

# M32C/83 GROUP DATA SHEET

## REV.1.02

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.



## Overview

# Overview

The M32C/83 is single-chip microcomputer that utilizes high-performance silicon gate CMOS technology with the M32C/80 series CPU core. The M32C/83 group is available in the 144-pin and 100-pin plastic molded QFP/LQFP package.

With 16-Mbyte address memory space, this microcomputer combines advanced instructions manipulation capabilities to process complex instructions by less bytes and execute instructions at higher speed.

It incorporates a multiplier and DMAC adequate to office automation, communication devices and industrial equipments and other high-speed processing applications.

## Applications

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

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

Overview .....	1	Three-phase Motor Control Timer Functions ..	163
Memory .....	19	Serial I/O .....	174
Central Processing Unit (CPU) .....	20	CAN Module .....	225
Reset .....	23	Intelligent I/O .....	264
SFR .....	26	A-D Converter .....	338
Processor Mode .....	48	D-A Converter .....	355
Bus .....	52	CRC Calculation .....	358
System Clock .....	65	X-Y Conversion .....	360
Protection .....	87	DRAMC .....	363
Interrupts .....	88	Programmable I/O Ports .....	370
Watchdog Timer .....	112	Usage Precaution .....	394
DMAC .....	115	Flash Memory Version .....	401
DMAC II .....	126	Electrical Characteristics .....	427
Timer .....	135	Register Index .....	464

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.

## Overview

## Performance Outline

Tables 1.1.1 and 1.1.2 list performance outline of the M32C/83 group.

**Table 1.1.1. M32C/83 Group Performance (144-Pin Package)**

Item		Performance
CPU	Basic instructions	108 instructions
	Shortest instruction execution time	33 ns (f(XIN)=30MHz)
	Operation mode	Single-chip, memory expansion and microprocessor modes
	Memory space	16M bytes
	Memory capacity	See Table 1.1.3.
Peripheral function	I/O port	124 pins (P0 to P15, P85 is used as an input port)
	Multifunction timer	
	Output	16 bits x 5 channels (TA0, TA1, TA2, TA3, TA4)
	Input	16 bits x 6 channels (TB0, TB1, TB2, TB3, TB4, TB5)
	Three-phase motor control output	1 circuit
	Intelligent I/O	4 groups
	Time measurement function	12 channels (group 0: 8 channels, group1: 4 channels)
	Waveform generation function	28 channels (group 0: 4 channels, group1,2,3: 8 channels each)
	Bit modulation PWM	16 channels (group 2,3: 8 channels each)
	Real-time port	16 channels (group 2,3: 8 channels each)
	Communication function	<ul style="list-style-type: none"> <li>• Clock synchronous serial I/O, UART: 2 channels (group 0 and 1)</li> <li>• HDLC data processing : 2 channels (group 0 and 1)</li> <li>• Clock synchronous variable length serial I/O: 1 channel (group 2)</li> <li>• IE bus<sup>1</sup> : 1 channel (group 2)</li> <li>• 8-bit or 16-bit clock synchronous serial I/O : 1 channel (group 3)</li> </ul>
	Serial I/O	5 channels (UART0 to UART4) Clock synchronous, Clock asynchronous, IE Bus <sup>1</sup> , I <sup>2</sup> C Bus <sup>2</sup>
	CAN module	1 channel, supporting CAN 2.0B specification
	A-D converter	10-bit A-D x 2 circuits (standard 18 inputs, maximum 34 inputs)
	D-A converter	8-bit D-A x 2 circuits
	DMAC	4 channels
	DMAC II	Activated by all relocatable vector interrupt factors Immediate transfer, arithmetic transfer and chain transfer functions
	DRAMC	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 channel (with prescaler)	
Interrupt	42 internal and 8 external sources, 5 software sources, interrupt priority level 7	
Clock generation circuit	4 circuits <ul style="list-style-type: none"> <li>• Main clock generation circuit</li> <li>• Sub clock generation circuit</li> </ul> The above circuits include an internal feedback resistance and external ceramic resonator/crystal oscillator. <ul style="list-style-type: none"> <li>• Ring oscillator (for the main clock oscillator stop detect function)</li> <li>• PLL frequency synthesizer</li> </ul>	
Electric characteristics	Supply voltage	4.2 to 5.5V (f(XIN)=30MHz, VDC on) 3.0 to 5.5V (f(XIN)=20MHz, VDC on) 3.0 to 3.6V (f(XIN)=20MHz, VDC off)
	Power consumption	38mA (f(XIN)=30MHz with no wait, Vcc=5V) 26mA (f(XIN)=20MHz with no wait, Vcc=3.3V)
	I/O characteristics	I/O withstand voltage Vcc I/O current 5mA
Operating ambient temperature		-20 to 85°C, -40 to 85°C
Device configuration		CMOS high performance silicon gate
Package		144-pin plastic mold QFP

Notes :

1. IE Bus is a trademark of NEC Corporation.
2. I<sup>2</sup>C Bus is a trademark of Koninklijke Philips Electronics N. V.

## Overview

Table 1.1.2. M32C/83 Group Performance (100-Pin Package)

Item		Performance
CPU	Basic instructions	108 instructions
	Shortest instruction execution time	33 ns (f(XIN)=30MHz)
	Operation mode	Single-chip, memory expansion and microprocessor modes
	Memory space	16M bytes
	Memory capacity	See Table 1.1.3.
Peripheral function	I/O port	88 pins (P0 to P10, P8s is used as an input port)
	Multifunction timer	
	Output	16 bits x 5 channels (TA0, TA1, TA2, TA3, TA4)
	Input	16 bits x 6 channels (TB0, TB1, TB2, TB3, TB4, TB5)
	Three-phase motor control output	1 circuit
	Intelligent I/O	4 groups
	Time measurement function	5 channels (group 0: 3 channels, group1: 2 channels)
	Waveform generation function	10 channels (group 0,3: 2 channels each, group1,2: 3 channels each)
	Bit modulation PWM	5 channels (group 2 : 3 channels, group 3 : 2 channels)
	Real time port	5 channels (group 2 : 3 channels, group 3 : 2 channels)
	Communication function	<ul style="list-style-type: none"> <li>• Clock synchronous serial I/O, UART: 2 channels (group 0 and 1)</li> <li>• HDLC data processing : 2 channels (group 0 and 1)</li> <li>• Clock synchronous variable length serial I/O:1 channel (group 2)</li> <li>• IE bus<sup>1</sup> : 1 channel (group 2)</li> </ul>
	Serial I/O	5 channels (UART0 to UART4) Clock synchronous, Clock asynchronous, IE Bus <sup>1</sup> , I <sup>2</sup> C Bus <sup>2</sup>
	CAN module	1 channel, supporting CAN 2.0B specification
	A-D converter	10-bit A-Dx 2 circuits (standard 10 inputs, maximum 26 inputs)
	D-A converter	8-bit D-A x 2 circuits
	DMAC	4 channels
	DMAC II	Activated by all relocatable vector interrupt factors Immediate transfer, arithmetic transfer and chain transfer functions
	DRAMC	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 channel (with prescaler)	
Interrupt	42 internal and 8 external sources, 5 software sources, interrupt priority level 7	
Clock generation circuit	4 circuits <ul style="list-style-type: none"> <li>• Main clock generation circuit</li> <li>• Sub clock generation circuit</li> </ul> The above circuits include an internal feedback resistance and external ceramic resonator/crystal oscillator. <ul style="list-style-type: none"> <li>• Ring oscillator (for the main clock oscillator stop detect function)</li> <li>• PLL frequency synthesizer</li> </ul>	
Electric characteristics	Supply voltage	4.2 to 5.5V (f(XIN)=30MHz, VDC on) 3.0 to 5.5V (f(XIN)=20MHz, VDC on) 3.0 to 3.6V (f(XIN)=20MHz, VDC off)
	Power consumption	38mA (f(XIN)=30MHz with no wait, Vcc=5V) 26mA (f(XIN)=20MHz with no wait, Vcc=3.3V)
	I/O characteristics	I/O withstand voltage Vcc I/O current 5mA
Operating ambient temperature		-20 to 85°C, -40 to 85°C
Device configuration		CMOS high performance silicon gate
Package		100-pin plastic mold QFP

Notes :

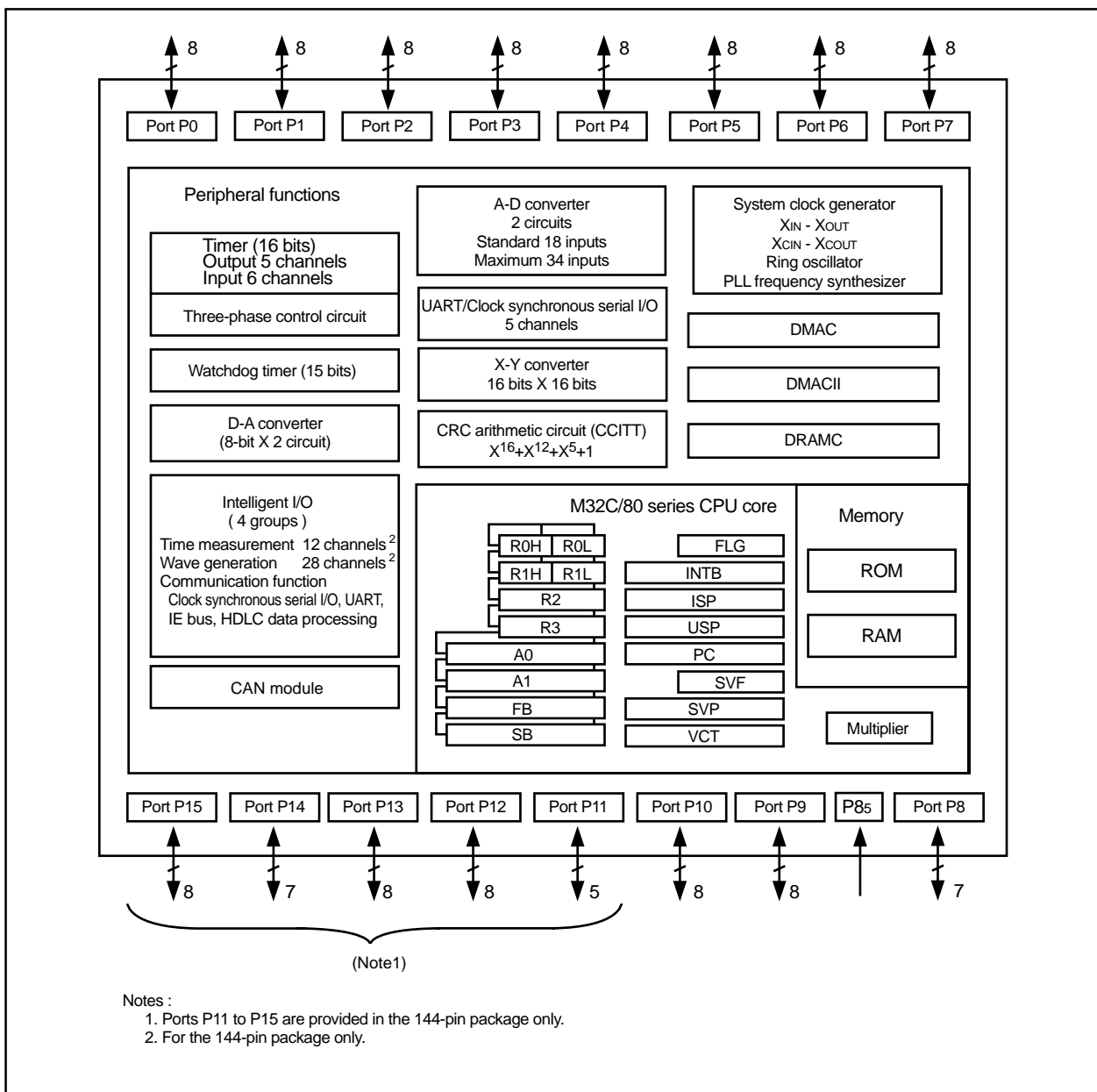
1. IE Bus is a trademark of NEC Corporation.
2. I<sup>2</sup>C Bus is a trademark of Koninklijke Philips Electronics N. V.

Overview

**Block Diagram**

Figure 1.1.1 shows a block diagram of the M32C/83 group.

The M32C/83 group microcomputer contains ROM and RAM as memory to store instructions and data, CPU to execute calculations and peripheral functions as interrupt, timer, serial I/O, D-A converter, DMAC, CRC calculation circuit, A-D converter, DRAMC, intelligent I/O and I/O ports.



**Figure 1.1.1. M32C/83 Group Block Diagram**

## Overview

### Product Information

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

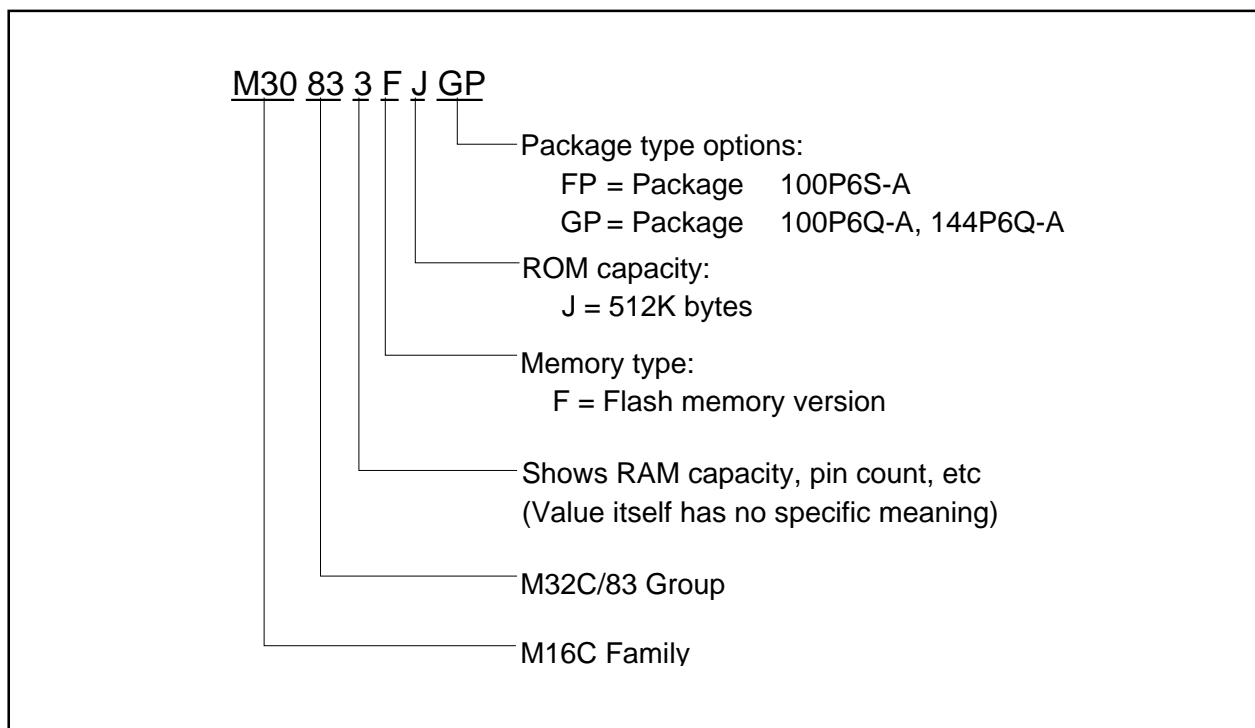
- (1) Support for the flash memory version
- (2) ROM/RAM capacity
- (3) Package
  - 100P6S-A : Plastic molded QFP
  - 100P6Q-A : Plastic molded QFP
  - 144P6Q-A : Plastic molded QFP

**Table 1.1.3. M32C/83 Group**

**As of December, 2002**

Type number	ROM capacity	RAM capacity	Package type	Remarks
M30835FJGP *	512K	31K	144P6Q-A	Flash memory version
M30833FJGP *			100P6Q-A	
M30833FJFP *			100P6S-A	

\* :New product



**Figure 1.1.2. Product Numbering System**

## Pin Assignments and Descriptions

Figures 1.1.3 to 1.1.5 show pin assignments (top view).

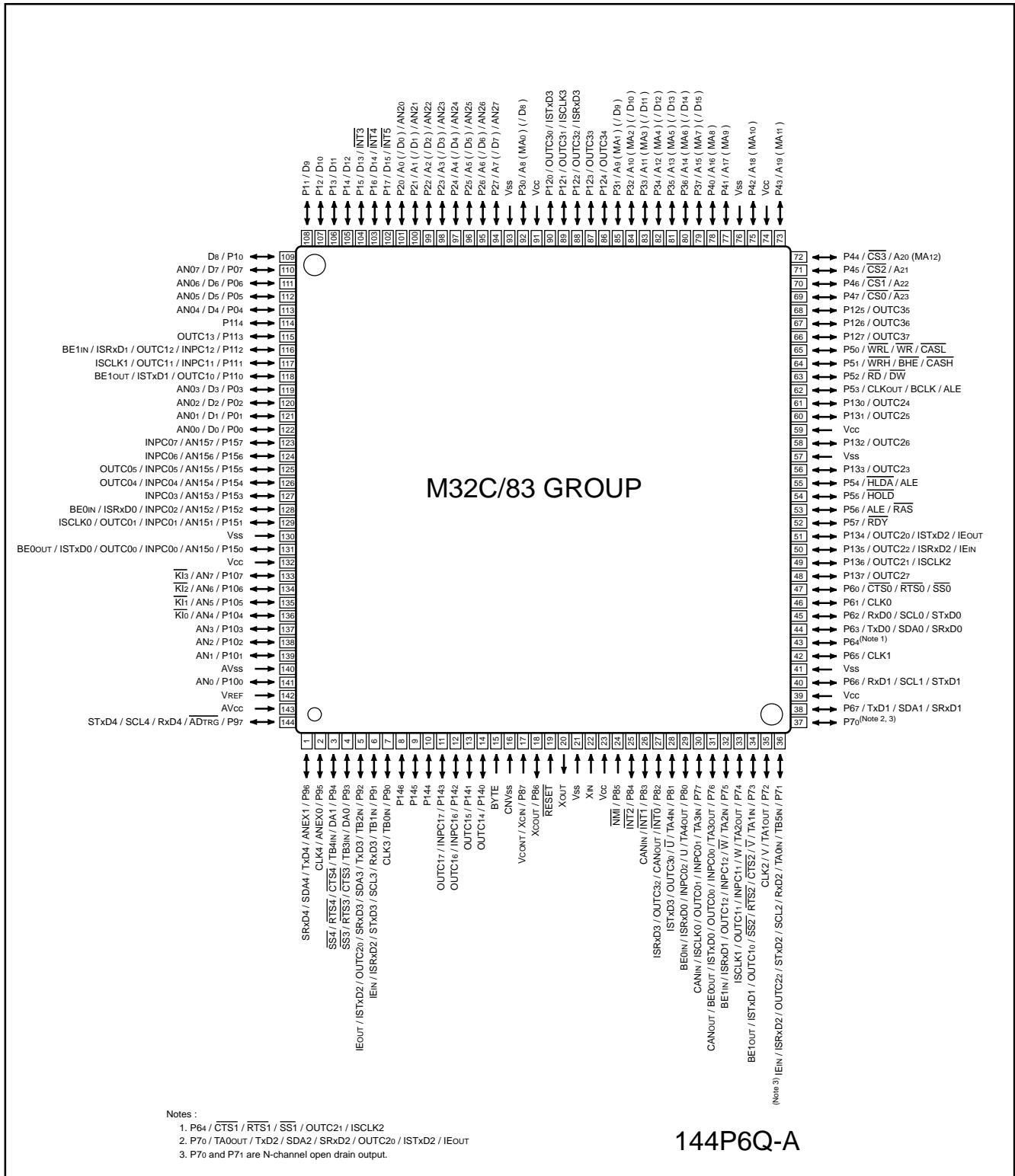


Figure 1.1.3. Pin Assignment for 144-Pin Package

## Overview

Table 1.1.4. Pin Characteristics for 144-Pin Package

Pin No	Control pin	Port	Interrupt pin	Timer pin	UART/CAN pin	Intelligent I/O pin	Analog pin	Bus control pin
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	OUTC2 <sub>0</sub> /IEOUT/ISTxD2		
6		P91		TB1IN	RxD3/SCL3/STxD3	IEIN/ISRxD2		
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	X <sub>CIN</sub> /V <sub>CONT</sub>	P87						
18	X <sub>COUT</sub>	P86						
19	RESET							
20	X <sub>OUT</sub>							
21	V <sub>SS</sub>							
22	X <sub>IN</sub>							
23	V <sub>CC</sub>							
24		P85	NMI					
25		P84	INT2					
26		P83	INT1		CAN <sub>IN</sub>			
27		P82	INT0		CAN <sub>OUT</sub>	OUTC3 <sub>2</sub> /ISRxD3		
28		P81	TA4 <sub>IN</sub> /U			OUTC3 <sub>0</sub> /ISTxD3		
29		P80	TA4 <sub>OUT</sub> /U			INPC0 <sub>2</sub> /ISRxD0/BE0 <sub>IN</sub>		
30		P77	TA3 <sub>IN</sub>		CAN <sub>IN</sub>	INPC0 <sub>1</sub> /OUTC0 <sub>1</sub> /ISCLK0		
31		P76	TA3 <sub>OUT</sub>		CAN <sub>OUT</sub>	INPC0 <sub>0</sub> /OUTC0 <sub>0</sub> /ISTxD0/BE0 <sub>OUT</sub>		
32		P75	TA2 <sub>IN</sub> /W			INPC1 <sub>2</sub> /OUTC1 <sub>2</sub> /ISRxD1/BE1 <sub>IN</sub>		
33		P74	TA2 <sub>OUT</sub> /W			INPC1 <sub>1</sub> /OUTC1 <sub>1</sub> /ISCLK1		
34		P73	TA1 <sub>IN</sub> /V		CTS2/RTS2/SS2	OUTC1 <sub>0</sub> /ISTxD1/BE1 <sub>OUT</sub>		
35		P72	TA1 <sub>OUT</sub> /V		CLK2			
36		P71	TB5 <sub>IN</sub> /TA0 <sub>IN</sub>		RxD2/SCL2/STxD2	OUTC2 <sub>2</sub> /ISRxD2/IEIN		
37		P70	TA0 <sub>OUT</sub>		TxD2/SDA2/SRxD2	OUTC2 <sub>0</sub> /ISTxD2/IEOUT		
38		P67			TxD1/SDA1/SRxD1			
39	V <sub>CC</sub>							
40		P66			RxD1/SCL1/STxD1			
41	V <sub>SS</sub>							
42		P65			CLK1			
43		P64			CTS1/RTS1/SS1	OUTC2 <sub>1</sub> /ISCLK2		
44		P63			TxD0/SDA0/SRxD0			
45		P62			RxD0/SCL0/STxD0			
46		P61			CLK0			
47		P60			CTS0/RTS0/SS0			
48		P137				OUTC2 <sub>7</sub>		

## Overview

Table 1.1.4. Pin Characteristics for 144-Pin Package (Continued)

Pin No	Control pin	Port	Interrupt pin	Timer pin	UART/CAN pin	Intelligent I/O pin	Analog pin	Bus control pin
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/ISRxD3		
89		P121				OUTC31/ISCLK3		
90		P120				OUTC30/ISTxD3		
91	Vcc							
92		P30						A8(MA0)/(D8)
93	Vss							
94		P27					AN27	A7/(D7)
95		P26					AN26	A6/(D6)
96		P25					AN25	A5/(D5)

Overview

**Table 1.1.4. Pin Characteristics for 144-Pin Package (Continued)**

Pin No	Control pin	Port	Interrupt pin	Timer pin	UART/CAN pin	Intelligent I/O pin	Analog pin	Bus control pin
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	$\overline{\text{INT5}}$					D15
103		P16	$\overline{\text{INT4}}$					D14
104		P15	$\overline{\text{INT3}}$					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	$\overline{\text{KI3}}$				AN7	
134		P106	$\overline{\text{KI2}}$				AN6	
135		P105	$\overline{\text{KI1}}$				AN5	
136		P104	$\overline{\text{KI0}}$				AN4	
137		P103					AN3	
138		P102					AN2	
139		P101					AN1	
140	AVss							
141		P100					AN0	
142							VREF	
143	AVcc							
144		P97			RxD4/SCL4/STxD4		$\overline{\text{ADTRG}}$	

Overview

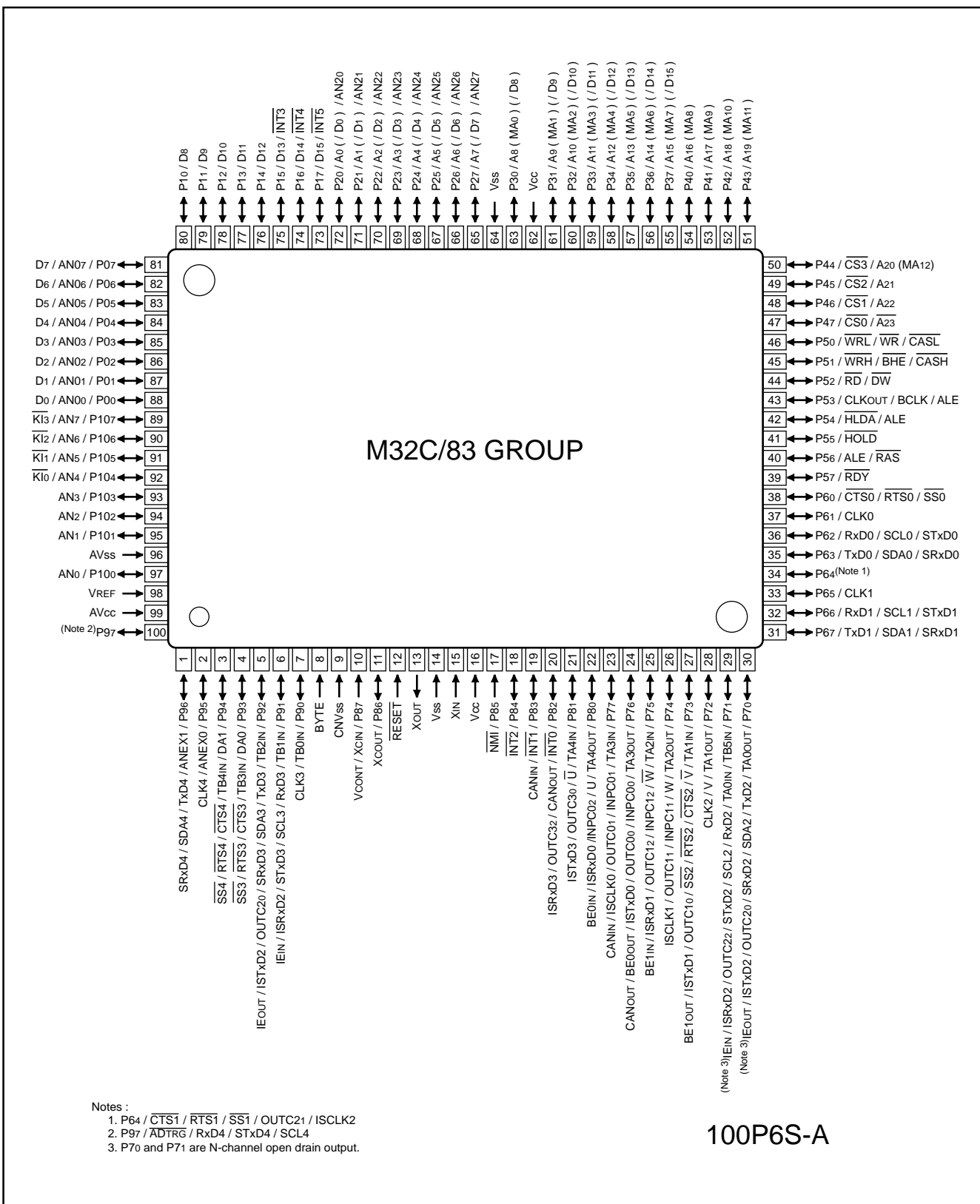


Figure 1.1.4. Pin assignment for 100-Pin Package

Overview

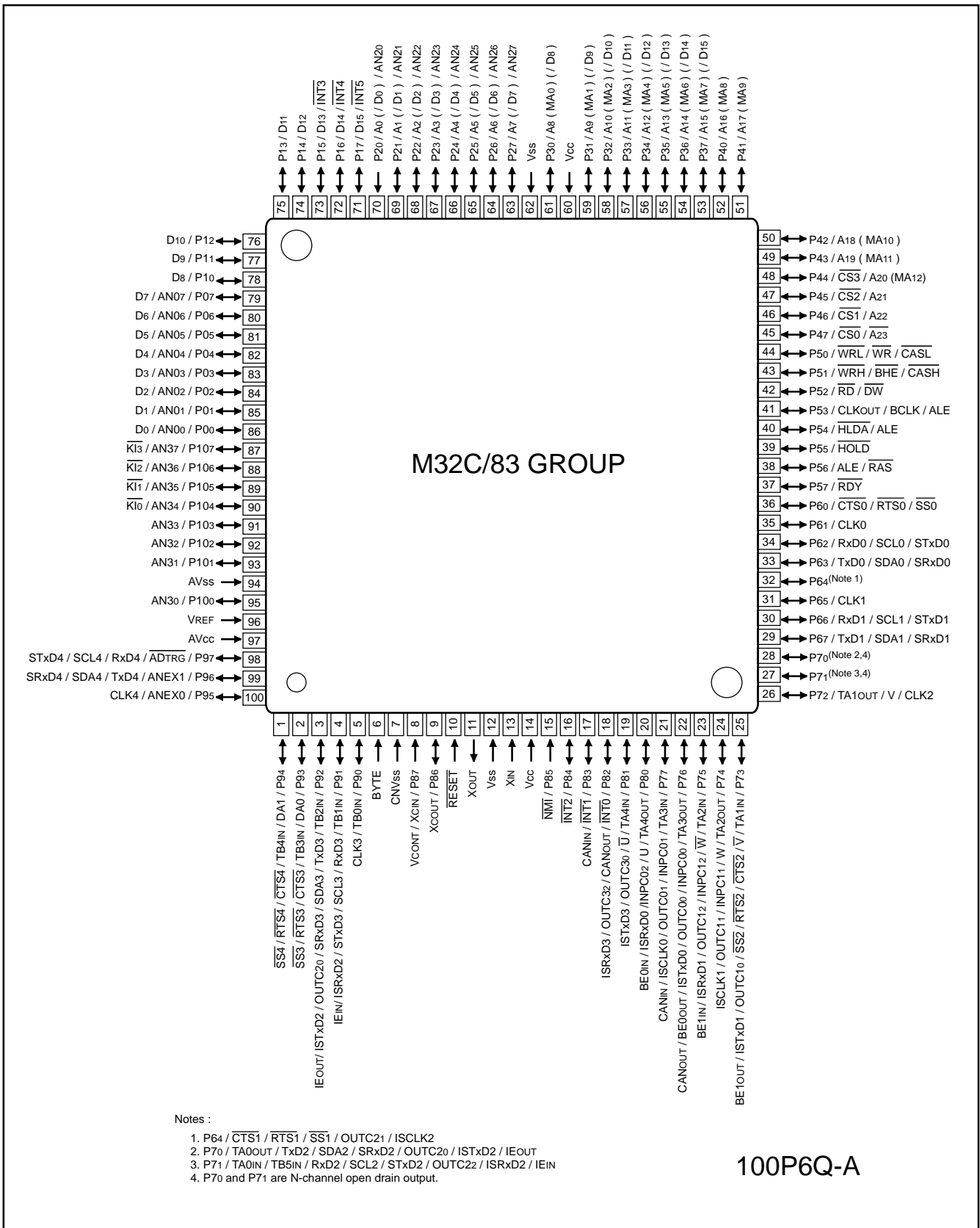


Figure 1.1.5. Pin Assignment for 100-Pin Package

Overview

**Table 1.1.5. Pin Characteristics for 100-Pin Package**

Package Pin No		Control pin	Port	Interrupt pin	Timer pin	UART/CAN pin	Intelligent I/O pin	Analog pin	Bus control pin
FP	GP								
1	99		P9 <sub>6</sub>			TxD4/SDA4/SRx4		ANEX1	
2	100		P9 <sub>5</sub>			CLK4		ANEX0	
3	1		P9 <sub>4</sub>		TB4IN	CTS4/RTS4/SS4		DA1	
4	2		P9 <sub>3</sub>		TB3IN	CTS3/RTS3/SS3		DA0	
5	3		P9 <sub>2</sub>		TB2IN	TxD3/SDA3/SRx3	OUTC2 <sub>0</sub> /IE <sub>OUT</sub> /ISTxD2		
6	4		P9 <sub>1</sub>		TB1IN	RxD3/SCL3/STxD3	IEIN/ISRxD2		
7	5		P9 <sub>0</sub>		TB0IN	CLK3			
8	6	BYTE							
9	7	CNV <sub>SS</sub>							
10	8	X <sub>CIN</sub> /V <sub>CONT</sub>	P8 <sub>7</sub>						
11	9	X <sub>COUT</sub>	P8 <sub>6</sub>						
12	10	RESET							
13	11	X <sub>OUT</sub>							
14	12	V <sub>SS</sub>							
15	13	X <sub>IN</sub>							
16	14	V <sub>CC</sub>							
17	15		P8 <sub>5</sub>	NMI					
18	16		P8 <sub>4</sub>	INT2					
19	17		P8 <sub>3</sub>	INT1		CANIN			
20	18		P8 <sub>2</sub>	INT0		CANOUT	OUTC3 <sub>2</sub> /ISRxD3		
21	19		P8 <sub>1</sub>		TA4IN/ $\bar{U}$		OUTC3 <sub>0</sub> /ISTxD3		
22	20		P8 <sub>0</sub>		TA4OUT/ $\bar{U}$		INPC0 <sub>2</sub> /ISRxD0/BE0IN		
23	21		P7 <sub>7</sub>		TA3IN	CANIN	INPC0 <sub>1</sub> /OUTC0 <sub>1</sub> /ISCLK0		
24	22		P7 <sub>6</sub>		TA3OUT	CANOUT	INPC0 <sub>0</sub> /OUTC0 <sub>0</sub> /ISTxD0/BE0OUT		
25	23		P7 <sub>5</sub>		TA2IN/ $\bar{W}$		INPC1 <sub>2</sub> /OUTC1 <sub>2</sub> /ISRxD1/BE1IN		
26	24		P7 <sub>4</sub>		TA2OUT/ $\bar{W}$		INPC1 <sub>1</sub> /OUTC1 <sub>1</sub> /ISCLK1		
27	25		P7 <sub>3</sub>		TA1IN/ $\bar{V}$	CTS2/RTS2/SS2	OUTC1 <sub>0</sub> /ISTxD1/BE1OUT		
28	26		P7 <sub>2</sub>		TA1OUT/ $\bar{V}$	CLK2			
29	27		P7 <sub>1</sub>		TB5IN/TA0IN	RxD2/SCL2/STxD2	OUTC2 <sub>2</sub> /ISRxD2/IEIN		
30	28		P7 <sub>0</sub>		TA0OUT	TxD2/SDA2/SRx2	OUTC2 <sub>0</sub> /ISTxD2/IEOUT		
31	29		P6 <sub>7</sub>			TxD1/SDA1/SRx1			
32	30		P6 <sub>6</sub>			RxD1/SCL1/STxD1			
33	31		P6 <sub>5</sub>			CLK1			
34	32		P6 <sub>4</sub>			CTS1/RTS1/SS1	OUTC2 <sub>1</sub> /ISCLK2		
35	33		P6 <sub>3</sub>			TxD0/SDA0/SRx0			
36	34		P6 <sub>2</sub>			RxD0/SCL0/STxD0			
37	35		P6 <sub>1</sub>			CLK0			
38	36		P6 <sub>0</sub>			CTS0/RTS0/SS0			
39	37		P5 <sub>7</sub>						RDY
40	38		P5 <sub>6</sub>						ALE/RAS
41	39		P5 <sub>5</sub>						HOLD
42	40		P5 <sub>4</sub>						HLDA/ALE
43	41		P5 <sub>3</sub>						CLKOUT/BCLK/ALE
44	42		P5 <sub>2</sub>						RD/DW
45	43		P5 <sub>1</sub>						WRH/BHE/CASH
46	44		P5 <sub>0</sub>						WRL/WR/CASL
47	45		P4 <sub>7</sub>						CS0/A23
48	46		P4 <sub>6</sub>						CS1/A22
49	47		P4 <sub>5</sub>						CS2/A21
50	48		P4 <sub>4</sub>						CS3/A20(MA12)

Overview

**Table 1.1.5. Pin Characteristics for 100-Pin Package (Continued)**

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

Overview

**Table 1.1.6. Pin Description (100-Pin and 144-Pin Packages)**

Symbol	Function	I/O type	Description
Vcc Vss	Power supply input	I I	Connect Vcc pin to 3.0 to 5.5 V. Connect Vss pin to 0 V.
CNVss	CNVss	I	Switches processor mode. Connect this pin to Vss to start up in single-chip the (memory expansion mode). Connect this pin to Vcc to start up in microprocessor mode.
RESET	Reset input	I	The microcomputer remains in a reset state when setting the pin to "L".
XIN XOUT	Clock input Clock output	I O	I/O pins for the main clock generation circuit. Connect a ceramic resonator or crystal oscillator between XIN pin and XOUT pin. To use an externally generated clock, input a clock from XIN pin to leave XOUT pin open.
BYTE	External data bus width select input	I	Switches a data bus in external memory space 3. Data bus in 16 bits long when setting this pin to "L" and 8 bits long when setting this pin to "H". Set it to either one. Connect this pin to Vss when an external bus is not used.
AVcc	Analog power supply input	I	Inputs the power supply for A-D converter and D-A converter. Connect this pin to Vcc.
AVss	Analog power supply input	I	Inputs the power supply for A-D converter and D-A converter. Connect this pin to Vss.
VREF	Reference voltage input	I	Inputs reference voltage for A-D converter.
P00 to P07	I/O port P0	I/O	8-bit I/O ports in CMOS, having a direction register to select an input or output. Each pin is set as an input port or output port. An input port in single-chip mode can be set for a pull-up or for no pull-up in 4-bit unit by program. When these pins are used as bus control signals in memory expansion mode and microprocessor mode, internal pull-up resistance cannot be selected. Ports used as input ports can be set for a pull-up or for no pull-up in these modes above.
D0 to D7	Data bus	I/O	Input and output data (D0 to D7) when setting these pins as data bus.
AN00 to AN07	Analog input pin	I	Analog input pins for A-D converter.
P10 to P17	I/O port P1	I/O	8-bit I/O ports equivalent to P0
INT3 to INT5	INT interrupt input pin	I	Input pins for INT interrupt
D8 to D15	Data bus	I/O	Input and output data (D8 to D15) when setting these pins as data bus.
P20 to P27	I/O port P2	I/O	8-bit I/O ports equivalent to P0
A0 to A7	Address bus	O	Output 8 low-order address bits (A0 to A7).
A0/D0 to A7/D7	Address bus/data bus	I/O	Input and output data (D0 to D7) and output 8 low-order address bits (A0 to A7) by time-sharing when setting these pins as multiplex bus.
AN20 to AN27	Analog input pin	I	Analog input pins for A-D converter.
P30 to P37	I/O port P3	I/O	I/O ports equivalent to P0
A8 to A15	Address bus	O	Output 8 middle-order address bits (A8 to A15).
A8/D8 to A15/D15	Address bus/data bus	I/O	Input and output data (D8 to D15) and output 8 middle-order address bits (A8 to A8) by time-sharing when setting an external 16-bit data bus as multiplex bus.
MA0 to MA7	Address bus	O	Output row addresses and column addresses by time-sharing when accessing the DRAM space.
P40 to P47	I/O port P4	I/O	8-bit I/O ports equivalent to P0
A16 to A22 A23	Address bus	O O	Output 8 high-order address bits (A16 to A22, A23). The uppermost bit (A23) inversed is also output.
CS0 to CS3	Chip-select	O	Output CS0 to CS3 signals. CS0 to CS3 are chip-select signals to specify access space.
MA8 to MA12	Address bus	O	Output row addresses and column addresses by time-sharing when accessing the DRAM space.

I : Input   O : Output   I/O : Input and output

Overview

Table 1.1.6. Pin Description (100-Pin and 144-Pin Packages) (Continued)

Symbol	Function	I/O type	Description
P50 to P57	I/O port P5	I/O	8-bit I/O ports equivalent to P0
CLKOUT	Clock output	O	Outputs a XIN divided by 8 or divided by 32 or a clock having the same frequency as XCIN from P53.
WRL WR WRH BHE RD BCLK HLDA HOLD ALE RDY	Bus control pin	O O O O O O O I O I	Output WRL, WRH, (WR, BHE), RD, BCLK, HLDA and ALE signals. WRL and WRH or BHE and WR can be switched by program. ■ WRL, WRH, RD is selected WRL signals is set to "L" when writing data to an even address in external memory space. WRH signal is set to "L" when writing data to an odd address in external memory space. RD signal is set to "L" when reading data in external memory space. ■ WR, BHE, RD is selected WR signal is set to "L" when writing data in external memory space. RD signal is set to "L" when reading data in external memory space. BHE signal is set to "L" when accessing data to an odd address. Select WR, BHE and RD for an external 8-bit data bus. While in an input level HOLD pin is set to "L", the microcomputer is placed in a hold state. In a hold state, HLDA outputs "L" level. ALE latches the address. While an input level of the RDY pin is set to "L", the microcomputer is placed in a wait state.
DW CASL CASH RAS	DRAM bus control pin	O O O O	DW signal is set to "L" when writing data to DRAM space. CASL and CASH signals indicate a timing to latch column address. CASL is set to "L" when accessing an even address. CASH is set to "L" when accessing an odd address. RAS signal latches row address.
P60 to P67	I/O port P6	I/O	8-bit I/O ports equivalent to P0.
CTS0, CTS1 RTS0, RTS1 SS0, SS1 CLK0, CLK1 RxD0, RxD1 SCL0, SCL1 STxD0, STxD1 TxD0, TxD1 SDA0, SDA1 SRxD0, SRxD1	UART pin	I O I I/O I I/O O O O I	I/O pins for UART0 (P60 to P63) and UART1 (P64 to P67)
ISCLK OUTC21	Intelligent I/O pin	I/O O	ISCLK2 inputs and outputs a clock for the intelligent I/O communication function. OUTC21 outputs a clock for the waveform generation function.

I : Input    O : Output    I/O : Input and output

Overview

**Table 1.1.6. Pin Description(100-Pin and 144-Pin Packages) (Continued)**

Symbol	Function	I/O type	Description
P7 <sub>0</sub> to P7 <sub>7</sub>	I/O port P7	I/O	8-bit I/O ports equivalent to P0 (except P7 <sub>0</sub> and P7 <sub>1</sub> for N-channel open drain outputs.)
TA0 <sub>OUT</sub> to TA3 <sub>OUT</sub> TA0 <sub>IN</sub> to TA3 <sub>IN</sub>	Timer A pin	I/O I	I/O ports for the timer A0 to timer A3.
TB5 <sub>IN</sub>	Timer B pin	I	Input pin for the timer B5
V, V̄ W, W̄	Three-phase motor control output pin	O	V-phase output pin W-phase output pin
CTS <sub>2</sub> RTS <sub>2</sub> SS <sub>2</sub> CLK <sub>2</sub> RxD <sub>2</sub> SCL <sub>2</sub> STxD <sub>2</sub> TxD <sub>2</sub> SDA <sub>2</sub> SRxD <sub>2</sub>	UART pin	I O I I/O I I/O O O O I	I/O pins for UART2
INPC0 <sub>0</sub> , INPC0 <sub>1</sub> , INPC1 <sub>1</sub> , INPC1 <sub>2</sub> OUTC0 <sub>0</sub> , OUTC0 <sub>1</sub> , OUTC1 <sub>0</sub> to OUTC1 <sub>2</sub> , OUTC2 <sub>0</sub> to OUTC2 <sub>2</sub> ISCLK <sub>0</sub> , ISCLK <sub>1</sub> ISTxD <sub>0</sub> to ISTxD <sub>2</sub> ISRxD <sub>1</sub> , ISRxD <sub>2</sub> IE <sub>OUT</sub> IE <sub>IN</sub> BE0 <sub>OUT</sub> BE1 <sub>OUT</sub> BE1 <sub>IN</sub>	Intelligent I/O pin	I O I/O O I O I O O I	INPC0 <sub>0</sub> , INPC0 <sub>1</sub> , INPC1 <sub>1</sub> and INPC1 <sub>2</sub> are input pins for the time measurement function. OUTC0 <sub>0</sub> , OUTC0 <sub>1</sub> and OUTC1 <sub>0</sub> to OUTC1 <sub>2</sub> , OUTC2 <sub>0</sub> , OUTC2 <sub>2</sub> are output pins for the waveform generation function. ISCLK <sub>0</sub> and ISCLK <sub>1</sub> input and output a clock for the intelligent I/O communication function. IE <sub>IN</sub> , ISRxD <sub>1</sub> , ISRxD <sub>2</sub> and BE1 <sub>IN</sub> input received data for the intelligent I/O communication function. IE <sub>OUT</sub> , ISTxD <sub>0</sub> to ISTxD <sub>2</sub> , BE0 <sub>OUT</sub> and BE1 <sub>OUT</sub> output transmit data for the intelligent I/O communication function.
CAN <sub>OUT</sub> CAN <sub>IN</sub>	CAN pin	O I	I/O pins for the CAN communication function
P8 <sub>0</sub> to P8 <sub>4</sub> , P8 <sub>6</sub> , P8 <sub>7</sub>	I/O port P8	I/O	I/O ports equivalent to P0
XCIN XCOUT	Sub clock	I O	I/O pins for a sub clock oscillation circuit. Connect a crystal oscillator between XCIN pin and XCOUT pin.
VCONT	Low-pass filter connect pin for PLL frequency synthesizer pin		Connect a low-pass filter to VCONT pin to use the PLL frequency synthesizer. Connect P8 <sub>6</sub> to V <sub>SS</sub> to make a PLL frequency stable.
TA4 <sub>OUT</sub> TA4 <sub>IN</sub>	Timer A pin	I/O I	I/O pins for the timer A4
U, Ū	Three phase motor control output pin	O	U-phase output pins
INT0 to INT2	INT interrupt input pin	I	Input pins for INT interrupt
INPC0 <sub>2</sub> ISRxD <sub>0</sub> BE0 <sub>IN</sub> OUTC3 <sub>0</sub> ISTxD <sub>3</sub> OUTC3 <sub>2</sub> ISRxD <sub>3</sub>	Intelligent I/O pin	I I I O O O I	INPC0 <sub>2</sub> is an input pin for the time measurement function. OUTC3 <sub>0</sub> and OUTC3 <sub>2</sub> are output pins for the waveform generation function. ISRxD <sub>0</sub> and BE0 <sub>IN</sub> input received data for the intelligent I/O communication function. ISTxD <sub>3</sub> outputs transmit data for the intelligent I/O communication function. ISRxD <sub>3</sub> inputs receive data for the intelligent I/O communication function.
CAN <sub>OUT</sub> CAN <sub>IN</sub>	CAN pin	O I	I/O pins for CAN communication function
P8 <sub>5</sub> /NMI	NMI interrupt input pin	I	Input pin for the NMI interrupt. Pin status can be read in the P8_5 bit in the P8 register.

