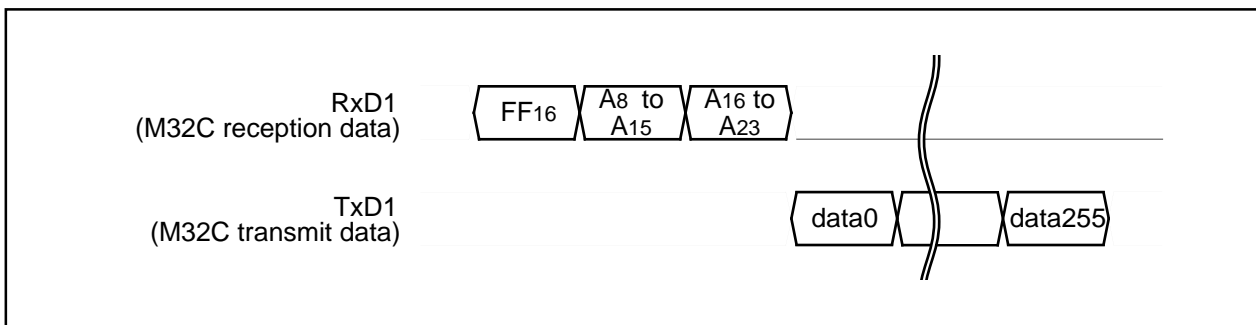


Figure 1.35.24. Timing for page read

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 "4116" command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) From the 4th byte onward, as write data (D0–D7) for the page (256 bytes) specified with addresses A8 to A23 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.

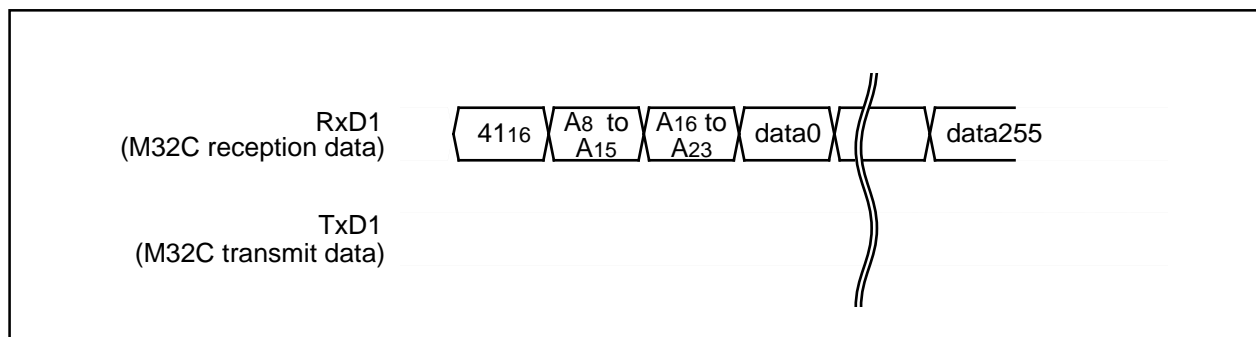


Figure 1.35.25. Timing for the page program

Block Erase Command

This command erases the data in the specified block. Execute the block erase command as explained here following.

- (1) Transfer the "20₁₆" command code with the 1st byte.
- (2) Transfer addresses A₈ to A₁₅ and A₁₆ to A₂₃ with the 2nd and 3rd bytes respectively.
- (3) Transfer the verify command code "D0₁₆" 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 A₁₆ to A₂₃.

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.

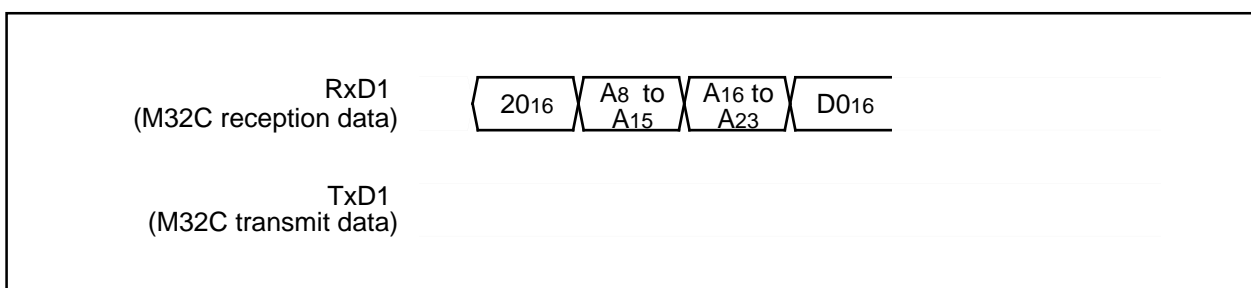


Figure 1.35.26. Timing for block erasing

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 "A7₁₆" command code with the 1st byte.
- (2) Transfer the verify command code "D0₁₆" 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 1.35.27. Timing for erasing all unlocked blocks

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.

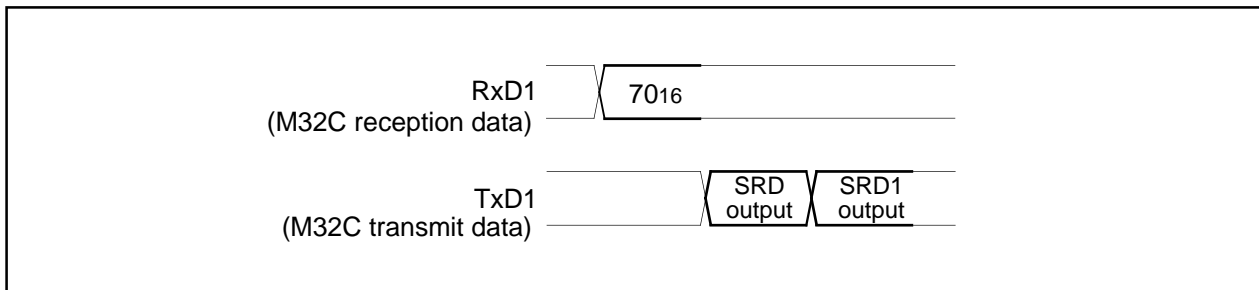


Figure 1.35.28. Timing for reading the status register

Clear Status Register Command

This command clears the bits (SR3–SR5) which are set when the status register operation ends in error. When the “5016” command code is sent with the 1st byte, the aforementioned bits are cleared.