I : Input    O : Output    I/O : Input and output

## Overview

Table 1.1.6. Pin Description(100-Pin and 144-Pin Packages) (Continued)

Symbol	Function	I/O type	Description
P90 to P97	I/O port P9	I/O	8-bit I/O ports equivalent to P0. PRCR register prevents PD9 and PS3 registers from rewriting.
TB0IN to TB4IN	Timer B pin	I	Input pins for timer B0 to B4
CTS3, CTS4 RTS3, RTS4 SS3, SS4 CLK3, CLK4 RxD3, RxD4 SCL3, SCL4 STxD3, STxD4 TxD3, TxD4 SDA3, SDA4 SRxD3, SRxD4	UART pin	I O I I/O I I/O O O O I	I/O pins for UART3 (P90 to P93) and UART 4 (P94 to P97)
DA0, DA1	D-A output pin	O	Input pins for D-A converter
ANEX0 ANEX1 ADTRG	A-D related pin	I/O I I	ANEX0 and ANEX1 are expanded analog I/O pins for A-D converter. ADTRG is an A-D trigger input pin.
OUTC20 ISTxD2 IEOUT IEIN ISRxD2	Intelligent I/O pin	O O O I I	OUTC20 is an output pin for the waveform generation function. ISTxD2 outputs transmit data for the intelligent I/O communication function. IEOUT outputs transmit data for the intelligent I/O communication function or IE mode. IEIN inputs receive data for the intelligent I/O communication function or IE mode. ISRxD2 inputs receive data for the intelligent I/O communication function.
P100 to P107	I/O port P10	I/O	8-bit I/O ports equivalent to P0
KI0 to KI3	Key input interrupt pin	I	Input pins for key input interrupt
AN0 to AN7	Analog input pin	I	Analog I/O pins for A-D converter

I : Input    O : Output    I/O : Input and output

Overview

**Table 1.1.6. Pin Description(144-Pin Package only) (Continued)**

Symbol	Function	I/O type	Description
P110 to P114	I/O port P11	I/O	5-bit I/O ports equivalent to P0
INPC11, INPC12 OUTC10 to OUTC13 ISCLK1 ISRxD1 BE1IN ISTxD1 BE1OUT	Intelligent I/O pin	I O I/O I O O	INPC11 and INPC12 are input pins for the time measurement function. OUTC10 to OUTC13 are output pins for the waveform generation function. ISCLK1 inputs and outputs a clock for the intelligent I/O communication function. ISRxD1 and BE1IN input receive data for the intelligent I/O communication function. ISTxD1 and BE1OUT output receive data for the intelligent I/O communication function.
P120 to P127	I/O port P12	I/O	8-bit I/O ports equivalent to P0
OUTC30 to OUTC37 ISCLK3 ISTxD3 ISRxD3	Intelligent I/O pin	O I/O O I	OUTC30 to OUTC37 are output pins for the waveform generation function. ISCLK3 outputs a clock for the intelligent I/O communication function. ISTxD3 outputs transmit data for the intelligent I/O communication function. ISRxD3 inputs receive data for the intelligent I/O communication function.
P130 to P137	I/O port P13	I/O	8-bit I/O ports equivalent to P0
OUTC20 to OUTC27 ISCLK2 ISRxD2 IEIN ISTxD2 IEOUT	Intelligent I/O pin	O I/O I I O O	OUTC20 to OUTC27 are output pins for the waveform generation function. ISCLK2 inputs and outputs a clock for the intelligent I/O communication function. ISTxD2 and IEOUT output transmit data for the intelligent I/O communication function. ISRxD1 and IEIN input receive data for the intelligent I/O communication function.
P140 to P146	I/O port P14	I/O	7-bit I/O ports equivalent to P0
INPC16 and INPC17 OUTC14 to OUTC17	Intelligent I/O pin	I O	INPC16 to INPC17 are input pins for the time measurement function. OUTC14 to OUTC17 are output pins for the waveform generation function.
P150 to P157	I/O port P15	I/O	8-bit I/O ports equivalent to P0
INPC00 to INPC07 OUTC00, OUTC01, OUTC04, OUTC05 ISCLK0 ISRxD0 BE0IN ISTxD0 BE0OUT	Intelligent I/O pin	I O I/O I I O O	INPC00 to INPC07 are input pins for the time measurement function. OUTC00, OUTC01, OUTC04 and OUTC05 are output pins for the waveform generation function. ISCLK0 inputs and outputs a clock for the intelligent I/O communication function. ISRxD0 and BE0IN input receive data for the intelligent I/O communication function. ISTxD0 and BE0OUT output transmit data for the intelligent I/O communication function.
AN150 to AN157	Analog input port	I	Analog input pin for A-D converter

I : Input    O : Output    I/O : Input and output

Memory

# Memory

Figure 1.2.1 shows a memory map of the M32C/83 group.

Total address space are 16M bytes from addresses 000000<sub>16</sub> to FFFFFFF<sub>16</sub>.

Internal ROM is allocated in lower addresses beginning with address FFFFFFF<sub>16</sub>. For example, a 64K-byte internal ROM is allocated in addresses from FF0000<sub>16</sub> to FFFFFFF<sub>16</sub>.

Fixed interrupt vectors are allocated in addresses from FFFFDC<sub>16</sub> to FFFFFFF<sub>16</sub>. The start address of each interrupt routine is stored there addresses. Refer to the section "Interrupts" for details.

Internal RAM is allocated in higher addresses beginning with address 000400<sub>16</sub>. For example, a 10K-byte internal RAM are allocated in addresses from 000400<sub>16</sub> to 002BFF<sub>16</sub>. Internal RAM stores data as well as a stack used to call subroutines and interrupts.

SFR is allocated in addresses from 000000<sub>16</sub> to 0003FF<sub>16</sub>. The peripheral function control registers such as I/O port, A-D conversion, serial I/O, timer are allocated here. All addresses, which have nothing allocated within SFR, are reserved space and cannot be used by users.

Special page vectors are allocated in addresses FFFE00<sub>16</sub> to FFFFDB<sub>16</sub>. This vector is used for the JMPS instruction and JSRS instruction. Refer to the Mitsubishi Electric publication "Software Manual" for the details.

In memory expansion mode and microprocessor mode, some space are reserved and cannot be used by users.

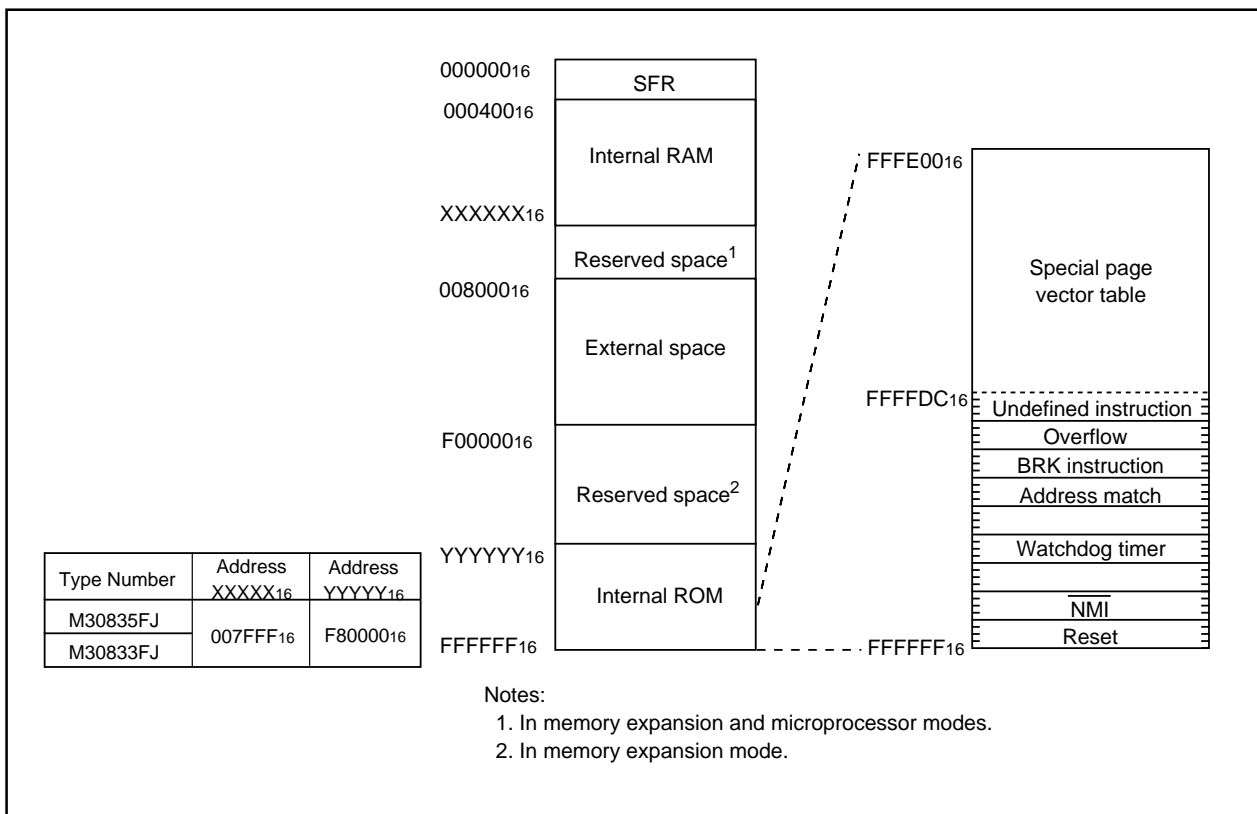


Figure 1.2.1. Memory Map

Central Processing Unit (CPU)

# Central Processing Unit (CPU)

Figure 1.3.1 shows the CPU registers.  
 Eight registers (R0, R1, R2, R3, A0, A1, SB and FB) out of twenty-eight CPU registers comprise a register bank. Two sets of register banks are provided.

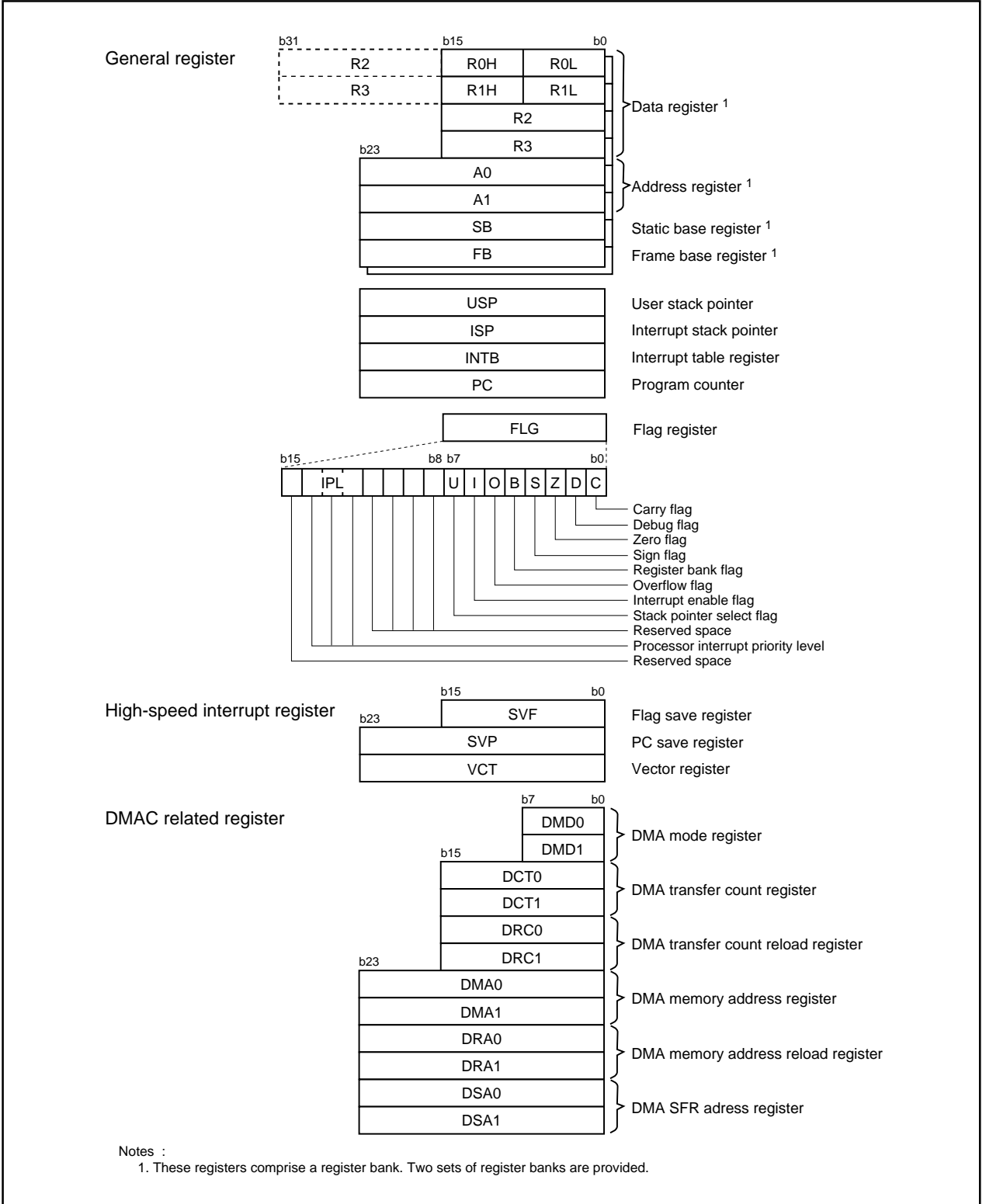


Figure 1.3.1. CPU Register

## Central Processing Unit (CPU)

---

### General Register

#### (1) Data Registers (R0, R1, R2 and R3)

R0 is 16 bits long to use primarily for transfer, arithmetic, calculation and logic as well as R1 to R3. R0 can be used as an 8-bit data register separated high-order(R0H) and low-order(R0L) as well as R1.

R0 can be combined with R2 to use as a 32-bit data register (R2R0) as well as R3R1.

#### (2) Address Registers (A0 and A1)

A0 is 24 bits long to use address register indirect addressing, address register relative addressing, transfer, arithmetic calculation and logic calculation, as well as A1.

#### (3) Static Base Register (SB)

SB is 24 bits long to use for SB relative addressing.

#### (4) Frame Base Register (FB)

FB is 24 bits long to use for FB relative addressing.

#### (5) Program Counter (PC)

PC is 24 bits long to indicate an address of an instruction to be executed.

#### (6) Interrupt Table Register (INTB)

INTB is 24 bits long to indicate a starting address of an interrupt vector table.

#### (7) User Stack Pointer (USP), Interrupt Stack Pointer (ISP)

The user stack pointer (USP) and interrupt stack pointer (ISP), which are 24 bits long, are provided as the stack pointer. The U flag can switch USP to ISP and vice versa. Refer to the paragraph "Flag Register (FLG)" about the U flag. USP and ISP should be set to an even number to execute an interrupt sequence efficiently.

### High-Speed Interrupt Registers

Registers associated to the high-speed interrupt are as follows. Refer to the paragraph "High-speed Interrupt" for details.

- Flag save register (SVF)
- PC save register (SVP)
- Vector register (VCT)

### DMAC-associated Registers

Registers associated to DMAC are as follows. Refer to the section "DMAC" for details.

- DMA mode register (DMD0, DMD1)
- DMA transfer count register (DCT0, DCT1)
- DMA transfer count reload register (DRC0, DRC1)
- DMA memory address register (DMA0, DMA1)
- DMA SFR address register (DSA0, DSA1)
- DMA memory address reload register (DRA0, DRA1)

## Central Processing Unit (CPU)

---

### Flag Register (FLG)

The flag register (FLG) is 16 bits long to show the CPU status.

#### (1) Carry Flag (C)

The C flag indicates carry and borrow status after an instruction is executed.

#### (2) Debug Flag (D)

The D flag is for debug only. It should be set to "0".

#### (3) Zero Flag (Z)

The Z flag is set to "1" when a zero value is obtained from an arithmetic calculation; otherwise, set to "0".

#### (4) Sign Flag (S)

The S flag is set to "1" when a negative value is obtained from an arithmetic calculation; otherwise, set to "0".

#### (5) Register Bank Select Flag (B)

The register bank 0 is selected when the B flag is set to "0". The register bank 1 is selected when this flag is set to "1".

#### (6) Overflow Flag (O)

The O flag is set to "1" when an arithmetic operation overflows; otherwise, is set to "0".

#### (7) Interrupt Enable Flag (I)

The I flag enables a maskable interrupt.

An interrupt is disabled when the I flag is set to "0" and is enabled when the I flag is set to "1". The I flag is set to "0" when an interrupt is acknowledged.

#### (8) Stack Pointer Select Flag (U)

ISP is selected when the U flag is set to "0". USP is selected when this flag is set to "1".

The U flag is set to "0" when the hardware interrupt is acknowledged or the INT instruction of software interrupt numbers 0 to 31 is executed.

#### (9) Processor Interrupt Priority Level (IPL)

IPL is 3 bits long to assign interrupt priority levels from level 0 to level 7.

If a requested interrupt has a greater priority than IPL, the interrupt is enabled.

#### (10) Reserved Space

When write, the reserved space should be set to "0". When read, its content is indeterminate.

## Reset

Hardware reset and software reset can be used to reset a microcomputer.

### Hardware Reset

#### 1. When the Power Supply is Stable

The  $\overline{\text{RESET}}$  pin is reset when the supply voltage meets recommended performance conditions and "L" is input to the  $\overline{\text{RESET}}$  pin (see Table 1.4.1). The  $\overline{\text{RESET}}$  pin should be in "H" after 20 or more clock cycles are input to the XIN pin with the  $\overline{\text{RESET}}$  pin in "L". The CPU and SFR are reset to run programs from an address indicated by a reset vector.

Internal RAM is not reset after reset. When the  $\overline{\text{RESET}}$  pin is set to "L" while writing to the internal RAM, the internal RAM becomes indeterminate.

#### 2. When the Power Supply is On

The  $\overline{\text{RESET}}$  pin is reset when the supply voltage for the VCC pin meet the recommended performance requirement. (See Table 1.4.1.)

Main clock oscillation is stable and 20 or more clocks are input to the XIN pin. The CPU and SFR are reset when the  $\overline{\text{RESET}}$  pin level changes "L" to "H". (Internal RAM is indeterminate.) Programs run from an address indicated.

### Software Reset

When the PM03 bit in the PM0 register is set to "1" (microcomputer reset), pins, the CPU and SFR are reset as well as a hardware reset. Programs run from an address indicated by a reset vector.

At software reset, processor mode is not exited and bus-associated registers are not changed.

The main clock should be selected as the CPU clock to set the PM03 bit to "1" when main clock oscillation is stable enough.

Figure 1.4.1 shows a reset sequence. Figure 1.4.3 shows the CPU register conditions after reset. Refer to the section "SFR" about SFR conditions after reset.

Reset

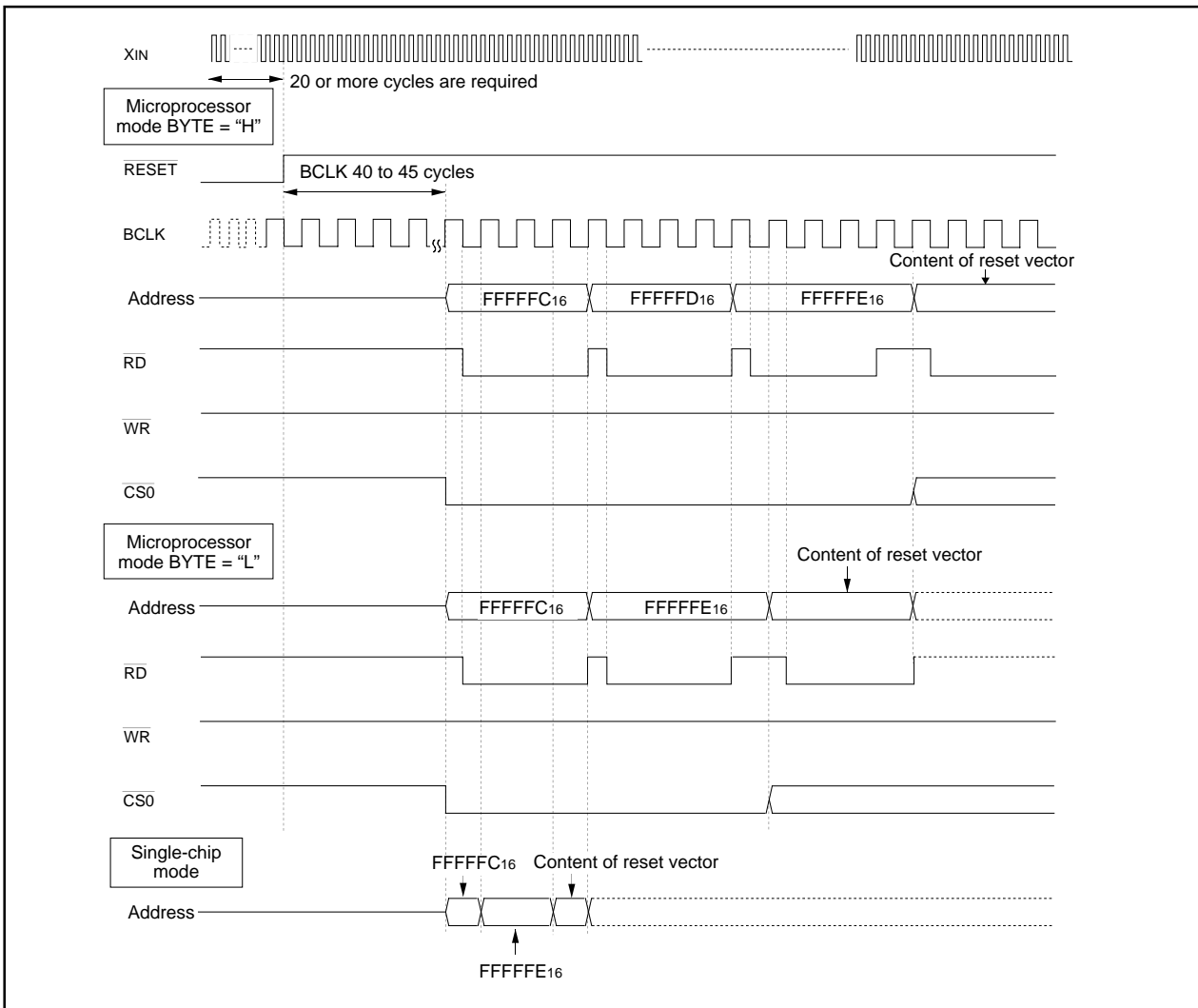


Figure 1.4.1. Reset Sequence

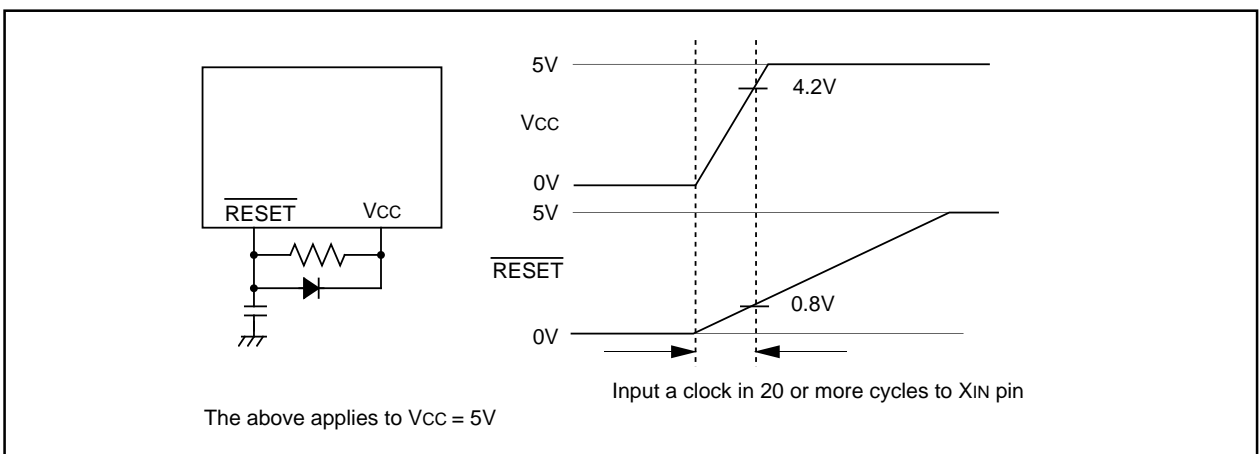


Figure 1.4.2. Reset Circuit

Reset

**Table 1.4.1. Pin Status**

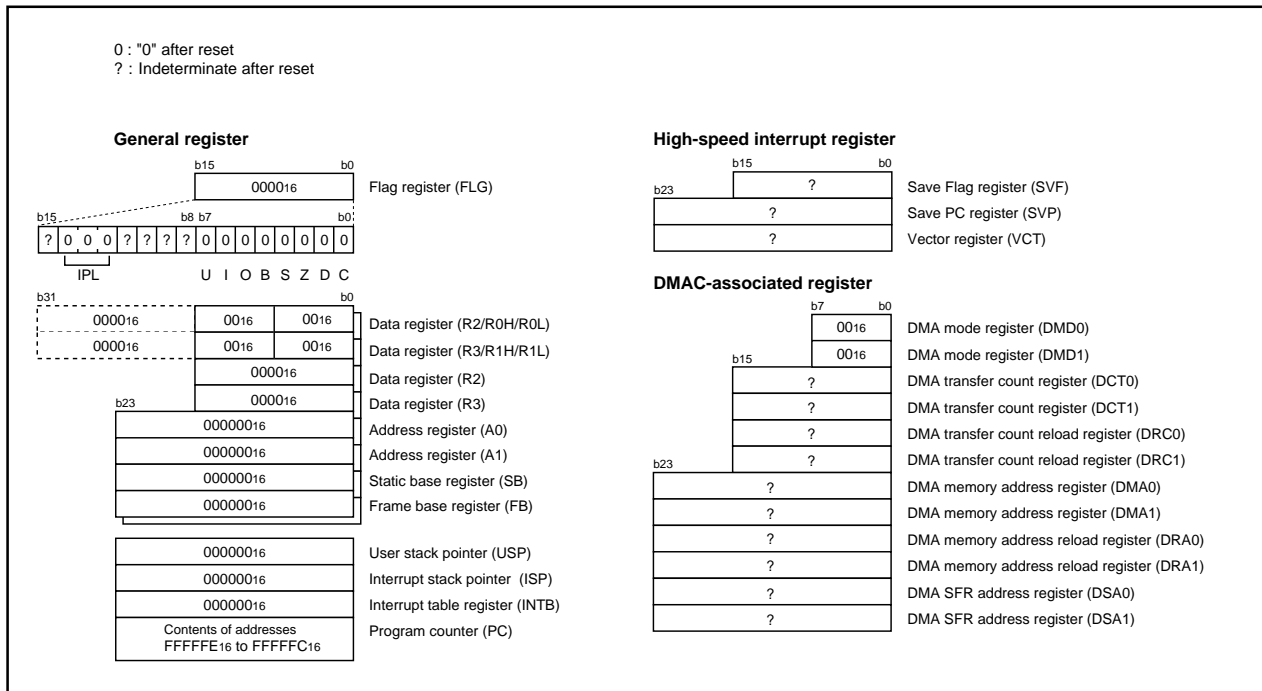
Pin name	Status		
	CNVss = Vss	CNVss = Vcc	
		BYTE = Vss	BYTE = Vcc
P0	Input port (high-impedance)	Data input (high-impedance)	
P1	Input port (high-impedance)	Data input (high-impedance)	Input port (high-impedance)
P2, P3, P4	Input port (high-impedance)	Address output (indeterminate)	
P50	Input port (high-impedance)	$\overline{WR}$ output (output "H")	
P51	Input port (high-impedance)	$\overline{BHE}$ output (indeterminate)	
P52	Input port (high-impedance)	$\overline{RD}$ output (output "H")	
P53	Input port (high-impedance)	BCLK output	
P54	Input port (high-impedance)	HLDA output (output value depends on an input to $\overline{HOLD}$ pin)	
P55	Input port (high-impedance)	$\overline{HOLD}$ input (high-impedance)	
P56	Input port (high-impedance)	$\overline{RAS}$ output	
P57	Input port (high-impedance)	$\overline{RDY}$ input (high-impedance)	
P6 to P15 <sup>1</sup>	Input port (high-impedance)	Input port (high-impedance)	

Notes :

1. Ports P11 to P15 are provided in the 144-pin package.

**Internal Space**

Figure 1.4.3 shows CPU register states after reset. Refer to the section "SFR" about a SFR state after reset.



**Figure 1.4.3. CPU Register after Reset**

## SFR

## SFR

Address	Register	Symbol	Value after RESET
0000 <sub>16</sub>			
0001 <sub>16</sub>			
0002 <sub>16</sub>			
0003 <sub>16</sub>			
0004 <sub>16</sub>	Processor mode register 0	PM0	1000 0000 <sub>2</sub> (CNVss pin ="L") 0000 0011 <sub>2</sub> (CNVss pin ="H")
0005 <sub>16</sub>	Processor mode register 1	PM1	0X00 0000 <sub>2</sub>
0006 <sub>16</sub>	System clock control register 0	CM0	0000 X000 <sub>2</sub>
0007 <sub>16</sub>	System clock control register 1	CM1	0010 0000 <sub>2</sub>
0008 <sub>16</sub>	Wait control register	WCR	1111 1111 <sub>2</sub>
0009 <sub>16</sub>	Address match interrupt enable register	AIER	XXXX 0000 <sub>2</sub>
000A <sub>16</sub>	Protect register	PRCR	XXXX 0000 <sub>2</sub>
000B <sub>16</sub>	External data bus width control register	DS	XXXX 1000 <sub>2</sub> (BYTE pin ="L") XXXX 0000 <sub>2</sub> (BYTE pin ="H")
000C <sub>16</sub>	Main clock division register	MCD	XXX0 1000 <sub>2</sub>
000D <sub>16</sub>	Oscillation stop detect register	CM2	0000 0000 <sub>2</sub>
000E <sub>16</sub>	Watchdog timer start register	WDTS	?? <sub>16</sub>
000F <sub>16</sub>	Watchdog timer control register	WDC	000? ???? <sub>2</sub>
0010 <sub>16</sub>			
0011 <sub>16</sub>	Address match interrupt register 0	RMAD0	000000 <sub>16</sub>
0012 <sub>16</sub>			
0013 <sub>16</sub>			
0014 <sub>16</sub>			
0015 <sub>16</sub>	Address match interrupt register 1	RMAD1	000000 <sub>16</sub>
0016 <sub>16</sub>			
0017 <sub>16</sub>	VDC control register for PLL	PLV	XXXX XX01 <sub>2</sub>
0018 <sub>16</sub>			
0019 <sub>16</sub>	Address match interrupt register 2	RMAD2	000000 <sub>16</sub>
001A <sub>16</sub>			
001B <sub>16</sub>	VDC control register 0	VDC0	0000 0000 <sub>2</sub>
001C <sub>16</sub>			
001D <sub>16</sub>	Address match interrupt register 3	RMAD3	000000 <sub>16</sub>
001E <sub>16</sub>			
001F <sub>16</sub>	VDC control register 1	VDC1	0000 0000 <sub>2</sub>
0020 <sub>16</sub>			
0021 <sub>16</sub>	Emulator interrupt vector table register	EIAD	F00000 <sub>16</sub>
0022 <sub>16</sub>			
0023 <sub>16</sub>	Emulator interrupt detect register	EITD	XXXX XX00 <sub>2</sub>
0024 <sub>16</sub>	Emulator protect register	EPRR	XXXX XXX0 <sub>2</sub>
0025 <sub>16</sub>	Emulator protect control register	EMU	XXXX X000 <sub>2</sub>
0026 <sub>16</sub>			
0027 <sub>16</sub>			
0028 <sub>16</sub>			
0029 <sub>16</sub>			
002A <sub>16</sub>			
002B <sub>16</sub>			
002C <sub>16</sub>			
002D <sub>16</sub>			
002E <sub>16</sub>			
002F <sub>16</sub>			

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

Users cannot use any symbols with \*. No access is allowed.

## SFR

Address	Register	Symbol	Value after RESET	
0030 <sub>16</sub>	ROM space set register	ROA	XXXX X000 <sub>2</sub>	*
0031 <sub>16</sub>	Debug monitor space set register	DBA	1111 0000 <sub>2</sub>	*
0032 <sub>16</sub>	Expanded space set register 0	EXA0	0000 0000 <sub>2</sub>	*
0033 <sub>16</sub>	Expanded space set register 1	EXA1	0000 0000 <sub>2</sub>	*
0034 <sub>16</sub>	Expanded space set register 2	EXA2	0000 0000 <sub>2</sub>	*
0035 <sub>16</sub>	Expanded space set register 3	EXA3	0000 0000 <sub>2</sub>	*
0036 <sub>16</sub>				
0037 <sub>16</sub>				
0038 <sub>16</sub>				
0039 <sub>16</sub>				
003A <sub>16</sub>				
003B <sub>16</sub>				
003C <sub>16</sub>				
003D <sub>16</sub>				
003E <sub>16</sub>				
003F <sub>16</sub>				
0040 <sub>16</sub>	DRAM control register	DRAMCONT	?XXX ???? <sub>2</sub>	
0041 <sub>16</sub>	DRAM refresh interval set register	REFCNT	?? <sub>16</sub>	
0042 <sub>16</sub>				
0043 <sub>16</sub>				
0044 <sub>16</sub>				
0045 <sub>16</sub>				
0046 <sub>16</sub>				
0047 <sub>16</sub>				
0048 <sub>16</sub>				
0049 <sub>16</sub>				
004A <sub>16</sub>				
004B <sub>16</sub>				
004C <sub>16</sub>				
004D <sub>16</sub>				
004E <sub>16</sub>				
004F <sub>16</sub>				
0050 <sub>16</sub>				
0051 <sub>16</sub>				
0052 <sub>16</sub>				
0053 <sub>16</sub>				
0054 <sub>16</sub>				
0055 <sub>16</sub>	Flash memory control register 2	FMR2	XXXX X0X0 <sub>2</sub>	*
0056 <sub>16</sub>	Flash memory control register 1	FMR1	XXXX XXX0 <sub>2</sub>	*
0057 <sub>16</sub>	Flash memory control register 0	FMR0	XX00 0001 <sub>2</sub>	
0058 <sub>16</sub>				
0059 <sub>16</sub>				
005A <sub>16</sub>				
005B <sub>16</sub>				
005C <sub>16</sub>				
005D <sub>16</sub>				
005E <sub>16</sub>				
005F <sub>16</sub>				

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

Users cannot use any symbols with \*. No access is allowed.

## SFR

Address	Register	Symbol	Value after RESET
0060 <sub>16</sub>			
0061 <sub>16</sub>			
0062 <sub>16</sub>			
0063 <sub>16</sub>			
0064 <sub>16</sub>			
0065 <sub>16</sub>			
0066 <sub>16</sub>			
0067 <sub>16</sub>			
0068 <sub>16</sub>	DMA0 interrupt control register	DM0IC	XXXX ?0002
0069 <sub>16</sub>	Timer B5 interrupt control register	TB5IC	XXXX ?0002
006A <sub>16</sub>	DMA2 interrupt control register	DM2IC	XXXX ?0002
006B <sub>16</sub>	UART2 receive /ACK interrupt control register	S2RIC	XXXX ?0002
006C <sub>16</sub>	Timer A0 interrupt control register	TA0IC	XXXX ?0002
006D <sub>16</sub>	UART3 receive /ACK interrupt control register	S3RIC	XXXX ?0002
006E <sub>16</sub>	Timer A2 interrupt control register	TA2IC	XXXX ?0002
006F <sub>16</sub>	UART4 receive /ACK interrupt control register	S4RIC	XXXX ?0002
0070 <sub>16</sub>	Timer A4 interrupt control register	TA4IC	XXXX ?0002
0071 <sub>16</sub>	UART0/UART3 bus conflict detect interrupt control register	BCN0IC/BCN3IC	XXXX ?0002
0072 <sub>16</sub>	UART0 receive/ACK interrupt control register	S0RIC	XXXX ?0002
0073 <sub>16</sub>	A-D0 conversion interrupt control register	AD0IC	XXXX ?0002
0074 <sub>16</sub>	UART1 receive/ACK interrupt control register	S1RIC	XXXX ?0002
0075 <sub>16</sub>	Intelligent I/O interrupt control register 0	IIO0IC	XXXX ?0002
0076 <sub>16</sub>	Timer B1 interrupt control register	TB1IC	XXXX ?0002
0077 <sub>16</sub>	Intelligent I/O interrupt control register 2	IIO2IC	XXXX ?0002
0078 <sub>16</sub>	Timer B3 interrupt control register	TB3IC	XXXX ?0002
0079 <sub>16</sub>	Intelligent I/O interrupt control register 4	IIO4IC	XXXX ?0002
007A <sub>16</sub>	INT5 interrupt control register	INT5IC	XX00 ?0002
007B <sub>16</sub>	Intelligent I/O interrupt control register 6	IIO6IC	XXXX ?0002
007C <sub>16</sub>	INT3 interrupt control register	INT3IC	XX00 ?0002
007D <sub>16</sub>	Intelligent I/O interrupt control register 8	IIO8IC	XXXX ?0002
007E <sub>16</sub>	INT1 interrupt control register	INT1IC	XX00 ?0002
007F <sub>16</sub>	Intelligent I/O interrupt control register 10/ CAN interrupt 1 control register	IIO10IC CAN1IC	XXXX ?0002
0080 <sub>16</sub>			
0081 <sub>16</sub>	Intelligent I/O interrupt control register 11/ CAN interrupt 2 control register	IIO11IC CAN2IC	XXXX ?0002
0082 <sub>16</sub>			
0083 <sub>16</sub>			
0084 <sub>16</sub>			
0085 <sub>16</sub>			
0086 <sub>16</sub>	A-D1 conversion interrupt control register	AD1IC	XXXX ?0002
0087 <sub>16</sub>			
0088 <sub>16</sub>	DMA1 interrupt control register	DM1IC	XXXX ?0002
0089 <sub>16</sub>	UART2 transmit /NACK interrupt control register	S2TIC	XXXX ?0002
008A <sub>16</sub>	DMA3 interrupt control register	DM3IC	XXXX ?0002
008B <sub>16</sub>	UART3 transmit /NACK interrupt control register	S3TIC	XXXX ?0002
008C <sub>16</sub>	Timer A1 interrupt control register	TA1IC	XXXX ?0002
008D <sub>16</sub>	UART4 transmit /NACK interrupt control register	S4TIC	XXXX ?0002
008E <sub>16</sub>	Timer A3 interrupt control register	TA3IC	XXXX ?0002
008F <sub>16</sub>	UART2 bus conflict detect interrupt control register	BCN2IC	XXXX ?0002

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

## SFR

Address	Register	Symbol	Value after RESET
0090 <sub>16</sub>	UART0 transmit /NACK interrupt control register	S0TIC	XXXX ?000 <sub>2</sub>
0091 <sub>16</sub>	UART1/UART4 bus conflict detect interrupt control register	BCN11C/BCN41C	XXXX ?000 <sub>2</sub>
0092 <sub>16</sub>	UART1 transmit/NACK interrupt control register	S1TIC	XXXX ?000 <sub>2</sub>
0093 <sub>16</sub>	Key input interrupt control register	KUPIC	XXXX ?000 <sub>2</sub>
0094 <sub>16</sub>	Timer B0 interrupt control register	TB0IC	XXXX ?000 <sub>2</sub>
0095 <sub>16</sub>	Intelligent I/O interrupt control register 1	IIO1IC	XXXX ?000 <sub>2</sub>
0096 <sub>16</sub>	Timer B2 interrupt control register	TB2IC	XXXX ?000 <sub>2</sub>
0097 <sub>16</sub>	Intelligent I/O interrupt control register 3	IIO3IC	XXXX ?000 <sub>2</sub>
0098 <sub>16</sub>	Timer B4 interrupt control register	TB4IC	XXXX ?000 <sub>2</sub>
0099 <sub>16</sub>	Intelligent I/O interrupt control register 5	IIO5IC	XXXX ?000 <sub>2</sub>
009A <sub>16</sub>	INT4 interrupt control register	INT4IC	XX00 ?000 <sub>2</sub>
009B <sub>16</sub>	Intelligent I/O interrupt control register 7	IIO7IC	XXXX ?000 <sub>2</sub>
009C <sub>16</sub>	INT2 interrupt control register	INT2IC	XX00 ?000 <sub>2</sub>
009D <sub>16</sub>	Intelligent I/O interrupt control register 9/ CAN interrupt 0 control register	IIO9IC CAN0IC	XXXX ?000 <sub>2</sub>
009E <sub>16</sub>	INT0 interrupt control register	INT0IC	XX00 ?000 <sub>2</sub>
009F <sub>16</sub>	Exit priority control register	RLVL	XX0X 0000 <sub>2</sub>
00A0 <sub>16</sub>	Interrupt request register 0	IIO0IR	0000 000X <sub>2</sub>
00A1 <sub>16</sub>	Interrupt request register 1	IIO1IR	0000 000X <sub>2</sub>
00A2 <sub>16</sub>	Interrupt request register 2	IIO2IR	0000 000X <sub>2</sub>
00A3 <sub>16</sub>	Interrupt request register 3	IIO3IR	0000 000X <sub>2</sub>
00A4 <sub>16</sub>	Interrupt request register 4	IIO4IR	0000 000X <sub>2</sub>
00A5 <sub>16</sub>	Interrupt request register 5	IIO5IR	0000 000X <sub>2</sub>
00A6 <sub>16</sub>	Interrupt request register 6	IIO6IR	0000 000X <sub>2</sub>
00A7 <sub>16</sub>	Interrupt request register 7	IIO7IR	0000 000X <sub>2</sub>
00A8 <sub>16</sub>	Interrupt request register 8	IIO8IR	0000 000X <sub>2</sub>
00A9 <sub>16</sub>	Interrupt request register 9	IIO9IR	0000 000X <sub>2</sub>
00AA <sub>16</sub>	Interrupt request register 10	IIO10IR	0000 000X <sub>2</sub>
00AB <sub>16</sub>	Interrupt request register 11	IIO11IR	0000 000X <sub>2</sub>
00AC <sub>16</sub>			
00AD <sub>16</sub>			
00AE <sub>16</sub>			
00AF <sub>16</sub>			
00B0 <sub>16</sub>	Interrupt enable register 0	IIO0IE	0000 0000 <sub>2</sub>
00B1 <sub>16</sub>	Interrupt enable register 1	IIO1IE	0000 0000 <sub>2</sub>
00B2 <sub>16</sub>	Interrupt enable register 2	IIO2IE	0000 0000 <sub>2</sub>
00B3 <sub>16</sub>	Interrupt enable register 3	IIO3IE	0000 0000 <sub>2</sub>
00B4 <sub>16</sub>	Interrupt enable register 4	IIO4IE	0000 0000 <sub>2</sub>
00B5 <sub>16</sub>	Interrupt enable register 5	IIO5IE	0000 0000 <sub>2</sub>
00B6 <sub>16</sub>	Interrupt enable register 6	IIO6IE	0000 0000 <sub>2</sub>
00B7 <sub>16</sub>	Interrupt enable register 7	IIO7IE	0000 0000 <sub>2</sub>
00B8 <sub>16</sub>	Interrupt enable register 8	IIO8IE	0000 0000 <sub>2</sub>
00B9 <sub>16</sub>	Interrupt enable register 9	IIO9IE	0000 0000 <sub>2</sub>
00BA <sub>16</sub>	Interrupt enable register 10	IIO10IE	0000 0000 <sub>2</sub>
00BB <sub>16</sub>	Interrupt enable register 11	IIO11IE	0000 0000 <sub>2</sub>
00BC <sub>16</sub>			
00BD <sub>16</sub>			
00BE <sub>16</sub>			
00BF <sub>16</sub>			

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

## SFR

Address	Register	Symbol	Value after RESET
00C0 <sup>16</sup> 00C1 <sup>16</sup>	Group 0 time measurement/waveform generation register 0	G0TM0/G0PO0	?? <sup>16</sup> ?? <sup>16</sup>
00C2 <sup>16</sup> 00C3 <sup>16</sup>	Group 0 time measurement/waveform generation register 1	G0TM1/G0PO1	?? <sup>16</sup> ?? <sup>16</sup>
00C4 <sup>16</sup> 00C5 <sup>16</sup>	Group 0 time measurement/waveform generation register 2	G0TM2/G0PO2	?? <sup>16</sup> ?? <sup>16</sup>
00C6 <sup>16</sup> 00C7 <sup>16</sup>	Group 0 time measurement/waveform generation register 3	G0TM3/G0PO3	?? <sup>16</sup> ?? <sup>16</sup>
00C8 <sup>16</sup> 00C9 <sup>16</sup>	Group 0 time measurement/waveform generation register 4	G0TM4/G0PO4	?? <sup>16</sup> ?? <sup>16</sup>
00CA <sup>16</sup> 00CB <sup>16</sup>	Group 0 time measurement/waveform generation register 5	G0TM5/G0PO5	?? <sup>16</sup> ?? <sup>16</sup>
00CC <sup>16</sup> 00CD <sup>16</sup>	Group 0 time measurement/waveform generation register 6	G0TM6/G0PO6	?? <sup>16</sup> ?? <sup>16</sup>
00CE <sup>16</sup> 00CF <sup>16</sup>	Group 0 time measurement/waveform generation register 7	G0TM7/G0PO7	?? <sup>16</sup> ?? <sup>16</sup>
00D0 <sup>16</sup>	Group 0 waveform generation control register 0	G0POCR0	0X00 X000 <sub>2</sub>
00D1 <sup>16</sup>	Group 0 waveform generation control register 1	G0POCR1	0X00 X000 <sub>2</sub>
00D2 <sup>16</sup>	Group 0 waveform generation control register 2	G0POCR2	0X00 X000 <sub>2</sub>
00D3 <sup>16</sup>	Group 0 waveform generation control register 3	G0POCR3	0X00 X000 <sub>2</sub>
00D4 <sup>16</sup>	Group 0 waveform generation control register 4	G0POCR4	0X00 X000 <sub>2</sub>
00D5 <sup>16</sup>	Group 0 waveform generation control register 5	G0POCR5	0X00 X000 <sub>2</sub>
00D6 <sup>16</sup>	Group 0 waveform generation control register 6	G0POCR6	0X00 X000 <sub>2</sub>
00D7 <sup>16</sup>	Group 0 waveform generation control register 7	G0POCR7	0X00 X000 <sub>2</sub>
00D8 <sup>16</sup>	Group 0 time measurement control register 0	G0TMCR0	0000 0000 <sub>2</sub>
00D9 <sup>16</sup>	Group 0 time measurement control register 1	G0TMCR1	0000 0000 <sub>2</sub>
00DA <sup>16</sup>	Group 0 time measurement control register 2	G0TMCR2	0000 0000 <sub>2</sub>
00DB <sup>16</sup>	Group 0 time measurement control register 3	G0TMCR3	0000 0000 <sub>2</sub>
00DC <sup>16</sup>	Group 0 time measurement control register 4	G0TMCR4	0000 0000 <sub>2</sub>
00DD <sup>16</sup>	Group 0 time measurement control register 5	G0TMCR5	0000 0000 <sub>2</sub>
00DE <sup>16</sup>	Group 0 time measurement control register 6	G0TMCR6	0000 0000 <sub>2</sub>
00DF <sup>16</sup>	Group 0 time measurement control register 7	G0TMCR7	0000 0000 <sub>2</sub>
00E0 <sup>16</sup> 00E1 <sup>16</sup>	Group 0 base timer register	G0BT	?? <sup>16</sup> ?? <sup>16</sup>
00E2 <sup>16</sup>	Group 0 base timer control register 0	G0BCR0	0000 0000 <sub>2</sub>
00E3 <sup>16</sup>	Group 0 base timer control register 1	G0BCR1	0000 0000 <sub>2</sub>
00E4 <sup>16</sup>	Group 0 time measurement prescaler register 6	G0TPR6	0000 0000 <sub>2</sub>
00E5 <sup>16</sup>	Group 0 time measurement prescaler register 7	G0TPR7	0000 0000 <sub>2</sub>
00E6 <sup>16</sup>	Group 0 function enable register	G0FE	0000 0000 <sub>2</sub>
00E7 <sup>16</sup>	Group 0 function select register	G0FS	0000 0000 <sub>2</sub>
00E8 <sup>16</sup> 00E9 <sup>16</sup>	Group 0 SI/O receive buffer register	G0RB	???? ???? <sub>2</sub> XX00 XXXX <sub>2</sub>
00EA <sup>16</sup>	Group 0 transmit buffer/receive data register	G0TB/G0DR	?? <sup>16</sup>
00EB <sup>16</sup>			
00EC <sup>16</sup>	Group 0 receive input register	G0RI	?? <sup>16</sup>
00ED <sup>16</sup>	Group 0 SI/O communication mode register	G0MR	0000 0000 <sub>2</sub>
00EE <sup>16</sup>	Group 0 transmit output register	G0TO	?? <sup>16</sup>
00EF <sup>16</sup>	Group 0 SI/O communication control register	G0CR	0000 X000 <sub>2</sub>

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

## SFR

Address	Register	Symbol	Value after RESET
00F0 <sub>16</sub>	Group 0 data compare register 0	G0CMP0	?? <sub>16</sub>
00F1 <sub>16</sub>	Group 0 data compare register 1	G0CMP1	?? <sub>16</sub>
00F2 <sub>16</sub>	Group 0 data compare register 2	G0CMP2	?? <sub>16</sub>
00F3 <sub>16</sub>	Group 0 data compare register 3	G0CMP3	?? <sub>16</sub>
00F4 <sub>16</sub>	Group 0 data mask register 0	G0MSK0	?? <sub>16</sub>
00F5 <sub>16</sub>	Group 0 data mask register 1	G0MSK1	?? <sub>16</sub>
00F6 <sub>16</sub>			
00F7 <sub>16</sub>			
00F8 <sub>16</sub> 00F9 <sub>16</sub>	Group 0 receive CRC code register	G0RCRC	?? <sub>16</sub> ?? <sub>16</sub>
00FA <sub>16</sub> 00FB <sub>16</sub>	Group 0 transmit CRC code register	G0TCRC	0000 0000 <sub>2</sub> 0000 0000 <sub>2</sub>
00FC <sub>16</sub>	Group 0 SI/O extended mode register	G0EMR	0000 0000 <sub>2</sub>
00FD <sub>16</sub>	Group 0 SI/O extended receive control register	G0ERC	0000 0000 <sub>2</sub>
00FE <sub>16</sub>	Group 0 SI/O special communication interrupt detect register	G0IRF	0000 00XX <sub>2</sub>
00FF <sub>16</sub>	Group 0 SI/O extended transmit control register	G0ETC	0000 0XXX <sub>2</sub>
0100 <sub>16</sub> 0101 <sub>16</sub>	Group 1 time measurement/waveform generation register 0	G1TM0/G1PO0	?? <sub>16</sub> ?? <sub>16</sub>
0102 <sub>16</sub> 0103 <sub>16</sub>	Group 1 time measurement/waveform generation register 1	G1TM1/G1PO1	?? <sub>16</sub> ?? <sub>16</sub>
0104 <sub>16</sub> 0105 <sub>16</sub>	Group 1 time measurement/waveform generation register 2	G1TM2/G1PO2	?? <sub>16</sub> ?? <sub>16</sub>
0106 <sub>16</sub> 0107 <sub>16</sub>	Group 1 time measurement/waveform generation register 3	G1TM3/G1PO3	?? <sub>16</sub> ?? <sub>16</sub>
0108 <sub>16</sub> 0109 <sub>16</sub>	Group 1 time measurement/waveform generation register 4	G1TM4/G1PO4	?? <sub>16</sub> ?? <sub>16</sub>
010A <sub>16</sub> 010B <sub>16</sub>	Group 1 time measurement/waveform generation register 5	G1TM5/G1PO5	?? <sub>16</sub> ?? <sub>16</sub>
010C <sub>16</sub> 010D <sub>16</sub>	Group 1 time measurement/waveform generation register 6	G1TM6/G1PO6	?? <sub>16</sub> ?? <sub>16</sub>
010E <sub>16</sub> 010F <sub>16</sub>	Group 1 time measurement/waveform generation register 7	G1TM7/G1PO7	?? <sub>16</sub> ?? <sub>16</sub>
0110 <sub>16</sub>	Group 1 waveform generation control register 0	G1POCR0	0X00 X000 <sub>2</sub>
0111 <sub>16</sub>	Group 1 waveform generation control register 1	G1POCR1	0X00 X000 <sub>2</sub>
0112 <sub>16</sub>	Group 1 waveform generation control register 2	G1POCR2	0X00 X000 <sub>2</sub>
0113 <sub>16</sub>	Group 1 waveform generation control register 3	G1POCR3	0X00 X000 <sub>2</sub>
0114 <sub>16</sub>	Group 1 waveform generation control register 4	G1POCR4	0X00 X000 <sub>2</sub>
0115 <sub>16</sub>	Group 1 waveform generation control register 5	G1POCR5	0X00 X000 <sub>2</sub>
0116 <sub>16</sub>	Group 1 waveform generation control register 6	G1POCR6	0X00 X000 <sub>2</sub>
0117 <sub>16</sub>	Group 1 waveform generation control register 7	G1POCR7	0X00 X000 <sub>2</sub>
0118 <sub>16</sub>	Group 1 time measurement control register 0	G1TMCR0	0000 0000 <sub>2</sub>
0119 <sub>16</sub>	Group 1 time measurement control register 1	G1TMCR1	0000 0000 <sub>2</sub>
011A <sub>16</sub>	Group 1 time measurement control register 2	G1TMCR2	0000 0000 <sub>2</sub>
011B <sub>16</sub>	Group 1 time measurement control register 3	G1TMCR3	0000 0000 <sub>2</sub>
011C <sub>16</sub>	Group 1 time measurement control register 4	G1TMCR4	0000 0000 <sub>2</sub>
011D <sub>16</sub>	Group 1 time measurement control register 5	G1TMCR5	0000 0000 <sub>2</sub>
011E <sub>16</sub>	Group 1 time measurement control register 6	G1TMCR6	0000 0000 <sub>2</sub>
011F <sub>16</sub>	Group 1 time measurement control register 7	G1TMCR7	0000 0000 <sub>2</sub>

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

## SFR

Address	Register	Symbol	Value after RESET
0120 <sub>16</sub>	Group 1 base timer register	G1BT	?? <sub>16</sub>
0121 <sub>16</sub>			?? <sub>16</sub>
0122 <sub>16</sub>	Group 1 base timer control register 0	G1BCR0	0000 0000 <sub>2</sub>
0123 <sub>16</sub>	Group 1 base timer control register 1	G1BCR1	0000 0000 <sub>2</sub>
0124 <sub>16</sub>	Group 1 time measurement prescaler register 6	G1TPR6	0000 0000 <sub>2</sub>
0125 <sub>16</sub>	Group 1 time measurement prescaler register 7	G1TPR7	0000 0000 <sub>2</sub>
0126 <sub>16</sub>	Group 1 function enable register	G1FE	0000 0000 <sub>2</sub>
0127 <sub>16</sub>	Group 1 function select register	G1FS	0000 0000 <sub>2</sub>
0128 <sub>16</sub>	Group 1 SI/O receive buffer register	G1RB	???? ???? <sub>2</sub>
0129 <sub>16</sub>			XX00 XXXX <sub>2</sub>
012A <sub>16</sub>	Group 1 transmit buffer/receive data register	G1TB/G1DR	?? <sub>16</sub>
012B <sub>16</sub>			
012C <sub>16</sub>	Group 1 receive input register	G1RI	?? <sub>16</sub>
012D <sub>16</sub>	Group 1 SI/O communication mode register	G1MR	0000 0000 <sub>2</sub>
012E <sub>16</sub>	Group 1 transmit output register	G1TO	?? <sub>16</sub>
012F <sub>16</sub>	Group 1 SI/O communication control register	G1CR	0000 X000 <sub>2</sub>
0130 <sub>16</sub>	Group 1 data compare register 0	G1CMP0	?? <sub>16</sub>
0131 <sub>16</sub>	Group 1 data compare register 1	G1CMP1	?? <sub>16</sub>
0132 <sub>16</sub>	Group 1 data compare register 2	G1CMP2	?? <sub>16</sub>
0133 <sub>16</sub>	Group 1 data compare register 3	G1CMP3	?? <sub>16</sub>
0134 <sub>16</sub>	Group 1 data mask register 0	G1MSK0	?? <sub>16</sub>
0135 <sub>16</sub>	Group 1 data mask register 1	G1MSK1	?? <sub>16</sub>
0136 <sub>16</sub>			
0137 <sub>16</sub>			
0138 <sub>16</sub>	Group 1 receive CRC code register	G1RCRC	?? <sub>16</sub>
0139 <sub>16</sub>			?? <sub>16</sub>
013A <sub>16</sub>	Group 1 transmit CRC code register	G1TCRC	0000 0000 <sub>2</sub>
013B <sub>16</sub>			0000 0000 <sub>2</sub>
013C <sub>16</sub>	Group 1 SI/O extended mode register	G1EMR	0000 0000 <sub>2</sub>
013D <sub>16</sub>	Group 1 SI/O extended receive control register	G1ERC	0000 0000 <sub>2</sub>
013E <sub>16</sub>	Group 1 SI/O special communication interrupt detect register	G1IRF	0000 00XX <sub>2</sub>
013F <sub>16</sub>	Group 1 SI/O extended transmit control register	G1ETC	0000 0XXX <sub>2</sub>
0140 <sub>16</sub>	Group 2 waveform generation register 0	G2PO0	?? <sub>16</sub>
0141 <sub>16</sub>			?? <sub>16</sub>
0142 <sub>16</sub>	Group 2 waveform generation register 1	G2PO1	?? <sub>16</sub>
0143 <sub>16</sub>			?? <sub>16</sub>
0144 <sub>16</sub>	Group 2 waveform generation register 2	G2PO2	?? <sub>16</sub>
0145 <sub>16</sub>			?? <sub>16</sub>
0146 <sub>16</sub>	Group 2 waveform generation register 3	G2PO3	?? <sub>16</sub>
0147 <sub>16</sub>			?? <sub>16</sub>
0148 <sub>16</sub>	Group 2 waveform generation register 4	G2PO4	?? <sub>16</sub>
0149 <sub>16</sub>			?? <sub>16</sub>
014A <sub>16</sub>	Group 2 waveform generation register 5	G2PO5	?? <sub>16</sub>
014B <sub>16</sub>			?? <sub>16</sub>
014C <sub>16</sub>	Group 2 waveform generation register 6	G2PO6	?? <sub>16</sub>
014D <sub>16</sub>			?? <sub>16</sub>
014E <sub>16</sub>	Group 2 waveform generation register 7	G2PO7	?? <sub>16</sub>
014F <sub>16</sub>			?? <sub>16</sub>

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

Address	Register	Symbol	Value after RESET
0150 <sub>16</sub>	Group 2 waveform generation control register 0	G2POCR0	0000 0000 <sub>2</sub>
0151 <sub>16</sub>	Group 2 waveform generation control register 1	G2POCR1	0000 0000 <sub>2</sub>
0152 <sub>16</sub>	Group 2 waveform generation control register 2	G2POCR2	0000 0000 <sub>2</sub>
0153 <sub>16</sub>	Group 2 waveform generation control register 3	G2POCR3	0000 0000 <sub>2</sub>
0154 <sub>16</sub>	Group 2 waveform generation control register 4	G2POCR4	0000 0000 <sub>2</sub>
0155 <sub>16</sub>	Group 2 waveform generation control register 5	G2POCR5	0000 0000 <sub>2</sub>
0156 <sub>16</sub>	Group 2 waveform generation control register 6	G2POCR6	0000 0000 <sub>2</sub>
0157 <sub>16</sub>	Group 2 waveform generation control register 7	G2POCR7	0000 0000 <sub>2</sub>
0158 <sub>16</sub>			
0159 <sub>16</sub>			
015A <sub>16</sub>			
015B <sub>16</sub>			
015C <sub>16</sub>			
015D <sub>16</sub>			
015E <sub>16</sub>			
015F <sub>16</sub>			
0160 <sub>16</sub> 0161 <sub>16</sub>	Group 2 base timer register	G2BT	?? <sub>16</sub> ?? <sub>16</sub>
0162 <sub>16</sub>	Group 2 base timer control register 0	G2BCR0	0000 0000 <sub>2</sub>
0163 <sub>16</sub>	Group 2 base timer control register 1	G2BCR1	0000 0000 <sub>2</sub>
0164 <sub>16</sub>	Base timer start register	BTSR	XXXX 0000 <sub>2</sub>
0165 <sub>16</sub>			
0166 <sub>16</sub>	Group 2 function enable register	G2FE	0000 0000 <sub>2</sub>
0167 <sub>16</sub>	Group 2 RTP output buffer register	G2RTP	0000 0000 <sub>2</sub>
0168 <sub>16</sub>			
0169 <sub>16</sub>			
016A <sub>16</sub>	Group 2 SI/O communication mode register	G2MR	00XX X000 <sub>2</sub>
016B <sub>16</sub>	Group 2 SI/O communication control register	G2CR	0000 X000 <sub>2</sub>
016C <sub>16</sub> 016D <sub>16</sub>	Group 2 SI/O transmit buffer register	G2TB	???? ???? <sub>2</sub> ???X X??? <sub>2</sub>
016E <sub>16</sub> 016F <sub>16</sub>	Group 2 SI/O receive buffer register	G2RB	???? ???? <sub>2</sub> XXX? XXXX <sub>2</sub>
0170 <sub>16</sub> 0171 <sub>16</sub>	Group 2 IE Bus address register	IEAR	???? ???? <sub>2</sub> XXXX ???? <sub>2</sub>
0172 <sub>16</sub>	Group 2 IE Bus control register	IECR	00XX X000 <sub>2</sub>
0173 <sub>16</sub>	Group 2 IE Bus transmit interrupt cause detect register	IETIF	XXX0 0000 <sub>2</sub>
0174 <sub>16</sub>	Group 2 IE Bus receive interrupt cause detect register	IERIF	XXX0 0000 <sub>2</sub>
0175 <sub>16</sub>			
0176 <sub>16</sub>			
0177 <sub>16</sub>			
0178 <sub>16</sub>	Input function select register	IPS	0000 0000 <sub>2</sub>
0179 <sub>16</sub>			
017A <sub>16</sub>	Group 3 SI/O communication mode register	G3MR	00XX 0000 <sub>2</sub>
017B <sub>16</sub>	Group 3 SI/O communication control register	G3CR	0000 X000 <sub>2</sub>
017C <sub>16</sub> 017D <sub>16</sub>	Group 3 SI/O transmit buffer register	G3TB	?? <sub>16</sub> ?? <sub>16</sub>
017E <sub>16</sub> 017F <sub>16</sub>	Group 3 SI/O receive buffer register	G3RB	?? <sub>16</sub> ?? <sub>16</sub>

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

## SFR

Address	Register	Symbol	Value after RESET
0180 <sub>16</sub> 0181 <sub>16</sub>	Group 3 waveform generation register 0	G3PO0	?? <sub>16</sub> ?? <sub>16</sub>
0182 <sub>16</sub> 0183 <sub>16</sub>	Group 3 waveform generation register 1	G3PO1	?? <sub>16</sub> ?? <sub>16</sub>
0184 <sub>16</sub> 0185 <sub>16</sub>	Group 3 waveform generation register 2	G3PO2	?? <sub>16</sub> ?? <sub>16</sub>
0186 <sub>16</sub> 0187 <sub>16</sub>	Group 3 waveform generation register 3	G3PO3	?? <sub>16</sub> ?? <sub>16</sub>
0188 <sub>16</sub> 0189 <sub>16</sub>	Group 3 waveform generation register 4	G3PO4	?? <sub>16</sub> ?? <sub>16</sub>
018A <sub>16</sub> 018B <sub>16</sub>	Group 3 waveform generation register 5	G3PO5	?? <sub>16</sub> ?? <sub>16</sub>
018C <sub>16</sub> 018D <sub>16</sub>	Group 3 waveform generation register 6	G3PO6	?? <sub>16</sub> ?? <sub>16</sub>
018E <sub>16</sub> 018F <sub>16</sub>	Group 3 waveform generation register 7	G3PO7	?? <sub>16</sub> ?? <sub>16</sub>
0190 <sub>16</sub>	Group 3 waveform generation control register 0	G3POCR0	0000 0000 <sub>2</sub>
0191 <sub>16</sub>	Group 3 waveform generation control register 1	G3POCR1	0000 0000 <sub>2</sub>
0192 <sub>16</sub>	Group 3 waveform generation control register 2	G3POCR2	0000 0000 <sub>2</sub>
0193 <sub>16</sub>	Group 3 waveform generation control register 3	G3POCR3	0000 0000 <sub>2</sub>
0194 <sub>16</sub>	Group 3 waveform generation control register 4	G3POCR4	0000 0000 <sub>2</sub>
0195 <sub>16</sub>	Group 3 waveform generation control register 5	G3POCR5	0000 0000 <sub>2</sub>
0196 <sub>16</sub>	Group 3 waveform generation control register 6	G3POCR6	0000 0000 <sub>2</sub>
0197 <sub>16</sub>	Group 3 waveform generation control register 7	G3POCR7	0000 0000 <sub>2</sub>
0198 <sub>16</sub> 0199 <sub>16</sub>	Group 3 waveform generation mask register 4	G3MK4	?? <sub>16</sub> ?? <sub>16</sub>
019A <sub>16</sub> 019B <sub>16</sub>	Group 3 waveform generation mask register 5	G3MK5	?? <sub>16</sub> ?? <sub>16</sub>
019C <sub>16</sub> 019D <sub>16</sub>	Group 3 waveform generation mask register 6	G3MK6	?? <sub>16</sub> ?? <sub>16</sub>
019E <sub>16</sub> 019F <sub>16</sub>	Group 3 waveform generation mask register 7	G3MK7	?? <sub>16</sub> ?? <sub>16</sub>
01A0 <sub>16</sub> 01A1 <sub>16</sub>	Group 3 base timer register	G3BT	?? <sub>16</sub> ?? <sub>16</sub>
01A2 <sub>16</sub>	Group 3 base timer control register 0	G3BCR0	0000 0000 <sub>2</sub>
01A3 <sub>16</sub>	Group 3 base timer control register 1	G3BCR1	0000 0000 <sub>2</sub>
01A4 <sub>16</sub>			
01A5 <sub>16</sub>			
01A6 <sub>16</sub>	Group 3 function enable register	G3FE	0000 0000 <sub>2</sub>
01A7 <sub>16</sub>	Group 3 RTP output buffer register	G3RTP	0000 0000 <sub>2</sub>
01A8 <sub>16</sub>			
01A9 <sub>16</sub>			
01AA <sub>16</sub>			
01AB <sub>16</sub>			
01AC <sub>16</sub>	Group 3 HDLC communication control register	HDLC	0000 0000 <sub>2</sub> *
01AD <sub>16</sub>	Group 3 SI/O communication flag register	G3FLG	XXXX XXX0 <sub>2</sub>
01AE <sub>16</sub> 01AF <sub>16</sub>	Group 3 HDLC transmit counter	HCNT	0000 0000 0000 0000 *

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

Users cannot use any symbols with \*. No access is allowed.

## SFR

Address	Register	Symbol	Value after RESET	
01B0 <sub>16</sub> 01B1 <sub>16</sub>	Group 3 HDLC address compare register 0	HADR0	0000 0000 <sub>2</sub> 0000 0000 <sub>2</sub>	*
01B2 <sub>16</sub> 01B3 <sub>16</sub>	Group 3 HDLC address mask register 0	HMSK0	0000 0000 <sub>2</sub> 0000 0000 <sub>2</sub>	*
01B4 <sub>16</sub> 01B5 <sub>16</sub>	Group 3 HDLC address compare register 1	HADR1	0000 0000 <sub>2</sub> 0000 0000 <sub>2</sub>	*
01B6 <sub>16</sub> 01B7 <sub>16</sub>	Group 3 HDLC address mask register 1	HMSK1	0000 0000 <sub>2</sub> 0000 0000 <sub>2</sub>	*
01B8 <sub>16</sub> 01B9 <sub>16</sub>	Group 3 HDLC address compare register 2	HADR2	0000 0000 <sub>2</sub> 0000 0000 <sub>2</sub>	*
01BA <sub>16</sub> 01BB <sub>16</sub>	Group 3 HDLC address mask register 2	HMSK2	0000 0000 <sub>2</sub> 0000 0000 <sub>2</sub>	*
01BC <sub>16</sub> 01BD <sub>16</sub>	Group 3 HDLC address compare register 3	HADR3	0000 0000 <sub>2</sub> 0000 0000 <sub>2</sub>	*
01BE <sub>16</sub> 01BF <sub>16</sub>	Group 3 HDLC address mask register 3	HMSK3	0000 0000 <sub>2</sub> 0000 0000 <sub>2</sub>	*
01C0 <sub>16</sub> 01C1 <sub>16</sub>	A-D1 register 0	AD10	???? ????? <sub>2</sub> XXXX XX?? <sub>2</sub>	
01C2 <sub>16</sub> 01C3 <sub>16</sub>	A-D1 register 1	AD11	???? ????? <sub>2</sub> XXXX XX?? <sub>2</sub>	
01C4 <sub>16</sub> 01C5 <sub>16</sub>	A-D1 register 2	AD12	???? ????? <sub>2</sub> XXXX XX?? <sub>2</sub>	
01C6 <sub>16</sub> 01C7 <sub>16</sub>	A-D1 register 3	AD13	???? ????? <sub>2</sub> XXXX XX?? <sub>2</sub>	
01C8 <sub>16</sub> 01C9 <sub>16</sub>	A-D1 register 4	AD14	???? ????? <sub>2</sub> XXXX XX?? <sub>2</sub>	
01CA <sub>16</sub> 01CB <sub>16</sub>	A-D1 register 5	AD15	???? ????? <sub>2</sub> XXXX XX?? <sub>2</sub>	
01CC <sub>16</sub> 01CD <sub>16</sub>	A-D1 register 6	AD16	???? ????? <sub>2</sub> XXXX XX?? <sub>2</sub>	
01CE <sub>16</sub> 01CF <sub>16</sub>	A-D1 register 7	AD17	???? ????? <sub>2</sub> XXXX XX?? <sub>2</sub>	
01D0 <sub>16</sub>				
01D1 <sub>16</sub>				
01D2 <sub>16</sub>				
01D3 <sub>16</sub>				
01D4 <sub>16</sub> 01D5 <sub>16</sub>	A-D1 control register 2	AD1CON2	X00X X000 <sub>2</sub>	
01D6 <sub>16</sub>	A-D1 control register 0	AD1CON0	0000 0000 <sub>2</sub>	
01D7 <sub>16</sub>	A-D1 control register 1	AD1CON1	XX00 0000 <sub>2</sub>	
01D8 <sub>16</sub>				
01D9 <sub>16</sub>				
01DA <sub>16</sub>				
01DB <sub>16</sub>				
01DC <sub>16</sub>				
01DD <sub>16</sub>				
01DE <sub>16</sub>				
01DF <sub>16</sub>				

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

Users cannot use any symbols with \*. No access is allowed.

## SFR

Address	Register	Symbol	Value after RESET
01E0 <sub>16</sub>	CAN0 message slot buffer 0 standard ID0	C0SLOT0_0	XXX? ???? <sub>2</sub>
01E1 <sub>16</sub>	CAN0 message slot buffer 0 standard ID1	C0SLOT0_1	XX?? ???? <sub>2</sub>
01E2 <sub>16</sub>	CAN0 message slot buffer 0 extended ID0	C0SLOT0_2	XXXX ???? <sub>2</sub>
01E3 <sub>16</sub>	CAN0 message slot buffer 0 extended ID1	C0SLOT0_3	?? <sub>16</sub>
01E4 <sub>16</sub>	CAN0 message slot buffer 0 extended ID2	C0SLOT0_4	XX?? ???? <sub>2</sub>
01E5 <sub>16</sub>	CAN0 message slot buffer 0 data length code	C0SLOT0_5	XXXX ???? <sub>2</sub>
01E6 <sub>16</sub>	CAN0 message slot buffer 0 data 0	C0SLOT0_6	?? <sub>16</sub>
01E7 <sub>16</sub>	CAN0 message slot buffer 0 data 1	C0SLOT0_7	?? <sub>16</sub>
01E8 <sub>16</sub>	CAN0 message slot buffer 0 data 2	C0SLOT0_8	?? <sub>16</sub>
01E9 <sub>16</sub>	CAN0 message slot buffer 0 data 3	C0SLOT0_9	?? <sub>16</sub>
01EA <sub>16</sub>	CAN0 message slot buffer 0 data 4	C0SLOT0_10	?? <sub>16</sub>
01EB <sub>16</sub>	CAN0 message slot buffer 0 data 5	C0SLOT0_11	?? <sub>16</sub>
01EC <sub>16</sub>	CAN0 message slot buffer 0 data 6	C0SLOT0_12	?? <sub>16</sub>
01ED <sub>16</sub>	CAN0 message slot buffer 0 data 7	C0SLOT0_13	?? <sub>16</sub>
01EE <sub>16</sub>	CAN0 message slot buffer 0 time stamp high-order	C0SLOT0_14	?? <sub>16</sub>
01EF <sub>16</sub>	CAN0 message slot buffer 0 time stamp low-order	C0SLOT0_15	?? <sub>16</sub>
01F0 <sub>16</sub>	CAN0 message slot buffer 1 standard ID0	C0SLOT1_0	XXX? ???? <sub>2</sub>
01F1 <sub>16</sub>	CAN0 message slot buffer 1 standard ID1	C0SLOT1_1	XX?? ???? <sub>2</sub>
01F2 <sub>16</sub>	CAN0 message slot buffer 1 extended ID0	C0SLOT1_2	XXXX ???? <sub>2</sub>
01F3 <sub>16</sub>	CAN0 message slot buffer 1 extended ID1	C0SLOT1_3	?? <sub>16</sub>
01F4 <sub>16</sub>	CAN0 message slot buffer 1 extended ID2	C0SLOT1_4	XX?? ???? <sub>2</sub>
01F5 <sub>16</sub>	CAN0 message slot buffer 1 data length code	C0SLOT1_5	XXXX ???? <sub>2</sub>
01F6 <sub>16</sub>	CAN0 message slot buffer 1 data 0	C0SLOT1_6	?? <sub>16</sub>
01F7 <sub>16</sub>	CAN0 message slot buffer 1 data 1	C0SLOT1_7	?? <sub>16</sub>
01F8 <sub>16</sub>	CAN0 message slot buffer 1 data 2	C0SLOT1_8	?? <sub>16</sub>
01F9 <sub>16</sub>	CAN0 message slot buffer 1 data 3	C0SLOT1_9	?? <sub>16</sub>
01FA <sub>16</sub>	CAN0 message slot buffer 1 data 4	C0SLOT1_10	?? <sub>16</sub>
01FB <sub>16</sub>	CAN0 message slot buffer 1 data 5	C0SLOT1_11	?? <sub>16</sub>
01FC <sub>16</sub>	CAN0 message slot buffer 1 data 6	C0SLOT1_12	?? <sub>16</sub>
01FD <sub>16</sub>	CAN0 message slot buffer 1 data 7	C0SLOT1_13	?? <sub>16</sub>
01FE <sub>16</sub>	CAN0 message slot buffer 1 time stamp high-order	C0SLOT1_14	?? <sub>16</sub>
01FF <sub>16</sub>	CAN0 message slot buffer 1 time stamp low-order	C0SLOT1_15	?? <sub>16</sub>
0200 <sub>16</sub> 0201 <sub>16</sub>	CAN0 control register 0	C0CTRL0	XX01 0X01 <sub>2</sub> <sup>1</sup> XXXX 0000 <sub>2</sub> <sup>1</sup>
0202 <sub>16</sub> 0203 <sub>16</sub>	CAN0 status register	C0STR	0000 0000 <sub>2</sub> <sup>1</sup> X000 0X01 <sub>2</sub> <sup>1</sup>
0204 <sub>16</sub> 0205 <sub>16</sub>	CAN0 extended ID register	C0IDR	0000 0000 <sub>2</sub> <sup>1</sup> 0000 0000 <sub>2</sub> <sup>1</sup>
0206 <sub>16</sub> 0207 <sub>16</sub>	CAN0 configuration register	C0CONR	0000 XXXX <sub>2</sub> <sup>1</sup> 0000 0000 <sub>2</sub> <sup>1</sup>
0208 <sub>16</sub> 0209 <sub>16</sub>	CAN0 time stamp register	C0TSR	0000 0000 <sub>2</sub> <sup>1</sup> 0000 0000 <sub>2</sub> <sup>1</sup>
020A <sub>16</sub>	CAN0 transmit error count register	C0TEC	0000 0000 <sub>2</sub> <sup>1</sup>
020B <sub>16</sub>	CAN0 receive error count register	C0REC	0000 0000 <sub>2</sub> <sup>1</sup>
020C <sub>16</sub> 020D <sub>16</sub>	CAN0 slot interrupt status register	C0SISTR	0000 0000 <sub>2</sub> <sup>1</sup> 0000 0000 <sub>2</sub> <sup>1</sup>
020E <sub>16</sub>			
020F <sub>16</sub>			

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

Notes :

1. Values are obtained by setting the SLEEP bit in the C0SLPR register to "1" (sleep mode exited) and allocating a clock to CAN module after reset.

Address	Register	Symbol	Value after RESET
0210 <sub>16</sub>	CAN0 slot interrupt mask register	C0SIMKR	0000 0000 <sub>2</sub> <sup>2</sup>
0211 <sub>16</sub>			0000 0000 <sub>2</sub> <sup>2</sup>
0212 <sub>16</sub>			
0213 <sub>16</sub>			
0214 <sub>16</sub>	CAN0 error interrupt mask register	C0EIMKR	XXXX X000 <sub>2</sub> <sup>2</sup>
0215 <sub>16</sub>	CAN0 error interrupt status register	C0EISTR	XXXX X000 <sub>2</sub> <sup>2</sup>
0216 <sub>16</sub>			
0217 <sub>16</sub>	CAN0 baud rate prescaler	C0BRP	0000 0001 <sub>2</sub> <sup>2</sup>
0218 <sub>16</sub>			
0219 <sub>16</sub>			
021A <sub>16</sub>			
021B <sub>16</sub>			
021C <sub>16</sub>			
021D <sub>16</sub>			
021E <sub>16</sub>			
021F <sub>16</sub>			
0220 <sub>16</sub>			
0221 <sub>16</sub>			
0222 <sub>16</sub>			
0223 <sub>16</sub>			
0224 <sub>16</sub>			
0225 <sub>16</sub>			
0226 <sub>16</sub>			
0227 <sub>16</sub>			
0228 <sub>16</sub>	CAN0 global mask register standard ID0	C0GMR0	XXX0 0000 <sub>2</sub> <sup>2</sup>
0229 <sub>16</sub>	CAN0 global mask register standard ID1	C0GMR1	XX00 0000 <sub>2</sub> <sup>2</sup>
022A <sub>16</sub>	CAN0 global mask register extended ID0	C0GMR2	XXXX 0000 <sub>2</sub> <sup>2</sup>
022B <sub>16</sub>	CAN0 global mask register extended ID1	C0GMR3	0000 0000 <sub>2</sub> <sup>2</sup>
022C <sub>16</sub>	CAN0 global mask register extended ID2	C0GMR4	XX00 0000 <sub>2</sub> <sup>2</sup>
022D <sub>16</sub>			
022E <sub>16</sub>			
022F <sub>16</sub>			
0230 <sub>16</sub>	CAN0 message slot 0 control register / CAN0 local mask register A standard ID0	C0MCTL0/ C0LMAR0	0000 0000 <sub>2</sub> <sup>2</sup> XXX0 0000 <sub>2</sub> <sup>2</sup>
0231 <sub>16</sub>	CAN0 message slot 1 control register / CAN0 local mask register A standard ID1	C0MCTL1/ C0LMAR1	0000 0000 <sub>2</sub> <sup>2</sup> XX00 0000 <sub>2</sub> <sup>2</sup>
0232 <sub>16</sub>	CAN0 message slot 2 control register / CAN0 local mask register A extended ID0	C0MCTL2/ C0LMAR2	0000 0000 <sub>2</sub> <sup>2</sup> XXXX 0000 <sub>2</sub> <sup>2</sup>
0233 <sub>16</sub>	CAN0 message slot 3 control register / CAN0 local mask register A extended ID1	C0MCTL3/ C0LMAR3	0000 0000 <sub>2</sub> <sup>2</sup> 0000 0000 <sub>2</sub> <sup>2</sup>
0234 <sub>16</sub>	CAN0 message slot 4 control register / CAN0 local mask register A extended ID2	C0MCTL4/ C0LMAR4	0000 0000 <sub>2</sub> <sup>2</sup> XX00 0000 <sub>2</sub> <sup>2</sup>
0235 <sub>16</sub>	CAN0 message slot 5 control register	C0MCTL5	0000 0000 <sub>2</sub> <sup>2</sup>
0236 <sub>16</sub>	CAN0 message slot 6 control register	C0MCTL6	0000 0000 <sub>2</sub> <sup>2</sup>
0237 <sub>16</sub>	CAN0 message slot 7 control register	C0MCTL7	0000 0000 <sub>2</sub> <sup>2</sup>
0238 <sub>16</sub>	CAN0 message slot 8 control register / CAN0 local mask register B standard ID0	C0MCTL8/ C0LMBR0	0000 0000 <sub>2</sub> <sup>2</sup> XXX0 0000 <sub>2</sub> <sup>2</sup>

(Note 1)

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

Notes :

- Addresses 0230<sub>16</sub> to 023F<sub>16</sub> are switched its function in the BankSel bit of C0CTLR1 register.
- Values are obtained by setting the SLEEP bit in the C0SLPR register to "1" (sleep mode exited) and allocating a clock to CAN module after reset.



## SFR

Address	Register	Symbol	Value after RESET
02C0 <sub>16</sub> 02C1 <sub>16</sub>	X0 register Y0 register	X0R,Y0R	?? <sub>16</sub> ?? <sub>16</sub>
02C2 <sub>16</sub> 02C3 <sub>16</sub>	X1 register Y1 register	X1R,Y1R	?? <sub>16</sub> ?? <sub>16</sub>
02C4 <sub>16</sub> 02C5 <sub>16</sub>	X2 register Y2 register	X2R,Y2R	?? <sub>16</sub> ?? <sub>16</sub>
02C6 <sub>16</sub> 02C7 <sub>16</sub>	X3 register Y3 register	X3R,Y3R	?? <sub>16</sub> ?? <sub>16</sub>
02C8 <sub>16</sub> 02C9 <sub>16</sub>	X4 register Y4 register	X4R,Y4R	?? <sub>16</sub> ?? <sub>16</sub>
02CA <sub>16</sub> 02CB <sub>16</sub>	X5 register Y5 register	X5R,Y5R	?? <sub>16</sub> ?? <sub>16</sub>
02CC <sub>16</sub> 02CD <sub>16</sub>	X6 register Y6 register	X6R,Y6R	?? <sub>16</sub> ?? <sub>16</sub>
02CE <sub>16</sub> 02CF <sub>16</sub>	X7 register Y7 register	X7R,Y7R	?? <sub>16</sub> ?? <sub>16</sub>
02D0 <sub>16</sub> 02D1 <sub>16</sub>	X8 register Y8 register	X8R,Y8R	?? <sub>16</sub> ?? <sub>16</sub>
02D2 <sub>16</sub> 02D3 <sub>16</sub>	X9 register Y9 register	X9R,Y9R	?? <sub>16</sub> ?? <sub>16</sub>
02D4 <sub>16</sub> 02D5 <sub>16</sub>	X10 register Y10 register	X10R,Y10R	?? <sub>16</sub> ?? <sub>16</sub>
02D6 <sub>16</sub> 02D7 <sub>16</sub>	X11 register Y11 register	X11R,Y11R	?? <sub>16</sub> ?? <sub>16</sub>
02D8 <sub>16</sub> 02D9 <sub>16</sub>	X12 register Y12 register	X12R,Y12R	?? <sub>16</sub> ?? <sub>16</sub>
02DA <sub>16</sub> 02DB <sub>16</sub>	X13 register Y13 register	X13R,Y13R	?? <sub>16</sub> ?? <sub>16</sub>
02DC <sub>16</sub> 02DD <sub>16</sub>	X14 register Y14 register	X14R,Y14R	?? <sub>16</sub> ?? <sub>16</sub>
02DE <sub>16</sub> 02DF <sub>16</sub>	X15 register Y15 register	X15R,Y15R	?? <sub>16</sub> ?? <sub>16</sub>
02E0 <sub>16</sub>	XY control register	XYC	XXXX XX00 <sub>2</sub>
02E1 <sub>16</sub>			
02E2 <sub>16</sub>			
02E3 <sub>16</sub>			
02E4 <sub>16</sub>	UART1 special mode register 4	U1SMR4	0000 0000 <sub>2</sub>
02E5 <sub>16</sub>	UART1 special mode register 3	U1SMR3	0000 0000 <sub>2</sub>
02E6 <sub>16</sub>	UART1 special mode register 2	U1SMR2	0000 0000 <sub>2</sub>
02E7 <sub>16</sub>	UART1 special mode register	U1SMR	0000 0000 <sub>2</sub>
02E8 <sub>16</sub>	UART1 transmit/receive mode register	U1MR	0000 0000 <sub>2</sub>
02E9 <sub>16</sub>	UART1 baud rate register	U1BRG	?? <sub>16</sub>
02EA <sub>16</sub> 02EB <sub>16</sub>	UART1 transmit buffer register	U1TB	???? ???? <sub>2</sub> XXXX XXX? <sub>2</sub>
02EC <sub>16</sub>	UART1 transmit/receive control register 0	U1C0	0000 1000 <sub>2</sub>
02ED <sub>16</sub>	UART1 transmit/receive control register 1	U1C1	0000 0010 <sub>2</sub>
02EE <sub>16</sub> 02EF <sub>16</sub>	UART1 receive buffer register	U1RB	???? ???? <sub>2</sub> ???? ?XX? <sub>2</sub>

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

## SFR

Address	Register	Symbol	Value after RESET
02F0 <sub>16</sub>			
02F1 <sub>16</sub>			
02F2 <sub>16</sub>			
02F3 <sub>16</sub>			
02F4 <sub>16</sub>	UART4 special mode register 4	U4SMR4	0000 0000 <sub>2</sub>
02F5 <sub>16</sub>	UART4 special mode register 3	U4SMR3	0000 0000 <sub>2</sub>
02F6 <sub>16</sub>	UART4 special mode register 2	U4SMR2	0000 0000 <sub>2</sub>
02F7 <sub>16</sub>	UART4 special mode register	U4SMR	0000 0000 <sub>2</sub>
02F8 <sub>16</sub>	UART4 transmit/receive mode register	U4MR	0000 0000 <sub>2</sub>
02F9 <sub>16</sub>	UART4 baud rate register	U4BRG	?? <sub>16</sub>
02FA <sub>16</sub>	UART4 transmit buffer register	U4TB	???? ???? <sub>2</sub>
02FB <sub>16</sub>			XXXX XXX <sub>2</sub>
02FC <sub>16</sub>	UART4 transmit/receive control register 0	U4C0	0000 1000 <sub>2</sub>
02FD <sub>16</sub>	UART4 transmit/receive control register 1	U4C1	0000 0010 <sub>2</sub>
02FE <sub>16</sub>	UART4 receive buffer register	U4RB	???? ???? <sub>2</sub>
02FF <sub>16</sub>			???? ?XX <sub>2</sub>
0300 <sub>16</sub>	Timer B3,B4,B5 count start flag	TBSR	000X XXXX <sub>2</sub>
0301 <sub>16</sub>			
0302 <sub>16</sub>	Timer A1-1 register	TA11	?? <sub>16</sub>
0303 <sub>16</sub>			?? <sub>16</sub>
0304 <sub>16</sub>	Timer A2-1 register	TA21	?? <sub>16</sub>
0305 <sub>16</sub>			?? <sub>16</sub>
0306 <sub>16</sub>	Timer A4-1 register	TA41	?? <sub>16</sub>
0307 <sub>16</sub>			?? <sub>16</sub>
0308 <sub>16</sub>	Three-phase PWM control register 0	INVC0	0000 0000 <sub>2</sub>
0309 <sub>16</sub>	Three-phase PWM control register 1	INVC1	0000 0000 <sub>2</sub>
030A <sub>16</sub>	Three-phase output buffer register 0	IDB0	XX11 1111 <sub>2</sub>
030B <sub>16</sub>	Three-phase output buffer register 1	IDB1	XX11 1111 <sub>2</sub>
030C <sub>16</sub>	Dead time timer	DTT	?? <sub>16</sub>
030D <sub>16</sub>	Timer B2 interrupt generation frequency set counter	ICTB2	XXXX ???? <sub>2</sub>
030E <sub>16</sub>			
030F <sub>16</sub>			
0310 <sub>16</sub>	Timer B3 register	TB3	?? <sub>16</sub>
0311 <sub>16</sub>			?? <sub>16</sub>
0312 <sub>16</sub>	Timer B4 register	TB4	?? <sub>16</sub>
0313 <sub>16</sub>			?? <sub>16</sub>
0314 <sub>16</sub>	Timer B5 register	TB5	?? <sub>16</sub>
0315 <sub>16</sub>			?? <sub>16</sub>
0316 <sub>16</sub>			
0317 <sub>16</sub>			
0318 <sub>16</sub>			
0319 <sub>16</sub>			
031A <sub>16</sub>			
031B <sub>16</sub>	Timer B3 mode register	TB3MR	00?? 0000 <sub>2</sub>
031C <sub>16</sub>	Timer B4 mode register	TB4MR	00?X 0000 <sub>2</sub>
031D <sub>16</sub>	Timer B5 mode register	TB5MR	00?X 0000 <sub>2</sub>
031E <sub>16</sub>			
031F <sub>16</sub>	External interrupt cause select register	IFSR	0000 0000 <sub>2</sub>

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

## SFR

Address	Register	Symbol	Value after RESET
0320 <sub>16</sub>			
0321 <sub>16</sub>			
0322 <sub>16</sub>			
0323 <sub>16</sub>			
0324 <sub>16</sub>	UART3 special mode register 4	U3SMR4	0000 0000 <sub>2</sub>
0325 <sub>16</sub>	UART3 special mode register 3	U3SMR3	0000 0000 <sub>2</sub>
0326 <sub>16</sub>	UART3 special mode register 2	U3SMR2	0000 0000 <sub>2</sub>
0327 <sub>16</sub>	UART3 special mode register	U3SMR	0000 0000 <sub>2</sub>
0328 <sub>16</sub>	UART3 transmit/receive mode register	U3MR	0000 0000 <sub>2</sub>
0329 <sub>16</sub>	UART3 baud rate register	U3BRG	?? <sub>16</sub>
032A <sub>16</sub> 032B <sub>16</sub>	UART3 transmit buffer register	U3TB	???? ????? <sub>2</sub> XXXX XXX? <sub>2</sub>
032C <sub>16</sub>	UART3 transmit/receive control register 0	U3C0	0000 1000 <sub>2</sub>
032D <sub>16</sub>	UART3 transmit/receive control register 1	U3C1	0000 0010 <sub>2</sub>
032E <sub>16</sub> 032F <sub>16</sub>	UART3 receive buffer register	U3RB	???? ????? <sub>2</sub> ???? ?XX? <sub>2</sub>
0330 <sub>16</sub>			
0331 <sub>16</sub>			
0332 <sub>16</sub>			
0333 <sub>16</sub>			
0334 <sub>16</sub>	UART2 special mode register 4	U2SMR4	0000 0000 <sub>2</sub>
0335 <sub>16</sub>	UART2 special mode register 3	U2SMR3	0000 0000 <sub>2</sub>
0336 <sub>16</sub>	UART2 special mode register 2	U2SMR2	0000 0000 <sub>2</sub>
0337 <sub>16</sub>	UART2 special mode register	U2SMR	0000 0000 <sub>2</sub>
0338 <sub>16</sub>	UART2 transmit/receive mode register	U2MR	0000 0000 <sub>2</sub>
0339 <sub>16</sub>	UART2 baud rate register	U2BRG	?? <sub>16</sub>
033A <sub>16</sub> 033B <sub>16</sub>	UART2 transmit buffer register	U2TB	???? ????? <sub>2</sub> XXXX XXX? <sub>2</sub>
033C <sub>16</sub>	UART2 transmit/receive control register 0	U2C0	0000 1000 <sub>2</sub>
033D <sub>16</sub>	UART2 transmit/receive control register 1	U2C1	0000 0010 <sub>2</sub>
033E <sub>16</sub> 033F <sub>16</sub>	UART2 receive buffer register	U2RB	???? ????? <sub>2</sub> ???? ?XX? <sub>2</sub>
0340 <sub>16</sub>	Count start flag	TABSR	0000 0000 <sub>2</sub>
0341 <sub>16</sub>	Clock prescaler reset flag	CPSRF	0XXX XXXX <sub>2</sub>
0342 <sub>16</sub>	One-shot start flag	ONSF	0000 0000 <sub>2</sub>
0343 <sub>16</sub>	Trigger select register	TRGSR	0000 0000 <sub>2</sub>
0344 <sub>16</sub>	Up-down flag	UDF	0000 0000 <sub>2</sub>
0345 <sub>16</sub>			
0346 <sub>16</sub> 0347 <sub>16</sub>	Timer A0 register	TA0	?? <sub>16</sub> ?? <sub>16</sub>
0348 <sub>16</sub> 0349 <sub>16</sub>	Timer A1 register	TA1	?? <sub>16</sub> ?? <sub>16</sub>
034A <sub>16</sub> 034B <sub>16</sub>	Timer A2 register	TA2	?? <sub>16</sub> ?? <sub>16</sub>
034C <sub>16</sub> 034D <sub>16</sub>	Timer A3 register	TA3	?? <sub>16</sub> ?? <sub>16</sub>
034E <sub>16</sub> 034F <sub>16</sub>	Timer A4 register	TA4	?? <sub>16</sub> ?? <sub>16</sub>

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

## SFR

Address	Register	Symbol	Value after RESET
0350 <sub>16</sub> 0351 <sub>16</sub>	Timer B0 register	TB0	?? <sub>16</sub> ?? <sub>16</sub>
0352 <sub>16</sub> 0353 <sub>16</sub>	Timer B1 register	TB1	?? <sub>16</sub> ?? <sub>16</sub>
0354 <sub>16</sub> 0355 <sub>16</sub>	Timer B2 register	TB2	?? <sub>16</sub> ?? <sub>16</sub>
0356 <sub>16</sub>	Timer A0 mode register	TA0MR	0000 0X00 <sub>2</sub>
0357 <sub>16</sub>	Timer A1 mode register	TA1MR	0000 0X00 <sub>2</sub>
0358 <sub>16</sub>	Timer A2 mode register	TA2MR	0000 0X00 <sub>2</sub>
0359 <sub>16</sub>	Timer A3 mode register	TA3MR	0000 0X00 <sub>2</sub>
035A <sub>16</sub>	Timer A4 mode register	TA4MR	0000 0X00 <sub>2</sub>
035B <sub>16</sub>	Timer B0 mode register	TB0MR	00?? 0000 <sub>2</sub>
035C <sub>16</sub>	Timer B1 mode register	TB1MR	00?X 0000 <sub>2</sub>
035D <sub>16</sub>	Timer B2 mode register	TB2MR	00?X 0000 <sub>2</sub>
035E <sub>16</sub>	Timer B2 special mode register	TB2SC	XXXX XXX0 <sub>2</sub>
035F <sub>16</sub>	Count source prescaler register	TCSPR	0XXX 0000 <sub>2</sub>
0360 <sub>16</sub>			
0361 <sub>16</sub>			
0362 <sub>16</sub>			
0363 <sub>16</sub>			
0364 <sub>16</sub>	UART0 special mode register 4	U0SMR4	0000 0000 <sub>2</sub>
0365 <sub>16</sub>	UART0 special mode register 3	U0SMR3	0000 0000 <sub>2</sub>
0366 <sub>16</sub>	UART0 special mode register 2	U0SMR2	0000 0000 <sub>2</sub>
0367 <sub>16</sub>	UART0 special mode register	U0SMR	0000 0000 <sub>2</sub>
0368 <sub>16</sub>	UART0 transmit/receive mode register	U0MR	0000 0000 <sub>2</sub>
0369 <sub>16</sub>	UART0 baud rate register	U0BRG	?? <sub>16</sub>
036A <sub>16</sub> 036B <sub>16</sub>	UART0 transmit buffer register	U0TB	???? ???? <sub>2</sub> XXXX XXX? <sub>2</sub>
036C <sub>16</sub>	UART0 transmit/receive control register 0	U0C0	0000 1000 <sub>2</sub>
036D <sub>16</sub>	UART0 transmit/receive control register 1	U0C1	0000 0010 <sub>2</sub>
036E <sub>16</sub> 036F <sub>16</sub>	UART0 receive buffer register	U0RB	???? ???? <sub>2</sub> ???? ?XX? <sub>2</sub>
0370 <sub>16</sub>			
0371 <sub>16</sub>			
0372 <sub>16</sub>			
0373 <sub>16</sub>			
0374 <sub>16</sub>			
0375 <sub>16</sub>			
0376 <sub>16</sub>	PLL control register 0	PLC0	0011 X100 <sub>2</sub>
0377 <sub>16</sub>	PLL control register 1	PLC1	XXXX 0000 <sub>2</sub>
0378 <sub>16</sub>	DMA0 cause select register	DM0SL	0X00 0000 <sub>2</sub>
0379 <sub>16</sub>	DMA1 cause select register	DM1SL	0X00 0000 <sub>2</sub>
037A <sub>16</sub>	DMA2 cause select register	DM2SL	0X00 0000 <sub>2</sub>
037B <sub>16</sub>	DMA3 cause select register	DM3SL	0X00 0000 <sub>2</sub>
037C <sub>16</sub> 037D <sub>16</sub>	CRC data register	CRCD	?? <sub>16</sub> ?? <sub>16</sub>
037E <sub>16</sub>	CRC input register	CRCIN	?? <sub>16</sub>
037F <sub>16</sub>			