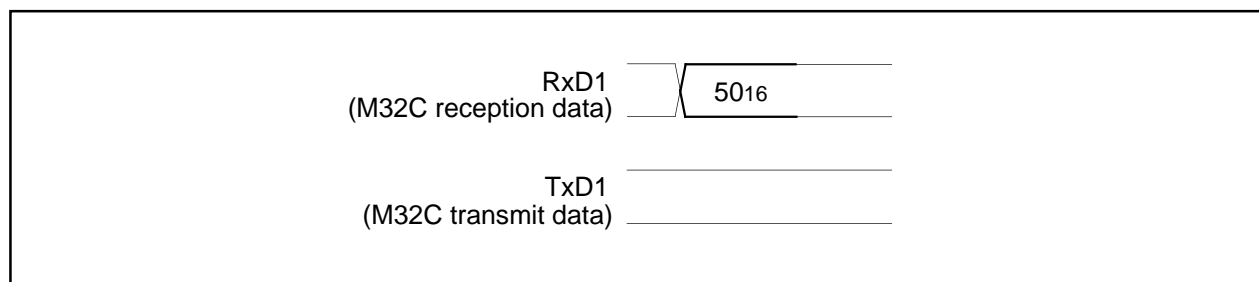


Figure 1.35.29. Timing for clearing the status register

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 “7116” command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 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 A8 to A23.

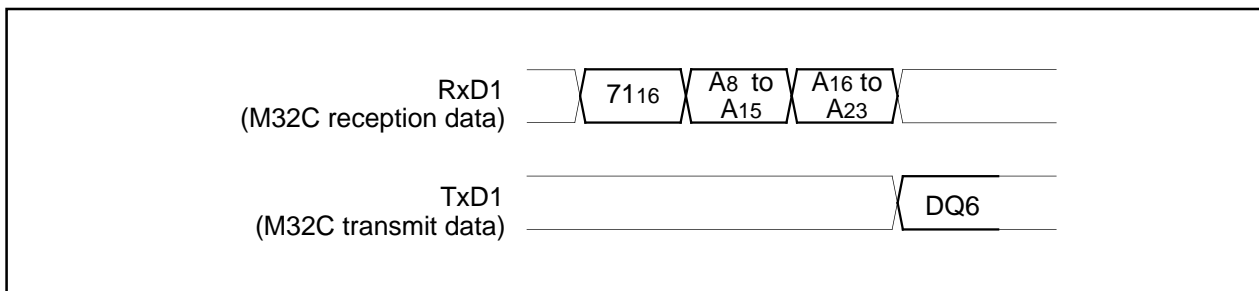


Figure 1.35.30. Timing for reading lock bit status

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 "7716" command code with the 1st byte.
- (2) Transfer addresses A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) Transfer the verify command code "D016" 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 A8 to A23.

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.

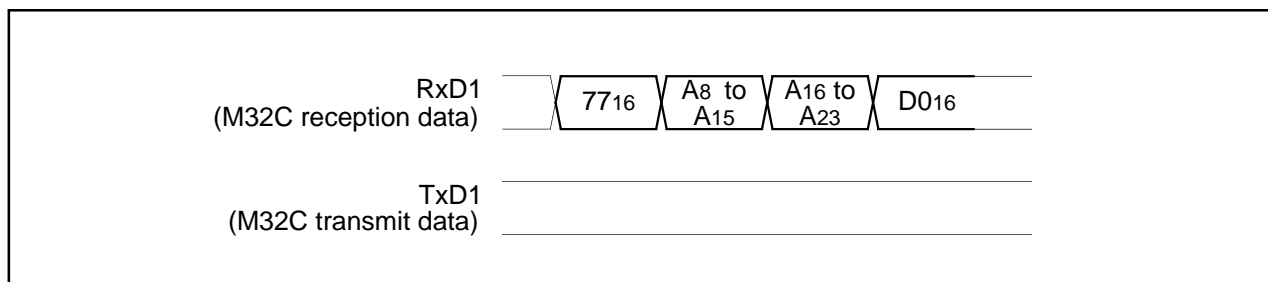


Figure 1.35.31. Timing for the lock bit program

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 "7A16" 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.

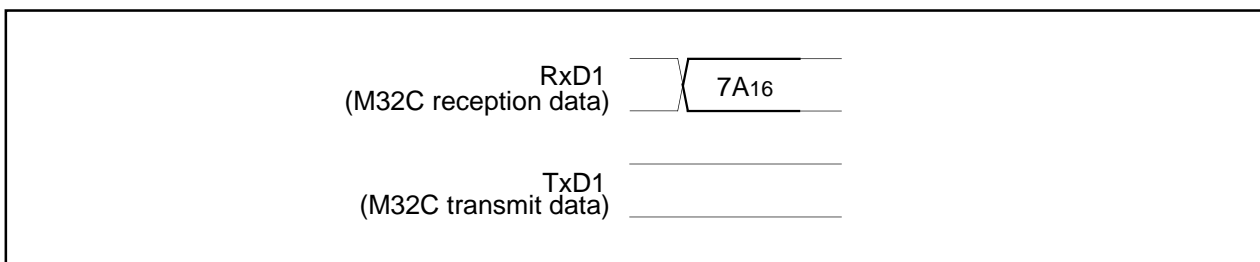


Figure 1.35.32. Timing for enabling the lock bit

Lock Bit Disable Command

This command disables the lock bit. The command code “7516” 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.

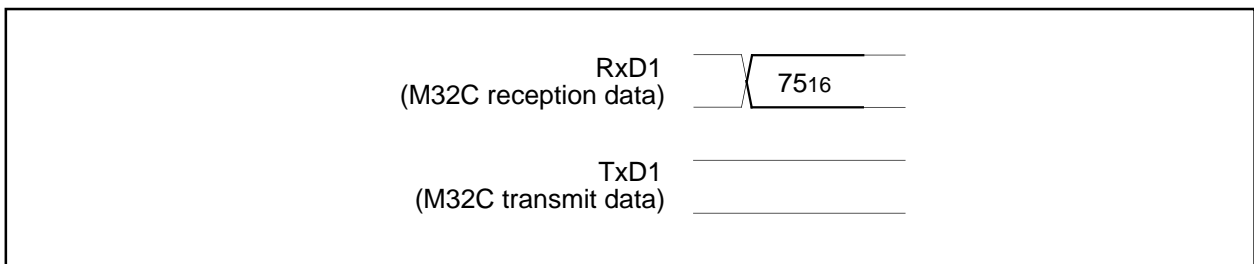


Figure 1.35.33. Timing for disabling the lock bit

ID Check

This command checks the ID code. Execute the boot ID check command as explained here following.

- (1) Transfer the “F516” command code with the 1st byte.
- (2) Transfer addresses A0 to A7, A8 to A15 and A16 to A23 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 1.35.34. Timing for the ID check

Download Command

This command downloads a program to the RAM for execution. Execute the download command as explained here following.

- (1) Transfer the "FA16" 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.