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

## SFR

Address	Register	Symbol	Value after RESET
0380 <sub>16</sub> 0381 <sub>16</sub>	A-D0 register 0	AD00	???? ????? XXXX XX???
0382 <sub>16</sub> 0383 <sub>16</sub>	A-D0 register 1	AD01	???? ????? XXXX XX???
0384 <sub>16</sub> 0385 <sub>16</sub>	A-D0 register 2	AD02	???? ????? XXXX XX???
0386 <sub>16</sub> 0387 <sub>16</sub>	A-D0 register 3	AD03	???? ????? XXXX XX???
0388 <sub>16</sub> 0389 <sub>16</sub>	A-D0 register 4	AD04	???? ????? XXXX XX???
038A <sub>16</sub> 038B <sub>16</sub>	A-D0 register 5	AD05	???? ????? XXXX XX???
038C <sub>16</sub> 038D <sub>16</sub>	A-D0 register 6	AD06	???? ????? XXXX XX???
038E <sub>16</sub> 038F <sub>16</sub>	A-D0 register 7	AD07	???? ????? XXXX XX???
0390 <sub>16</sub>			
0391 <sub>16</sub>			
0392 <sub>16</sub>			
0393 <sub>16</sub>			
0394 <sub>16</sub> 0395 <sub>16</sub>	A-D0 control register 2	AD0CON2	X000 0000 <sub>2</sub>
0396 <sub>16</sub>	A-D0 control register 0	AD0CON0	0000 0000 <sub>2</sub>
0397 <sub>16</sub>	A-D0 control register 1	AD0CON1	0000 0000 <sub>2</sub>
0398 <sub>16</sub> 0399 <sub>16</sub>	D-A register 0	DA0	?? <sub>16</sub>
039A <sub>16</sub> 039B <sub>16</sub>	D-A register 1	DA1	?? <sub>16</sub>
039C <sub>16</sub> 039D <sub>16</sub>	D-A control register	DACON	XXXX XX00 <sub>2</sub>
039E <sub>16</sub>			
039F <sub>16</sub>			

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

## SFR

&lt;144-pin package&gt;

Address	Register	Symbol	Value after RESET
03A0 <sub>16</sub>	Function select register A8	PS8	X000 0000 <sub>2</sub>
03A1 <sub>16</sub>	Function select register A9	PS9	0000 0000 <sub>2</sub>
03A2 <sub>16</sub>			
03A3 <sub>16</sub>			
03A4 <sub>16</sub>			
03A5 <sub>16</sub>			
03A6 <sub>16</sub>			
03A7 <sub>16</sub>			
03A8 <sub>16</sub>			
03A9 <sub>16</sub>			
03AA <sub>16</sub>			
03AB <sub>16</sub>			
03AC <sub>16</sub>			
03AD <sub>16</sub>			
03AE <sub>16</sub>			
03AF <sub>16</sub>	Function select register C	PSC	00X0 0000 <sub>2</sub>
03B0 <sub>16</sub>	Function select register A0	PS0	0000 0000 <sub>2</sub>
03B1 <sub>16</sub>	Function select register A1	PS1	0000 0000 <sub>2</sub>
03B2 <sub>16</sub>	Function select register B0	PSL0	0000 0000 <sub>2</sub>
03B3 <sub>16</sub>	Function select register B1	PSL1	0000 0000 <sub>2</sub>
03B4 <sub>16</sub>	Function select register A2	PS2	00X0 0000 <sub>2</sub>
03B5 <sub>16</sub>	Function select register A3	PS3	0000 0000 <sub>2</sub>
03B6 <sub>16</sub>	Function select register B2	PSL2	00X0 0000 <sub>2</sub>
03B7 <sub>16</sub>	Function select register B3	PSL3	0000 0000 <sub>2</sub>
03B8 <sub>16</sub>			
03B9 <sub>16</sub>	Function select register A5	PS5	XXX0 0000 <sub>2</sub>
03BA <sub>16</sub>			
03BB <sub>16</sub>			
03BC <sub>16</sub>	Function select register A6	PS6	0000 0000 <sub>2</sub>
03BD <sub>16</sub>	Function select register A7	PS7	0000 0000 <sub>2</sub>
03BE <sub>16</sub>			
03BF <sub>16</sub>			
03C0 <sub>16</sub>	Port P6 register	P6	?? <sub>16</sub>
03C1 <sub>16</sub>	Port P7 register	P7	?? <sub>16</sub>
03C2 <sub>16</sub>	Port P6 direction register	PD6	0000 0000 <sub>2</sub>
03C3 <sub>16</sub>	Port P7 direction register	PD7	0000 0000 <sub>2</sub>
03C4 <sub>16</sub>	Port P8 register	P8	?? <sub>16</sub>
03C5 <sub>16</sub>	Port P9 register	P9	?? <sub>16</sub>
03C6 <sub>16</sub>	Port P8 direction register	PD8	00X0 0000 <sub>2</sub>
03C7 <sub>16</sub>	Port P9 direction register	PD9	0000 0000 <sub>2</sub>
03C8 <sub>16</sub>	Port P10 register	P10	?? <sub>16</sub>
03C9 <sub>16</sub>	Port P11 register	P11	XXX? ???? <sub>2</sub>
03CA <sub>16</sub>	Port P10 direction register	PD10	0000 0000 <sub>2</sub>
03CB <sub>16</sub>	Port P11 direction register	PD11	XXX0 0000 <sub>2</sub>
03CC <sub>16</sub>	Port P12 register	P12	?? <sub>16</sub>
03CD <sub>16</sub>	Port P13 register	P13	?? <sub>16</sub>
03CE <sub>16</sub>	Port P12 direction register	PD12	0000 0000 <sub>2</sub>
03CF <sub>16</sub>	Port P13 direction register	PD13	0000 0000 <sub>2</sub>

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

## SFR

&lt;144-pin package&gt;

Address	Register	Symbol	Value after RESET
03D0 <sub>16</sub>	Port P14 register	P14	X??? ???? <sub>2</sub>
03D1 <sub>16</sub>	Port P15 register	P15	?? <sub>16</sub>
03D2 <sub>16</sub>	Port P14 direction register	PD14	X000 0000 <sub>2</sub>
03D3 <sub>16</sub>	Port P15 direction register	PD15	0000 0000 <sub>2</sub>
03D4 <sub>16</sub>			
03D5 <sub>16</sub>			
03D6 <sub>16</sub>			
03D7 <sub>16</sub>			
03D8 <sub>16</sub>			
03D9 <sub>16</sub>			
03DA <sub>16</sub>	Pull-up control register 2	PUR2	0000 0000 <sub>2</sub>
03DB <sub>16</sub>	Pull-up control register 3	PUR3	0000 0000 <sub>2</sub>
03DC <sub>16</sub>	Pull-up control register 4	PUR4	XXXX 0000 <sub>2</sub>
03DD <sub>16</sub>			
03DE <sub>16</sub>			
03DF <sub>16</sub>			
03E0 <sub>16</sub>	Port P0 register	P0	?? <sub>16</sub>
03E1 <sub>16</sub>	Port P1 register	P1	?? <sub>16</sub>
03E2 <sub>16</sub>	Port P0 direction register	PD0	0000 0000 <sub>2</sub>
03E3 <sub>16</sub>	Port P1 direction register	PD1	0000 0000 <sub>2</sub>
03E4 <sub>16</sub>	Port P2 register	P2	?? <sub>16</sub>
03E5 <sub>16</sub>	Port P3 register	P3	?? <sub>16</sub>
03E6 <sub>16</sub>	Port P2 direction register	PD2	0000 0000 <sub>2</sub>
03E7 <sub>16</sub>	Port P3 direction register	PD3	0000 0000 <sub>2</sub>
03E8 <sub>16</sub>	Port P4 register	P4	?? <sub>16</sub>
03E9 <sub>16</sub>	Port P5 register	P5	?? <sub>16</sub>
03EA <sub>16</sub>	Port P4 direction register	PD4	0000 0000 <sub>2</sub>
03EB <sub>16</sub>	Port P5 direction register	PD5	0000 0000 <sub>2</sub>
03EC <sub>16</sub>			
03ED <sub>16</sub>			
03EE <sub>16</sub>			
03EF <sub>16</sub>			
03F0 <sub>16</sub>	Pull-up control register 0	PUR0	0000 0000 <sub>2</sub>
03F1 <sub>16</sub>	Pull-up control register 1	PUR1	XXXX 0000 <sub>2</sub>
03F2 <sub>16</sub>			
03F3 <sub>16</sub>			
03F4 <sub>16</sub>			
03F5 <sub>16</sub>			
03F6 <sub>16</sub>			
03F7 <sub>16</sub>			
03F8 <sub>16</sub>			
03F9 <sub>16</sub>			
03FA <sub>16</sub>			
03FB <sub>16</sub>			
03FC <sub>16</sub>			
03FD <sub>16</sub>			
03FE <sub>16</sub>			
03FF <sub>16</sub>	Port control register	PCR	XXXX XXX0 <sub>2</sub>

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

## SFR



&lt;100-pin package&gt;

Address	Register	Symbol	Value after RESET	
03A0 <sub>16</sub>				(Note 2)
03A1 <sub>16</sub>				
03A2 <sub>16</sub>				
03A3 <sub>16</sub>				
03A4 <sub>16</sub>				
03A5 <sub>16</sub>				
03A6 <sub>16</sub>				
03A7 <sub>16</sub>				
03A8 <sub>16</sub>				
03A9 <sub>16</sub>				
03AA <sub>16</sub>				
03AB <sub>16</sub>				
03AC <sub>16</sub>				
03AD <sub>16</sub>				
03AE <sub>16</sub>				
03AF <sub>16</sub>	Function select register C	PSC	0X00 0000 <sub>2</sub>	
03B0 <sub>16</sub>	Function select register A0	PS0	0000 0000 <sub>2</sub>	
03B1 <sub>16</sub>	Function select register A1	PS1	0000 0000 <sub>2</sub>	
03B2 <sub>16</sub>	Function select register B0	PSL0	0000 0000 <sub>2</sub>	
03B3 <sub>16</sub>	Function select register B1	PSL1	0000 0000 <sub>2</sub>	
03B4 <sub>16</sub>	Function select register A2	PS2	00X0 0000 <sub>2</sub>	
03B5 <sub>16</sub>	Function select register A3	PS3	0000 0000 <sub>2</sub>	
03B6 <sub>16</sub>	Function select register B2	PSL2	00X0 0000 <sub>2</sub>	
03B7 <sub>16</sub>	Function select register B3	PSL3	0000 0000 <sub>2</sub>	
03B8 <sub>16</sub>				
03B9 <sub>16</sub>				(Note 2)
03BA <sub>16</sub>				
03BB <sub>16</sub>				
03BC <sub>16</sub>				(Note 2)
03BD <sub>16</sub>				
03BE <sub>16</sub>				
03BF <sub>16</sub>				
03C0 <sub>16</sub>	Port P6 register	P6	?? <sub>16</sub>	
03C1 <sub>16</sub>	Port P7 register	P7	?? <sub>16</sub>	
03C2 <sub>16</sub>	Port P6 direction register	PD6	0000 0000 <sub>2</sub>	
03C3 <sub>16</sub>	Port P7 direction register	PD7	0000 0000 <sub>2</sub>	
03C4 <sub>16</sub>	Port P8 register	P8	?? <sub>16</sub>	
03C5 <sub>16</sub>	Port P9 register	P9	?? <sub>16</sub>	
03C6 <sub>16</sub>	Port P8 direction register	PD8	00X0 0000 <sub>2</sub>	
03C7 <sub>16</sub>	Port P9 direction register	PD9	0000 0000 <sub>2</sub>	
03C8 <sub>16</sub>	Port P10 register	P10	?? <sub>16</sub>	
03C9 <sub>16</sub>				(Note 2)
03CA <sub>16</sub>	Port P10 direction register	PD10	0000 0000 <sub>2</sub>	
03CB <sub>16</sub>				(Note 1)
03CC <sub>16</sub>				(Note 2)
03CD <sub>16</sub>				(Note 2)
03CE <sub>16</sub>				(Note 1)
03CF <sub>16</sub>				(Note 1)

X : Nothing is assigned ? : Indetermination

Blank columns are all reserved space. No use is allowed.

Notes :

-  Address space 03CB<sub>16</sub>, 03CE<sub>16</sub> and 03CF<sub>16</sub> should be set to "FF<sub>16</sub>" in the 100-pin package.
-  No address space 03A0<sub>16</sub>, 03A1<sub>16</sub>, 03B9<sub>16</sub>, 03BD<sub>16</sub>, 03C9<sub>16</sub>, 03CC<sub>16</sub> and 03CD<sub>16</sub> is provided in the 100-pin package.




SFR

<100-pin package>

Address	Register	Symbol	Value after RESET	
03D0 <sub>16</sub>				(Note 3)
03D1 <sub>16</sub>				
03D2 <sub>16</sub>				(Note 1)
03D3 <sub>16</sub>				
03D4 <sub>16</sub>				
03D5 <sub>16</sub>				
03D6 <sub>16</sub>				
03D7 <sub>16</sub>				
03D8 <sub>16</sub>				
03D9 <sub>16</sub>				
03DA <sub>16</sub>	Pull-up control register 2	PUR2	0000 0000 <sub>2</sub>	
03DB <sub>16</sub>	Pull-up control register 3	PUR3	0000 0000 <sub>2</sub>	
03DC <sub>16</sub>				(Note 2)
03DD <sub>16</sub>				
03DE <sub>16</sub>				
03DF <sub>16</sub>				
03E0 <sub>16</sub>	Port P0 register	P0	?? <sub>16</sub>	
03E1 <sub>16</sub>	Port P1 register	P1	?? <sub>16</sub>	
03E2 <sub>16</sub>	Port P0 direction register	PD0	0000 0000 <sub>2</sub>	
03E3 <sub>16</sub>	Port P1 direction register	PD1	0000 0000 <sub>2</sub>	
03E4 <sub>16</sub>	Port P2 register	P2	?? <sub>16</sub>	
03E5 <sub>16</sub>	Port P3 register	P3	?? <sub>16</sub>	
03E6 <sub>16</sub>	Port P2 direction register	PD2	0000 0000 <sub>2</sub>	
03E7 <sub>16</sub>	Port P3 direction register	PD3	0000 0000 <sub>2</sub>	
03E8 <sub>16</sub>	Port P4 register	P4	?? <sub>16</sub>	
03E9 <sub>16</sub>	Port P5 register	P5	?? <sub>16</sub>	
03EA <sub>16</sub>	Port P4 direction register	PD4	0000 0000 <sub>2</sub>	
03EB <sub>16</sub>	Port P5 direction register	PD5	0000 0000 <sub>2</sub>	
03EC <sub>16</sub>				
03ED <sub>16</sub>				
03EE <sub>16</sub>				
03EF <sub>16</sub>				
03F0 <sub>16</sub>	Pull-up control register 0	PUR0	0000 0000 <sub>2</sub>	
03F1 <sub>16</sub>	Pull-up control register 1	PUR1	XXXX 0000 <sub>2</sub>	
03F2 <sub>16</sub>				
03F3 <sub>16</sub>				
03F4 <sub>16</sub>				
03F5 <sub>16</sub>				
03F6 <sub>16</sub>				
03F7 <sub>16</sub>				
03F8 <sub>16</sub>				
03F9 <sub>16</sub>				
03FA <sub>16</sub>				
03FB <sub>16</sub>				
03FC <sub>16</sub>				
03FD <sub>16</sub>				
03FE <sub>16</sub>				
03FF <sub>16</sub>	Port control register	PCR	XXXX XXX0 <sub>2</sub>	

X : Nothing is assigned ? : Indetermination  
 Blank columns are all reserved space. No use is allowed.

Notes :

1.  Address space 03D2<sub>16</sub> and 03D3<sub>16</sub> should be set to "FF<sub>16</sub>" in the 100-pin package.
2.  Address space 03DC<sub>16</sub> should be set to "00<sub>16</sub>" in the 100-pin package.
3.  No address space 03D0<sub>16</sub> and 03D1<sub>16</sub> is provided in the 100-pin package.

## Processor Mode

### (1) Types of Processor Mode

Single-chip mode, memory expansion mode or microprocessor mode can be selected as a processor mode. Some pin functions, memory map and access space vary depending on the selected processor mode.

#### 1. Single-chip Mode

In single-chip mode, internal memory space (the SFR, internal RAM and internal ROM) can be accessed. All I/O ports can be used.

#### 2. Memory Expansion Mode

In memory expansion mode, both external and internal memory spaces can be accessed.

Some pins function as pins for bus control signal. The BYTE pin and register settings determine how many pins are assigned for these pin functions. (Refer to the section "Bus" for details.)

#### 3. Microprocessor Mode

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

Some pins function as pins for bus control signal. The BYTE pin and register settings determine how many pins are assigned for these pin functions. (Refer to the section "Bus" for details.)

### (2) Setting Processor Mode

The CNVss pin and the PM01 to PM00 bits in the PM0 register are combined to set a processor mode. Avoid setting "012" in the PM01 to PM00 bits.

If the PM01 to PM00 bits are rewritten, a mode corresponding to the PM01 to PM00 bits is selected regardless of a CNVss pin level.

- Avoid changing the PM01 to PM00 bits when the PM02 to PM07 bits in the PM0 register are rewritten.
- Avoid shifting to microprocessor mode while the CPU is executing a program in the internal ROM.
- Avoid shifting to single-chip mode while the CPU is executing a program in an external memory space.

Figures 1.6.1 and 1.6.2 show the PM0 register and PM1 register. Figure 1.6.3 shows a memory map in each processor mode.

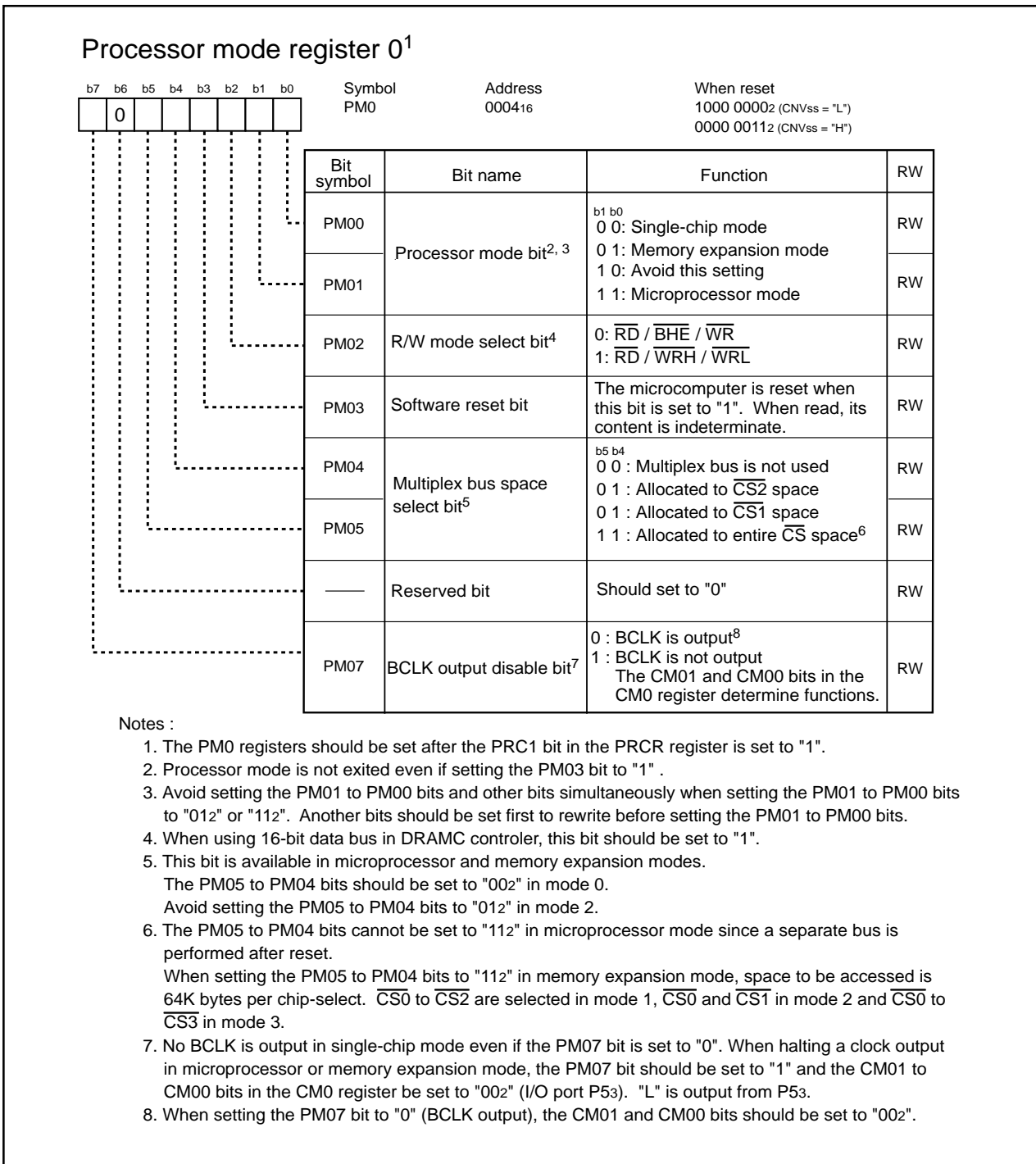
#### 1. Applying Vss to CNVss Pin

The microcomputer enters single-chip mode after reset. The PM01 to PM00 bits should be set to "012" (memory expansion mode) to switch to memory expansion mode after an operation start.

#### 2. Applying Vcc to CNVss Pin

The microcomputer enters microprocessor mode after reset.

**Processor Mode**



**Figure 1.6.1. PM0 Register**

Processor Mode

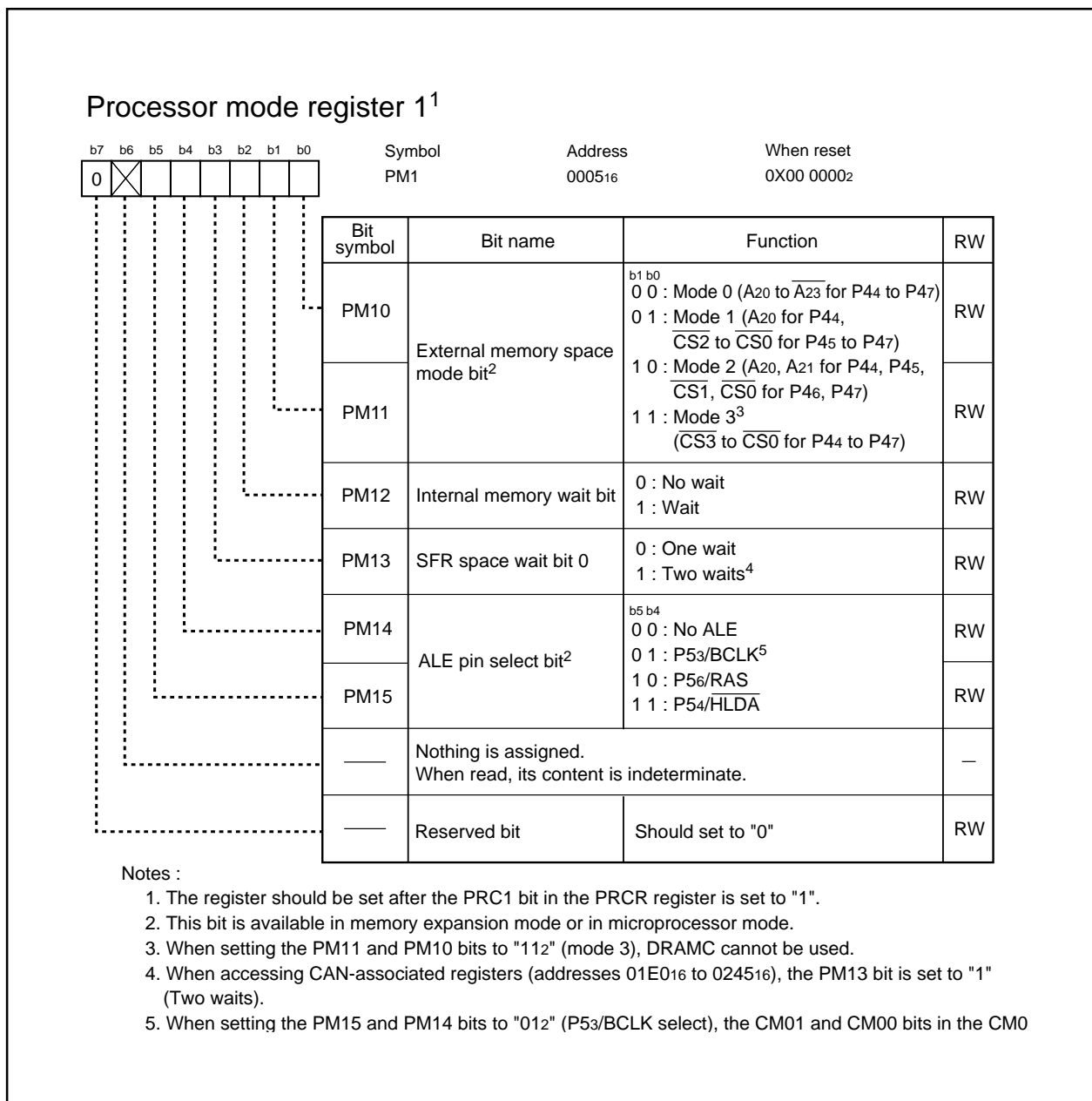


Figure 1.6.2. PM1 Register

Processor Mode

Address	Single-chip mode			Memory expansion mode			Microprocessor mode				
	SFR Internal RAM	SFR Internal RAM Reserved space	Reserved space	Mode 0	Mode 1	Mode 2	Mode 3	Mode 0	Mode 1	Mode 2	Mode 3
00000016											
00040016											
00080016											
10000016				External space 0	CS1 2M bytes <sup>1</sup> External space 0	CS1 4M bytes <sup>2</sup> External space 0	CS1, 1M byte External space 0	External space 0	CS1 2M bytes <sup>1</sup> External space 0	CS1 4M bytes <sup>2</sup> External space 0	CS1, 1M byte External space 0
20000016				External space 1	CS2 2M bytes External space 1		CS2, 1M byte External space 1	External space 1	CS2 2M bytes External space 1		CS2, 1M byte External space 1
40000016				DRAM connectable space 0, 0.5 to 8M bytes (Available as external space when DRAM is not used) (External space 2)	DRAM connectable space 0, 0.5 to 8M bytes (When open space is under 8M bytes, cannot use the rest of this space) (External space 2)	DRAM connectable space 0, 0.5 to 8M bytes (When open space is under 8M bytes, cannot use the rest of this space) (External space 2)	(Cannot use as DRAM space or external space)	DRAM connectable space 0, 0.5 to 8M bytes (Available as external space when DRAM is not used) (External space 2)	DRAM connectable space 0, 0.5 to 8M bytes (When open space is under 8M bytes, cannot use the rest of this space) (External space 2)	DRAM connectable space 0, 0.5 to 8M bytes (When open space is under 8M bytes, cannot use the rest of this space) (External space 2)	No use (Cannot use as DRAM space or external space)
C0000016				External space 3	CS0 2M bytes External space 3	CS0 3M bytes External space 3	CS3, 1M byte External space 2	External space 3	No use	CS0 4M bytes External space 3	CS3, 1M byte External space 2
E0000016				Reserved space	No use		No use				No use
F0000016				Internal ROM	Reserved space	Reserved space	CS0, 1M byte External space 3		CS0 2M bytes External space 3		
FFFFFF16				Internal ROM	Internal ROM	Internal ROM	Internal ROM	Internal ROM			CS0, 1M byte External space 3

Notes :  
1. 20000016-00800016=2016K bytes. 32K bytes less than 2M bytes.  
2. 40000016-00800016=4064K bytes. 32K bytes less than 4M bytes.

Each CS0 to CS3 can be set for 0 to 3 wait(s) in the WCR register.

Figure 1.6.3. Memory Map in Each Processor Mode

## Bus

In memory expansion mode or microprocessor mode, some pins function as bus control pins to input and output data from and to external devices. A0 to A22,  $\overline{A}23$ , D0 to D15, MA0 to MA12, CS0 to CS3,  $\overline{WR}/\overline{WR}/\overline{CASL}$ ,  $\overline{WRH}/\overline{BHE}/\overline{CASH}$ ,  $\overline{RD}/\overline{DW}$ , BCLK/ALE,  $\overline{HLDA}/\overline{ALE}$ ,  $\overline{HOLD}$ , ALE/RAS,  $\overline{RDY}$  are included as bus control pins.

### Bus Settings

The BYTE pin, the DS register, the PM05 to PM04 bits in the PM0 register and the PM11 to PM10 bits in the PM1 register determine bus settings.

Table 1.7.1 lists how to changing bus setting. Figure 1.7.1 shows the DS register.

Table 1.7.1. Bus Settings

Bus setting	Changed by
Selecting external address bus width	DS register
Selecting bus width after reset	BYTE pin (external space 3 only)
Selecting between separate bus or multiplex bus	PM05 to PM04 bits in PM0 register
Number of chip-select	PM11 to PM10 bits in PM1 register

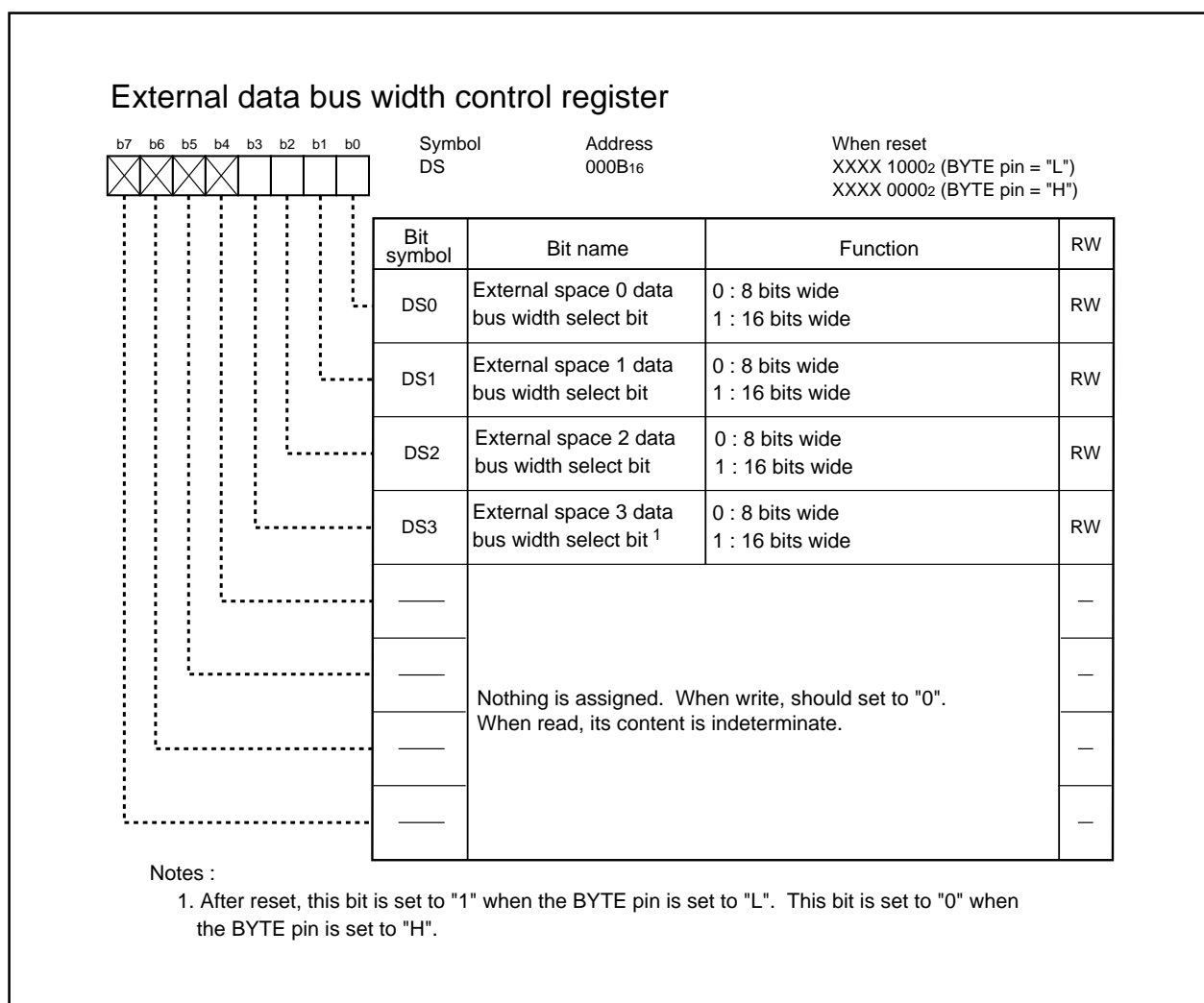


Figure 1.7.1. DS Register

### (1) Selecting External Address Bus

The number of address bus for external output, the number of chip-selects and chip-select space vary depending on each external space mode. The PM11 to PM10 bits in the PM1 register determine an external space mode.

With DRAMC, row addresses and column addresses are multiplexed to output in a DRAM space.

### (2) Selecting External Data Bus

8 bits or 16 bits can be selected for external data bus in the DS register per external space. Data bus in the external space 3, after reset, becomes 16 bits wide when an input to the BYTE pin is set to "L" and 8 bits wide when it is set to "H". Avoid changing the BYTE pin input level during operation. Internal bus is always 16 bits wide.

### (3) Selecting Separate/Multiplex Bus

The PM05 to PM04 bits in the PM0 register determine either a separate or multiplex bus as bus format .

#### • Separate Bus

Data and address are separated for input and output. The DS register determines an external data bus width, 8-bit data bus or 16-bit, per external space. When all DS<sub>i</sub> bits in the DS register (i=0 to 3) are set to "0" (8-bit data bus), port P<sub>0</sub> becomes a data bus and port P<sub>1</sub> becomes a programmable I/O port. When setting one of the DS<sub>i</sub> bits to "1" (16-bit data bus), ports P<sub>0</sub> and P<sub>1</sub> become the data bus. When setting the DS<sub>i</sub> bits to "0", port P<sub>1</sub> is indeterminate.

With a separate bus, the WCR register determines a software wait status.

#### • Multiplex Bus

Data and address are timeshared for input and output. D<sub>0</sub> to D<sub>7</sub> are multiplexed with A<sub>0</sub> to A<sub>7</sub> in 8-bit space selected by the DS<sub>i</sub> bit. D<sub>0</sub> to D<sub>15</sub> are multiplexed with A<sub>0</sub> to A<sub>15</sub> in 16-bit space selected by the DS<sub>i</sub> bit. In the multiplex bus space, the WCR register selects two waits or three waits. Two-wait access is automatically selected even if either no wait, one wait or two waits is selected. Refer to the paragraph "(4) Bus Timing" for details.

In memory expansion mode, when the PM05 to PM04 bits in the PM register are set to "112" (allocated to entire CS space), only 16 bits from A<sub>0</sub> to A<sub>15</sub> are output as an address.

The PM05 to PM04 bits cannot set to "112" in microprocessor mode. See Table 1.7.2 for details.

Table 1.7.2. Each Processor Mode and Port Function

Processor mode	Single-chip mode	Memory expansion mode/microprocessor mode				Memory expansion mode	
		"012", "102" ( $\overline{CS1}$ or $\overline{CS2}$ as multiplex bus.) Another as separate bus)		"002" (Separate bus)		"112" <sup>1</sup> (All space multiplex bus)	
Data bus width of space to be accessed		All 8-bit external space	Some 16-bit external space	All 8-bit external space	Some 16-bit external space	All 8-bit external space	Some 16-bit external space
P00 to P07	I/O port	Data bus D <sub>0</sub> to D <sub>7</sub>	Data bus D <sub>0</sub> to D <sub>7</sub>	Data bus D <sub>0</sub> to D <sub>7</sub>	Data bus D <sub>0</sub> to D <sub>7</sub>	I/O port	I/O port
P10 to P17	I/O port	I/O port	Data bus D <sub>8</sub> to D <sub>15</sub>	I/O port	Data bus D <sub>8</sub> to D <sub>15</sub>	I/O port	I/O port
P20 to P27	I/O port	Address bus data bus <sup>2</sup> A <sub>0</sub> /D <sub>0</sub> to A <sub>7</sub> /D <sub>7</sub>	Address bus data bus <sup>2</sup> A <sub>0</sub> /D <sub>0</sub> to A <sub>7</sub> /D <sub>7</sub>	Address bus A <sub>0</sub> /D <sub>0</sub> to A <sub>7</sub> /D <sub>7</sub>	Address bus A <sub>0</sub> /D <sub>0</sub> to A <sub>7</sub> /D <sub>7</sub>	Address bus data bus A <sub>0</sub> /D <sub>0</sub> to A <sub>7</sub> /D <sub>7</sub>	Address bus data bus A <sub>0</sub> /D <sub>0</sub> to A <sub>7</sub> /D <sub>7</sub>
P30 to P37	I/O port	Address bus A <sub>8</sub> to A <sub>15</sub>	Address bus/data bus <sup>2</sup> A <sub>8</sub> /D <sub>8</sub> to A <sub>15</sub> /D <sub>15</sub>	Address bus A <sub>8</sub> to A <sub>15</sub>	Address bus A <sub>8</sub> to A <sub>15</sub>	Address bus A <sub>8</sub> to A <sub>15</sub>	Address bus/data bus A <sub>8</sub> /D <sub>8</sub> to A <sub>15</sub> /D <sub>15</sub>
P40 to P43	I/O port	Address bus A <sub>16</sub> to A <sub>19</sub>	Address bus A <sub>16</sub> to A <sub>19</sub>	Address bus A <sub>16</sub> to A <sub>19</sub>	Address bus A <sub>16</sub> to A <sub>19</sub>	I/O port	I/O port
P44 to P46	I/O port	$\overline{CS}$ (chip-select) or address bus (A <sub>23</sub> ) (Refer to the paragraph "Bus control" for details) <sup>5</sup>					
P47	I/O port	$\overline{CS}$ (chip-select) or address bus (A <sub>23</sub> ) (Refer to the paragraph "Bus control" for details) <sup>5</sup>					
P50 to P53	I/O port	$\overline{RD}$ , $\overline{WRL}$ , $\overline{WRH}$ and $\overline{BCLK}$ output or $\overline{RD}$ , $\overline{BHE}$ , $\overline{WR}$ and $\overline{BCLK}$ output (Refer to the paragraph "Bus control" for details) <sup>3,4</sup>					
P54	I/O port	$\overline{HDLA}$ <sup>3</sup>	$\overline{HDLA}$ <sup>3</sup>	$\overline{HDLA}$ <sup>3</sup>	$\overline{HDLA}$ <sup>3</sup>	$\overline{HDLA}$ <sup>3</sup>	$\overline{HDLA}$ <sup>3</sup>
P55	I/O port	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$
P56	I/O port	$\overline{RAS}$ <sup>3</sup>	$\overline{RAS}$ <sup>3</sup>	$\overline{RAS}$ <sup>3</sup>	$\overline{RAS}$ <sup>3</sup>	$\overline{RAS}$ <sup>3</sup>	$\overline{RAS}$ <sup>3</sup>
P57	I/O port	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$

## Notes :

- Avoid setting the PM05 to PM04 bits to "112" (all  $\overline{CS}$  space as multiplex bus) in microprocessor mode because running a separate bus after reset.  
When selecting "112" in memory expansion mode, address bus accesses with 64K bytes per chip-select.
- Address bus is selected in separate bus configuration.
- The ALE output pin should be selected by the PM15 to PM14 bits in the PM1 register. Either " $\overline{WRL}$ ,  $\overline{WRH}$ " or " $\overline{BHE}$ ,  $\overline{WR}$ " should be selected by the PM02 bit in the PM0 register.
- When selecting the DRAMC and accessing the DRAM space,  $\overline{CASL}$ ,  $\overline{CASH}$ ,  $\overline{DW}$  and  $\overline{BCLK}$  output occurs.
- The  $\overline{CS}$  signal and address bus should be determined by the PM11 to PM10 bits in the PM1 register.

## Bus Control

Signals, which are required to access to external devices, and software wait are provided as follows. The signals are available in memory expansion mode and microprocessor mode only.

### (1) Address Bus and Data Bus

Address bus is assigned to access a 16M-byte address space with 24 control pins as A0 to A22 and  $\overline{A23}$ .  $\overline{A23}$  is an inversed output of the highest-order address bit.

Data bus is a signal to input and output data. The DS register selects 8-bit data bus as D0 to D7 or 16-bit data bus as D0 to D15 for each external space. When setting the BYTE pin to "H", data bus in the external memory space 3 is set as an 8-bit data bus after reset. When setting the BYTE pin to "L", data bus in the external memory space 3 is set as a 16-bit data bus.

When changing single-chip mode to memory expansion mode, address bus is indeterminate till accessing an external memory space.

When accessing a DRAM space with DRAMC, row addresses and column addresses are multiplexed into A8 to A20.

### (2) Chip-Select Signal

The chip-select signal is shared with A0 to A22 and  $\overline{A23}$ . The PM11 to PM10 bits in the PM1 register determine chip-select space and the number of chip-select outputs. Four chip-select signals maximum can be output.

In microprocessor mode, the chip-select signal is not output after reset except  $\overline{A23}$  as chip-select signal. "L" is output while CS<sub>i</sub> (i=0 to 3) accesses the corresponding external space. "H" is output when CS<sub>i</sub> accesses internal space and another external memory space. Figure 1.7.2 shows an example of the address bus and chip-select signal outputs.

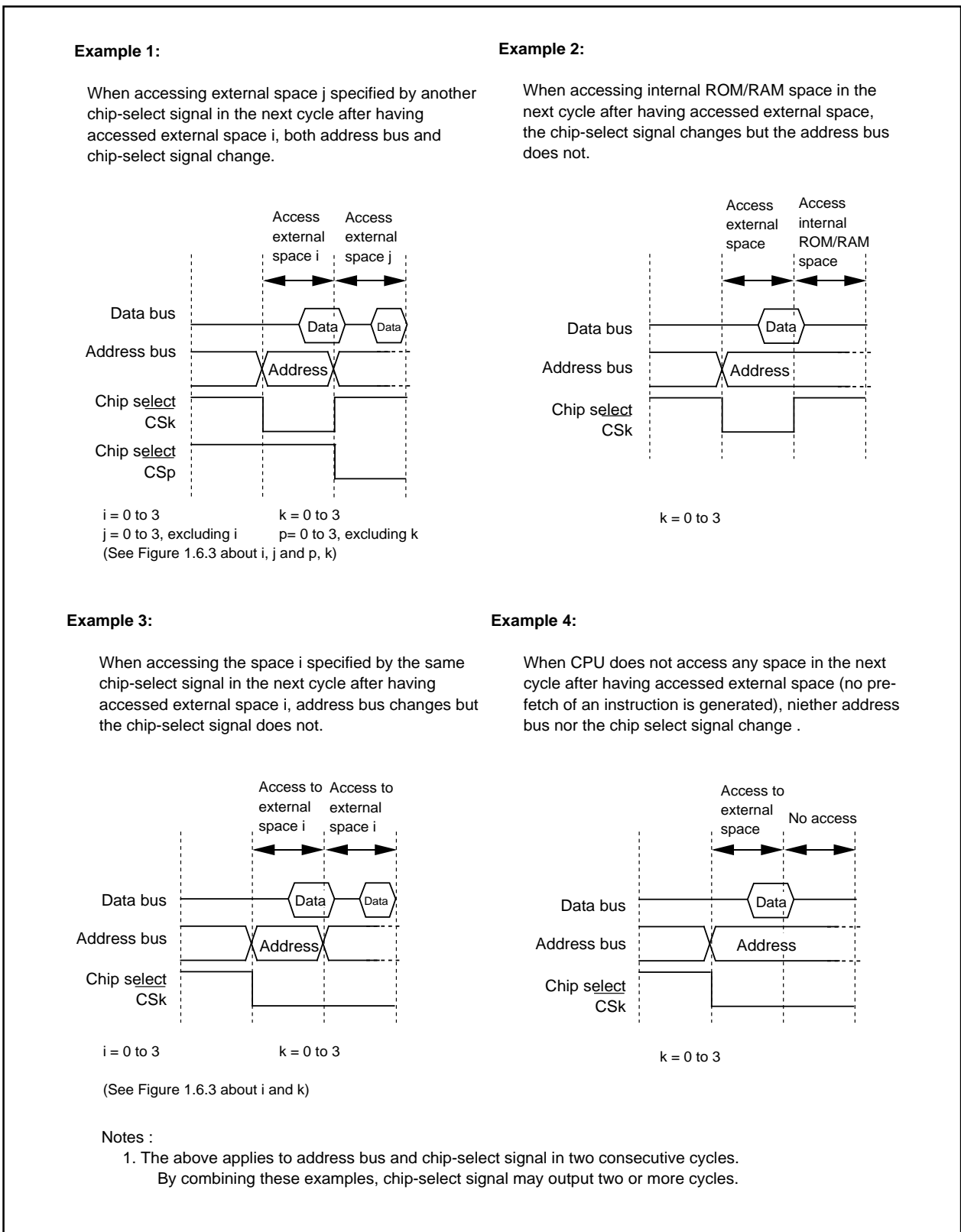


Figure 1.7.2. Address Bus and Chip-Select Signal Outputs (Separate bus)

### (3) Read and Write Signals

With a 16-bit data bus, the PM02 bit in the PM0 register determines the read and write signals, a combination of the  $\overline{RD}$ ,  $\overline{WR}$  and  $\overline{BHE}$  signals or the  $\overline{RD}$ ,  $\overline{WRL}$  and  $\overline{WRH}$  signals. When the DS3 to DS0 bits in the DS register is set to "0" (all 8-bit data bus in external memory space), the PM02 bit should be set to "0" ( $\overline{RD}$ ,  $\overline{WR}$  and  $\overline{BHE}$ ). When setting some of the DS3 to DS0 bits to "1" (16-bit data bus) to access an 8-bit space, a combination of  $\overline{RD}$ ,  $\overline{WR}$  and  $\overline{BHE}$  is automatically selected regardless of the PM02 bit. Tables 1.7.3 and 1.7.4 list each signal operations.

$\overline{RD}$ ,  $\overline{WR}$  and  $\overline{BHE}$  are combined for read and write signals after reset.

When changing a combination of  $\overline{RD}$ ,  $\overline{WRL}$  and  $\overline{WRH}$ , the PM02 bit should be set first to write to an external memory.

When accessing the DRAM with a 16-bit bus, the PM02 bit should be set to "1" ( $\overline{RD}$ ,  $\overline{WRL}$  and  $\overline{WRH}$ ).

**Table 1.7.3.  $\overline{RD}$ ,  $\overline{WRL}$  and  $\overline{WRH}$  Signals**

Data bus	$\overline{RD}$	$\overline{WRL}$	$\overline{WRH}$	Status of external data bus
16 bits	L	H	H	Read data
	H	L	H	Write 1-byte data to even address
	H	H	L	Write 1-byte data to odd address
	H	L	L	Write data to both even and odd addresses
8 bits	H	L <sup>1</sup>	Not used	Write 1-byte data
	L	H <sup>1</sup>	Not used	Read 1-byte data

Notes :

1. The  $\overline{WR}$  signal is set.

**Table 1.7.4.  $\overline{RD}$ ,  $\overline{WR}$  and  $\overline{BHE}$  Signals**

Data bus	$\overline{RD}$	$\overline{WR}$	$\overline{BHE}$	A0	Status of external data bus
16 bits	H	L	L	H	Write 1-byte data to odd address
	L	H	L	H	Read 1-byte data from odd address
	H	L	H	L	Write 1-byte data to even address
	L	H	H	L	Read 1-byte 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 bits	H	L	Not used	H / L	Write 1-byte data
	L	H	Not used	H / L	Read 1-byte data

#### (4) Bus Timing

Bus cycle for the internal ROM and internal RAM are basically one BCLK cycle. When the PM12 bit in the PM1 register is set to "1" (wait), the bus cycles are two BCLK cycles.

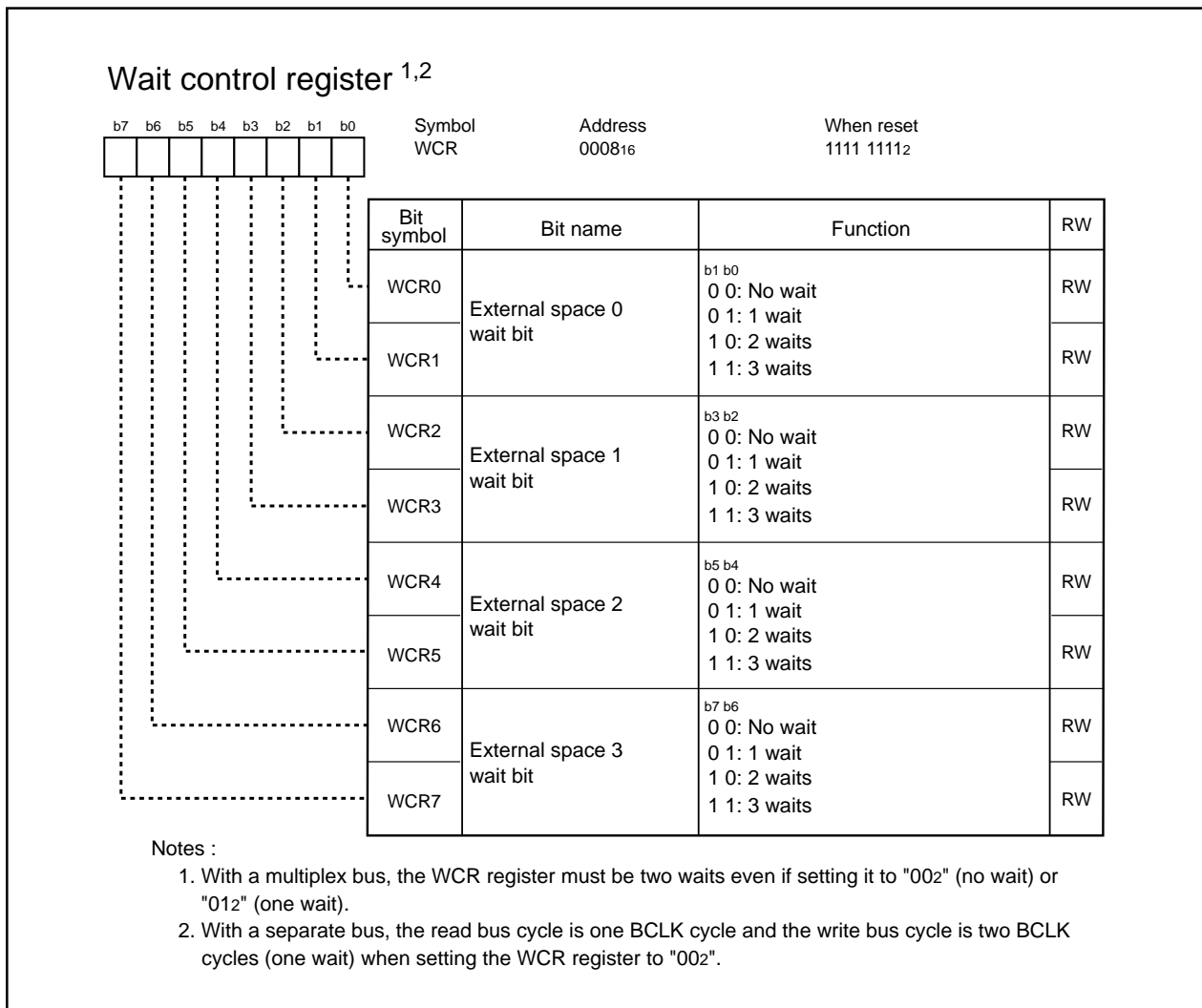
Bus cycles for the SFR is basically two BCLK cycles. When the PM13 bit in the PM1 register is set to "1" (2 waits), the bus cycles are three BCLK cycles. When accessing CAN-associated registers (addresses 01E0<sub>16</sub> to 0245<sub>16</sub>), the PM13 bit should be set to "1".

Bus cycle for an external space is basically one BCLK cycle. The WCR register inserts wait(s), equivalent to one to three BCLK cycles, into an external space. Bus cycles are two BCLK cycles if selecting one wait. Bus cycles are four BCLK cycles if selecting three waits.

If applicable to the followings, bus cycles vary from the ones selected by the WCR register. Figure 1.7.5 shows each bit status and bus cycle.

- Write cycle with a separate bus and no wait.
- Read cycle and write cycle with a multiplex bus and no wait.
- Read cycle and write cycle with a multiplex bus and one wait.

Figure 1.7.3 shows the WCR register. Figures 1.7.4 and 1.7.5 show bus timing in an external space.



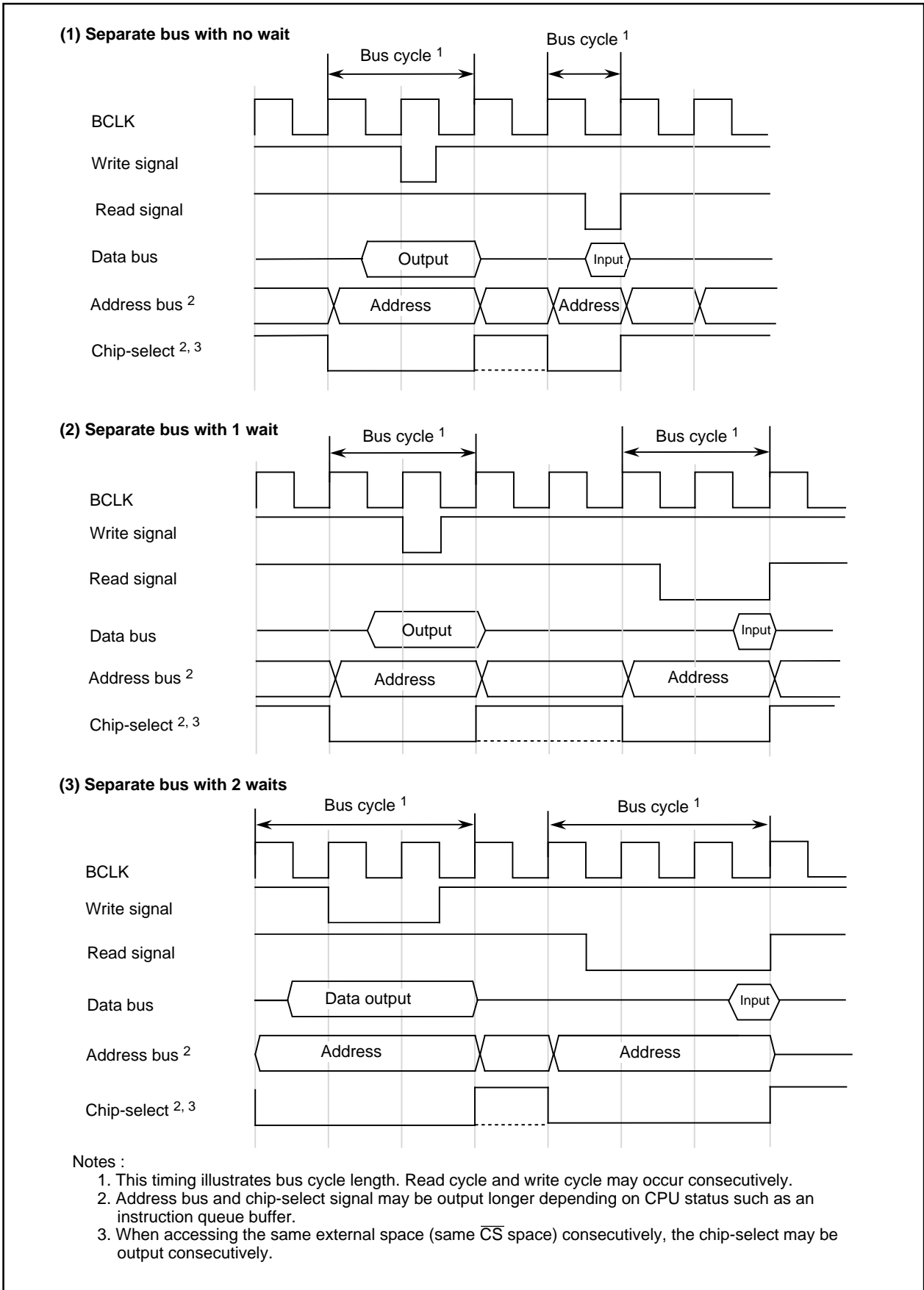
**Figure 1.7.3. WCR Register**

**Table 1.7.5. Software Wait and Bus Cycle**

Space	External Bus status	PM1 register		WCR register	Bus cycle
		PM13 bit	PM12 bit	WCRj to WCRi bits	
SFR	_____	0	_____	_____	2 BCLK cycles
		1			3 BCLK cycles
Internal ROM/RAM	_____	_____	0	_____	1 BCLK cycle
			1		2 BCLK cycles
External memory	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

$i = 0, 2, 4, 6$   $j = i + 1$

**Bus**



**Figure 1.7.4. External Bus Operation with Software Wait (1)**

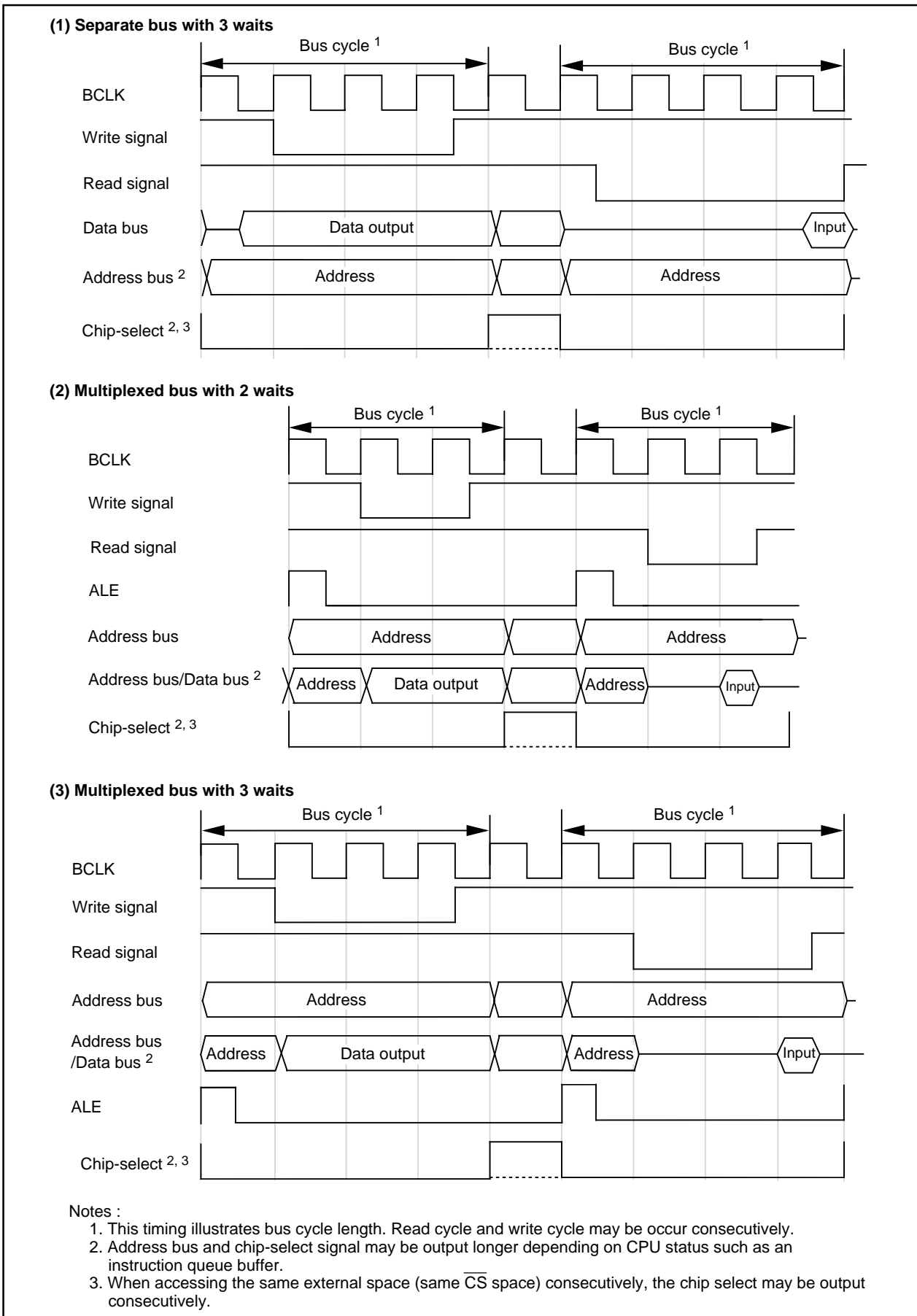


Figure 1.7.5. External Bus Operation with Software Wait (2)

### (5) ALE Signal

The ALE signal latches an address of multiplex bus. An address should be latched at a falling edge of ALE. The PM15 to PM14 bits in the PM1 register determine an output pin for the ALE signal. ALE signal outputs to an internal space and external space.

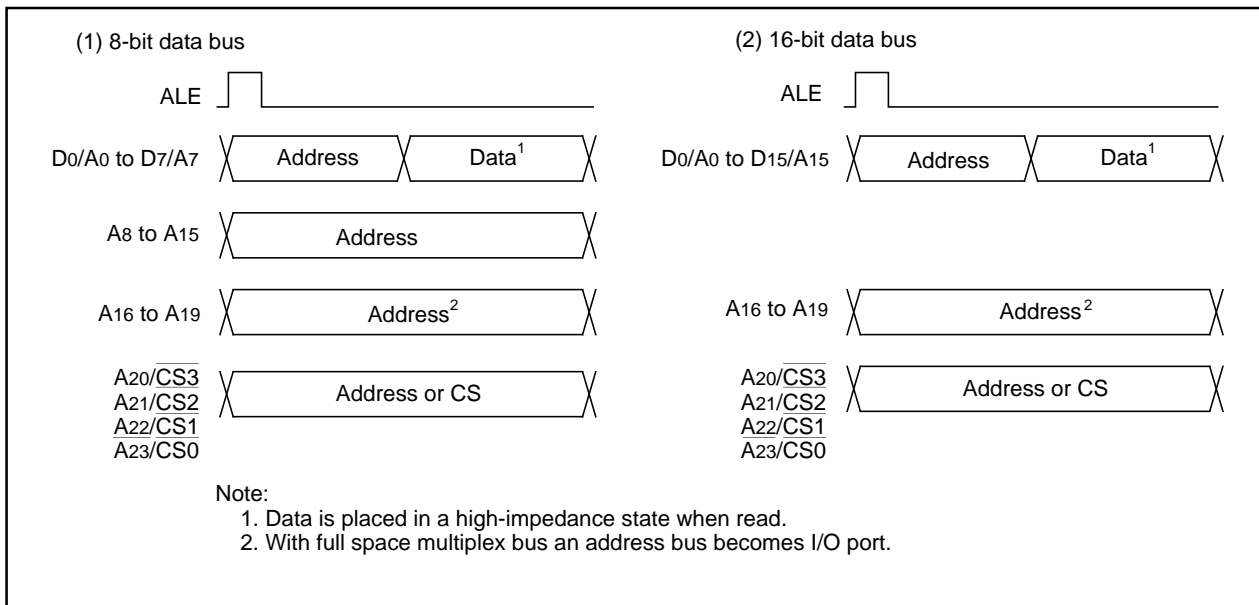


Figure 1.7.6. ALE Signal and Address/data Bus

### (6) RDY Signal

The RDY signal facilitates an access to external devices which need a longer access time. When setting the RDY pin to "L" on the falling edge of last BCLK of bus cycle, a wait is inserted into the bus cycle. Then when setting the RDY pin to "H" on the falling edge of the BCLK, the reset of bus cycle is resumed. Table 1.7.6 lists a microcomputer state when the RDY signal inserts a wait into the bus cycle. Figure 1.7.7 shows an example of the RD signal that is output longer by the RDY signal.

Table 1.7.6. Microcomputer Status in a Wait State<sup>1</sup>

Item	State
Oscillation	On
RD signal, WR signal, address bus, data bus, CS ALE signal, HLDA, programmable I/O ports	Maintains status when RDY signal is received
Internal peripheral circuits	On

Note:

1. The RDY signal cannot be received immediately before a software wait.

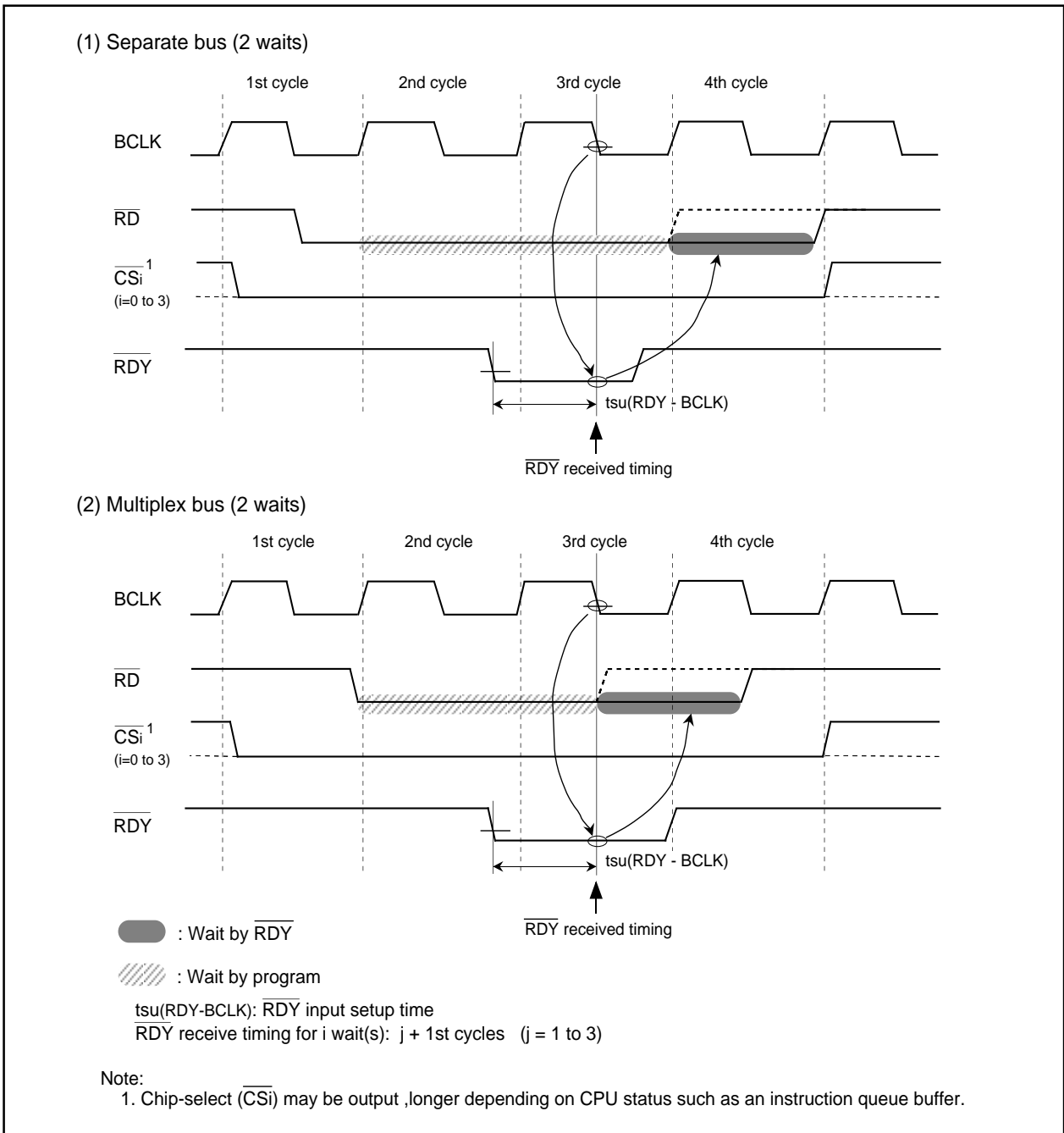


Figure 1.7.7.  $\overline{RD}$  Signal Output Longer by  $\overline{RDY}$  Signal

### (7) $\overline{HOLD}$ Signal

The  $\overline{HOLD}$  signal transfers a bus privileges from the CPU to external circuits. When setting the  $\overline{HOLD}$  pin to "L", the microcomputer becomes in a hold state after a bus access at the time is completed. The microcomputer remains in a hold state while the  $\overline{HOLD}$  pin is set to "L". The  $\overline{HLDA}$  pin outputs "L". Table 1.7.7 shows a microcomputer state in a hold state.

Bus is used in the following priority:  $\overline{HOLD}$ , DMAC, CPU.

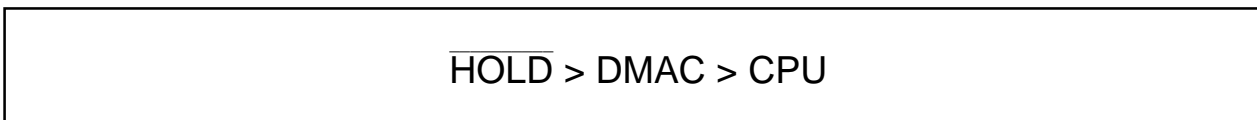


Figure 1.7.8. Priority to Use Bus

**Table 1.7.7. Microcomputer Status in a Hold State**

Item	Status
Oscillation	On
$\overline{RD}$ , $\overline{WR}$ , $\overline{WRL}$ , $\overline{WRH}$ , address bus, data bus, $\overline{CS}$ , $\overline{BHE}$	High-impedance
Programmable I/O ports: P0 to P15	Maintains status when HOLD signal is received
$\overline{HLDA}$	Output "L"
Internal peripheral circuits	On (except the watchdog timer stops)
ALE signal	Output "L"

**(8) External Bus Status when Accessing Internal Space**

Table 1.7.8 shows external bus status when accessing internal space.

**Table 1.7.8. External Bus State when Accessing Internal Space**

Item	State when accessing SFR, internal ROM and internal RAM	
Address bus	Maintains address of external space accessed immediately before	
Data bus	When read	High-impedance
	When write	High-impedance
$\overline{RD}$ , $\overline{WR}$ , $\overline{WRL}$ , $\overline{WRH}$	Output "H"	
$\overline{BHE}$	Maintains in an external space state accessed immediately before	
$\overline{CS}$	Output "H"	
ALE	Output ALE	

**(9) BCLK Output**

The CPU clock is a clock to operate the CPU. When combining the PM07 bit in the PM0 register set to "0" (BCLK output) and the CM01 to CM00 bits in the CM0 register set to "002", the CPU clock signal is output from P53 as BCLK.

No BCLK is output in single-chip mode. Refer to the section "System Clock" for details.

**(10) DRAM Control Signals ( $\overline{RAS}$ ,  $\overline{CASL}$ ,  $\overline{CASH}$  and  $\overline{DW}$ )**

The DRAM control signals control DRAM. The DRAM control signals are output when the AR0 to AR2 bits in the DRAMCONT register determines a DRAM space. Table 1.7.9 lists each signal operation.

**Table 1.7.9.  $\overline{RAS}$ ,  $\overline{CASL}$ ,  $\overline{CASH}$  and  $\overline{DW}$  Signals**

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

## System Clock

## System Clock

### Clock Generation Circuit

Four circuits are incorporated to generate the system clock signal :

- Main clock oscillation circuit
- Sub clock oscillation circuit
- Ring oscillator
- PLL frequency synthesizer

Table 1.8.1 lists specifications of the clock generation circuit. Figure 1.8.1 shows a block diagram of the clock generation circuit. Figures 1.8.2 to 1.8.8 show registers to control the clock.

**Table 1.8.1. Clock Generation Circuit Specifications**

Item	Main clock generation circuit	Sub clock generation circuit	Ring oscillator	PLL frequency synthesizer
Application	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Peripheral function clock source</li> </ul>	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Timer A and B clock source</li> </ul>	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Peripheral function clock source</li> </ul>	<ul style="list-style-type: none"> <li>• CPU clock source</li> <li>• Peripheral function clock source</li> </ul>
Clock frequency	Up to 30 MHz	32.768 kHz	Approximately 1 MHz	20 MHz, 30 MHz
Connectable or additional circuit oscillator	<ul style="list-style-type: none"> <li>• Ceramic resonator</li> <li>• Crystal oscillator</li> </ul>	<ul style="list-style-type: none"> <li>• Crystal oscillator</li> </ul>	_____	<ul style="list-style-type: none"> <li>• Low pass filter</li> </ul>
Pins for oscillator or for additional circuit	XIN, XOUT	XCIN, XCOUT	_____	VCOUT (connect to Low pass filter) P86 (connect to Vss)
Oscillation stop/restart function	Available	Available	Available	Available
Oscillator status after reset	Oscillating	Stopped	Stopped	Stopped
Other	Externally generated clock can be input	Externally generated clock can be input. With a sub clock oscillation circuit, PLL frequency synthesizer cannot be used.	When main clock oscillation stops, a ring oscillator starts oscillating automatically and becomes a clock source for CPU and peripheral functions.	With PLL frequency synthesizer, a sub clock cannot be used.

System Clock

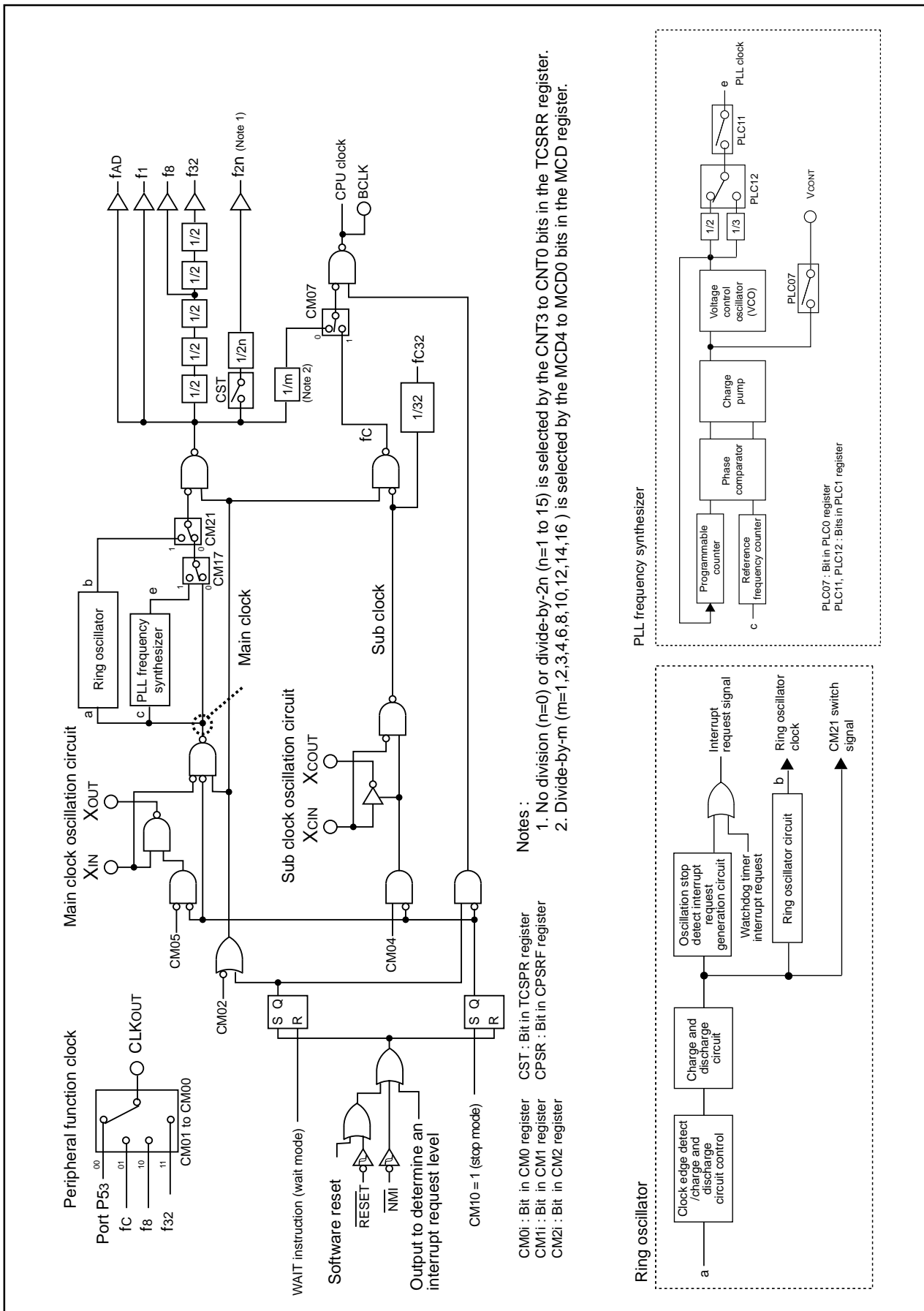


Figure 1.8.1. Clock Generation Circuit

System Clock

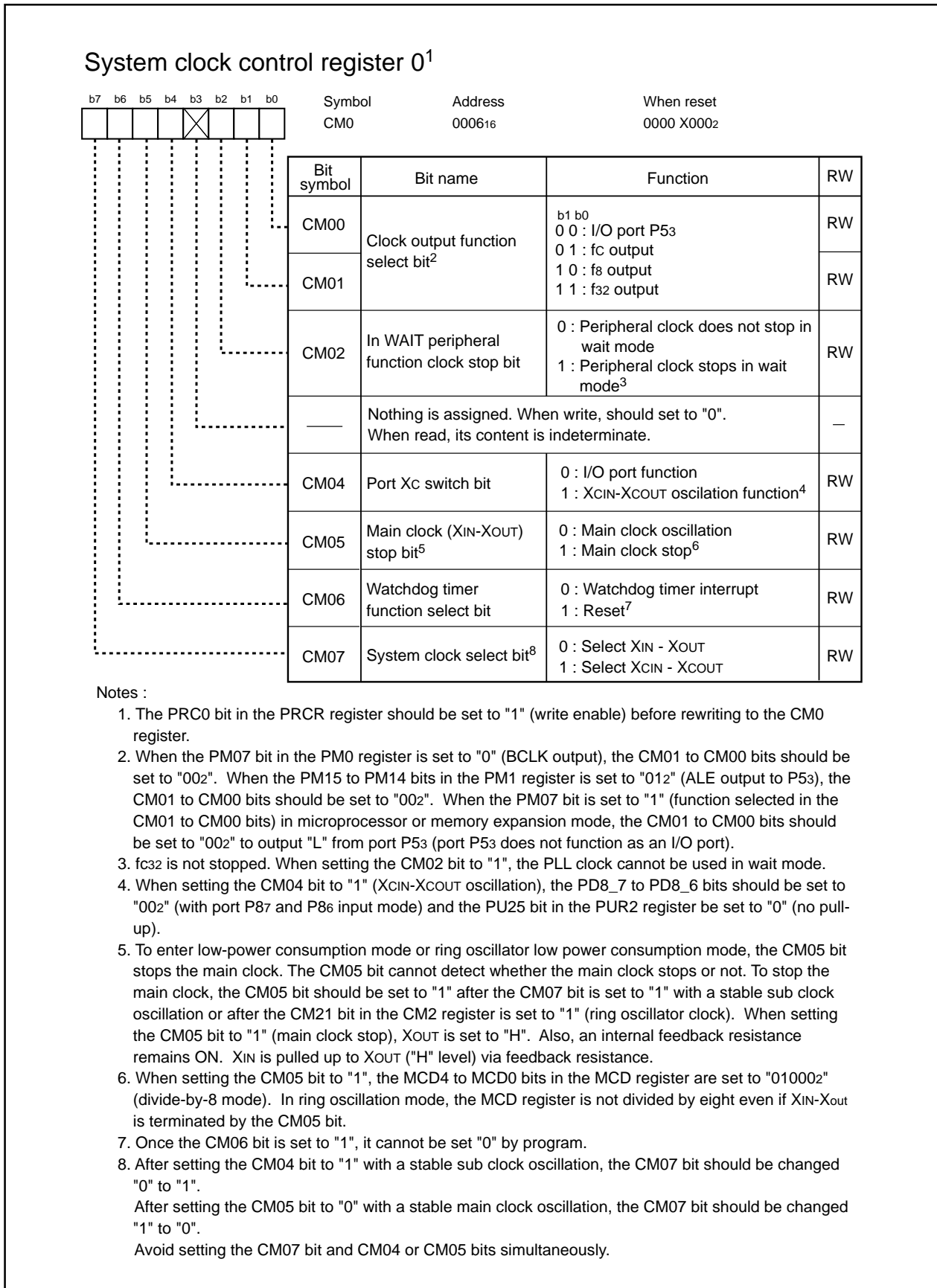


Figure 1.8.2. CM0 register

System Clock

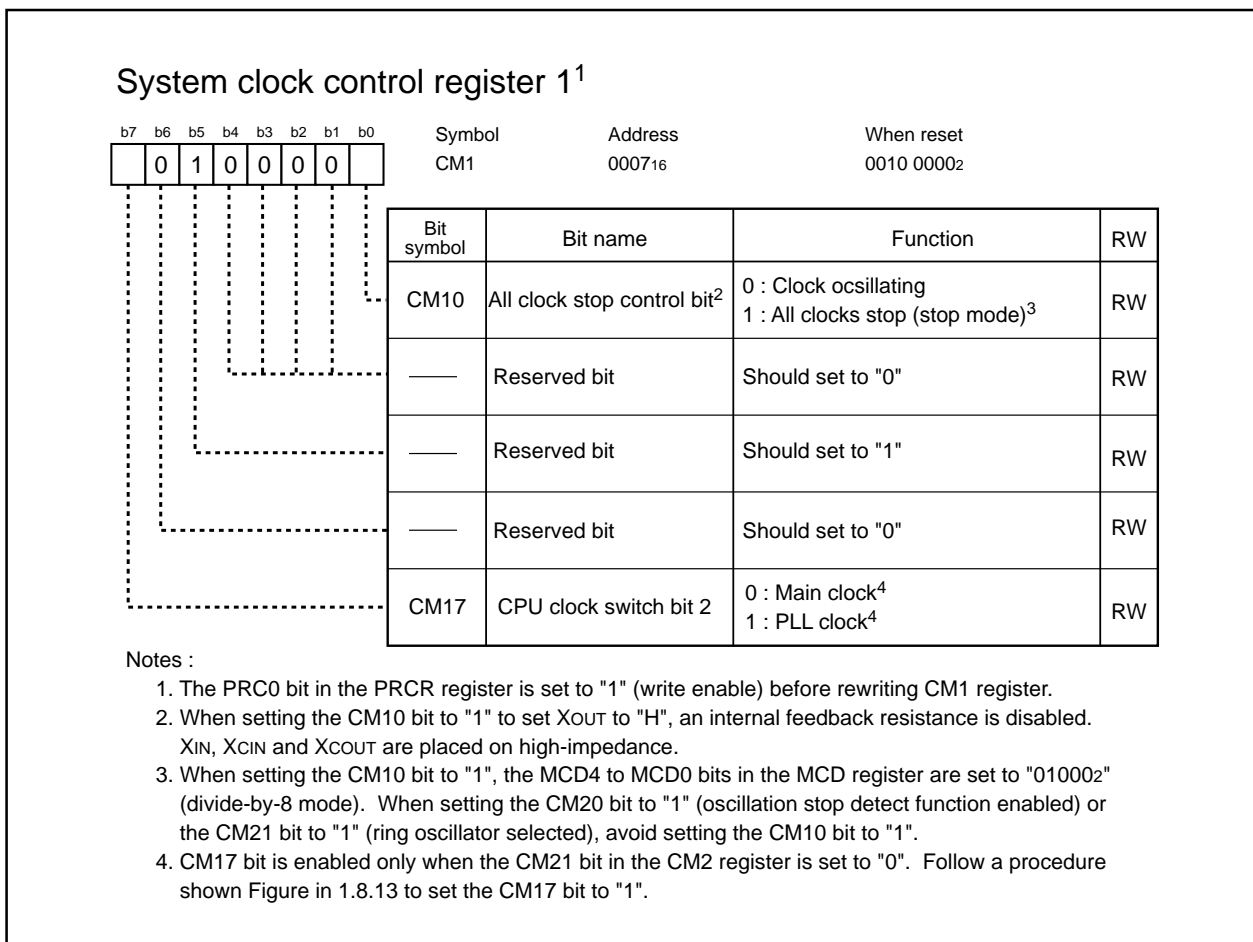


Figure 1.8.3. CM1 Register

System Clock

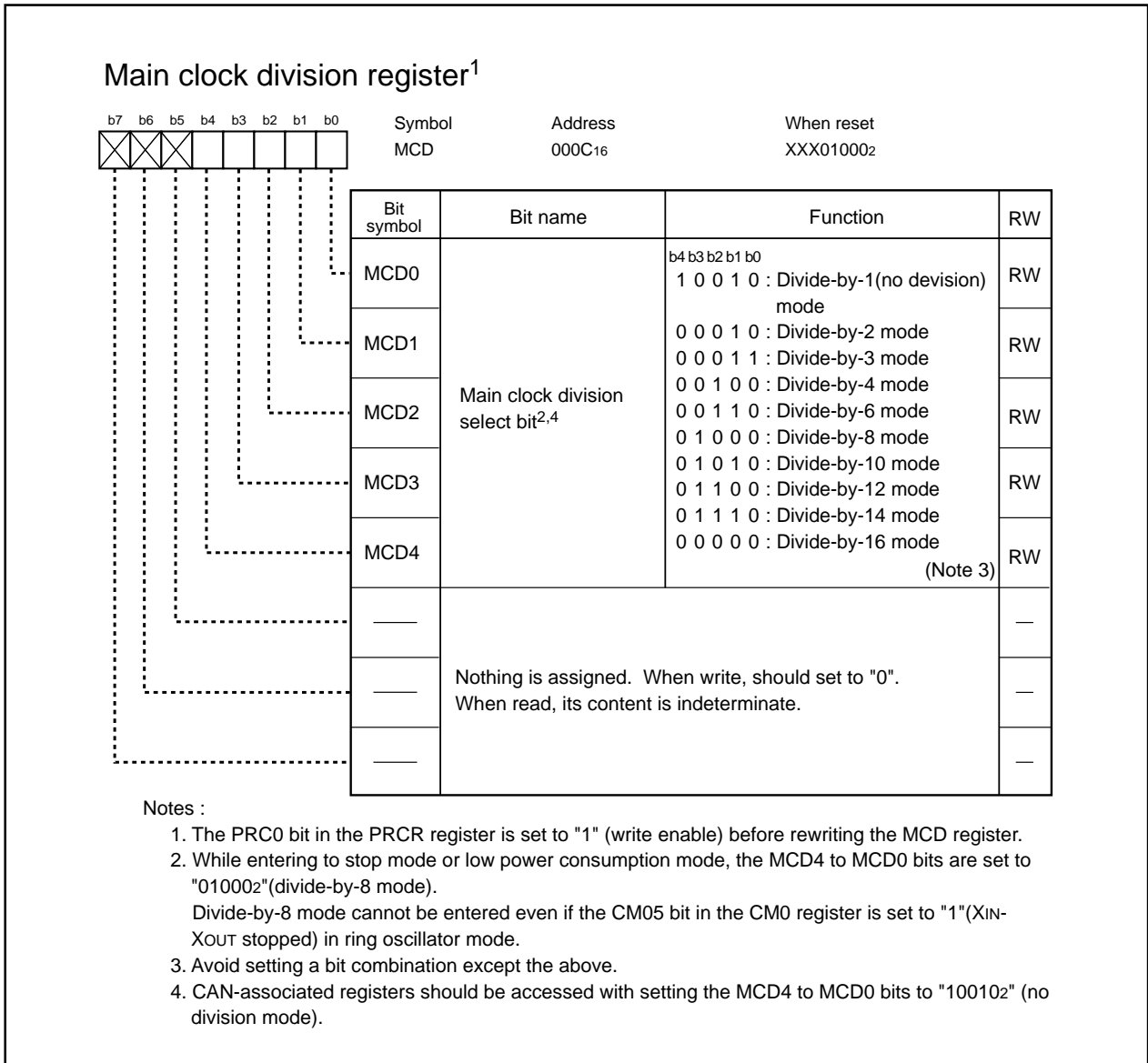


Figure 1.8.4. MCD Register

System Clock

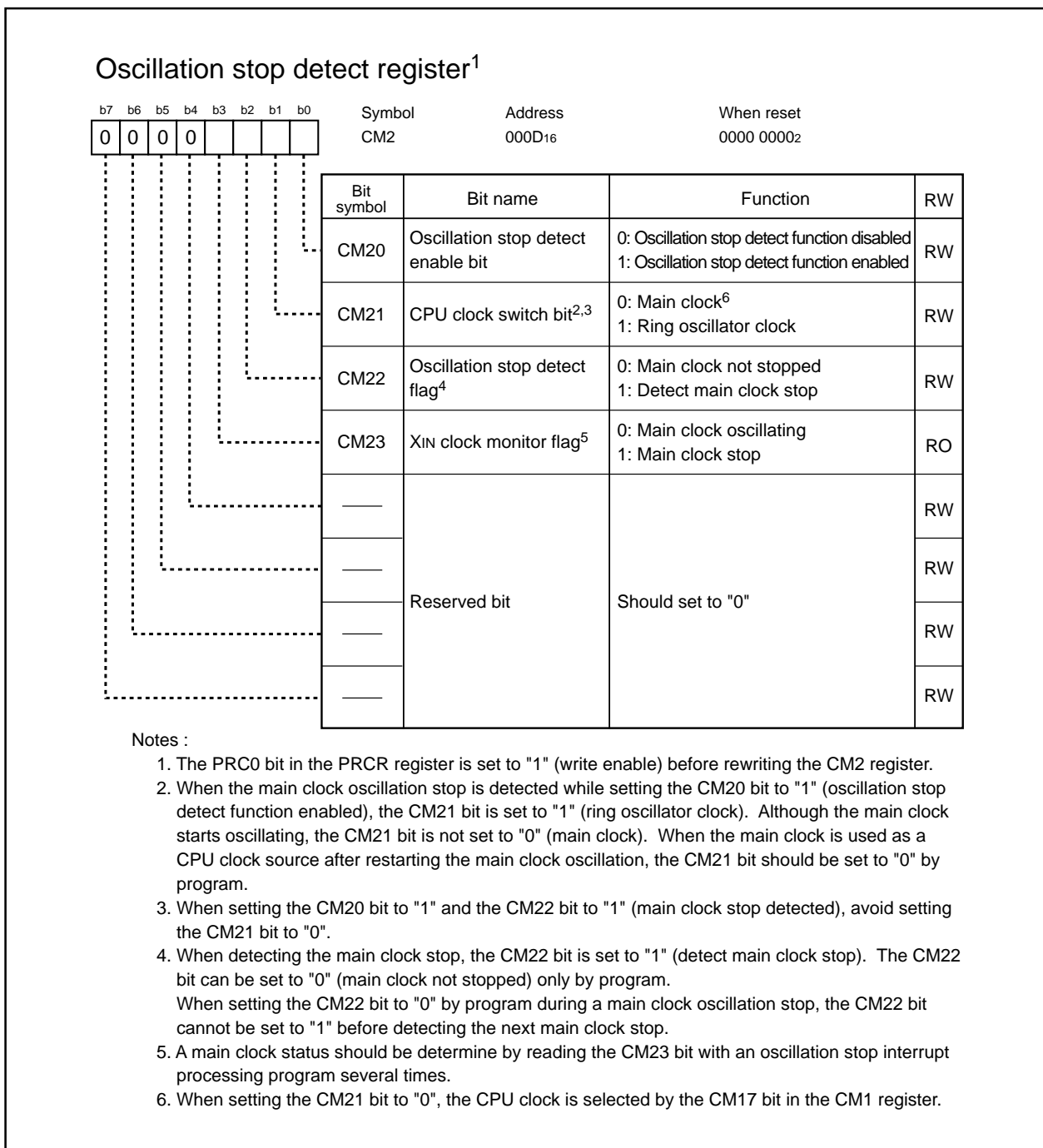


Figure 1.8.5. CM2 Register

System Clock

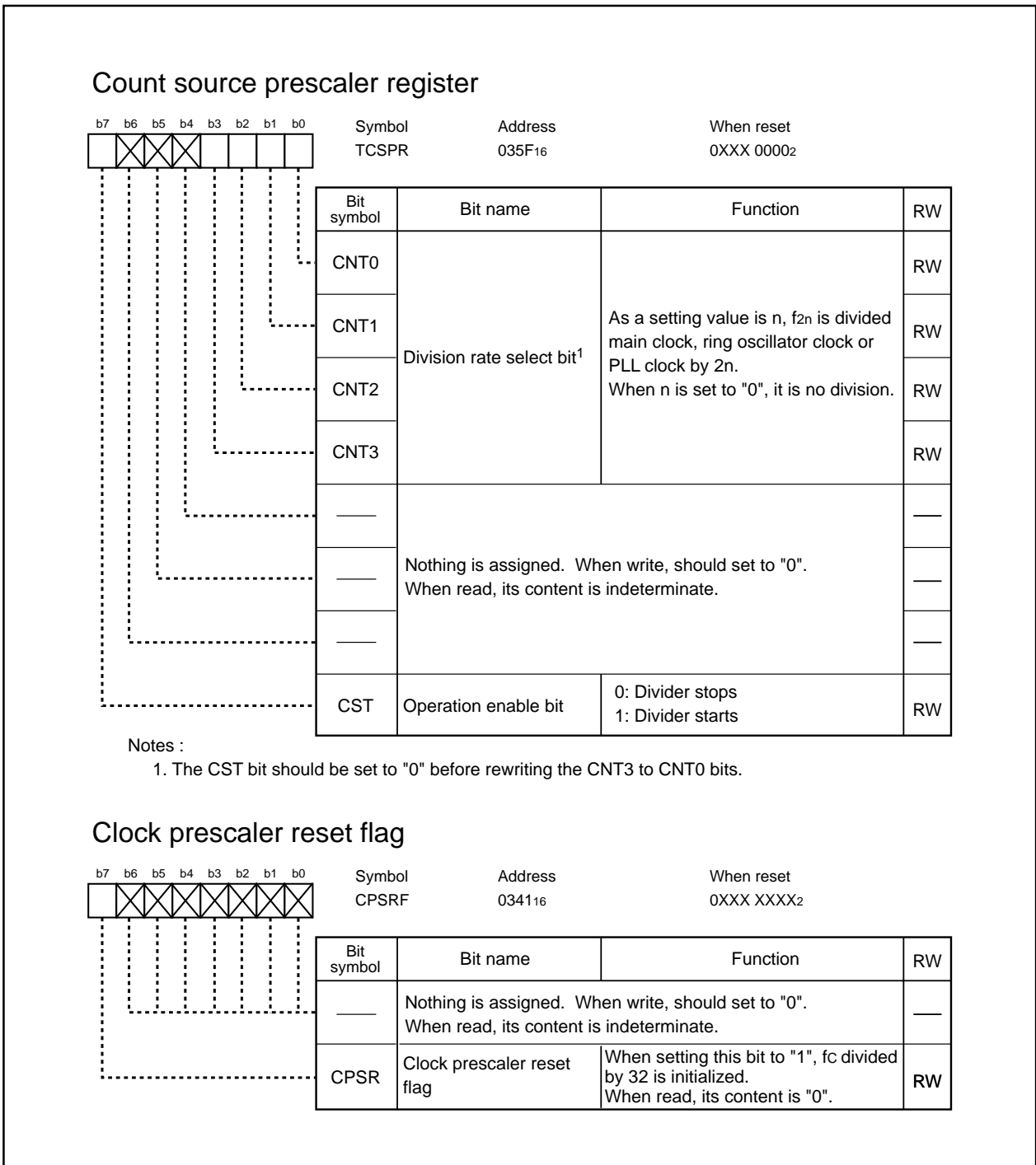
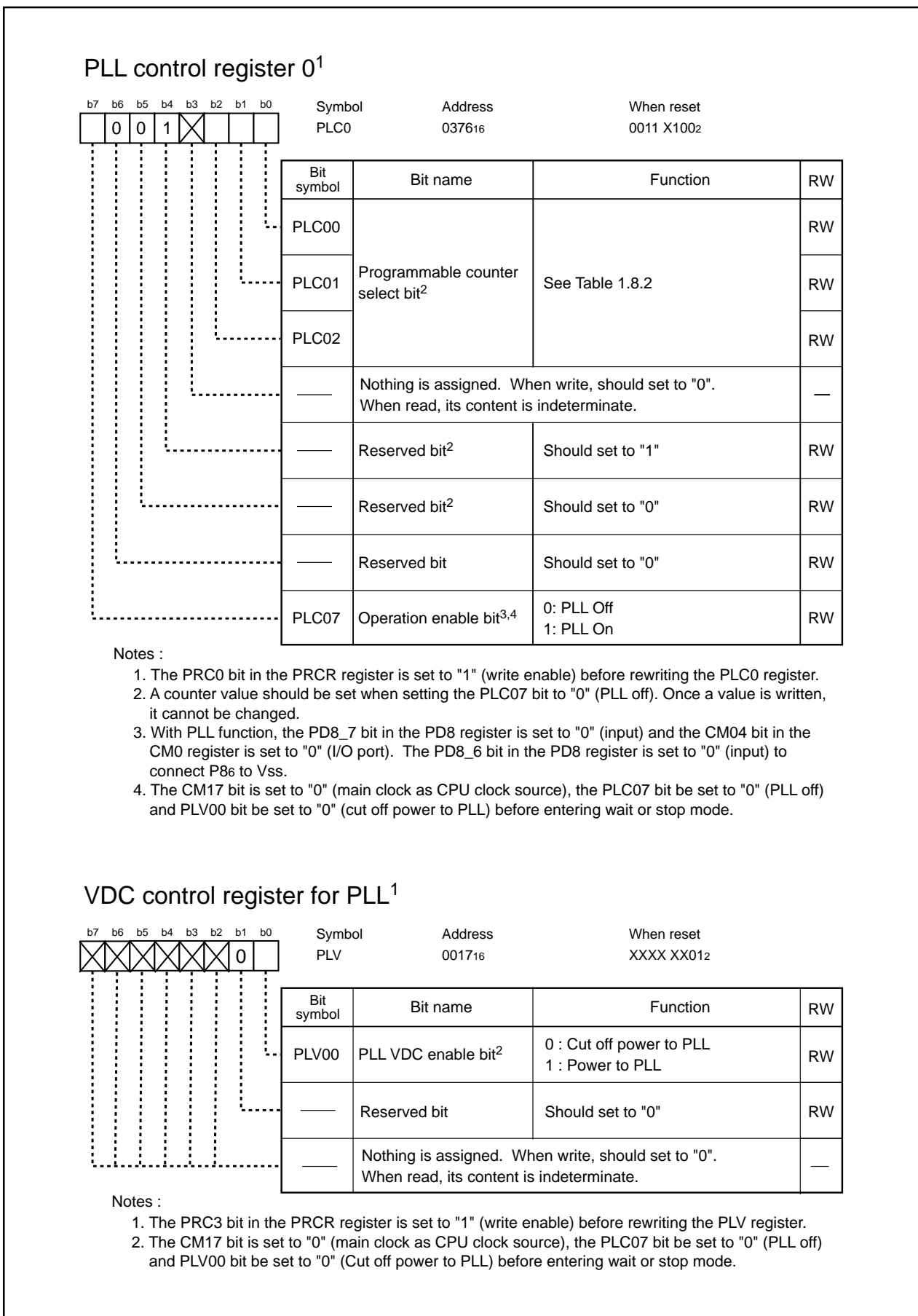


Figure 1.8.6. TCSR and CPSRF Registers

**System Clock**



**Figure 1.8.7. PLC0 and PLV Registers**

System Clock

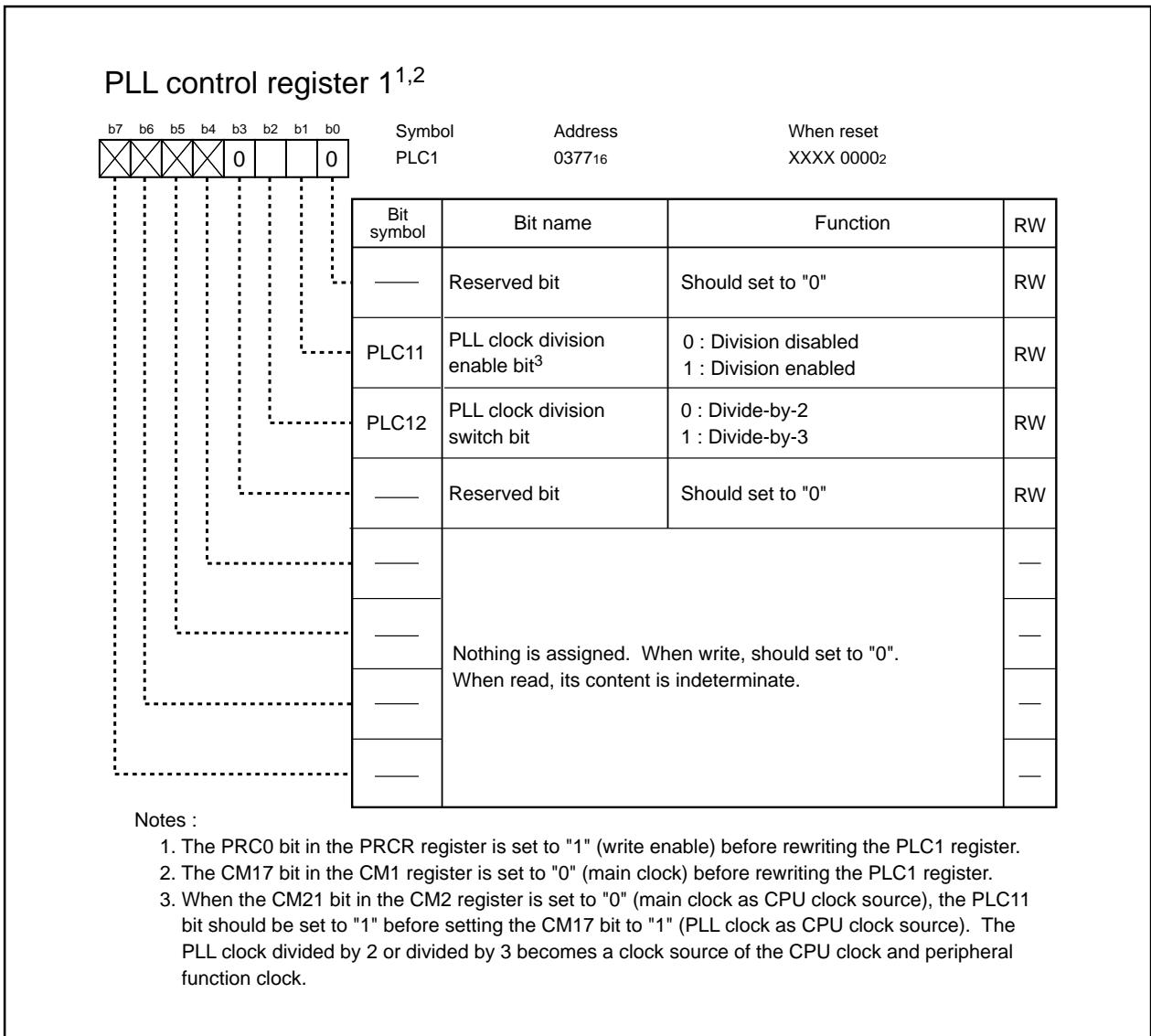


Figure 1.8.8. PLC1 Register

## System Clock

## 1. Main Clock

Main clock oscillation circuit generates the main clock. The main clock becomes a clock source of the CPU clock and peripheral function clock.