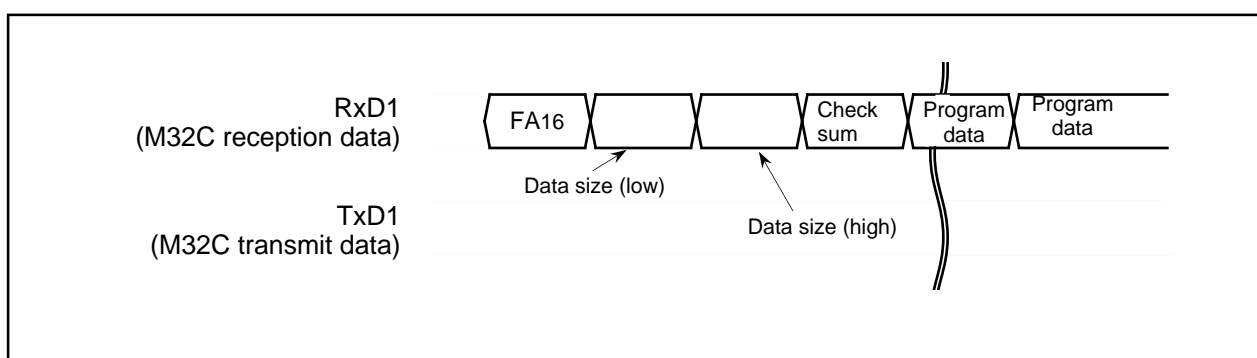


Figure 1.35.35. Timing for download

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.

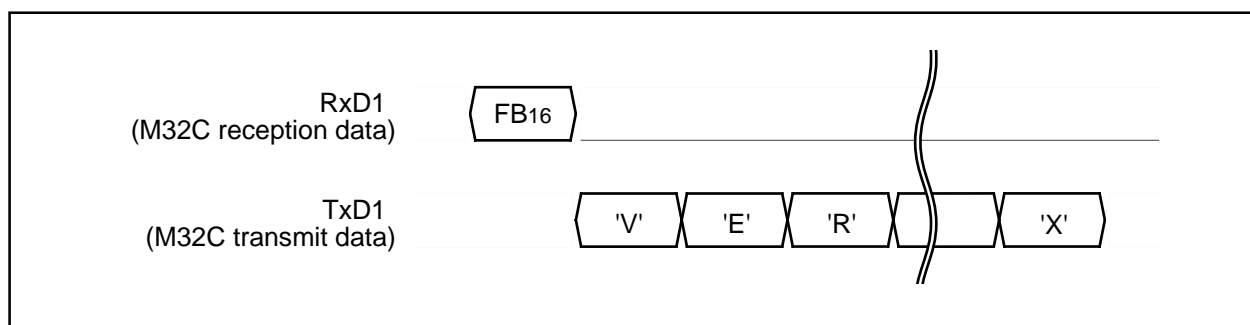


Figure 1.35.36. Timing for version information output

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 A8 to A15 and A16 to A23 with the 2nd and 3rd bytes respectively.
- (3) From the 4th byte onward, data (D0–D7) for the page (256 bytes) specified with addresses A8 to A23 will be output sequentially from the smallest address first, in sync with the rise of the clock.

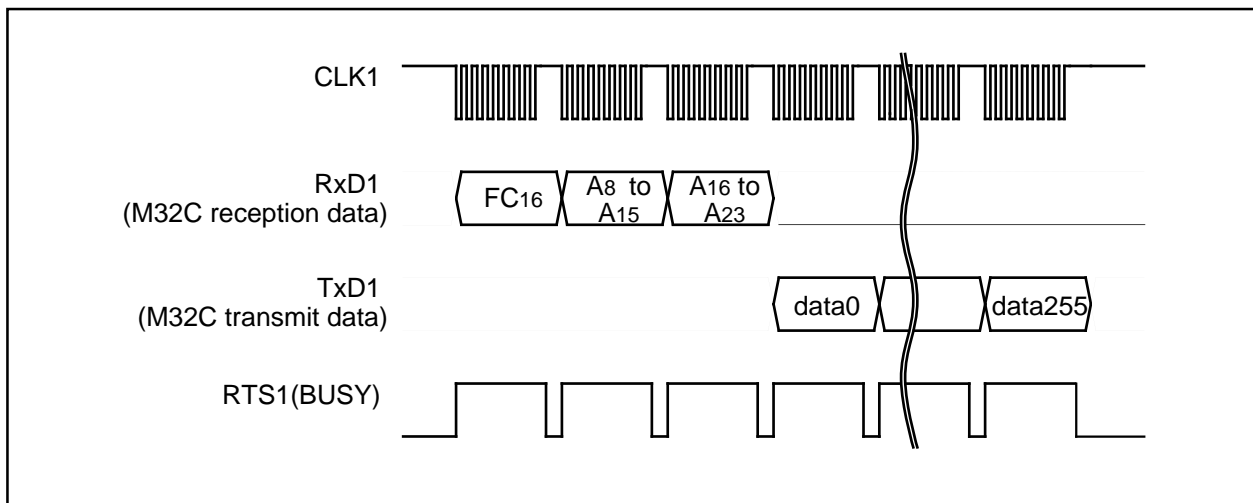


Figure 1.35.37. Timing for boot ROM area output

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 "FD16" 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.

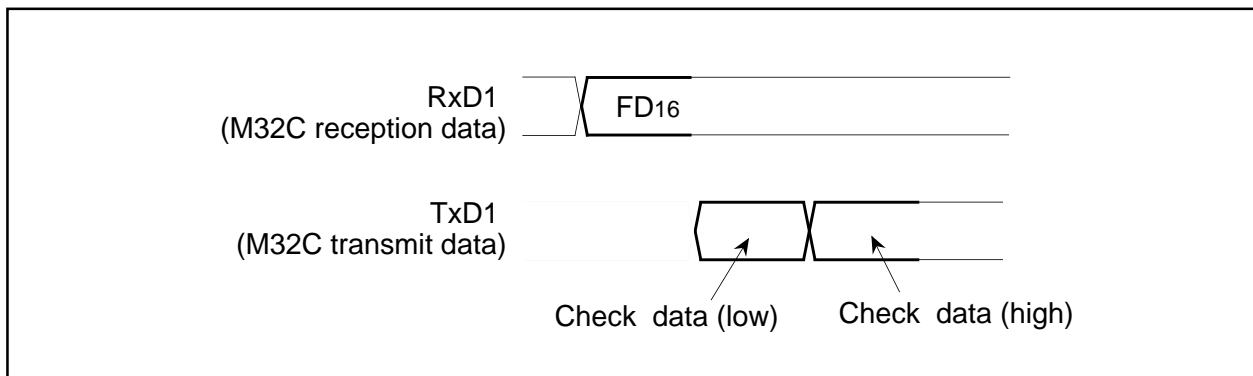


Figure 1.35.38. Timing for the read check data

Baud Rate 9600

This command changes baud rate to 9,600 bps. Execute it as follows.