The main clock oscillation circuit is configured by connecting an oscillator or resonator between the XIN and XOUT pins. The circuit has a built-in feedback resistance. The feedback resistance is separated from an oscillation circuit in stop mode to reduce power consumption. An externally generated clock can be input to the XIN pin in the main clock oscillation circuit. Figure 1.8.9 shows an example of a main clock circuit connection. Circuit constant varies depending on each oscillator. Circuit constant recommended by each oscillation manufacturer should be used.

The main clock is divided by eight to become the CPU clock after reset.

The CM05 bit in the CM0 register is set to "1" (oscillation stop in main clock oscillation circuit) to reduce power consumption after switching a CPU clock source to the sub clock or ring oscillator clock. In this case, XOUT is set to "H". XIN is pulled up by XOUT via a feedback resistance since a built-in feedback resistance remains on. When an externally generated clock is input to the XIN pin, the main clock does not stop even if the CM05 bit is set to "1". The main clock should be terminated externally if needed.

All clocks, including the main clock, stop in stop mode. Refer to the paragraph "Power Dissipation Control" for details.

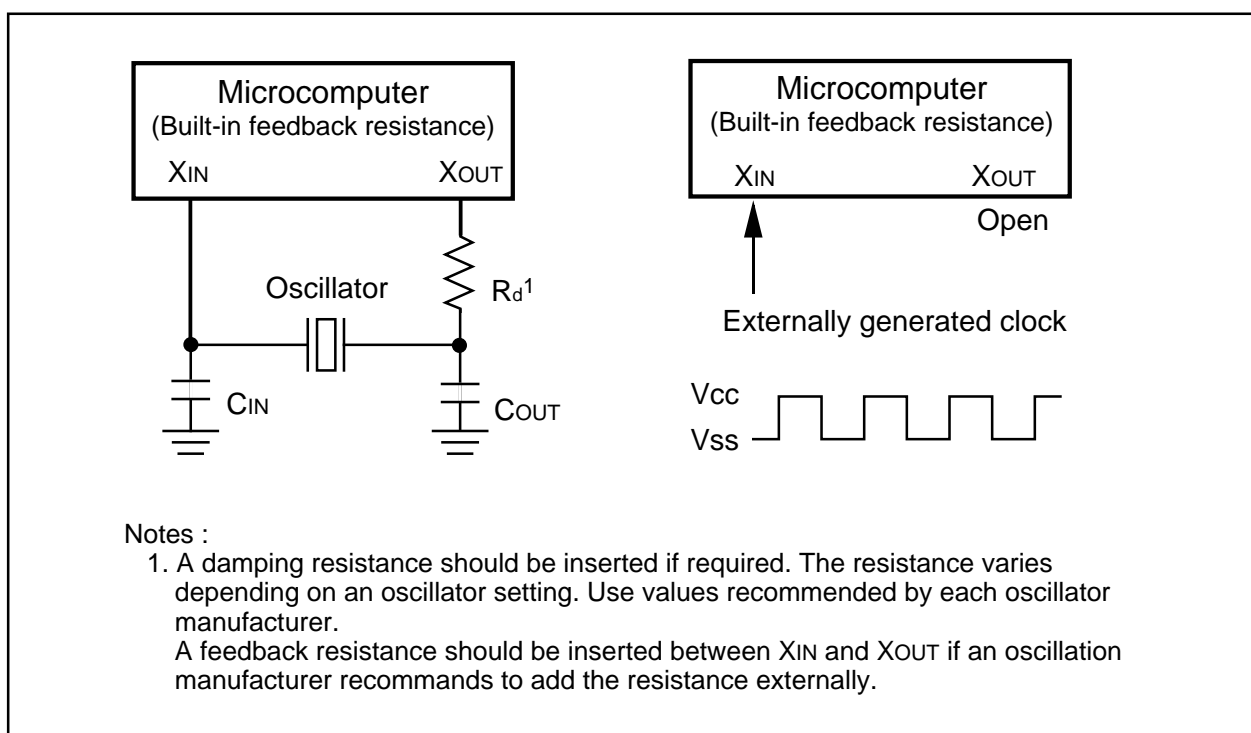


Figure 1.8.9. Main Clock Circuit Connection

## System Clock

### 2. Sub Clock

Sub clock oscillation circuit generates the sub clock. The sub clock becomes a clock source of the CPU clock and a count source of the timers A and B. The same frequency  $f_c$  as the sub clock can be output from the CLKOUT pin.

The sub clock oscillation circuit is configured by connecting an oscillator between the XCIN and XCOUT pins. The circuit has a built-in feedback resistance. The feedback resistance is separated from an oscillation circuit in stop mode to reduce power consumption. An externally generated clock can be input to the XCIN pin. Figure 1.8.10 shows an example of a sub clock circuit connection. Circuit constant varies depending on each oscillator. Circuit constant recommended by each oscillation manufacturer should be used.

The sub clock stops after reset. Feedback resistance is separated from an oscillation circuit. When the PD8\_6 and PD8\_7 bits in the PD8 register is set to "0" (input mode) and the PU25 bit in the PUR2 register is set to "0" (no pull-up), the CM04 bit in the CM0 register is set to "1" (XCIN-XCOUT select). The sub clock oscillation circuit starts oscillating. To input an externally generated clock to the XCIN pin, when setting the PD8\_6 bit to "0" and the PU25 bit to "0", the CM04 bit is set to "1". The clock input to the XCIN pin becomes a clock source of the sub clock.

When a sub clock oscillation is stable to set the CM07 bit of CM0 register to "1" (XCIN-XCOUT select), the sub clock becomes the CPU clock.

All clocks, including the sub clock, stop in stop mode. Refer to the paragraph "Power Dissipation Control" for details.

the XCIN and XCOUT pins shares pins with the VCONT and P86 pins. When the sub clock is used, PLL frequency synthesizer cannot be used.

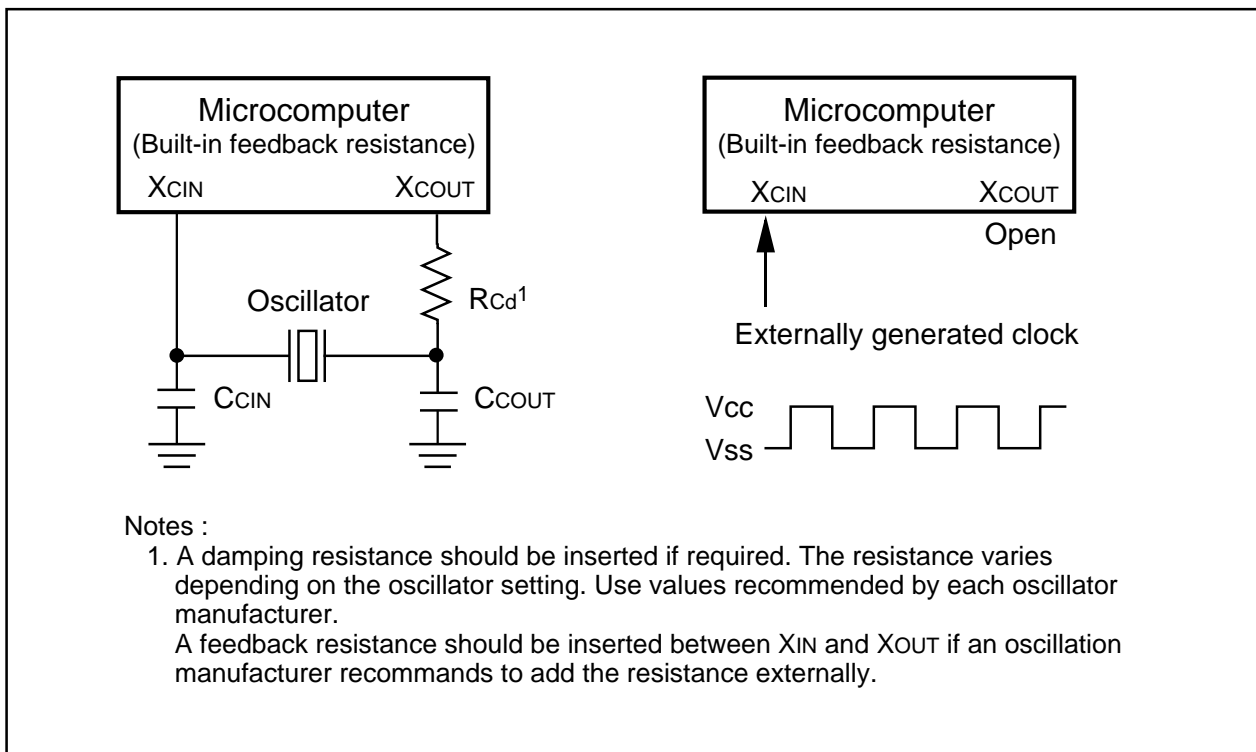


Figure 1.8.10. Sub Clock Connection Circuit

### 3. Ring Oscillator Clock

Ring oscillator generates the ring oscillator clock as an approximate 1MHz clock. The ring oscillator clock becomes a clock source of the CPU clock and peripheral function clock.

The ring oscillator clock stops after reset. When the CM21 bit in the CM2 register is set to "1" (ring oscillator clock), the ring oscillator clock starts oscillating. Instead of the main clock, the ring oscillator clock becomes a clock source for the CPU clock and peripheral function clock.

#### (1) Oscillation Stop Detect Function

When the main clock is terminated by external factors, the ring oscillator automatically starts operating to generate another clock.

When setting the CM 20 bit to "1" (oscillation stop detect function enabled), an oscillation stop detect interrupt request is generated as soon as the main clock stops. Simultaneously, the ring oscillator starts oscillating. The ring oscillator clock, instead of the main clock, becomes a clock source for the CPU clock and peripheral function clock. Associated bits are set as follows:

- CM21 bit = 1 (Ring oscillator clock becomes a clock source of the CPU clock.)
- CM22 bit = 1 (Main clock stop is detected.)
- CM23 bit = 1 (The Main clock stops.) (See Figure 1.8.15)

#### (2) How to Use Oscillation Stop Detect Function

- The oscillation stop detect interrupt shares vectors with the watchdog timer interrupt. When using both oscillation stop detect interrupt and watchdog timer interrupt, the CM22 bit should be read by an interrupt processing program to determine which interrupt request is used.
- When an oscillation stop is detected and the main clock resume its oscillation, the main clock should be set as a clock source of the CPU clock and peripheral function clock. Figure 1.8.11 shows a procedure to switch the ring oscillator clock to the main clock.
- In low-speed mode, when the main clock is stopped by setting the CM20 bit to "1", an oscillation stop detect interrupt request is generated. Simultaneously, the ring oscillator clock starts oscillating. The CPU clock remains unchanged as the sub clock. The ring oscillator clock becomes a clock source of the peripheral function clock.
- To enter wait mode while using the oscillation stop detect interrupt function, the CM02 bit should be set to "0" (peripheral function clock does not stop in wait mode).
- When an oscillation stop detect interrupt request is generated in wait mode, wait mode cannot be exited by the oscillation stop detect interrupt. After its exit from wait mode, the oscillation stop detect interrupt is executed at first and then interrupt, used for its exit from wait mode, are executed.
- The oscillation stop detect function is provided against a main clock stop caused by external factors. When the main clock is terminated by program in stop mode or the CM05 bit is set to "1" (main clock oscillation stop), the CM20 bit should be set to "0" (oscillation stop detect function disabled).
- When a main clock frequency is less than or equal to 2MHz, the oscillation stop detect function is not available. The CM20 bit should be set to "0".

## System Clock

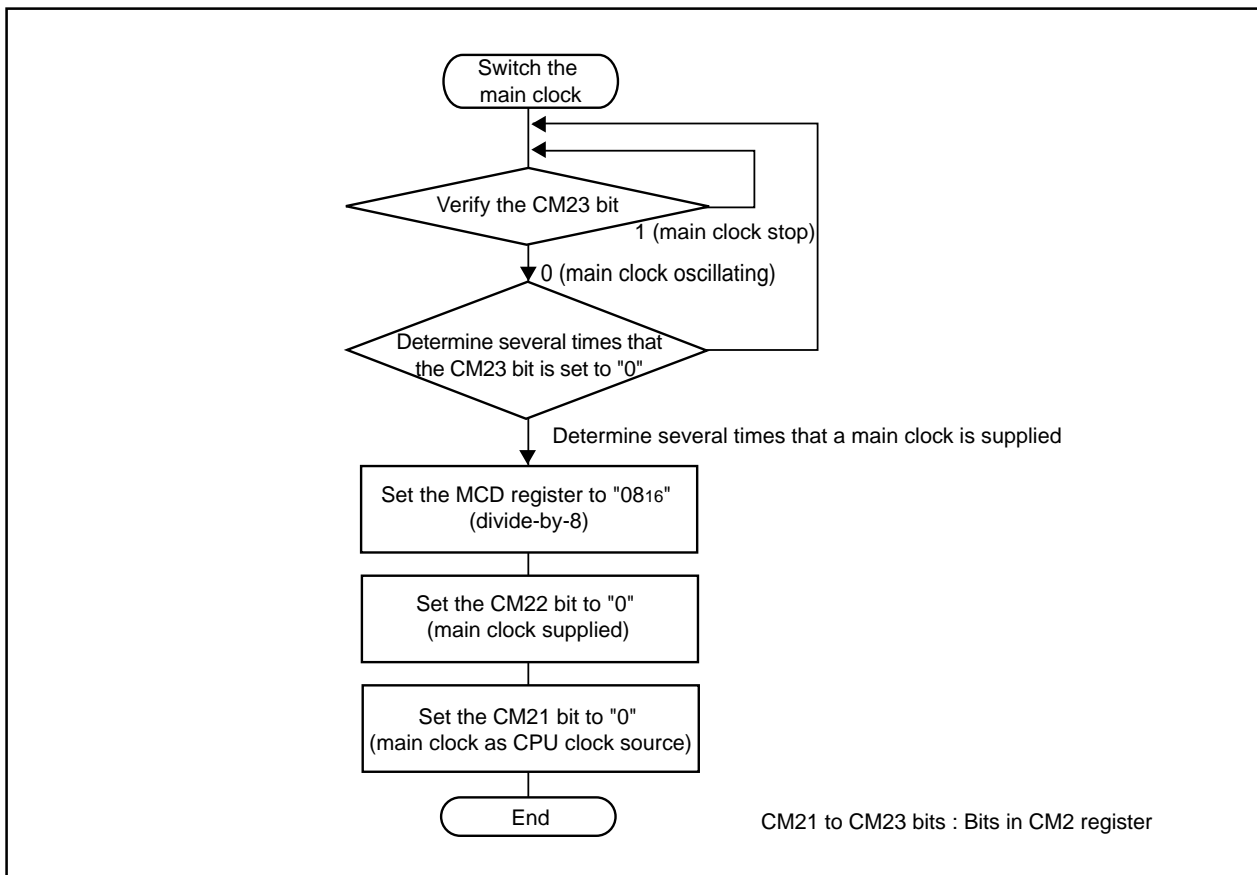


Figure 1.8.11. Switching Procedure from Ring Oscillator Clock to Main Clock

#### 4. PLL Clock

The PLL frequency synthesizer generates the PLL clock, based on the main clock. The PLL clock can be used as a clock source for the CPU clock or peripheral function clock.

With the PLL frequency synthesizer, a resistance and capacitor should be connected to the VCONT pin. The PD8\_6 and PD8\_7 bits in the PD8 register should be set to "0" (input mode) and the CM04 bit be set to "0" (the XCIN and XCOUT pins as ports). After that, the VCONT pin should be connected to the circuit shown in Figure 1.8.12. The P86 pin be connected to Vss. The PLV00 bit in the PLV register should be set to "1" (power to PLL).

The PLL frequency synthesizer stops after reset. When setting the PLC07 bit to "1" (PLL start), the PLL frequency synthesizer starts operating. It takes 20 ms(5V operation) to 50ms(3.3V operation) as a waiting time for the PLL clock to be stable.

When the PLL clock is used as a clock source for the CPU clock or peripheral function clock, the PLL clock is divided by two or divided by three to be a clock source of the CPU clock or peripheral function instead of the main clock. When the PLL clock is used as a clock source for the CPU clock or peripheral function clock, each bit should be set as shown in Table 1.8.2. Figure 1.8.13 shows a procedure to make the PLL clock into CPU clock source.

When entering wait or stop mode, the CM17 bit should be set to "0" (main clock as CPU clock source). The PLC07 bit in the PLC0 register should be set to "0" (PLL stop) and the PLV00 bit be set to "0" (no power to PLL) before entering wait or stop mode.

The VCONT and P86 pins share pins with XCIN and XCOUT pins. When using the PLL frequency synthesizer, the sub clock cannot be used.

System Clock

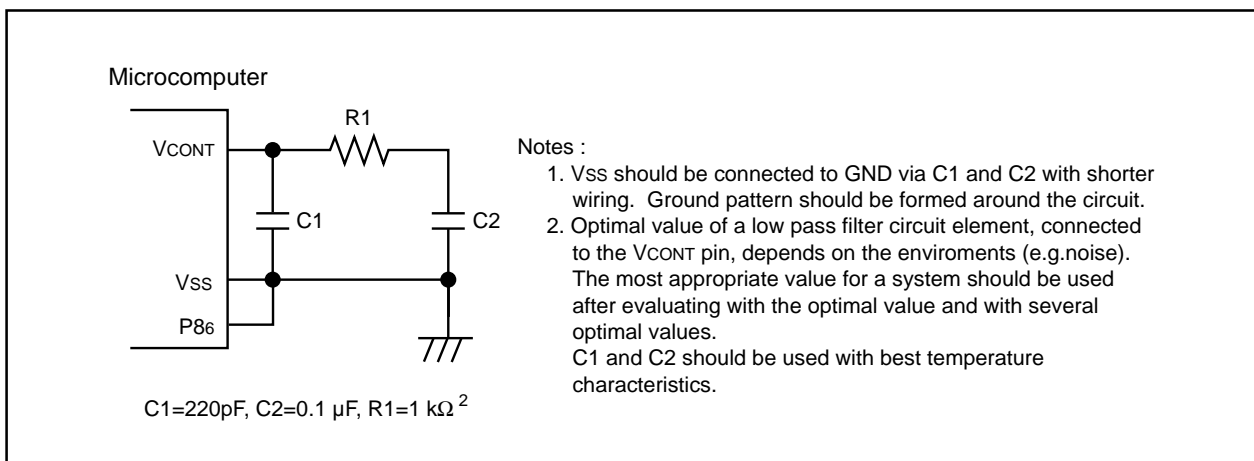


Figure 1.8.12. External Circuit with PLL Frequency Synthesizer

Table 1.8.2. Settings to Use PLL Clock as CPU Clock Source

f(XIN)	PLC0 register			PLC1 register	PLL clock
	PLC02	PLC01	PLC00	PLC12	
10 MHz	0	1	1	0	30 MHz
				1	20 MHz

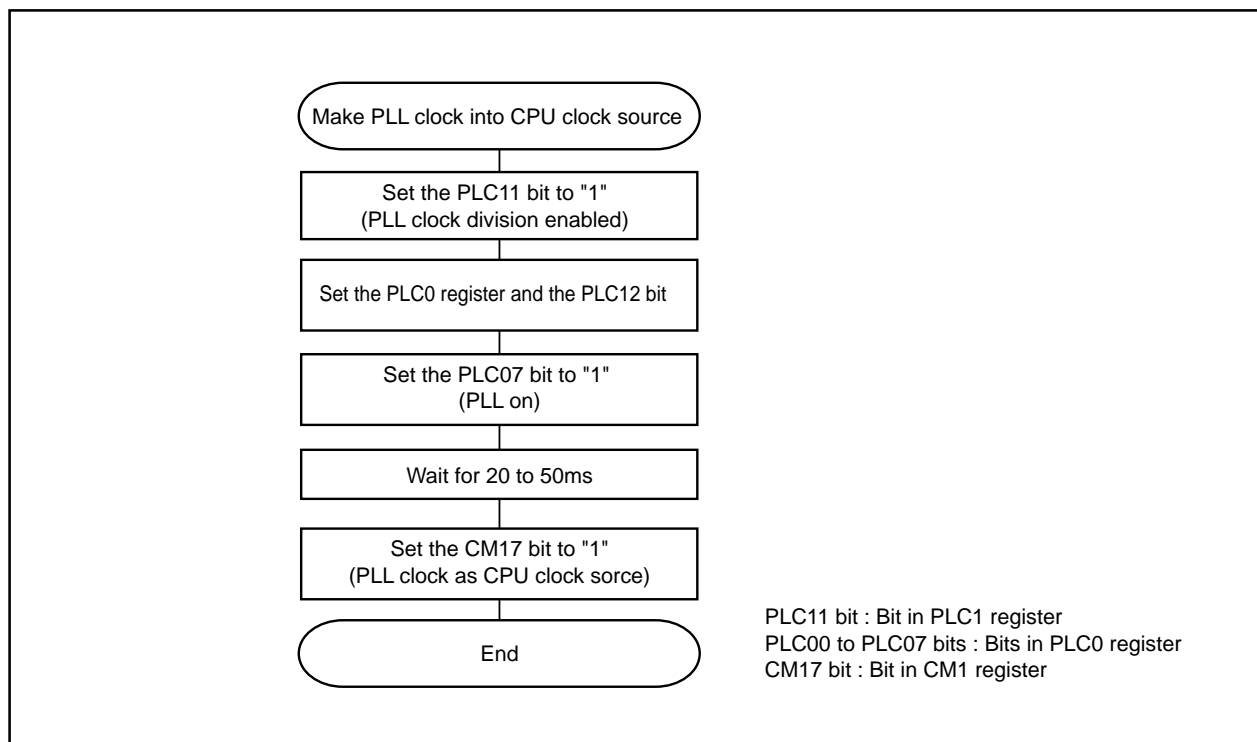


Figure 1.8.13. Procedure to Use PLL Clock as CPU Clock Source

## System Clock

### CPU Clock and BCLK

The CPU clock becomes a CPU operation clock and also a count source of the watchdog timer. The CPU clock is the main clock divided by eight after reset. In memory expansion or microprocessor mode, the clock having the same frequency as the CPU clock can be output from the BCLK pin as BCLK. Refer to the paragraph "Clock Output Function" for details.

The main clock, sub clock, ring oscillator clock or PLL clock divided by two or three can be selected as a clock source for the CPU clock. Table 1.8.3 shows a CPU clock source and bit settings.

When the main clock, ring oscillator clock or PLL clock divided by two or three is selected as a clock source of the CPU clock, the selected clock divided by 1 (no division), 2, 3, 4, 6, 8, 10, 12, 14 or 16 becomes the CPU clock. The MCD register selects the clock division.

When entering stop mode or low power consumption mode (except for the CPU clock as the ring oscillator clock), the MCD register should be set to "0816" (divide-by-8 mode). Consequently, when the main clock starts, the CPU clock enters middle-speed mode (divide-by-8).

**Table 1.8.3. CPU Clock Source and Bit Settings**

CPU clock source	CM0 register	CM2 register	CM1 register
	CM07	CM21	CM17
Main clock	0	0	0
Sub clock	1	0	0
Ring oscillator clock	0	1	0
PLL clock	0	0	1

### Peripheral Function Clock

The peripheral function clock becomes an operation clock or count source of the peripheral functions except the watchdog timer.

#### 1. f<sub>1</sub>, f<sub>8</sub>, f<sub>32</sub> and f<sub>2n</sub>

f<sub>1</sub>, f<sub>8</sub>, f<sub>32</sub> and f<sub>2n</sub> are the main clock<sup>1</sup> or ring oscillator clock divided by 1, 8, 32 or 2n (n=1 to 15, except no division when n=0). The CM21 bit determines which clock is selected.

When setting the CM02 bit to "1" (peripheral function stop in wait mode) to enter wait mode, these clocks stop. The clocks also stop in low-power consumption mode.

f<sub>1</sub>, f<sub>8</sub> and f<sub>2n</sub> are used for an operation clock of the serial I/O and a count source of the timers A and B. The CNT3 to CNT0 bits in the TCSPR register selects a f<sub>2n</sub> division. f<sub>1</sub> is also used for an operation clock of the intelligent I/O.

The CLKOUT pin outputs f<sub>8</sub> and f<sub>32</sub>. Refer to the paragraph "Clock Output Function" for details.

#### 2. f<sub>AD</sub>

f<sub>AD</sub> is an operation clock of the A-D convertor and has the same frequency as the main clock<sup>1</sup> and ring oscillator clock. The CM21 bit determines which clock is selected.

When setting the CM02 bit to "1" (peripheral function stop in wait mode) to enter wait mode, f<sub>AD</sub> stops. f<sub>AD</sub> stops in low power consumption mode.

Notes :

1. It is the PLL clock divided by two when setting the CM17 bit to "1" (PLL clock as CPU clock source).

## System Clock

### 3. fc32

fc32 is the sub clock divided by 32. fc32 is used for a count source of the timers A and B. fc32 is available when the sub clock runs.

### Clock Output Function

The CLKOUT pin outputs fc, f8 or f32.

In memory expansion and microprocessor modes, the BCLK pin can output a clock having the same frequency as the CPU clock as BCLK.

Table 1.8.4 lists a CLKOUT pin function in single-chip mode. Table 1.8.5 lists a CLKOUT pin function in memory expansion and microprocessor modes.

**Table 1.8.4. CLKOUT Pin in Single-Chip Mode**

PM0 register <sup>1</sup>		CM0 register <sup>2</sup>		CLKOUT pin function
PM07		CM01	CM00	
—		0	0	P53 I/O port
1		0	1	fc output
1		1	0	f8 output
1		1	1	f32 output

- : "0" and "1" can be set either.

Notes :

1. The PM0 register should be set after the PRC1 bit in the PRCR register is set to "1" (write enable).
2. The CM0 register should be set after the PRC0 bit in the PRCR register is set to "1" (write enable).

**Table 1.8.5. CLKOUT Pin in Memory Expansion Mode and Microprocessor Mode**

PM1 register <sup>1</sup>		PM0 register <sup>1</sup>		CM0 register <sup>2</sup>		CLKOUT pin function
PM15	PM14	PM07		CM01	CM00	
002, 102, 112,		0		0 <sup>3</sup>	0 <sup>3</sup>	BCLK output
		1		0	0	"L" output (not P53)
		1		0	1	fc output
		1		1	0	f8 output
		1		1	1	f32 output
0	1	—		0 <sup>3</sup>	0 <sup>3</sup>	ALE output

- : "0" and "1" can be set either.

Notes :

1. The PM0 register should be set after the PRC1 bit in the PRCR register is set to "1" (write enable).
2. The CM0 register should be set after the PRC0 bit in the PRCR register is set to "1" (write enable).
3. When setting the PM07 bit to "0" (selected in the CM01 to CM00 bits) or the PM15 to PM14 bits to "012" (P53/BCLK), the CM01 to CM00 bits should be set to "002" (I/O port P53).

### Power Dissipation Control

Power dissipation control contains three modes.

All modes, except wait mode and stop mode, are called normal operation mode in this paragraph. Figure 1.8.14 shows a block diagram of status transition in wait mode and stop mode. Figure 1.8.15 shows a block diagram of status transition in all modes.

## System Clock

---

### 1. Normal Operation Mode

Normal operation mode has six modes.

In normal operation mode, the CPU clock and peripheral function clock operates the CPU and peripheral function. Power dissipation is enabled by controlling a CPU clock frequency. The more CPU clock frequency goes up, the more processing power increases. The more CPU clock frequency goes down, the more power consumption reduces. When unnecessary oscillation circuits stop, power consumption reduces further more.

#### (1) High-Speed Mode

The main clock<sup>1</sup> becomes the CPU clock and a clock source for the peripheral function clock. When the sub clock runs, fc32 can be used for a count source of the timers A and B.

#### (2) Medium-Speed Mode

The main clock divided by 2, 3, 4, 6, 8, 10, 12, 14, or 16 becomes the CPU clock. The main clock becomes a clock source for the peripheral function clock. When the sub clock runs, fc32 can be used for a count source of the timers A and B.

#### (3) Low-Speed Mode

The sub clock becomes the CPU clock. The main clock becomes a count source of the peripheral function clock. fc32 can be used for a count source of the timers A and B.

#### (4) Low-Power Consumption Mode

Low-power consumption mode is entered when the main clock stops in low-speed mode. The sub clock becomes the CPU clock. Only fc32 can be used as the peripheral function clock. In low-power consumption mode, the MCD register should be set to "0816" (divide-by-8 mode). When the main clock starts operating next time, middle-speed mode (divide-by-8 mode) is entered.

#### (5) Ring Oscillator Mode

The ring oscillator clock divided by 1(no division), 2, 3, 4, 6, 8, 10, 12, 14, or 16 becomes the CPU clock. The ring oscillator clock becomes a clock source for the peripheral function clock. When the sub clock runs, fc32 can be used for a count source of the timers A and B.

#### (6) Ring Oscillator Low-Power Consumption Mode

Ring oscillator low-power consumption mode is entered when the main clock stops in ring oscillator mode. The ring oscillator clock divided by 1(no division), 2, 3, 4, 6, 8, 10, 12, 14, or 16 becomes the CPU clock. The ring oscillator clock becomes a clock source for the peripheral function clock. When the sub clock runs, fc32 can be used for a count source of the timers A and B.

The CPU clock should be switched after the clock to be switched is stable. Especially the sub clock takes a longer time<sup>2</sup> for a sub clock oscillation stability. Enough waiting time should be taken for the clock to be stable by program after power-on or an exit from stop mode.

When switching the ring oscillator to the main clock, medium-speed mode should be entered (divide-by-8) after the main clock is divided by eight in ring oscillator mode (MCD register=0816).

Avoid entering ring oscillator mode or ring oscillator low-power consumption mode from low-speed mode or low power consumption mode and vice versa.

Notes :

1. When setting the CM17 bit to "1" (PLL clock as CPU clock source), it is the PLL clock divided by two or three.
2. Contact each oscillator manufacturer about a stable time for oscillation.

## System Clock

### 2. Wait Mode

In wait mode, the CPU clock stops and the CPU and watchdog timer, operated by the CPU clock, also stop. Since the main clock, sub clock, ring oscillator clock and PLL clock continue to run, the peripheral function using these clocks also continue to operate.

#### (a) Peripheral Function Clock Stop Function

When setting the CM02 bit to "1" (peripheral function clock stop in wait mode), f1, f8, f32, f2n and fAD stop in wait mode. Power consumption can be reduced. fc32 does not stop.

#### (b) Entering Wait Mode

Wait mode is entered when the WAIT instruction is executed.

When setting the CM17 bit to "1" (PLL clock as CPU clock source), the CM17 bit should be set to "0" (main clock as CPU clock source) first. Then the PLC07 bit should be set to "0" (PLL stop) and the PLV00 bit be set to "0" (no power to PLL) before entering wait mode.

#### (c) Pin Status in Wait Mode

Table 1.8.6 lists pin status in wait mode.

#### (d) Exiting Wait Mode

Wait mode is exited by a hardware reset,  $\overline{\text{NMI}}$  interrupt or peripheral function interrupt.

When using a hardware reset or  $\overline{\text{NMI}}$  interrupt to exit wait mode, the ILVL2 to ILVL0 bits for the peripheral function interrupt should be set to "0002" (interrupt disabled) before executing the WAIT instruction.

The CM02 bit affected the peripheral function interrupt. When setting the CM02 bit to "0" (peripheral function clock does not stop in wait mode), all peripheral function interrupts can be used to exit wait mode. When setting the CM02 bit to "1" (peripheral function clock stops in wait mode), the peripheral functions, which use the peripheral function clock, stop. The peripheral function interrupt with external signals can be used to exit wait mode.

Table 1.8.7 shows interrupts to be used to exit wait mode and usage conditions.

**Table 1.8.6. Pin Status in Wait Mode**

Pin		Memory expansion mode Microprocessor mode	Single-chip mode
Address bus, data bus, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$ , $\overline{\text{BHE}}$		Maintains status before entering wait mode	/
$\overline{\text{RD}}$ , $\overline{\text{WR}}$ , $\overline{\text{WRL}}$ , $\overline{\text{WRH}}$ , $\overline{\text{DW}}$ , $\overline{\text{CASL}}$ , $\overline{\text{CASH}}$		"H" 1	
$\overline{\text{RAS}}$		"H" 1	
$\overline{\text{HLDA}}$ , $\overline{\text{BCLK}}$		"H"	
ALE		"L"	
Port		Maintains status before entering wait mode	
CLKOUT	When fc is selected	Clock output	
	When f8, f32 are selected	When the CM02 bit in the CM0 register is set to "0" (peripheral function clock not stop in wait mode), the clock is output. When the CM02 bit is set to "1" (peripheral function clock stopped in wait mode), status immediately before entering wait mode is maintained.	

Notes :

1. In self-refresh operation with the DRAMC,  $\overline{\text{CAS}}$  and  $\overline{\text{RAS}}$  are set to "L".

## System Clock

Table 1.8.7. Interrupts to Exit Wait Mode

Interrupt	When CM02=0	When CM02=1
NMI interrupt	Available	Available
Serial I/O interrupt	Available with the internal and external clocks	Available only when the external clock is used
Key input interrupt	Available	Available
A-D conversion interrupt	Available in single or single-sweep mode	Avoid using
Timer A interrupt Timer B interrupt	Available in all modes	Available in event counter mode or when a count source is fc32
$\overline{\text{INT}}$ interrupt	Available	Available
CAN interrupt	Available	Avoid using
Intelligent I/O interrupt	Available	Avoid using

When the peripheral function interrupt is used to exit wait mode, the followings should be set before executing the WAIT instruction.

- (1) The RLVL2 to RLVL0 bits in the RLVL register should have the same value as the IPL in the FLG register has.
- (2) The ILVL2 to ILVL0 bits in the interrupt control registers, for the peripheral function interrupt used to exit wait mode, should have larger value than the RLVL2 to RLVL0 bits have.  
The ILVL2 to ILVL0 bits in all other interrupt control registers, for peripheral function interrupts which are not used to exit wait mode, should be set to "0002" (interrupt disabled).
- (3) The I flag should be set to "1".
- (4) Peripheral functions, which are used to exit wait mode, start operating.

When using the peripheral function interrupt to exit wait mode, the CPU clock resumes to be provided by generating an interrupt request and running an interrupt routine.

The CPU clock in exiting wait mode with the peripheral function interrupt is the same clock as the CPU clock in executing the WAIT instruction.

### 3. Stop Mode

In stop mode, all oscillation stop. The CPU clock and peripheral function clock stop and the CPU and peripheral function, operated by these clock, also stop. Stop mode needs the least power to work. When Vcc is more or equal to 2.5V, the internal RAM continue to operate.

The peripheral functions operated by external signals do not stop. Interrupts used to exit stop mode are  $\overline{\text{NMI}}$  interrupt, key input interrupt and  $\overline{\text{INT}}$  interrupt.

#### (1) Entering Stop Mode

When the CM10 bit in the CM1 register is set to "1" (all clocks stops), stop mode is entered. The MCD register is simultaneously set to "0816" (divide-by-8 mode).

Avoid entering stop mode when the CM21 bit in the CM2 register is set to "1" (ring oscillator clock as CPU clock source). Stop mode should be entered after setting the CM20 bit to "0" (oscillation stop detect function disabled) and the CM21 bit to "0" (main clock as CPU clock source).

To enter stop mode, when the CM17 bit is set to "1" (PLL clock as CPU clock source), the CM17 bit should be set to "0" (main clock as CPU clock source) at first. Then the PLC07 bit should be set to "0" (PLL off) and the PLV00 bit be set to "0" (no power to PLL).

#### (2) Pin Status in Stop Mode

Table 1.8.8 lists pin status in stop mode.

#### (3) Exiting Stop Mode

Stop mode is exited by a hardware reset,  $\overline{\text{NMI}}$  interrupt or peripheral function interrupt.

When using a hardware reset or  $\overline{\text{NMI}}$  interrupt to exit wait mode, all ILVL2 to ILVL0 bits in the interrupt control registers for the peripheral function interrupt should be set to "0002" (interrupt disabled) to set the CM10 bit to "1".

When the peripheral function interrupt is used to exit stop mode, the followings should be set before setting the CM10 bit to "1".

- (1) The RLVL2 to RLVL0 bits in the RLVL register should have the same value as the IPL bit has.
- (2) The ILVL2 to ILVL0 bits in the interrupt control registers, for the peripheral function interrupt used to exit stop mode, should have larger value than the RLVL2 to RLVL0 bits have.  
The ILVL2 to ILVL0 bits in all other interrupt control registers, for the peripheral function interrupts which are not used to exit stop mode, should be set to "0002" (interrupt disabled).
- (3) The I flag should be set to "1".
- (4) Peripheral functions, which are used to exit wait mode starts operating.

When using a peripheral function interrupt to exit wait mode, the CPU clock resumes to be provided by generating an interrupt request and running an interrupt routine.

The CPU clock in exiting stop mode by the peripheral function interrupt or  $\overline{\text{NMI}}$  interrupt is as follows, according to the CPU clock before entering stop mode.

- When the sub clock is set as the CPU clock before entering stop mode : Sub clock
- When the main clock is set as the CPU clock before entering stop mode : Main clock divided by eight

System Clock

Table 1.8.8. Pin Status in Stop Mode

Pin		Memory expansion mode Microprocessor mode	Single-chip mode
Address bus, data bus, $\overline{CS0}$ to $\overline{CS3}$ , $\overline{BHE}$		Maintains status before entering stop mode	/
$\overline{RD}$ , $\overline{WR}$ , $\overline{WRL}$ , $\overline{WRH}$ , $\overline{DW}$ , $\overline{CASL}$ , $\overline{CASH}$		"H" 1	
$\overline{RAS}$		"H" 1	
$\overline{HLDA}$ , $\overline{BCLK}$		"H"	
ALE		"H"	
Port		Maintains status before entering stop mode	
CLKOUT	When f <sub>c</sub> selected	"H"	
	When f <sub>8</sub> , f <sub>32</sub> selected	Maintains status before entering stop mode	
XIN		High-impedance	
XOUT		"H"	
XCIN, XCOU		High-impedance	

Notes :

1. In self-refresh operation with the DRAMC,  $\overline{CAS}$  and  $\overline{RAS}$  are set to "L".

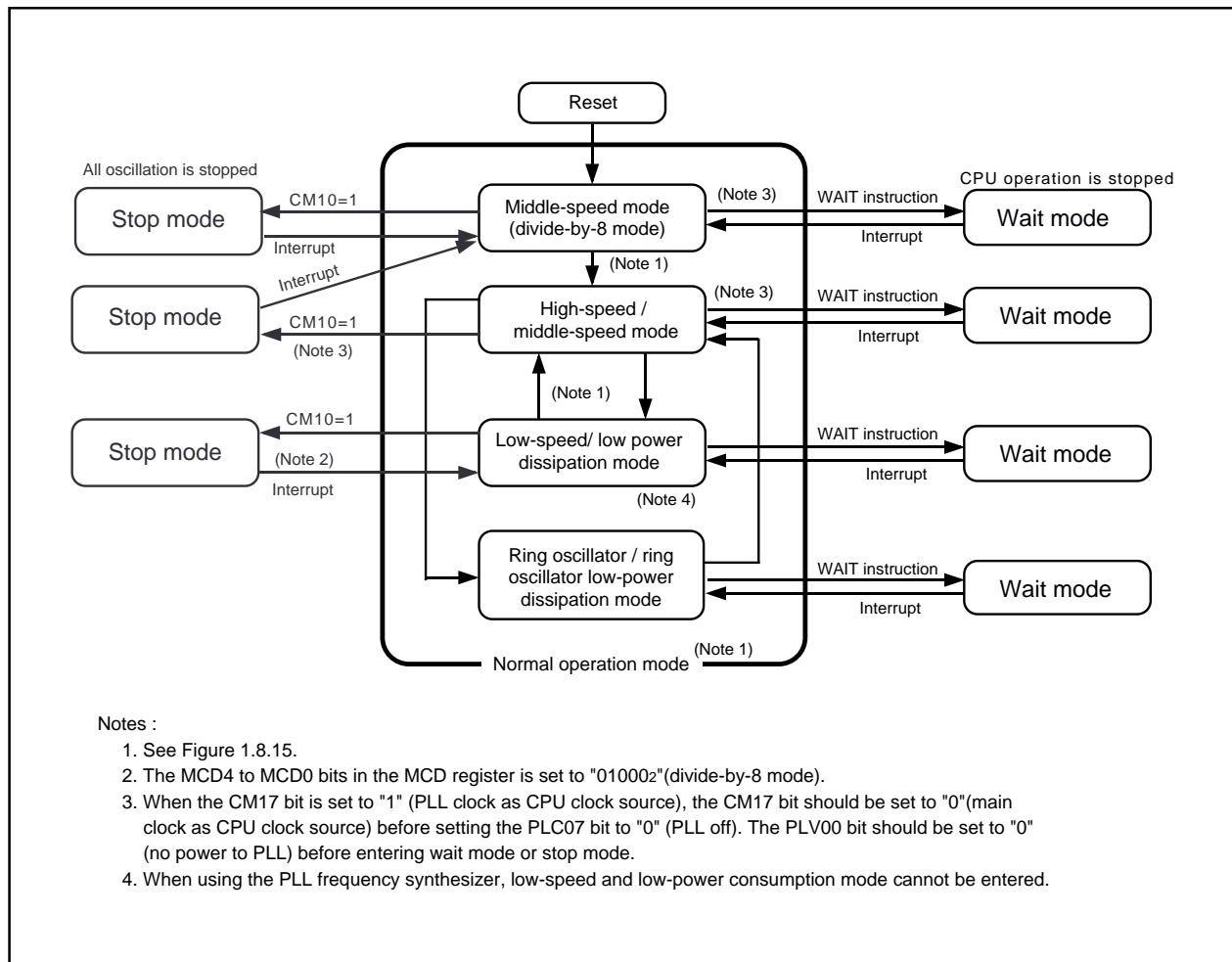


Figure 1.8.14. Status Transition in Wait Mode and Stop Mode

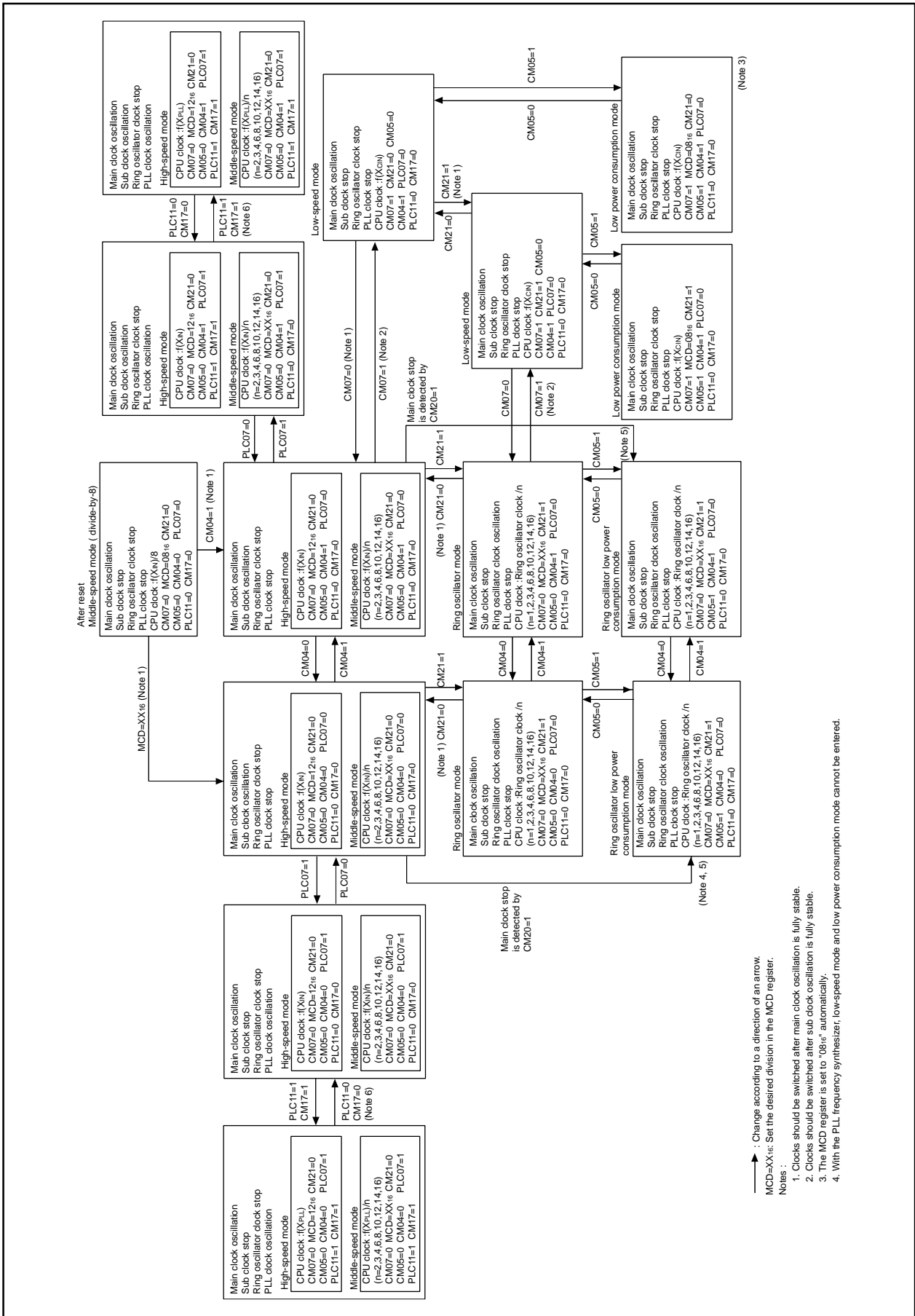


Figure 1.8.15. Status Transition

Protection

# Protection

The protection function protects important registers from rewriting easily when a program runs out of control.