- (1) Transfer the "B016" command code with the 1st byte.
- (2) After the "B016" check code is output with the 2nd byte, change the baud rate to 9,600 bps.

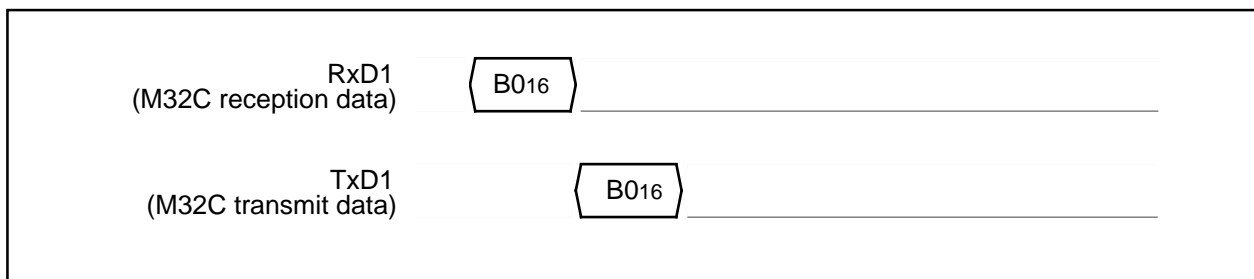


Figure 1.35.39. Timing of baud rate 9600

Baud Rate 19200

This command changes baud rate to 19,200 bps. Execute it as follows.

- (1) Transfer the "B116" command code with the 1st byte.
- (2) After the "B116" check code is output with the 2nd byte, change the baud rate to 19,200 bps.

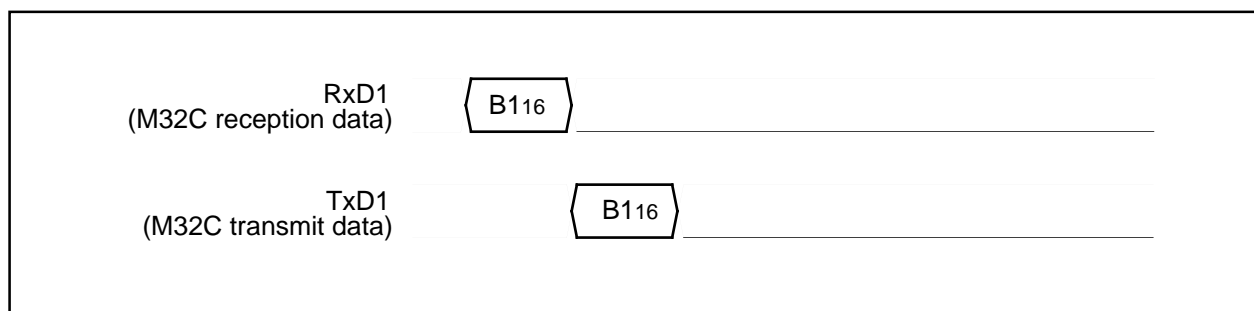


Figure 1.35.40. Timing of baud rate 19200

Baud Rate 38400

This command changes baud rate to 38,400 bps. Execute it as follows.

- (1) Transfer the "B216" command code with the 1st byte.
- (2) After the "B216" check code is output with the 2nd byte, change the baud rate to 38,400 bps.

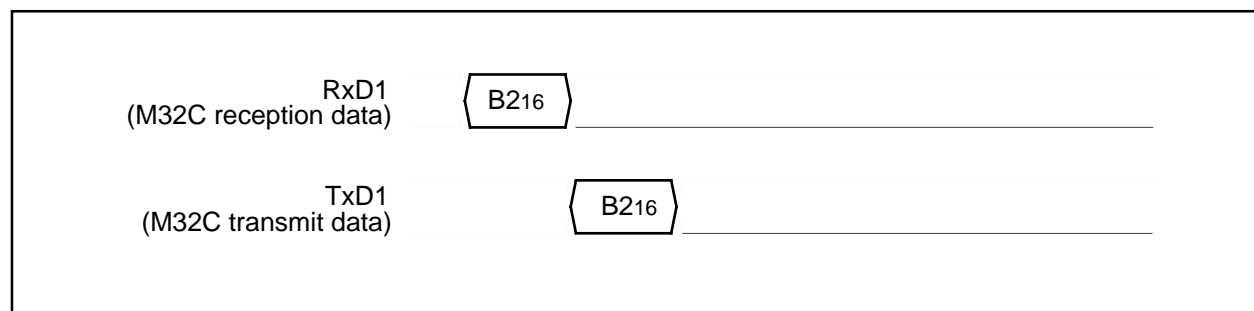


Figure 1.35.41. Timing of baud rate 38400

Baud Rate 57600

This command changes baud rate to 57,600 bps. Execute it as follows.

- (1) Transfer the "B316" command code with the 1st byte.
- (2) After the "B316" check code is output with the 2nd byte, change the baud rate to 57,600 bps.

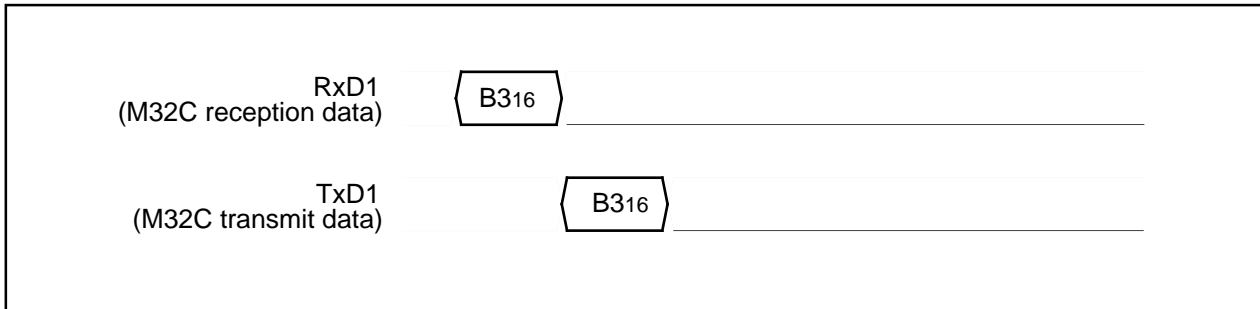


Figure 1.35.42. Timing of baud rate 57600

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.

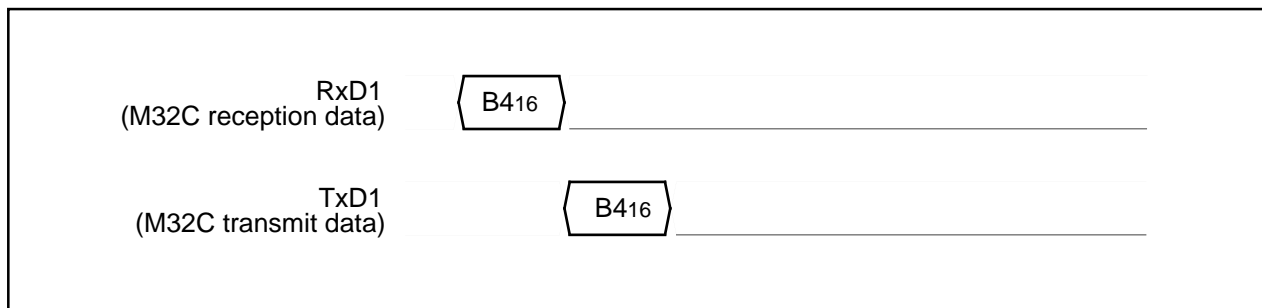


Figure 1.35.43. Timing of baud rate 115200

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 0FFFFDF₁₆, 0FFFFE3₁₆, 0FFFFEB₁₆, 0FFFFEF₁₆, 0FFFFF3₁₆, 0FFFFF7₁₆ and 0FFFFFB₁₆. Write a program into the flash memory, which already has the ID code set for these addresses.

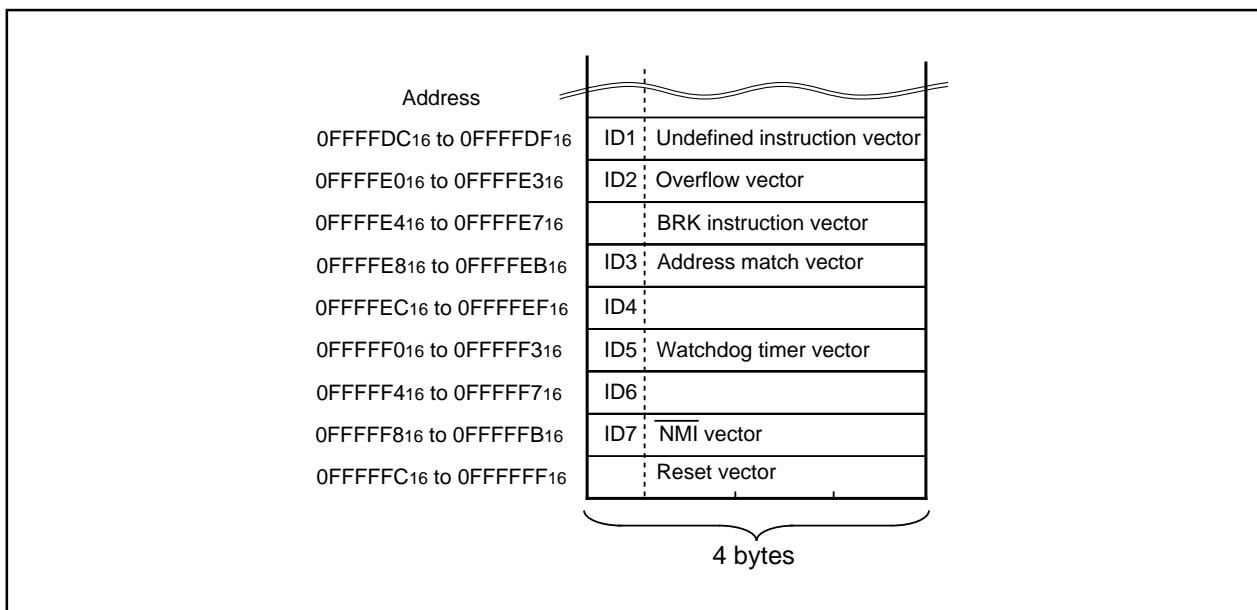


Figure 1.35.44. ID code storage addresses

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.