Figure 1.8.16 shows the PRCR register. The PRCR register protects the following registers.

Registers protected by the PRC0 bit : CM0, CM1, CM2, MCD, PLC0 and PLC1 registers

Registers protected by the PRC1 bit : PM0, PM1, INVC0 and INVC1 registers

Registers protected by the PRC2 bit : PD9 and PS3 registers

Registers protected by the PRC3 bit : PLV, VDC0 and VDC1 registers

When setting the PRC2 bit to "1" (write enable) and trying to write into any addresses, the PRC2 bit is automatically set to "0" (write disable). The PD9 and PS3 registers should be set subsequent to the instruction which sets the PRC2 bit to "1" (write enable). Avoid interrupt and DMA transfer between the instruction to set the PRC2 bit to "1" and the next instruction. The PRC0, PRC1 and PRC3 bits are not set to "0" even if trying to write into any addresses. The PRC0, PRC1 and PRC3 bits should be set to "0" by program.

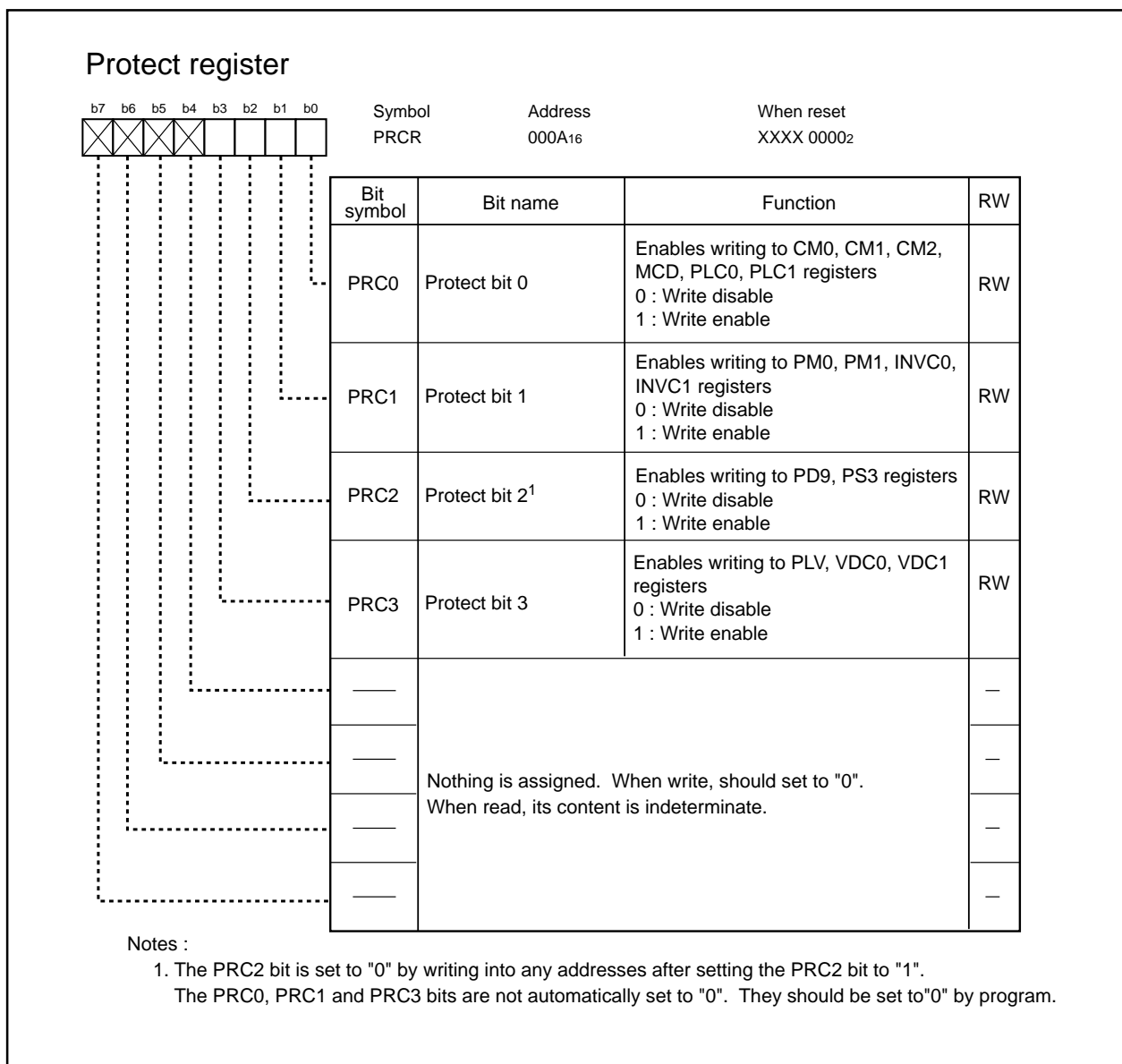
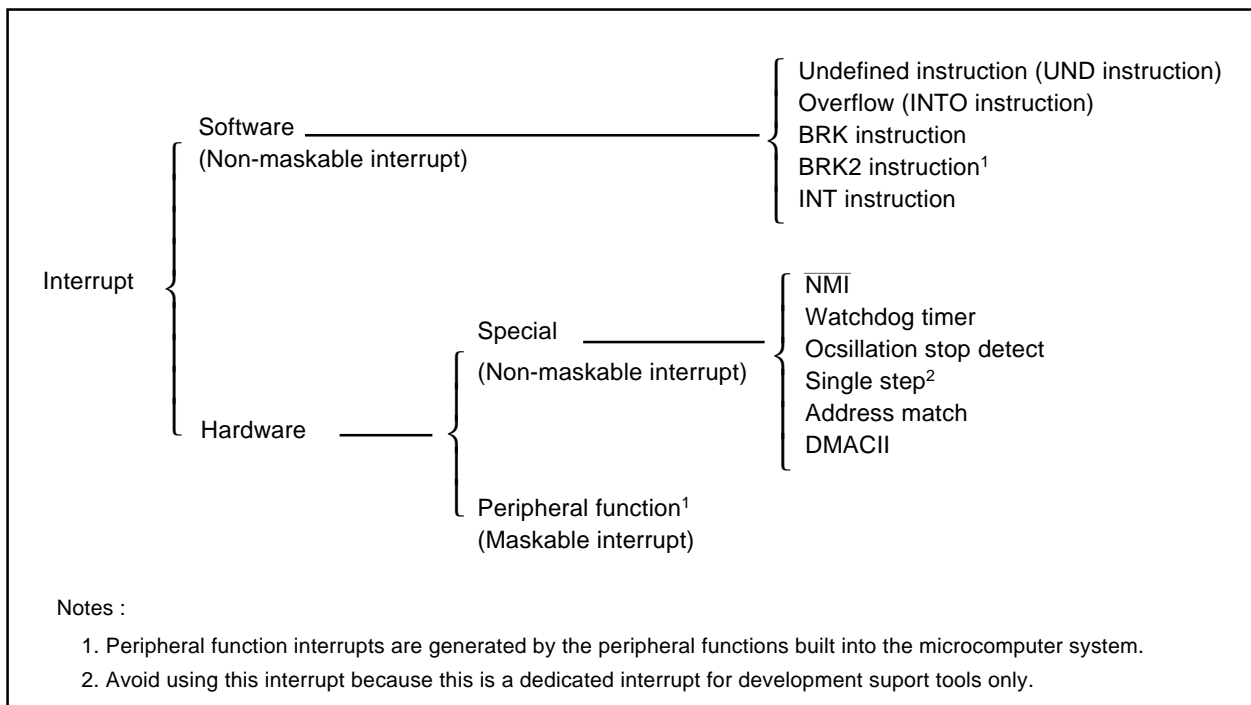


Figure 1.8.16. PRCR Register

# Interrupts

## Types of Interrupts

Figure 1.9.1 shows types of interrupts.



**Figure 1.9.1. Interrupts**

### • Maskable Interrupt

The I flag can change an interrupt enabled to an interrupt disabled and vice versa.

An interrupt priority under interrupt priority level order **can be changed**.

### • Non-maskable Interrupt

The I flag can change an interrupt enabled to an interrupt disabled and vice versa.

An interrupt priority under interrupt priority level order **cannot be changed**.

## Software Interrupts

Software interrupts are generated by executing an instruction. The software interrupts are non-maskable interrupts.

### (1) Undefined Instruction Interrupt

The undefined instruction interrupt is generated when the UND instruction is executed.

### (2) Overflow Interrupt

The overflow interrupt is generated when the INTO instruction is executed by setting the O flag to "1" (arithmetic operation overflow).

Instructions to set the O flag are :

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

## Interrupts

---

### (3) BRK Interrupt

The BRK interrupt is generated when the BRK instruction is executed.

### (4) BRK2 Interrupt

The BRK2 interrupt is generated when the BRK2 instruction is executed.

Avoid using this interrupt. For a development support tool only.

### (5) INT Instruction Interrupt

The INT instruction interrupt is generated when the INT instruction is executed. The INT instruction can select software interrupt numbers 0 to 63. Since software interrupt numbers 7 to 54 and 57 are assigned to the peripheral function interrupt, the INT instruction can operate the same interrupt routine as used for the peripheral function interrupt.

When executing the INT instruction, the FLG register and PC are saved to the stack. PC also stores a relocatable vector in the specified software interrupt number. Where the stack is saved varies depending on a software interrupt number. ISP is selected with the software interrupt numbers 0 to 31 (setting the U flag to "0"). SP, which is set before executing the INT instruction, is selected with the software interrupt numbers 32 to 63.

With the peripheral function interrupt, the FLG register is saved and the U flag is set to "0" (ISP select) when an interrupt request is acknowledged. With software interrupt numbers 32 to 54 and 57, SP to be used varies, depending on an interrupt which is generated by a peripheral function interrupt request or by the INT instruction.

## Hardware Interrupts

Special interrupt and peripheral function interrupt are available as hardware interrupts.

### (1) Special Interrupt

Special interrupt is an non-maskable interrupt.

- **NMI Interrupt**

The  $\overline{\text{NMI}}$  interrupt is generated when  $\overline{\text{NMI}}$  pin input changes "H" to "L". Refer to the paragraph " $\overline{\text{NMI}}$  interrupt" for details.

- **Watchdog Timer Interrupt**

The watchdog timer interrupt is generated by the watchdog timer. Refer to the section "Watchdog timer" for details.

- **Oscillation Stop Detect Interrupt**

The oscillation stop detect interrupt is generated when a main clock oscillation stop is detected. Refer to the section "System clock" for details.

- **Single-Step Interrupt**

Avoid using the single-step interrupt. For development support tool only.

- **Address Match Interrupt**

The address match interrupt is generated just before executing an instruction stored into an address that the RMADi register indicates when the AIERi bit in the AIER register (i=0 to 3) is set to "1" (address match interrupt enabled). A starting address of an instruction should be set in the RMADi register. Address match interrupt is not generated when setting an address such as table data or while executing an instruction. Refer to the paragraph "Address match interrupt" for details.

## Interrupts

### (2) Peripheral Function Interrupt

The peripheral function interrupt is generated by the peripheral functions built into the microcomputer. Interrupt vector table for the peripheral function interrupt is the same as the one for software interrupt numbers 7 through 54 and 57 used with the INT instruction. The peripheral function interrupt is a maskable interrupt.

See Table 1.9.2 about how the peripheral function interrupt is generated. Refer to explanations on each function for details.

### High-Speed Interrupts

The high-speed interrupt executes an interrupt sequence in 5 cycles and a return from the interrupt in 3 cycles.

When the FSIT bit in the RLVL register is set to "1" (interrupt priority 7 available for the high-speed interrupt), the ILVL2 to ILVL0 bits in the interrupt control registers can be set to "1112" (level 7) to use the high-speed interrupt.

Only one interrupt can be set as the high-speed interrupt. With the high-speed interrupt, avoid setting multiple interrupts as level 7 interrupt. The DMA II bit in the RLVL register should be set to "0" (interrupt priority level 7 available for interrupts).

A starting address of a high-speed interrupt routine should be set in the VCT register.

When the high-speed interrupt is acknowledged, the FLG register is saved into the SVF register and PC are saved into the SVP registers. A program is executed from an address indicated by the VCT register. The FREIT instruction should be executed to return from high-speed interrupt routine.

The saved values in the SVF and SVP registers are restored to the FLG register and PC by executing the FREIT instruction.

The same registers are simultaneously used in the high-speed interrupt and in DMA2 and DMA3. With the high-speed interrupt, neither DMA2 nor DMA3 is available. DMA0 and DMA1 are available, instead.

### Interrupts and Interrupt Vectors

There is four bytes in one vector. A starting address of interrupt routine should be set in each vector table. When an interrupt request is acknowledged, a program is executed from an address set in interrupt vectors. Figure 1.9.2 shows the interrupt vector.

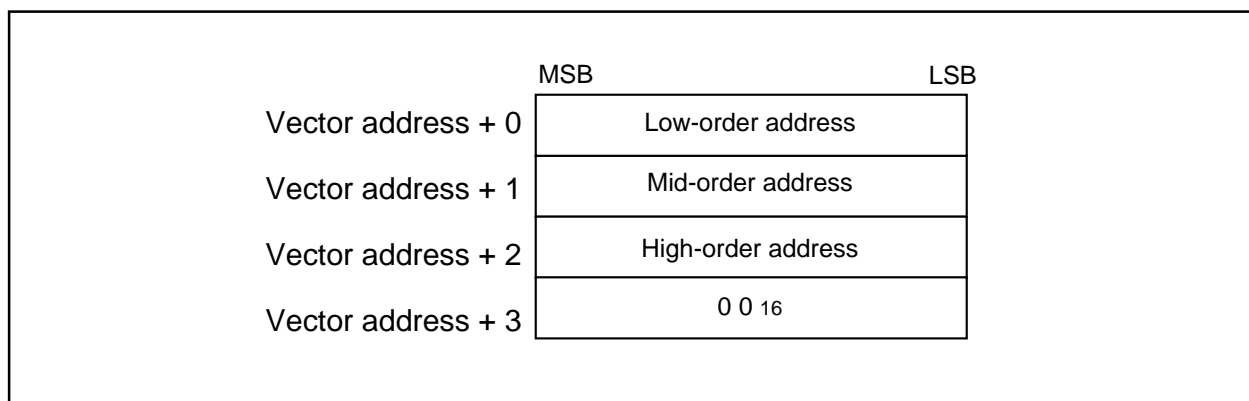


Figure 1.9.2. Interrupt Vector

## Interrupts

### (1) Fixed Vector Tables

Fixed vector tables are allocated in addresses from FFFFDC<sub>16</sub> to FFFFFFF<sub>16</sub>. Table 1.9.1 lists fixed vector tables. Refer to the paragraph "Functions to inhibit rewriting flash memory" about fixed vectors of flash memory.

**Table 1.9.1. Fixed Vector Table**

Interrupt generated by	Vector addresses Address (L) to address (H)	Remarks	Reference
Undefined instruction	FFFFDC <sub>16</sub> to FFFFDF <sub>16</sub>		M32C/80 series software manual
Overflow	FFFFE0 <sub>16</sub> to FFFF E3 <sub>16</sub>		
BRK instruction	FFFFE4 <sub>16</sub> to FFFF E7 <sub>16</sub>	If a content of FFFF E7 <sub>16</sub> is FF <sub>16</sub> , a program is executed from an address stored into software interrupt number 0 in variable vector table	
Address matching	FFFFE8 <sub>16</sub> to FFFF EB <sub>16</sub>		
-	FFFFEC <sub>16</sub> to FFFF EF <sub>16</sub>	Reserved space	
Watchdog timer	FFFFF0 <sub>16</sub> to FFFF F3 <sub>16</sub>	These address are shared by the watchdog timer and oscillation stop detect interrupt	Watchdog timer
-	FFFFF4 <sub>16</sub> to FFFF F7 <sub>16</sub>	Reserved space	
NMI	FFFFF8 <sub>16</sub> to FFFF FB <sub>16</sub>		
Reset	FFFFFC <sub>16</sub> to FFFF FF <sub>16</sub>		Reset

### (2) Relocatable Vector Tables

Relocatable vector tables occupy 256 bytes from a starting address set in the INTB register. Table 1.9.2 lists relocatable vector tables.

An even address should be set as a starting address of vector table set in the INTB register to increase interrupt sequence executing rate.

## Interrupts

Table 1.9.2. Relocatable Vector Tables

Interrupt generated by	Vector table address Address(L)to address(H) <sup>1</sup>	Software interrupt number	Reference
BRK instruction <sup>2</sup>	+0 to +3 (0000 <sub>16</sub> to 0003 <sub>16</sub> )	0	M32C/80 series
Reserved space	+4 to +27 (0004 <sub>16</sub> to 0027 <sub>16</sub> )	1 to 6	software manual
A-D 1	+28 to +31 (001C <sub>16</sub> to 001F <sub>16</sub> )	7	A-D converter
DMA0	+32 to +35 (0020 <sub>16</sub> to 0023 <sub>16</sub> )	8	DMAC
DMA1	+36 to +39 (0024 <sub>16</sub> to 0027 <sub>16</sub> )	9	
DMA2	+40 to +43 (0028 <sub>16</sub> to 002B <sub>16</sub> )	10	
DMA3	+44 to +47 (002C <sub>16</sub> to 002F <sub>16</sub> )	11	
Timer A0	+48 to +51 (0030 <sub>16</sub> to 0033 <sub>16</sub> )	12	Timer A
Timer A1	+52 to +55 (0034 <sub>16</sub> to 0037 <sub>16</sub> )	13	
Timer A2	+56 to +59 (0038 <sub>16</sub> to 003B <sub>16</sub> )	14	
Timer A3	+60 to +63 (003C <sub>16</sub> to 003F <sub>16</sub> )	15	
Timer A4	+64 to +67 (0040 <sub>16</sub> to 0043 <sub>16</sub> )	16	
UART0 transmission, NACK <sup>3</sup>	+68 to +71 (0044 <sub>16</sub> to 0047 <sub>16</sub> )	17	Serial I/O
UART0 reception, ACK <sup>3</sup>	+72 to +75 (0048 <sub>16</sub> to 004B <sub>16</sub> )	18	
UART1 transmission, NACK <sup>3</sup>	+76 to +79 (004C <sub>16</sub> to 004F <sub>16</sub> )	19	
UART1 reception, ACK <sup>3</sup>	+80 to +83 (0050 <sub>16</sub> to 0053 <sub>16</sub> )	20	
Timer B0	+84 to +87 (0054 <sub>16</sub> to 0057 <sub>16</sub> )	21	Timer B
Timer B1	+88 to +91 (0058 <sub>16</sub> to 005B <sub>16</sub> )	22	
Timer B2	+92 to +95 (005C <sub>16</sub> to 005F <sub>16</sub> )	23	
Timer B3	+96 to +99 (0060 <sub>16</sub> to 0063 <sub>16</sub> )	24	
Timer B4	+100 to +103 (0064 <sub>16</sub> to 0067 <sub>16</sub> )	25	
INT5	+104 to +107 (0068 <sub>16</sub> to 006B <sub>16</sub> )	26	Interrupt
INT4	+108 to +111 (006C <sub>16</sub> to 006F <sub>16</sub> )	27	
INT3	+112 to +115 (0070 <sub>16</sub> to 0073 <sub>16</sub> )	28	
INT2	+116 to +119 (0074 <sub>16</sub> to 0077 <sub>16</sub> )	29	
INT1	+120 to +123 (0078 <sub>16</sub> to 007B <sub>16</sub> )	30	
INT0	+124 to +127 (007C <sub>16</sub> to 007F <sub>16</sub> )	31	
Timer B5	+128 to +131 (0080 <sub>16</sub> to 0083 <sub>16</sub> )	32	Timer B
UART2 transmission, NACK <sup>3</sup>	+132 to +135 (0084 <sub>16</sub> to 0087 <sub>16</sub> )	33	Serial I/O
UART2 reception, ACK <sup>3</sup>	+136 to +139 (0088 <sub>16</sub> to 008B <sub>16</sub> )	34	
UART3 transmission, NACK <sup>3</sup>	+140 to +143 (008C <sub>16</sub> to 008F <sub>16</sub> )	35	
UART3 reception, ACK <sup>3</sup>	+144 to +147 (0090 <sub>16</sub> to 0093 <sub>16</sub> )	36	
UART4 transmission/NACK <sup>3</sup>	+148 to +151 (0094 <sub>16</sub> to 0097 <sub>16</sub> )	37	
UART4 reception, ACK <sup>3</sup>	+152 to +155 (0098 <sub>16</sub> to 009B <sub>16</sub> )	38	

## Interrupts

Table 1.9.2. Relocatable Vector Tables (Continued)

Interrupt generated by	Vector table address Address(L)to address(H) <sup>1</sup>	Software interrupt number	Reference	
Bus conflict detect, start condition detect, stop condition detect, (UART2) <sup>5</sup> , fault error <sup>4</sup>	+156 to +159 (009C <sub>16</sub> to 009F <sub>16</sub> )	39	Serial I/O	
Bus conflict detect, start condition detect, stop condition detect, (UART3/UART0) <sup>5</sup> , fault error <sup>4</sup>	+160 to +163 (00A0 <sub>16</sub> to 00A3 <sub>16</sub> )	40		
Bus conflict detect, start condition detect, stop condition detect, (UART4/UART1) <sup>5</sup> , fault error <sup>4</sup>	+164 to +167 (00A4 <sub>16</sub> to 00A7 <sub>16</sub> )	41		
A-D0	+168 to +171 (00A8 <sub>16</sub> to 00AB <sub>16</sub> )	42	A-D converter	
Key input	+172 to +175 (00AC <sub>16</sub> to 00AF <sub>16</sub> )	43	Interrupts	
Intelligent I/O interrupt 0	+176 to +179 (00B0 <sub>16</sub> to 00B3 <sub>16</sub> )	44	Intelligent I/O CAN	
Intelligent I/O interrupt 0	+180 to +183 (00B4 <sub>16</sub> to 00B7 <sub>16</sub> )	45		
Intelligent I/O interrupt 0	+184 to +187 (00B8 <sub>16</sub> to 00BB <sub>16</sub> )	46		
Intelligent I/O interrupt 0	+188 to +191 (00BC <sub>16</sub> to 00BF <sub>16</sub> )	47		
Intelligent I/O interrupt 0	+192 to +195 (00C0 <sub>16</sub> to 00C3 <sub>16</sub> )	48		
Intelligent I/O interrupt 0	+196 to +199 (00C4 <sub>16</sub> to 00C7 <sub>16</sub> )	49		
Intelligent I/O interrupt 0	+200 to +203 (00C8 <sub>16</sub> to 00CB <sub>16</sub> )	50		
Intelligent I/O interrupt 0	+204 to +207 (00CC <sub>16</sub> to 00CF <sub>16</sub> )	51		
Intelligent I/O interrupt 0	+208 to +211 (00D0 <sub>16</sub> to 00D3 <sub>16</sub> )	52		
Intelligent I/O interrupt 9, CAN 0	+212 to +215 (00D4 <sub>16</sub> to 00D7 <sub>16</sub> )	53		
Intelligent I/O interrupt 10, CAN 1	+216 to +219 (00D8 <sub>16</sub> to 00DB <sub>16</sub> )	54		
Intelligent I/O interrupt 10, CAN 2	+228 to +231 (00E4 <sub>16</sub> to 00E7 <sub>16</sub> )	57		
INT instruction <sup>2</sup>	+0 to +3 (0000 <sub>16</sub> to 0003 <sub>16</sub> ) to +252 to +255 (00FC <sub>16</sub> to 00FF <sub>16</sub> )	0 to 63		Interrupts

## Notes :

1. This is a relative address to the one in the INTB register.
2. Interrupts are disabled by another way except by the I flag.
3. In IIC mode, NACK/ACK or start/stop condition detection causes an g to be generation.
4. When the  $\overline{SS}$  pin is selected, fault error causes an interrupt to be generated.
5. The IFSR6 bit in the IFSR register should be set to determine whether these addresses are used for an interrupt in UART0 or UART3.  
The IFSR7 bit in the IFSR register should be set to determine whether these addresses are used for an interrupt int UART1 or UART4.

## Interrupts

### Interrupt Request Reception

Software interrupts and special interrupt are generated when conditions to generate an interrupt are met. The peripheral g interrupt is generated by meeting all conditions below.

- I flag = "1"
- IR bit = "1"
- ILVL2 to ILVL0 bits > IPL

The I flag, IPL, IR bit and ILVL2 to ILVL0 bits are all independent of each other, and does not affect any other bits. The I flag and IPL are allocated in FLG register. The IR bit and ILVL2 to ILVL0 bits are in the interrupt control register.

#### (1) I Flag and IPL

The I flag makes a maskable interrupt disabled or enabled. When setting the I flag to "1" (enable), all maskable interrupts are enabled; when setting the I flag to "0" (disable), they are disabled. The I flag is automatically set to "0" after reset.

IPL is configured with three bits to determine the interrupt priority level from level 0 to level 7.

If a requested interrupt has higher priority than IPL has, an interrupt is enabled.

Table 1.9.3 lists interrupt enable levels associated with IPL.

**Table 1.9.4. Interrupt Priority Levels Field Encoding**

IPL2	IPL1	IPL0	Interrupt priority levels
0	0	0	level 1 and above are enabled
0	0	1	level 2 and above are enabled
0	1	0	level 3 and above are enabled
0	1	1	level 4 and above are enabled
1	0	0	level 5 and above are enabled
1	0	1	level 6 and above are enabled
1	1	0	level 7 and above are enabled
1	1	1	All maskable interrupts are disabled

#### (2) Interrupt Control Register and RLVL Register

The Peripheral function interrupts use interrupt control registers to control each interrupt. Figures 1.9.3 and 1.9.4 show the interrupt control register. Figure 1.9.5 shows the RLVL register.

Interrupts

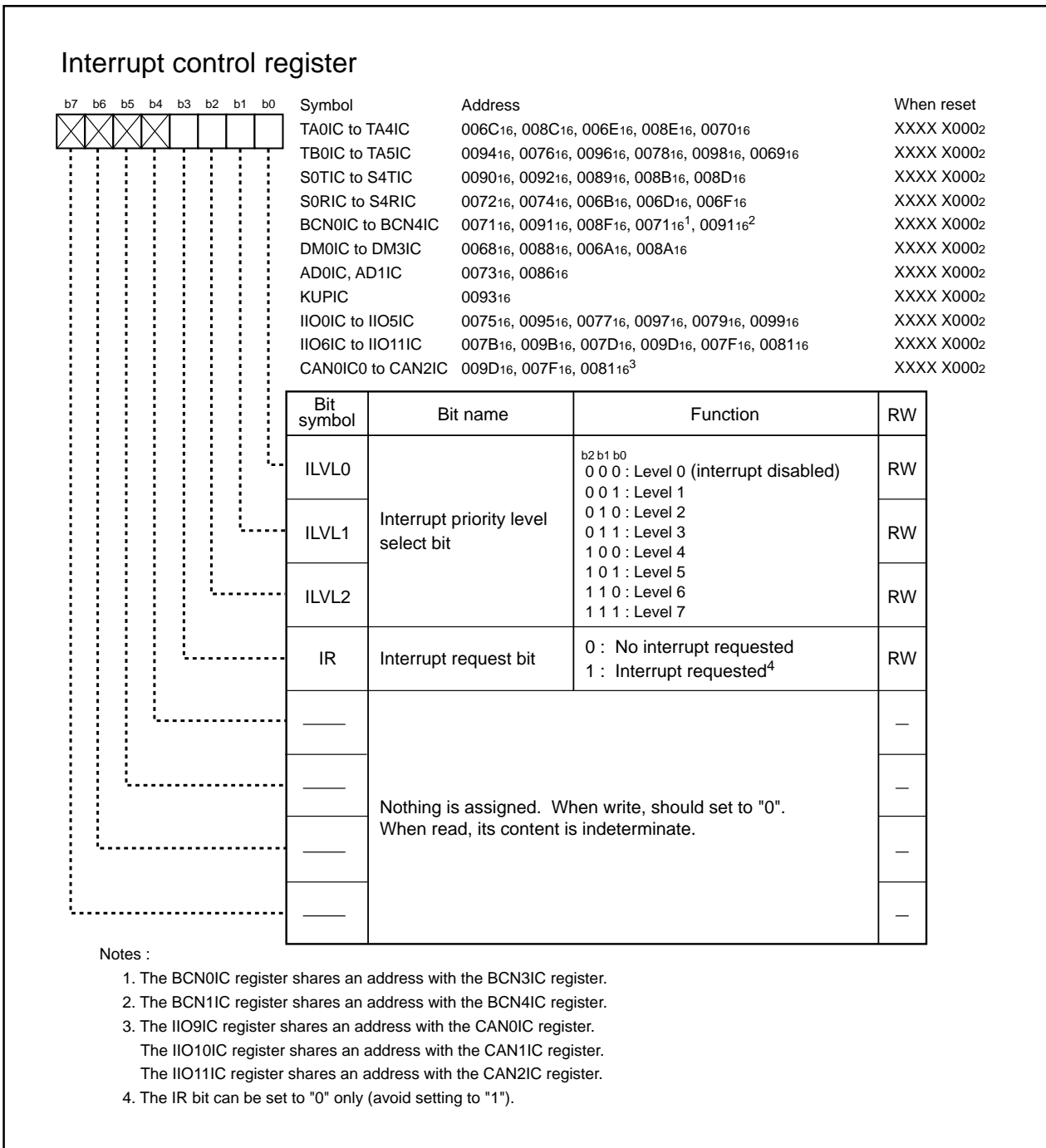


Figure 1.9.3. Interrupt Control Register (1)

Interrupts

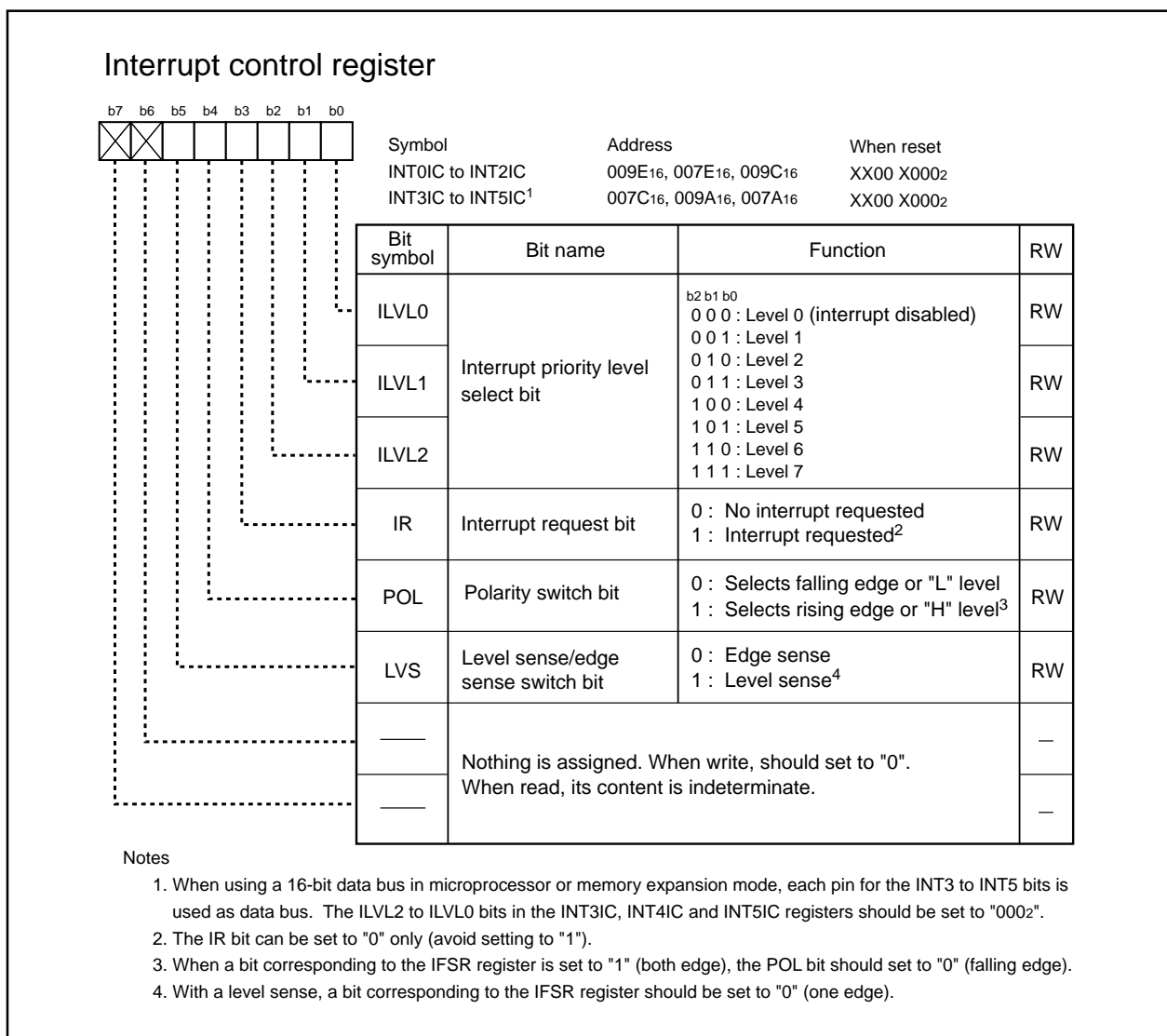


Figure 1.9.4. Interrupt Control Register (2)

• **ILVL2 to ILVL0 Bits**

The ILVL2 to ILVL0 bits determines the interrupt priority level. The greater interrupt priority level becomes, the higher interrupt priority level gets.

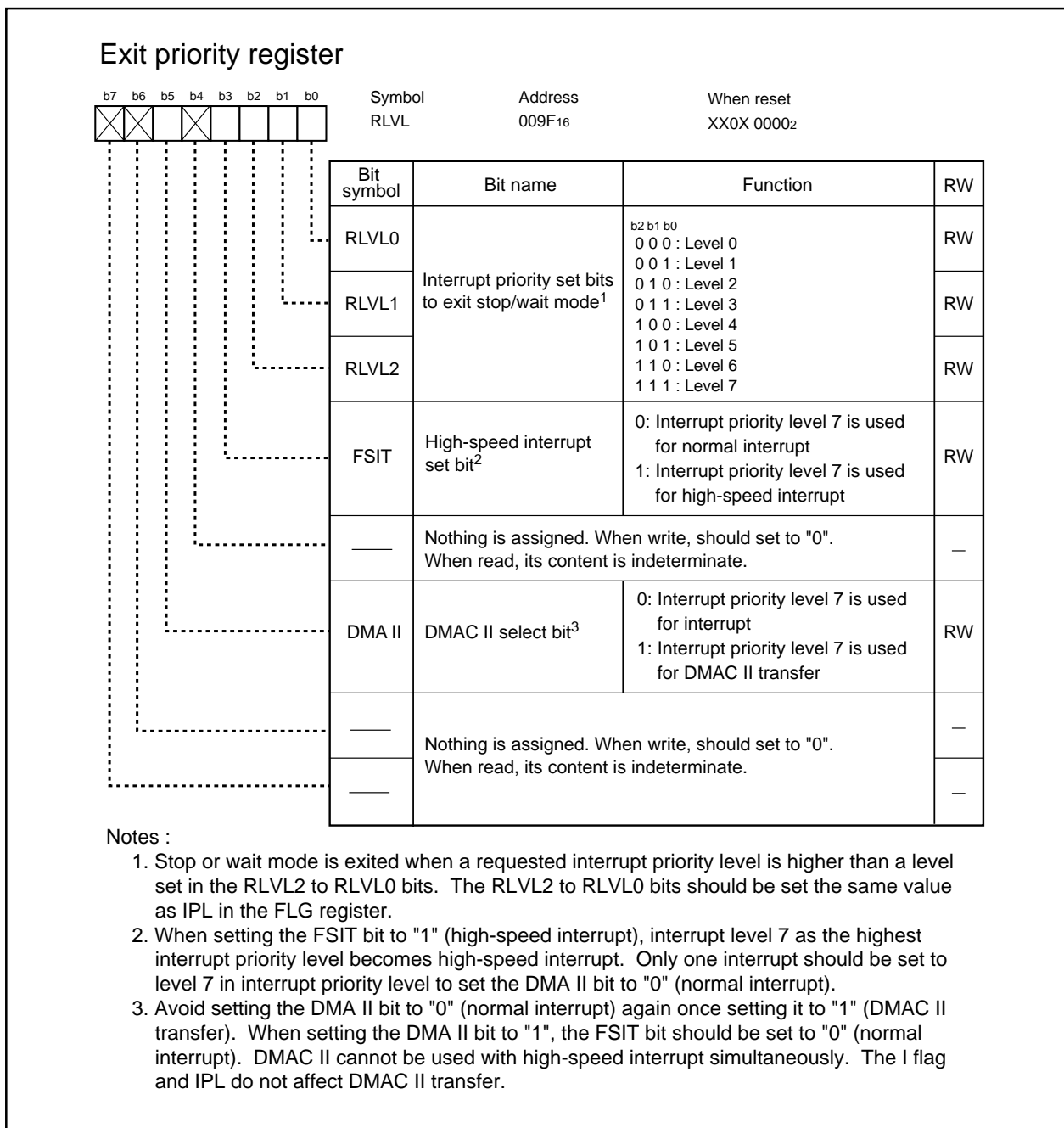
When an interrupt request is generated, its interrupt priority level is compared to IPL. This interrupt is enabled only when its interrupt priority level is greater than IPL. When setting the ILVL2 to ILVL0 bits to "00z" (level 0), its interrupt is disabled.

• **IR Bit**

The IR bit is set to "1" by hardware when an interrupt request is generated. The IR bit is set to "0" by hardware after an interrupt request is acknowledged and jumps to an interrupt vector.

The IR bit can be set to "0" by program (avoid setting to "1").

Interrupts



**Figure 1.9.5. RLVL Register**

• **RLVL2 to RLVL0 Bits**

When using an interrupt to exit stop or wait mode, the RLVL2 to RLVL0 bits should be set the same value as IPL in the FLG register before entering stop or wait mode. Interrupt priority level to exit stop or wait mode should be higher than the level set in the RLVL2 to RLVL0 bits.

### (3) Interrupt Sequence

Interrupt sequence is handled between an interrupt request acknowledgment and interrupt routine execution.

An interrupt request is generated while executing an instruction. Then the CPU determines its interrupt priority level after the instruction is completed. Regarding the SCMPU, SIN, SMOVB, SMOVF, SMOVU, SSTR, SOUT or RMPA instruction, the CPU suspends the instruction being executed to determine the interrupt priority level. The CPU starts handling the interrupt sequence from the next cycle after the CPU determination.

The interrupt sequence is handled as follows:

- (1) The CPU obtains interrupt information (interrupt number and interrupt request level) by reading address 000000<sub>16</sub> (address 000002<sub>16</sub> for the high-speed interrupt). After this, the IR bit associated with an interrupt is set to "0" (interrupt requested).
- (2) The FLG register, prior to an interrupt sequence handling, is saved to a temporary register<sup>1</sup> within the CPU.
- (3) Each bit in the FLG register is set to the followings:
  - The I flag is set to "0" (interrupt disabled)
  - The D flag is set to "0" (single-step disabled)
  - The U flag is set to "0" (ISP selected)
- (4) A temporary register within the CPU is saved to the stack space and to the SVF register for the high-speed interrupt.
- (5) Content of PC is saved to the stack space and to the SVP register for the high-speed interrupt.
- (6) An interrupt priority level of an acknowledged interrupt in IPL is set.
- (7) A relocatable vector corresponding to an acknowledged interrupt is stored into PC.

After the interrupt sequence handling is completed, the instruction is resumed executing from a starting address of an interrupt routine.

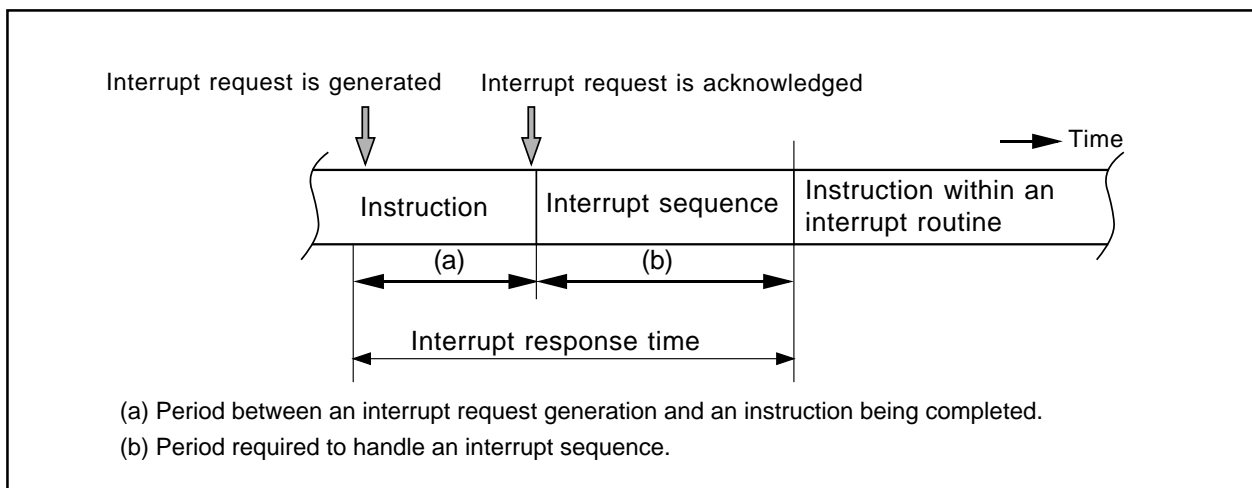
Notes :

1. Temporary register cannot be modified by users.

## Interrupts

**(4) Interrupt Response Time**

Figure 1.9.6 shows an interrupt response time. Interrupt response time is a period between an interrupt generation and an execution of the first instruction within an interrupt routine. An interrupt response time comprises a period from an interrupt request generation to an instruction completed ((a) on figure 1.9.6) and the period required to handle an interrupt sequence ((b) on Figure 1.9.6).



**Figure 1.9.6. Interrupt Response Time**

Time (a) varies depending on each instruction being executed. The DIV instruction requires the longest time for (a). It takes 40 cycles when an immediate value or register is set as a divisor .

When a 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 the number of a wait/waits set in a divisor space. Y is the number of wait/waits set in a space that stores indirect addresses. If X and Y are in an odd address or in 8-bit bus space, the X and Y value should be doubled.

Table 1.9.4 lists time (b).

## Interrupts

**Table 1.9.4 Handling Timer for Interrupt Sequence**

Interrupt	Interrupt vector address	16-bit bus	8-bit bus
Peripheral function	Even address	14 cycles	16 cycles
	Odd address <sup>1</sup>	16 cycles	16 cycles
INT instruction	Even address	12 cycles	14 cycles
	Odd address <sup>1</sup>	14 cycles	14 cycles
$\overline{\text{NMI}}$ Watchdog timer Undefined instruction Address match	Even address <sup>2</sup>	13 cycles	15 cycles
Overflow	Even address <sup>2</sup>	14 cycles	16 cycles
BRK instruction (variable vector table)	Even address	17 cycles	19 cycles
	Odd address <sup>1</sup>	19 cycles	19 cycles
BRK instruction (fixed vector table)	Even address <sup>2</sup>	19 cycles	21 cycles
High-speed interrupt	Vector table is internal register	5 cycles	

Notes :

1. Interrupt vectors should be allocated in even addresses.
2. Vectors are fixed to even addresses.

**(5) Changes of IPL When Interrupt Request is Acknowledged**

When an peripheral function request is acknowledged, the CPU sets the acknowledged interrupt priority level in IPL.

Software interrupt and special interrupt have no interrupt priority level. If acknowledging an interrupt request that has no interrupt priority level, a value shown in Table 1.9.5 is set in IPL as an interrupt priority level.

**Table 1.9.5 Interrupts without Interrupt Priority Levels and IPL**

Interrupt sources	Level that is set to IPL
Watchdog timer, $\overline{\text{NMI}}$ , Oscillation stop detect	7
Reset	0
Software, Address match	Not changed

**(6) Saving a Register**

In an interrupt sequence, the FLG register and PC are saved to the stack space.

The FLG register is saved to the stack space. Next, 16 high-order bits and 16 low-order bits of PC extended to 32 bits are saved. Figure 1.9.7 shows a stack status before and after an interrupt request is acknowledged.

Other required registers are saved by program at the beginning of an interrupt routine. The PUSHM instruction can save all registers, except SP, by single s.

Refer to the paragraph "High-speed interrupt" about the high-speed interrupt.