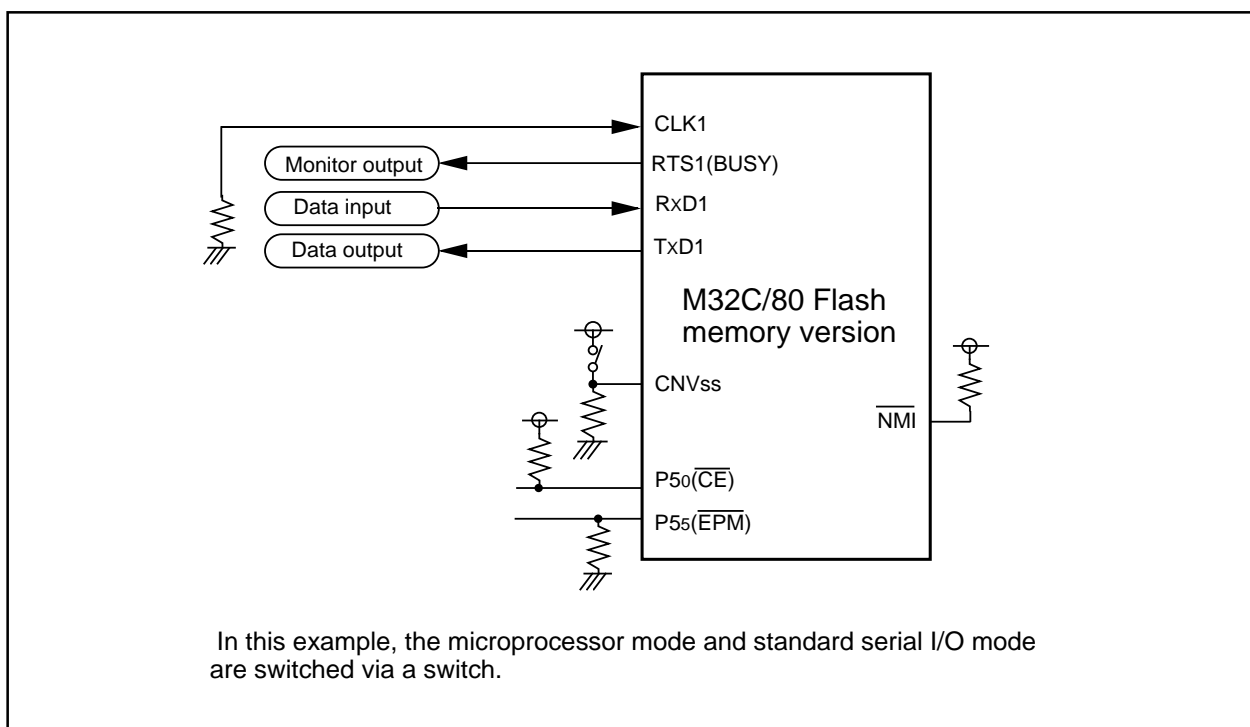


Figure 1.35.45. Example circuit application for the standard serial I/O mode 2

REVISION HISTORY

M32C/83 GROUP DATA SHEET

Rev. Date	Page	Description	
		Error	Correct
B1 1/8/2001			100-pin version is added. Flash memory version is added. Others
B1 30/8/ 2001	2,3	Tables 1.1.1 and 1.1.2 Interrupt: 12 internal/external sources (intelligent I/O and CAN module) Supply voltage	Delete 3.0 to 3.6V (f(XIN)=20MHz without wait) add
	3	A-D converter 10 bits (8 channels) x 2 circuits, max 26 inputs	10 bits x 2 circuits, standard 10 inputs, max 26 inputs
	7	Table 1.1.3 Pin 26	CANIN addition
	10-12	Figures 1.1.4, 1.1.5, Table 1.1.7	CANIN is added to Pin 17(GP) and pin 19(FP)
	11	Figure 1.1.5 Pin 97 AN00	AN0
	12	Pin 32 (FP) Vcc Pin 34 (FP) Vss	Delete Delete
	13	Vcc position to pin 64(FP) Vss position to pin 66(FP) RxD4/SCL4/STxD4 position to pin 98 (FP)	Pin 62 Pin 64 Pin 100
	14	Table 1.1.5 AN20 to AN27 AN30 to AN37	AN00 to AN07 AN20 to AN27
	17	Table 1.1.12 P120 to P127 ISCLK description AN10 to AN17	Delete AN150 to AN157
	18	Figure 1.1.6 System clock oscillation circuit	PLL oscillation stop detect addition
	28, 29	Figure 1.4.3 (122), (167) (123), (168)	Group0 receive buffer register, Group1 receive buffer register Group0 transmit buffer/receive data register, Group1 transmit buffer/receive data register
	46	Note 1: Addresses 03C9 ₁₆ , 03CB ₁₆ to 03D3 ₁₆	Addresses 03A0 ₁₆ , 03A1 ₁₆ , 03B9 ₁₆ , 03BC ₁₆ , 03BD ₁₆ , 03C9 ₁₆ , 03CB ₁₆ to 03D3 ₁₆
	48	Figure 1.6.1 Note 2	Addition. Displace after the former Note 2
	70	Figure 1.8.6 When reset of PLL control register 0 0X11 0100	0011 0100
	72	Figure 1.8.8 Count value set bit Count start bit Count stop/start Note 2	Division rate select bit Operation enable bit Divider stops/starts Delete
	76	Line 10 Addition	Stop mode is canceled before setting this bit to "1".
	77	Line 8 1:Sub clock is selected	1: Clock from ring oscillator is selected
	135	Figure 1.14.2 Values that can be set Pulse width modulation mode (8-bit PWM) 00 ₁₆ to FF ₁₆ (High-order and low-order address)	00 ₁₆ to FE ₁₆ (High-order address) 00 ₁₆ to FF ₁₆ (Low-order address)
	230	Line 5, Bit 1 TrmActive	TrmData
	266	Table 1.23.11 Waveform generate control register 1 when clock synchronous serial I/O -	√
280	Table 1.23.17 Note 1:	When the transfer clock and transfer data are transmission, transfer clock is set to at least 6 divisions of	

REVISION HISTORY

M32C/83 GROUP DATA SHEET

Rev. Date	Page	Description	
		Error	Correct
			the base timer clock. Except this, transfer clock is set to at least 20 divisions of the base timer clock. Addition
	285	Note 2 Figure 1.23.37	Delay timing of base timer
	284	Table 1.24.1 A-D conversion start condition • Timer B2 interrupt	• Timer B2 interrupt occurrences frequency counter overflow
B2 Feb/1/ 2002	2, 3, 4	Table 1.1.1, 1.1.2 Clock generating circuit 4 built-in...circuit PLL freq. synthe. Power consumption 29mA 44mA	3 built-in clock generation circuits Delete 26mA 38mA
	6,10, 11	Fig 1.1.3-1.1.5	Note: P70 and P71 are N-channel...output.-> Add
	18	Fig 1.1.6 System clock generator PLL Oscillation stop detection	Delete Ring oscillator
	24	7th line	Since the value.....due to the interruption. -> Add
	27	Fig 1.4.3 (1) (2) Processor mode register 1 (3) System clock control register 0 (10) Oscillation stop detect register (17) VDC control register 1 (21) DRAM refresh interval set register (46) CAN interrupt 1 control register (47) CAN interrupt 2 control register	XX00 X000 -> X000 00XX 80 -> 0000 X000 XXXX 0000 -> 00 Add XXXX ?000 -> ?? Add Add
	28	Fig 1.4.3 (2) (70) CAN interrupt 0 control register	Add
	28-31	Fig 1.4.3(2) (97)-(104), Fig 1.4.3(3) (142)-(149), Fig 1.4.3(4) (187)-(194), Fig 1.4.3(5) (222)-(229) Group 0 -3 time measurement/ waveform generation register 0-7	00 -> ??
	29, 30	Fig 1.4.3(3) (124), Fig 1.4.3(4) (169) Group 0,1 SI/O communication buffer register Fig 1.4.3(3) (125), Fig 1.4.3(4) (170) Group 0,1 receive data register (129) Group 0 SI/O comm cont register (186) Group 1 SI/O expansion trans cont register	Group 0,1 SI/O receive buffer register Group 0,1 transmit buffer/receive data register X000 XXX -> 000 X011 0000 00XX -> 0000 0XXX
	31	Fig 1.4.3(5) (238)-(241) Group 3 waveform generate mask register 4-7	00 -> ??
	32	Fig 1.4.3(6) (270)-(308) (270)-(302)	Note added Reset value changed
	33	Fig 1.4.3(7) (309)-(338) (314)-(318),(321),(323),(329),(331),(336) (337) CAN0 clock control register	Note added Reset values changed CAN0 sleep control register
	36	Fig 1.4.3(10) (461) A-D control register 2	X000 XXX0 -> X000 0000

REVISION HISTORY

M32C/83 GROUP DATA SHEET

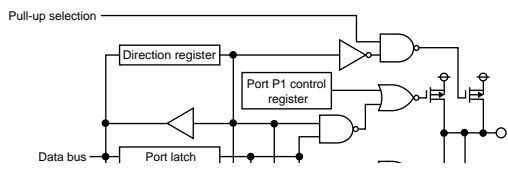
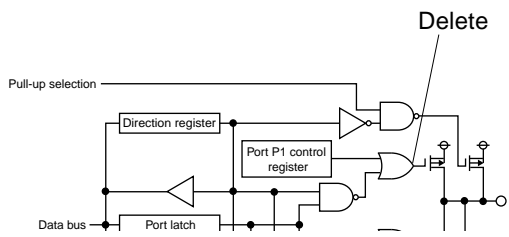
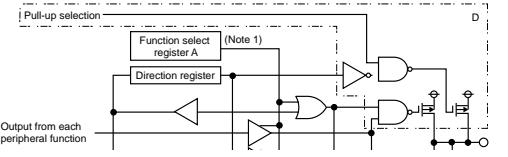
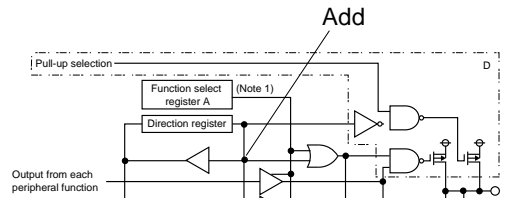
Rev. Date	Page	Description	
		Error	Correct
	38	Address 007F ₁₆ Address 0081 ₁₆ Address 009D ₁₆	CAN interrupt 1 control register added CAN interrupt 2 control register added CAN interrupt 0 control register added
	61	(10) Software wait, 11th line SFR area is accessed.....with "2 waits".	Add
	67	Fig 1.8.2 System clock control register 0 When reset: 08 ₁₆ Note 3: When selecting fc,.....as input port.	0000 X000 ₂ Delete
	79	Fig 1.8.9 Note 7: When using PLL.....cannot be used.	Delete
	90	Fig 1.9.3, Symbol CAN0iCi	CANiIC
	110	Table 1.11.1, DMA request factors	Intelligent I/O interrupt -> add
	128	Fig 1.12.4, the number of cycles	Change
	133	Fig 1.14.3, Timer Ai mode register, MR0 Port output.....registers A and B.	Port output.....registers A, B and C.
	137, 138, 142, 144	Table 1.14.1, 1.14.2, 1.14.4, 1.14.5 TAiOUT pin function	Function select register C -> add
	137	Fig 1.14.7 Timer Ai mode register bit 2 (MR0) Location of Note 3 (b7, b6): 11	Function select register C -> add 10
	139	Fig 1.14.8 Timer Ai mode register bit 2 (MR0)	Function select register C -> add
	143, 145	Fig 1.14.11, 1.14.12 Timer Ai mode register bit 2 (MR0) Location of Note 3 (b7, b6): 11	Function select register C -> add 10
	159	Fig 1.16.5 Timer Ai mode register bit 2 (MR0)	Function select register C -> add
	161	Fig 1.16.6 Reload register n = 1 to 255	Reload register
	172	Fig 1.17.4 UARTi transmit/receive control register 0 Note 2	Function select register C -> add
	173	Fig 1.17.5 UARTi transmit/receive control register 1 Function of bit 7: Error signal output enable bit	Set to "0"
	199	Fig 1.22.1 Clock control register Time stamp count register	Sleep control register Time stamp register
	200	Fig 1.22.3 Bit 4 0: Forced reset Bit 10 Time stamp count reset bit	0: Reset requested Time stamp counter reset bit
	201	5th line: In no case will the CAN module be Bit 3: BasicCAN mode bit	In no case will the CAN be..... Bit 3: BasicCAN mode select bit
	202	Bit 8,9: Timestamp prescaler bits Bit 11, 1st line: Receive Error Counter Transmit Error Counter	Bit 8, 9: Timestamp prescaler select bits Receive Error Counter Register Transmit Error Counter Register
	209	Fig 1.22.8 bit 4: Reserved bit	Sampling number
	210	6. CAN0 configuration register	Explanation of Bit 4 -> add

REVISION HISTORY

M32C/83 GROUP DATA SHEET

Rev. Date	Page	Description	
		Error	Correct
	211	Note:1 Setting the COCTRL0 register's Reset0 bit to 1 resets <u>the CAN protocol control unit, with</u> the COTSR register thereby initialized to 0000 ₁₆ . Also, setting the TSReset (timestamp <u>count</u> reset) bit to 1 initializes the COTSR register to 0000 ₁₆ on-the-fly (<u>while the CAN protocol control unit remains operating</u>).	Note 1: Setting the COCTRL0 register's Reset0 <u>and</u> Reset1 bits to 1 resets <u>the CAN, and</u> the COTSR register is thereby initialized to 0000 ₁₆ . Also, setting the TSReset (timestamp <u>counter</u> reset) bit to 1 initializes the COTSR register to 0000 ₁₆ on-the-fly (<u>while the CAN remains operating: CAN0 status register's State Reset bit is "0"</u>).
	212	Tq period = (C0BRP+1)	Tq period = (C0BRP+1)/CPU clock
	220	Fig 1.22.19 b0 b2	b2 b1
	226	Fig 1.22.25 bit 0 bit 1, When transmit, TrmData bit 3 bit 6, 7, Transmit request flag	Note 2 -> add When transmit, TrmActive Note 2 -> add Transmit request bit
	229	Fig 1.22.26, explanation of function	Change
	230, 231, 232	Fig 1.22.27, 1.22.28, 1.22.29 Explanation of function	Message slot j (j=0 to 15) -> change
	233	Fig 1.22.30, CAN0 message slot butter i data m Symbol C0SLOT0_m (m=0 to 3) C0SLOT0_m (m=4 to 7) C0SLOT1_m (m=0 to 3) C0SLOT1_m (m=4 to 7)	C0SLOT0_n (n=m+6, m=0 to 3) C0SLOT0_n (n=m+6, m=4 to 7) C0SLOT1_n (n=m+6, m=0 to 3) C0SLOT1_n (n=m+6, m=4 to 7)
	235	Table 1.23.1 Group 2, WG register Group 3 Comm shift register	- -> 8chs 16bits x 2chs -> -
	240	Fig 1.23.5, Group i base timer cont reg 0 Bit 2 to bit 6, explanations on fPLL	Delete
	245	Table 1.23.2, Count reset condition, Group 2, 3 (3) Reset request circuit	(3) Reset request circuit (group 2 only)
	245	Fig 1.23.10 fPLL	Delete
	246	Fig 1.23.11	Newly added
	248	Fig 1.23.13, the values when reset: 00 ₁₆	0000 ₁₆
	249	Table 1.23.3, select function, digital filter function Strips off pulses less than 3 cycles long from f1 and the base timerclock.	Pulses will pass when they match either f1 or the base timerclock 3 times.
	250	Fig 1.23.14, (c)	Change
	252	Fig 1.23.16, reset values for both registers	0000 ₁₆ -> XXXX ₁₆
	256	Fig 1.23.20, When WG register is "xxxb ₁₆ "	When WG register is "xxxa ₁₆ "
	270	Table 1.23.12 Transmission start condition • Write data to transmit buffer register Interrupt request generation timing •When transmitting - When SI/O transmit buffer register is..... •When receiving When.....to SI/O communication buffer register	• Write data to transmit buffer - When transmit buffer is When.....to SI/O receive buffer register

Rev. Date	Page	Description	
		Error	Correct
	270	Select function This.....TxD pin output and RxD pin input.	This.....ISTxD pin output and ISRxD pin input.
	271	Table 1.23.13, Transfer clock input •Selects I/O with function.....	•Select I/O port with function.....
	271	Fig 1.23.31 Write to communication buffer (Input to INPC2/ISRxD0 pin)	Write to receive buffer (Input to INPCi2/ISRxDi pin (i=0, 1))
	272	Table 1.23.14 Transmission start condition • Write data to transmit buffer register Interrupt request generation timing •When transmitting - When SI/O transmit buffer register is..... •When receiving When....to SI/O communication buffer register Error detection • Overrun error:before contents of receive buffer register....	• Write data to transmit buffer - When transmit buffer is When.....to SI/O receive buffer register before contents o SI/O receive buffer register.....
	273	Fig 1.23.32 Write to communication buffer	Write to receive buffer
	273	Fig 1.23.33 (Input to INPC2/ISRxD0 pin)	(Input to INPCi2/ISRxDi pin (i=0, 1))
	279	Table 1.23.17 Transmission start condition • Write data to transmit buffer register Reception start condition • Write data to transmit buffer register Interrupt request generation timing •When receiving When....to SI/O communication buffer register Select function This.....TxD pin output and RxD pin input.	• Write data to SI/O transmit buffer register • Write data to SI/O transmit buffer register When.....to SI/O receive buffer register This.....ISTxD pin output and ISRxD pin input.
	286	Fig 1.24.4, A-D control register 2 When reset: X000 XXX02	X000 00002
	287, 288	Fig 1.24.5, Note 4 and Fig 1.24.6, Note 3 by A-D sweep pin select bits.....by analog input port select bits.....
	292	(e) Replace function of input pin 2nd line:of A-D0 and A-D2.of A-D0 and A-D1.
	293	(f) , at the end of 2nd line (g) 3rd line:, input via AN0 to AN7 is.....	as AN0.....respectively. -> add , input via AN0 to AN7 is.....
	294	Table 1.24.9 P00 analog input P01 analog input	P95 analog input P96 analog input
	312	Fig 1.29.1, P00 to P07, P20 to P27: -	○

Rev. Date	Page	Description	
		Error	Correct
313	Fig 1.29.2	 <p>Circuit (C) P15 to P17, Circuit (B): -</p>	 <p>Delete</p> <p>Delete ○</p>
314	Fig 1.29.3	 <p>P121, P122, Circuit (B): -</p>	 <p>Add</p> <p>○</p>
326	Fig 1.29.16, Pull-up register 2, Note 1		Delete
331	Table 1.29.5 Bit 0, 1: Three-phase PWM output (\overline{U}) Bit 1, 0: Three-phase PWM output (\overline{U})		1: Three-phase PWM output (\overline{U}) 0: Three-phase PWM output (\overline{U})
331	Table 1.29.6, PS4 PSL4 Bit 1, UART0 Bit 2, UART4 Bit 3, UART1 Bit 4, 5 UART1 A4 B4		PS3 PSL3 UART3 UART3 UART3 UART4 A3 B3
334	VDC		Add
337	A-D Converter 1st line: A-D 1st line: A-D 2nd line:and to bit 0 of A-D control register 2...		A-D i (i=0,1) A-D iand to each bit of A-D i control register 2.....
340	(3) External interrupt • Level sense, 2nd line: (When X_{IN} =20MHz and...) 3rd line: (....., at least 250 ns.....) • When the polarity of INT0 to INT5 pins is.....		(When X_{IN} =30MHz and) (....., at least 233 ns.....) • When the polarity of INT0 and INT5 pins is.....
341	Reducing power consumption, (2) 1st line, last line: AN04, AN07		AN4, AN7
343	Table 1.30.3 GOCR 00EF16 G1RI 012F16 U0BRG 036116 U0TB 036316, 036216 U1BRG 036916 U1TB 036B16, 036A16		G0RI 00EC16 G1RI 012C16 U0BRG 036916 U0TB 036B16, 036A16 U1BRG 02E916 U1TB 02EB16, 02EA16
343	Notes on CNVss pin reset at "H" level		Add

REVISION HISTORY

M32C/83 GROUP DATA SHEET

Rev. Date	Page	Description	
		Error	Correct
	344-380	Electric characteristics	Add
	385	Fig 1.34.1, Address 0377 ₁₆ Bit 0: RY/B \bar{Y} status bit	Address 0057 ₁₆ RY/B \bar{Y} signal status bit
	385	Flash memory control register (address 0057 ₁₆) 1st line:the RY/B \bar{Y} status flag.....the RY/B \bar{Y} signal status bit.....
	390	13th line of Page Program Command (41 ₁₆) and Fig 1.34.3: RY/B \bar{Y} status flag	RY/B \bar{Y} signal status bit
	391	11th line of Block Erase Command (20 ₁₆ /D0 ₁₆) and Fig 1.34.4: RY/B \bar{Y} status flag	RY/B \bar{Y} signal status bit
	392	Fig 1.34.5: RY/B \bar{Y} status flag	RY/B \bar{Y} signal status bit
	400	3rd paragraph, 1st line, set the CLK ₁ pin to "H" level and...., set the CLK ₁ pin to "H" level and the Tx _{D1} pin to "L" level, and.....
	400	3rd paragraph, 2nd line The CLK ₁ pin is connected to V _{cc}resistance.	Add
	401	P67 When using standard.....transfer.	Add
	419	Fig 1.35.22, Data output	Pulled down
	421	How frequency is identified, 2nd line: (2 - 20MHz)	(2 - 30MHz)

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 Semiconductor product distributor for further details on these materials or the products contained therein.

MITSUBISHI SEMICONDUCTORS
M32C/83 Group DATA SHEET REV. B2

February First Edition 2002

Edited by
Committee of editing of Mitsubishi Semiconductor DATA SHEET

Published by
Mitsubishi Electric Corp., Kitaitami Works

This book, or parts thereof, may not be reproduced in any form without
permission of Mitsubishi Electric Corporation.

©2002 MITSUBISHI ELECTRIC CORPORATION