Interrupts

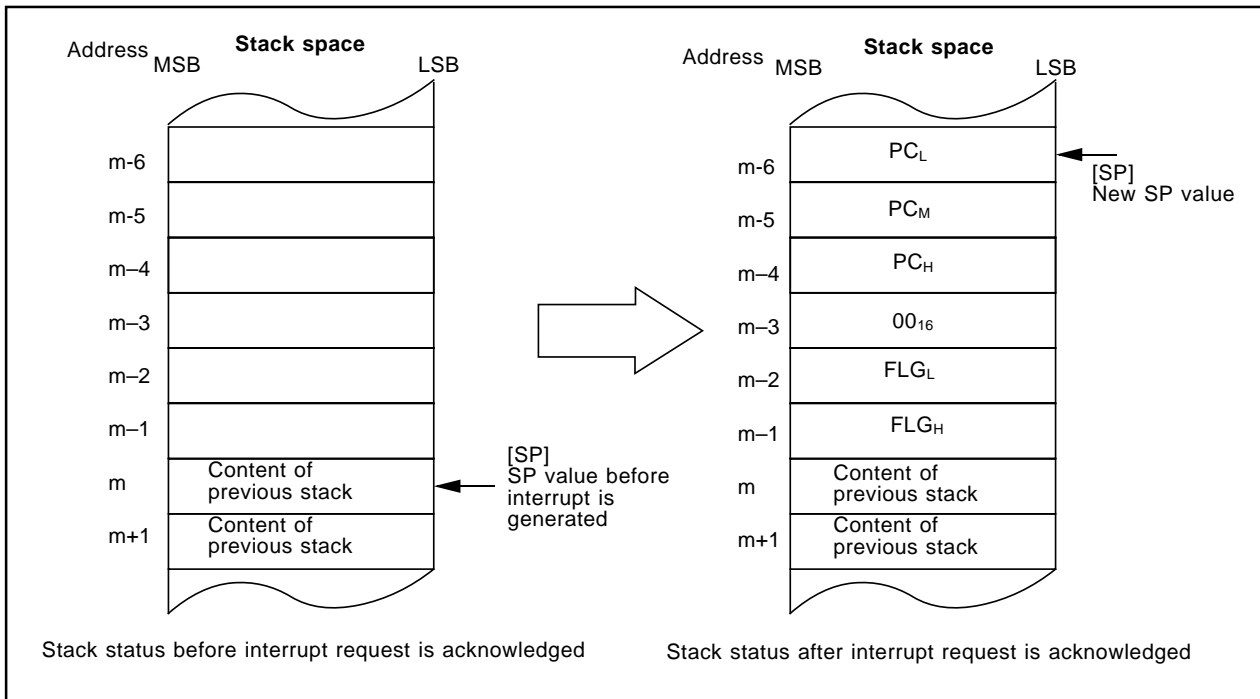


Figure 1.9.7. Stack Status

**(7) Return from Interrupt Routine**

When executing the REIT instruction in the end of an interrupt routine, the FLG register and PC which have been saved to the stack space are automatically restored. A program which is executed before an interrupt request is acknowledged, resumes to continue processing. Refer to the paragraph "High-speed interrupt" about the high-speed interrupt.

If a register is saved by program in an interrupt routine, it should be restored by the POPM instruction or others before executing the REIT and FREIT instructions. A register bank is switched to the prior to an interrupt sequence by the REIT or FREIT instruction.

**(8) Interrupt Priority**

If two or more interrupt requests are sampled at the same sampling points (in timing to detect whether an interrupt request is in or not), an interrupt with the highest priority is acknowledged.

The ILVL2 to ILVL0 bits determine a desired priority level for a maskable interrupt (peripheral function interrupt).

Special interrupts such as a reset (reset takes the highest priority) and watchdog timer interrupt have their priority levels set in hardware. Figure 1.9.8 shows priority levels of hardware interrupts.

The interrupt priority affects software interrupts. The microcomputer jumps to an interrupt routine when executing an instruction.

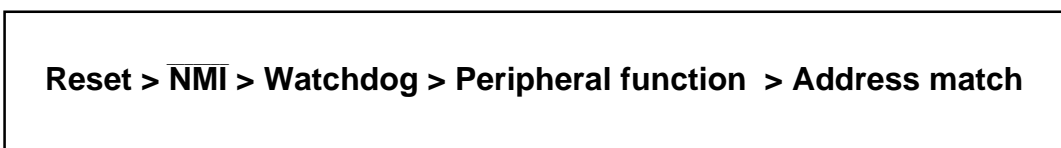


Figure 1.9.8. Interrupt Priority

## Interrupts

---

### **(9) Interrupt Priority Select Circuit**

The interrupt priority select circuit determines the highest priority interrupt when two or more interrupt requests are sampled at the same sampling point.

Figure 1.9.9 shows the interrupt priority select circuit.

Interrupts

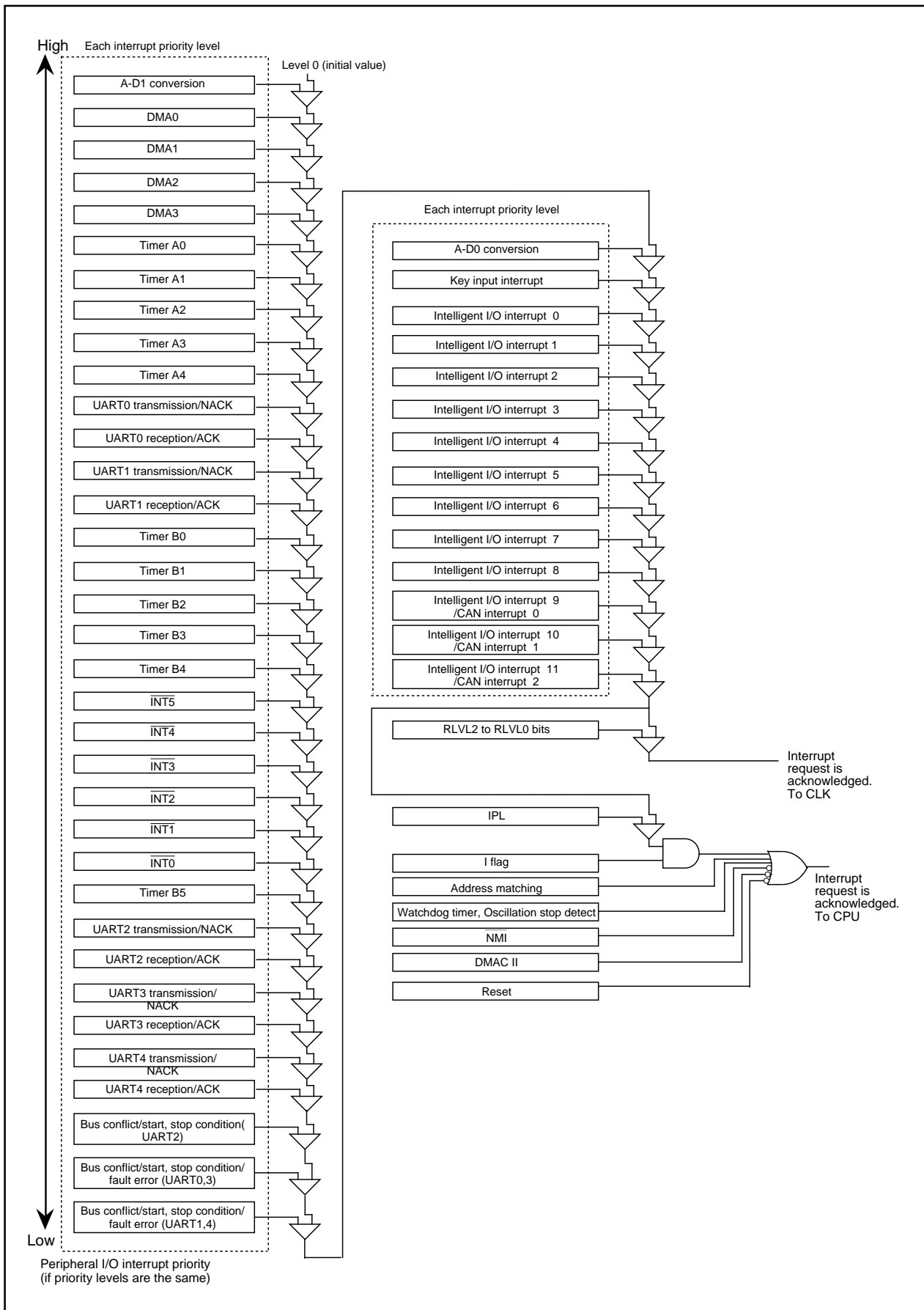


Figure 1.9.9. Interrupt Priority Select Circuit

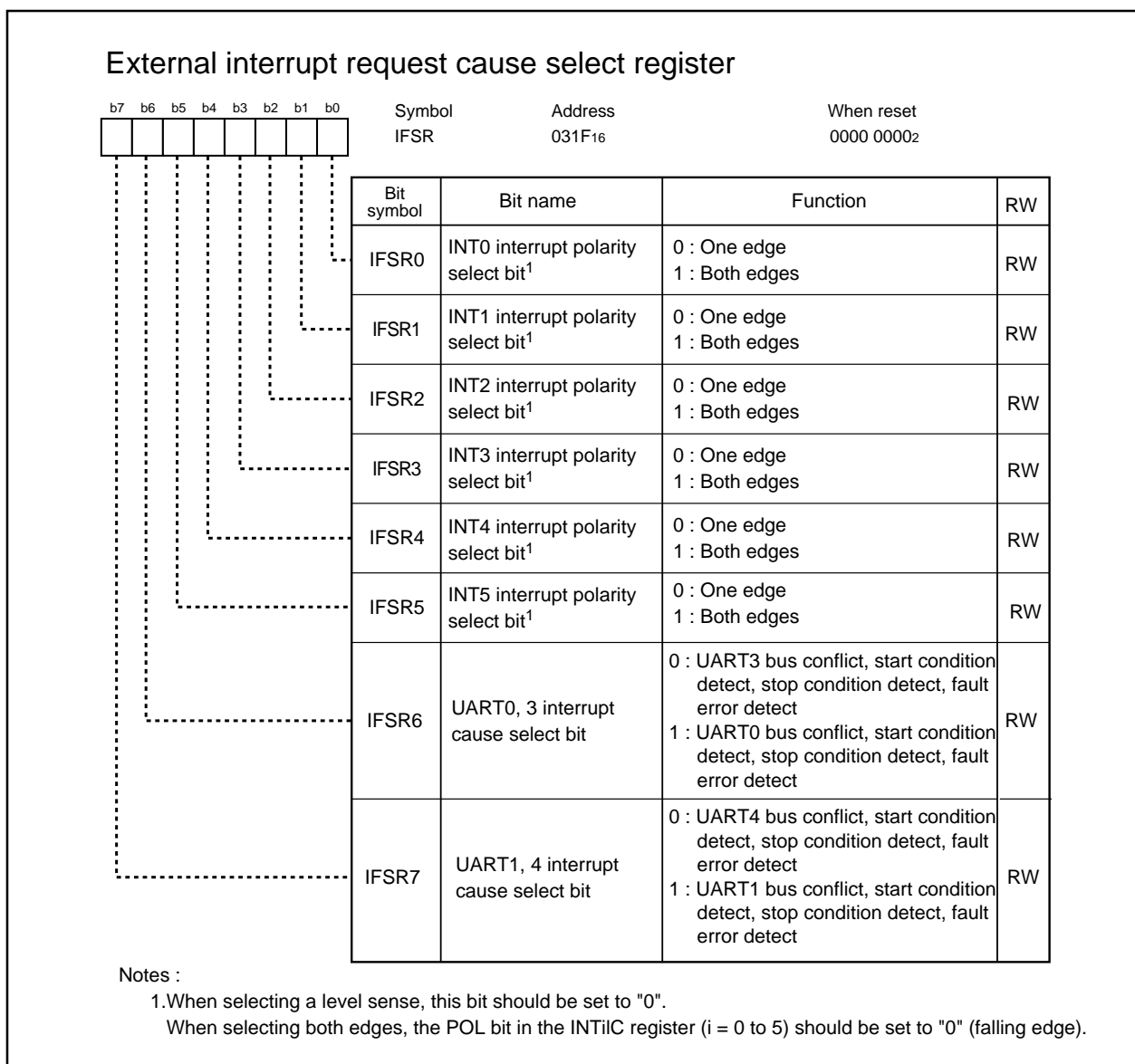
Interrupts

**INT Interrupt**

The  $\overline{INT}_i$  interrupts ( $i = 0$  to  $5$ ) are generated by an external input. The LVS bit in the  $INT_iIC$  register determines a level sense to generate an interrupt in an input signal level. The LVS bit also determines an edge sense to generate an interrupt at an edge. The POL bit in the  $INT_iIC$  register can determine polarity. With an edge sense, when the  $IFSR_i$  bit in the IFSR register is set to "1", an interrupt is generated on both rising and falling edges of an external input interrupt. If setting the  $IFSR_i$  bit to "1", the POL bit in a corresponding register should be set to "0" (falling edge).

With a level sense, the  $IFSR_i$  bit should be set to "0" (single edge). When the  $\overline{INT}_i$  pin input level reaches a level set in the POL bit, the IR bit in the  $INT_iIC$  register is set to "1". The IR bit remains unchanged in "1" even if the  $\overline{INT}_i$  pin is changed. The IR bit is set to "0" when the  $\overline{INT}_i$  interrupt is acknowledged or when the IR bit is written to "0" by program.

Figure 1.9.10 shows the IFSR register.



**Figure 1.9.10. IFSR Register**

## Interrupts

### NMI Interrupt

The  $\overline{\text{NMI}}$  interrupt is generated when input to the P85/ $\overline{\text{NMI}}$  pin changes "H" to "L". The  $\overline{\text{NMI}}$  interrupt is a non-maskable external interrupt. Even though the P85/ $\overline{\text{NMI}}$  pin is used as the  $\overline{\text{NMI}}$  interrupt input pin, the P8\_5 bit in the P85 register indicates input level for this pin.

Notes:

When the  $\overline{\text{NMI}}$  function is not used, the  $\overline{\text{NMI}}$  pin should be connected (or pulled up) to VCC via a resistance. The  $\overline{\text{NMI}}$  interrupt is non-maskable. Because it cannot be disabled, a pin must be pulled up.

### Key Input Interrupt

A key input interrupt request is generated when one of signals that are input to the P104 to P107 pins for input mode falls. The key input interrupt can be also used as key-on wakeup function to exit wait or stop mode. With the key input interrupt, avoid using P104 to P107 as A-D input ports. Figure 1.9.11 shows a block diagram of the key input interrupt. When any pins that the direction register enables to input are input "L", inputs to other pins are not detected as an interrupt.

When the PSC\_7 bit in the PSC register<sup>1</sup> is set to "1" (key input interrupt disabled), the key input interrupt is not generated regardless of settings in the interrupt control register. When setting the PSC\_7 bit to "0", no input from a port pin is available even if the direction register is set to be input.

Notes:

1. Refer to the section "Programmable I/O Ports" about the PSC register.

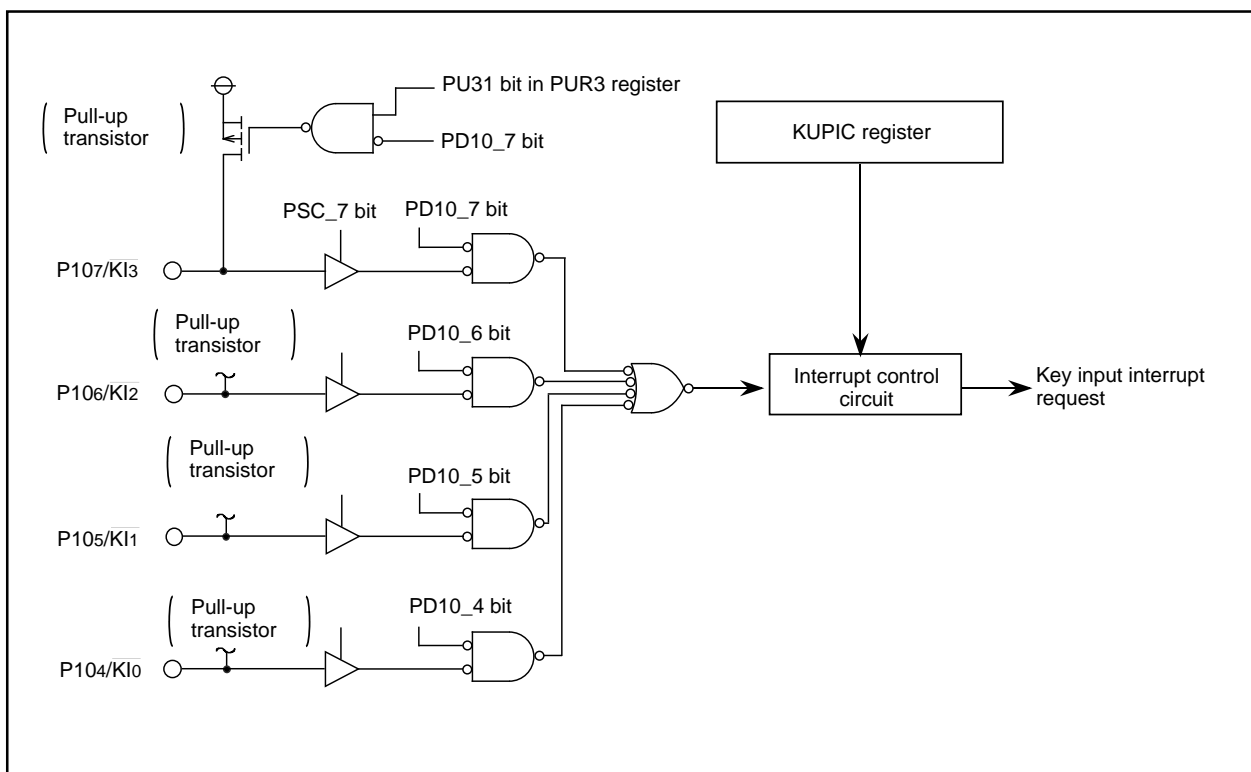


Figure 1.9.11. Key Input Interrupt

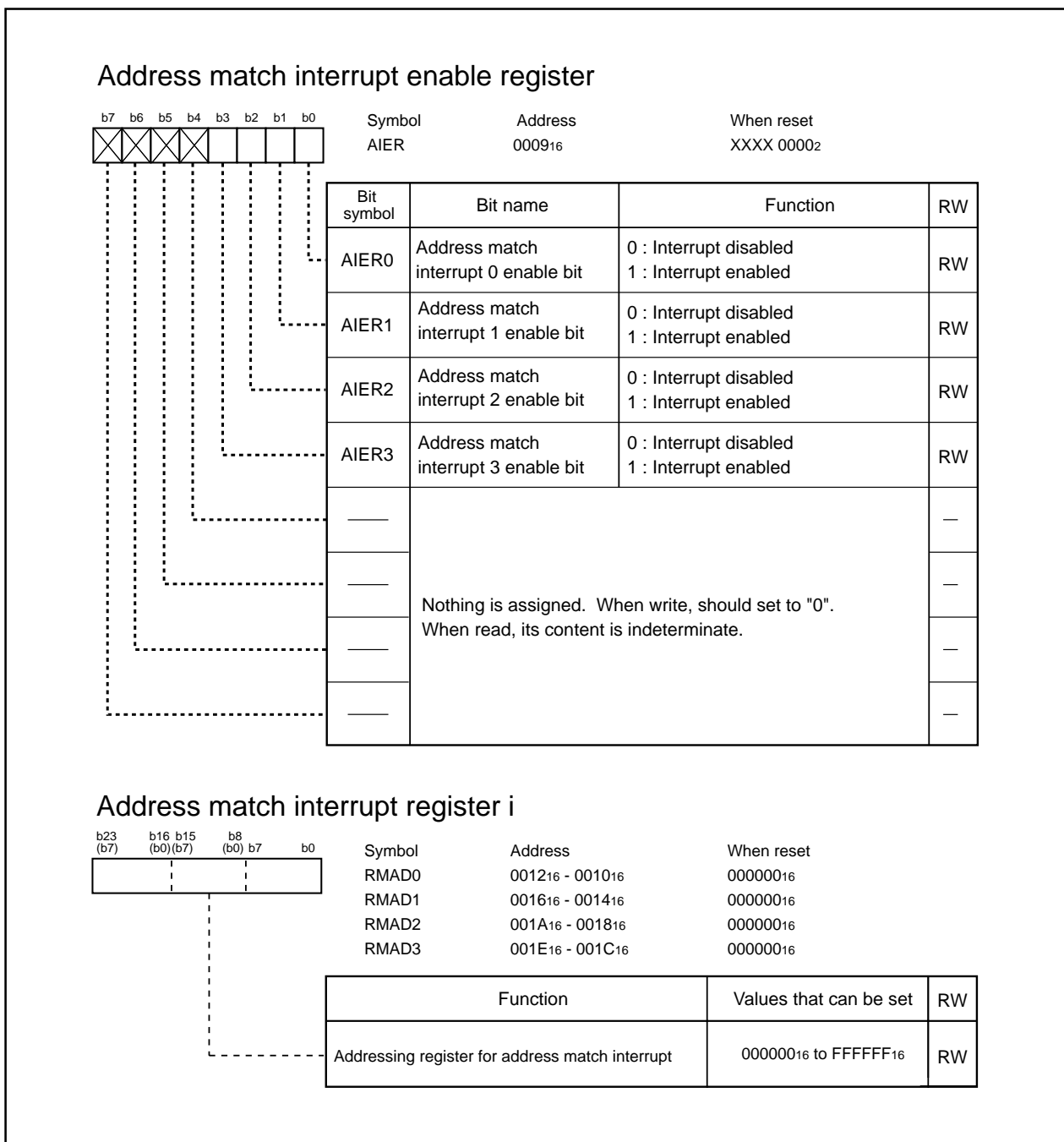
Interrupts

**Address Match Interrupt**

The address match interrupt is generated immediately before executing an instruction in address set by the RMADi register (i = 0 to 3). The address match interrupt can be set in four addresses. The AIERi bit in the AIER register can determine whether an interrupt is enabled or disabled. The I flag and IPL do not affect the address match interrupt.

Figure 1.9.12 shows registers associated with the address match interrupt.

A starting address of an instruction should be set in the RMADi register. The address match interrupt is not generated while an executing an instruction or when setting an address such for a table data.



**Figure 1.9.12. AIER Register and RMAD0 to RMAD3 Registers**

## Interrupts

## Intelligent I/O Interrupt and CAN Interrupt

The intelligent I/O interrupt and CAN interrupt are allocated to software interrupt numbers 44 to 54 and 57. Figure 1.9.13 shows a block diagram of the intelligent I/O interrupt and CAN interrupt. Figure 1.9.14 shows the IIOiIR register (i = 0 to 15). Figure 1.9.15 shows the IIOiIE register.

With the intelligent I/O interrupt or CAN interrupt, the IRLT bit in the IIOiIE register should be set to "1" (used interrupt request for interrupt).

Various interrupt requests generate the intelligent I/O interrupt. When an interrupt request is generated with each intelligent I/O function, a corresponding bit in the IIOiIR register is set to "1" (interrupt request). When a corresponding bit in the IIOiIE register is set to "1" (interrupt enabled), the IR bit in the corresponding IIOiIC register is set to "1" (interrupt request).

When a bit in the IIOiIR register is set to "1" by another interrupt request and a corresponding bit in the IIOiIE register is set to "1". The IR bit remains unchanged in "1" after setting the IR bit "0" to "1", .

No bit in the IIOiIR register is automatically set to "0" even if an interrupt is acknowledged. Each bit should be set to "0" by program. If leaving these bits remained in "1", all interrupt requests are disabled.

The CAN interrupt use bit 7 in the IIO9IR to IIO11IR registers and bit 7 in the IIO9IE to IIO11IE registers. The IIO9IC to IIO11IC registers share addresses with the CAN0IC to CAN2IC registers. Refer to the paragraph "CAN interrupt" in "CAN module" about the CAN interrupt.

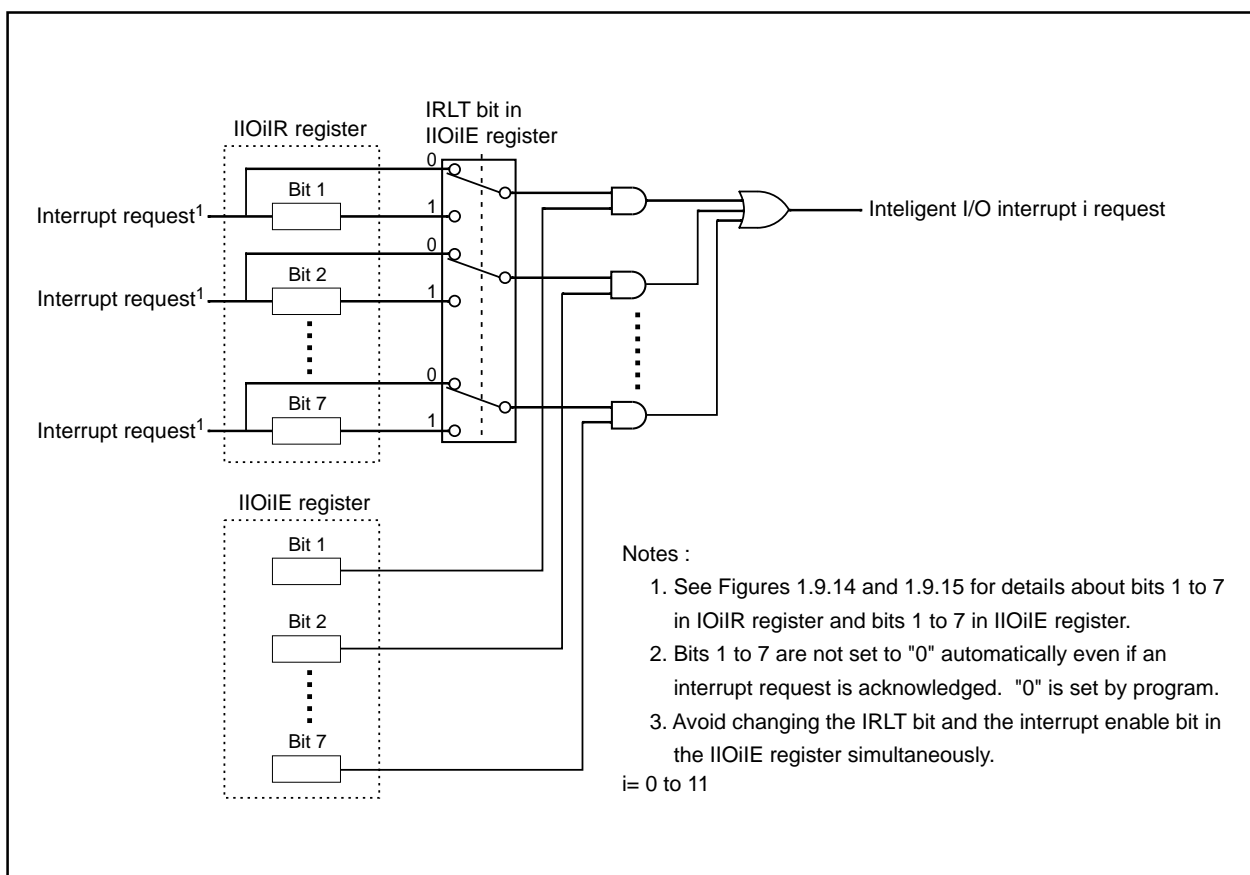


Figure 1.9.13. Intelligent I/O Interrupt and CAN Interrupt

With the intelligent I/O interrupt or CAN interrupt to activate DMA II, the IRLT bit in the IIOiIE register should be set to "0" (used interrupt for DMAC, DMAC II) to enable an interrupt request that the IIOiIE register uses.

Interrupts

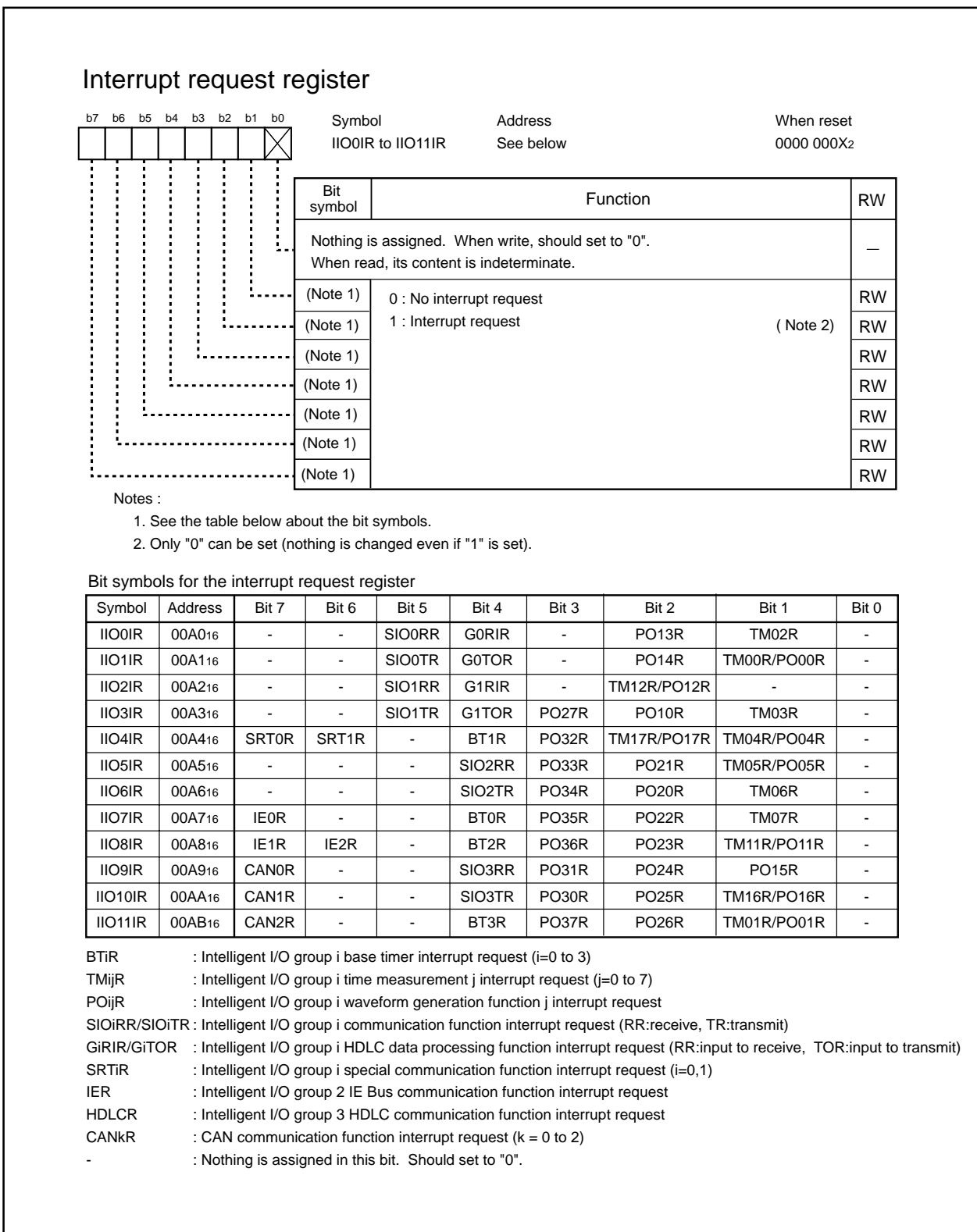
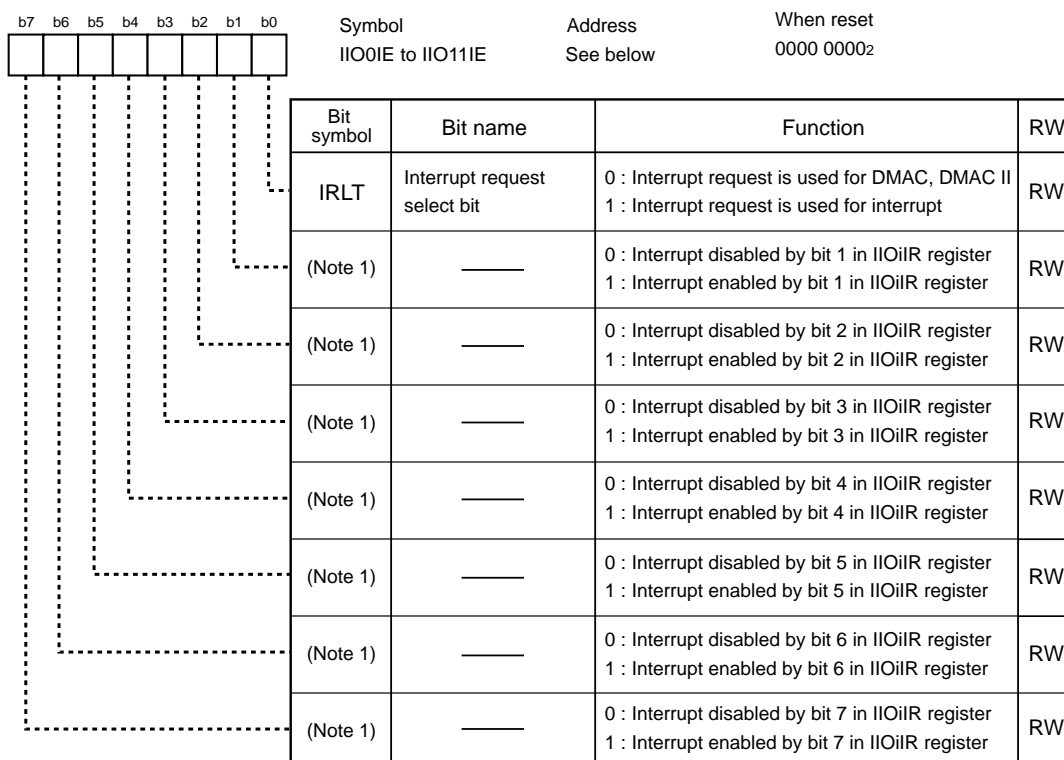


Figure 1.9.14. IIO0IR to IIO11IR Registers

Interrupts

Interrupt enable register



Notes :

1. See the table below about the bit symbols.

Bit symbols for the interrupt enable register

Symbol	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IIO0IE	00B0 <sub>16</sub>	-	-	SIO0RE	G0RIE	-	PO13E	TM02E	IRLT
IIO1IE	00B1 <sub>16</sub>	-	-	SIO0TE	G0TOE	-	PO14E	TM00E/PO00E	IRLT
IIO2IE	00B2 <sub>16</sub>	-	-	SIO1RE	G1RIE	-	TM12E/PO12E	-	IRLT
IIO3IE	00B3 <sub>16</sub>	-	-	SIO1TE	G1TOE	PO27E	PO10E	TM03E	IRLT
IIO4IE	00B4 <sub>16</sub>	SRT0E	SRT1E	-	BT1E	PO32E	TM17E/PO17E	TM04E/PO04E	IRLT
IIO5IE	00B5 <sub>16</sub>	-	-	-	SIO2RE	PO33E	PO21E	TM05E/PO05E	IRLT
IIO6IE	00B6 <sub>16</sub>	-	-	-	SIO2TE	PO34E	PO20E	TM06E	IRLT
IIO7IE	00B7 <sub>16</sub>	IE0E	-	-	BT0E	PO35E	PO22E	TM07E	IRLT
IIO8IE	00B8 <sub>16</sub>	IE1E	IE2E	-	BT2E	PO36E	PO23E	TM11E/PO11E	IRLT
IIO9IE	00B9 <sub>16</sub>	CAN0E	-	-	SIO3RE	PO31E	PO24E	PO15E	IRLT
IIO10IE	00BA <sub>16</sub>	CAN1E	-	-	SIO3TE	PO30E	PO25E	TM16E/PO16E	IRLT
IIO11IE	00BB <sub>16</sub>	CAN2E	-	-	BT3E	PO37E	PO26E	TM01E/PO01E	IRLT

- BTiE : Intelligent I/O group i base timer interrupt request (i=0 to 3)
- TMijE : Intelligent I/O group i time measurement j interrupt request (j=0 to 7)
- POijE : Intelligent I/O group i waveform generation function j interrupt request
- SIOiRE/SIOiTE : Intelligent I/O group i communication function interrupt request (RR:receive, TR:transmit)
- GiRIE/GiTOE : Intelligent I/O group i HDLC data processing function interrupt request (RR:input to receive, TOR:input to transmit)
- SRTiE : Intelligent I/O group i special communication function interrupt request (i=0,1)
- IEE : Intelligent I/O group 2 IE Bus communication function interrupt request
- HDLCE : Intelligent I/O group 3 HDLC communication function interrupt request
- CANkE : CAN communication function interrupt request (k = 0 to 2)
- : Nothing is assigned in this bit. Should set to "0".

Figure 1.9.15. IIO0IE to IIO11IE Registers

## Precautions for Interrupts

### (1) SP Setting

- SP is reset to "00000016" after reset. The microcomputer runs out of control if an interrupt is acknowledged before setting SP. SP should be set before an interrupt is acknowledged. With the  $\overline{\text{NMI}}$  interrupt, SP should be reset at the beginning of program. All interrupts including the  $\overline{\text{NMI}}$  interrupt are acknowledged when executing the first instruction after reset. An even number should be set in SP to increase an operating rate for interrupt sequence.

### (2) $\overline{\text{NMI}}$ Interrupt

- The  $\overline{\text{NMI}}$  interrupt cannot be obstructed. The  $\overline{\text{NMI}}$  pin should be connected (pulled up) to VCC via a resistor if not used.
- A  $\overline{\text{NMI}}$  pin value can be read by the P8\_5 bit in the P8 register. This bit should be read only when identifying pin levels after the  $\overline{\text{NMI}}$  interrupt is generated.
- At least two CPU clock cycles + 300ns should be input as "L" width to the  $\overline{\text{NMI}}$  pin.

### (3) $\overline{\text{INT}}$ Interrupt

- Edge sense

At least 250ns should be input as "L" or "H" width to the  $\overline{\text{INT}}_i$  pins ( $i = 0$  to 5) regardless of the CPU clock.

- Level sense

At least one CPU clock cycle + 200ns should be input as "L" or "H" width to the  $\overline{\text{INT}}_i$  pins. (At least 234ns when  $X_{IN} = 30\text{MHz}$  and no division.)

- When a polarity of the  $\overline{\text{INT}}_i$  pins is switched, the IR bit in the INTiIC register may be set to "1". The IR bit should be set to "0" (no interrupt request) after switching.

Figure 1.9.16 shows a procedure to switch an  $\overline{\text{INT}}$  interrupt requests.

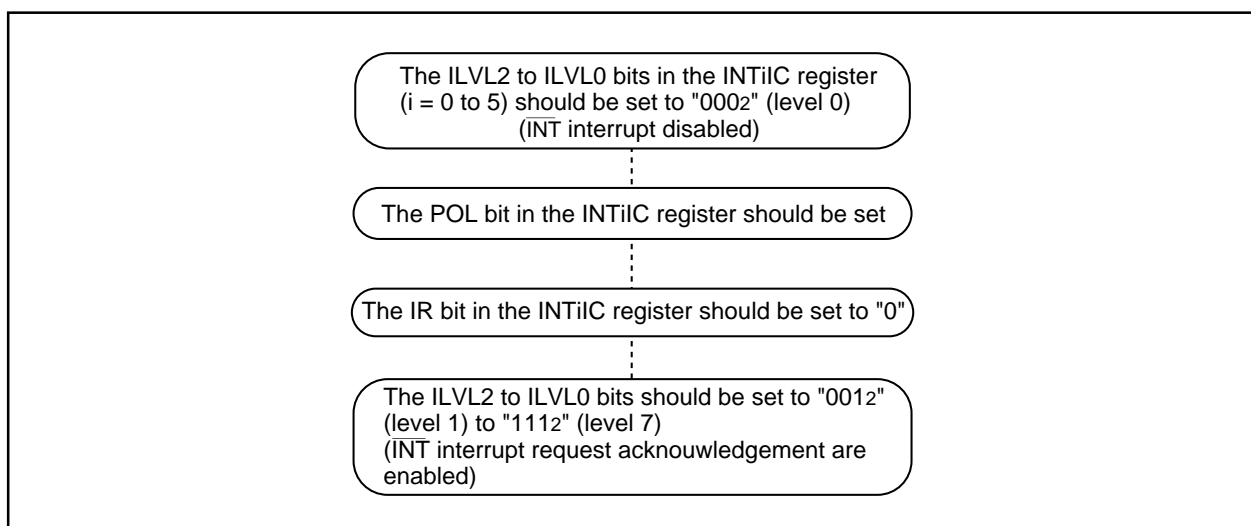


Figure 1.9.16. Switching Procedure for  $\overline{\text{INT}}$  Interrupt request

## Interrupts

---

### 4. Changing Interrupt Control Register

The below procedure should be taken when changing the interrupt control register under an interrupt-inhibited condition.

#### (1) Bits Except Interrupt Request Bit

An interrupt may be disabled to leave the IR bit unchanged in "0" when a corresponding interrupt is generated during an gg executing. If that is a problem, the below instructions should be used to change the register.

AND, OR, BCLR, BSET

#### (2) Setting Interrupt Request Bit

When setting the IR bit to "0" (no interrupt request), the IR bit may remain unchanged in "1", depending on an used instruction. If that is a problem, the below instruction should be used to change the register.

MOV

### 5. Changing IIOiR Register (i = 0 to 15)

When bits 1 to 7 in the IIOiR register are set to "0" (no interrupt request), the below instructions should be used to change the register.

AND, BCLR

## Watchdog Timer

# Watchdog Timer

The watchdog timer detects that a program is out of control. The watchdog timer contains a 15-bit counter, which is decremented by the CPU clock that a prescaler divides. The CM06 bit in the CM0 register determines a watchdog timer interrupt request or reset is generated if an underflow occurs in the watchdog timer. No other value than "1" (reset) is written into the CM06 bit. Once the CM06 bit is set to "1", it cannot be changed to "0" (the watchdog timer interrupt) by program. The CM06 bit is set to "0" only after reset. When the main clock or ring oscillator clock runs as the CPU clock, the WDC7 bit in the WDC register can determine whether a prescaler is divided either by 16 or by 128. When the sub clock runs as the CPU clock, a prescaler is divided by 2 regardless of the WDC7 bit. Watchdog timer cycle is calculated as follows. Margins of error, due to a prescaler, can arise in a watchdog timer cycle.

When the main clock or ring oscillator clock is selected as the CPU clock

$$\text{Watchdog timer cycle} = \frac{\text{Prescaler divided by 16 or 128} \times \text{counter value of watchdog timer (32768)}}{\text{CPU clock}}$$

When the sub clock is selected as the CPU clock

$$\text{Watchdog timer cycle} = \frac{\text{Prescaler divided by 2} \times \text{counter value of watchdog timer (32768)}}{\text{CPU clock}}$$

For example, with a 30MHz CPU clock and a prescaler divided by 16, watchdog timer cycle is approximate 17.5 ms.

The watchdog timer is reset when the WDTS register is set and a watchdog timer interrupt request is generated. A prescaler is reset only when the microcomputer is reset. Both watchdog timer and prescaler are stopped after reset. They start counting by writing to the WDTS register.

The watchdog timer and prescaler stop in stop mode, wait mode and a hold status. They resume counting from a value held when the modes or state are exited.

Figure 1.10.1 shows a block diagram of the watchdog timer. Figures 1.10.2 and 1.10.3 show registers associated with the watchdog timer.

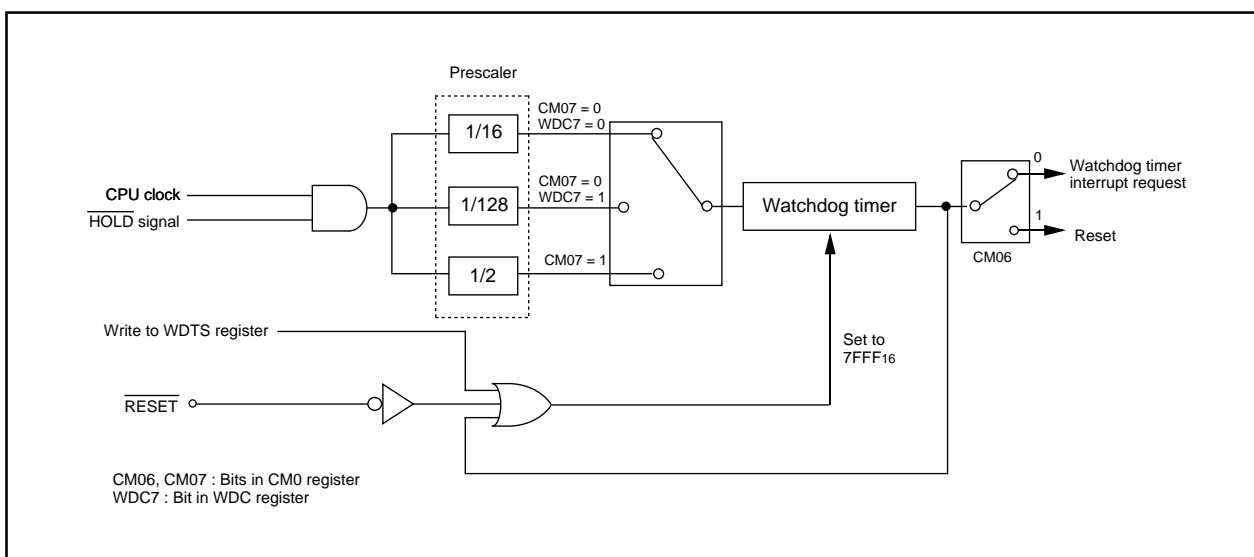


Figure 1.10.1. Watchdog Timer Block Diagram

## Watchdog Timer

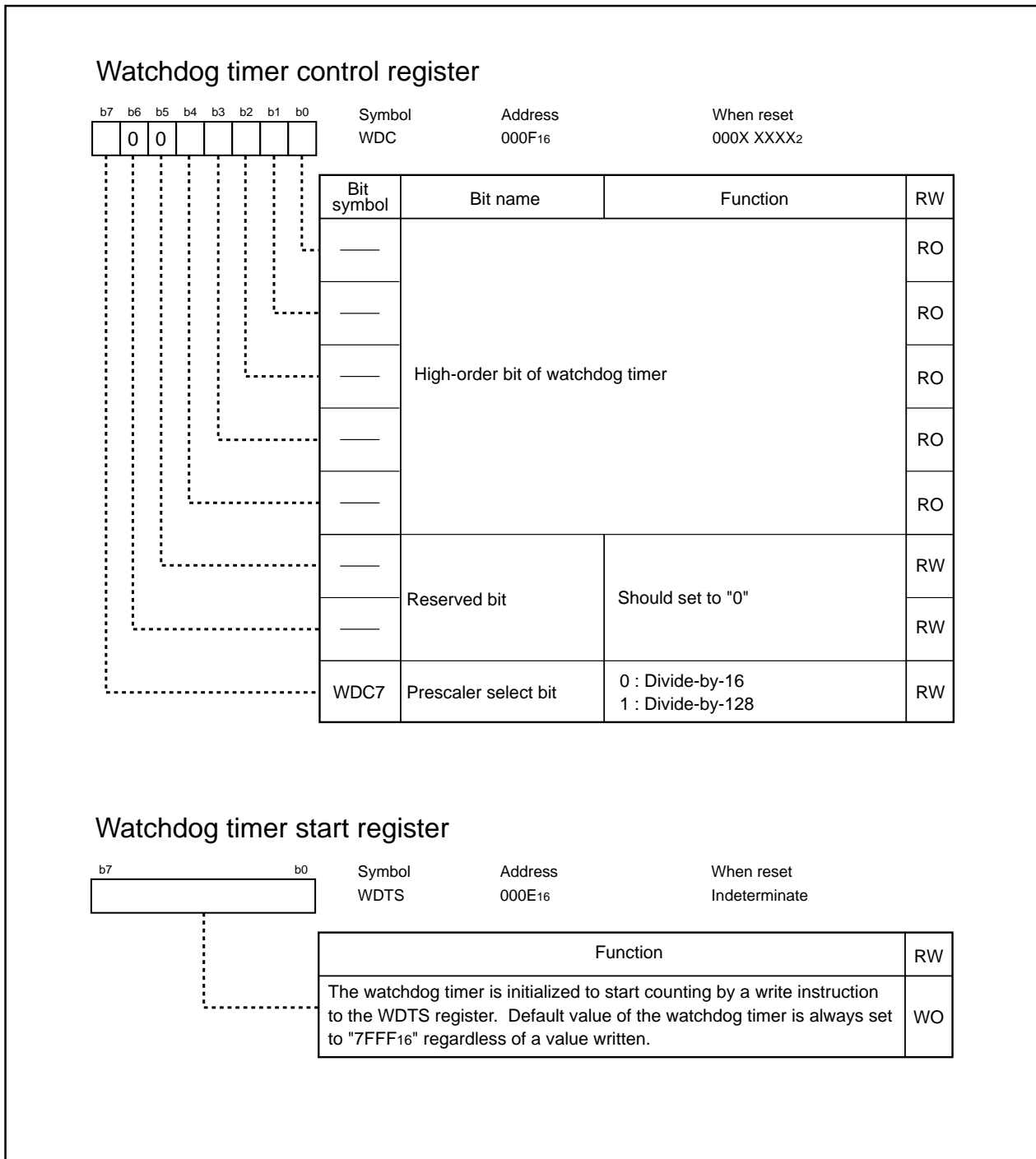
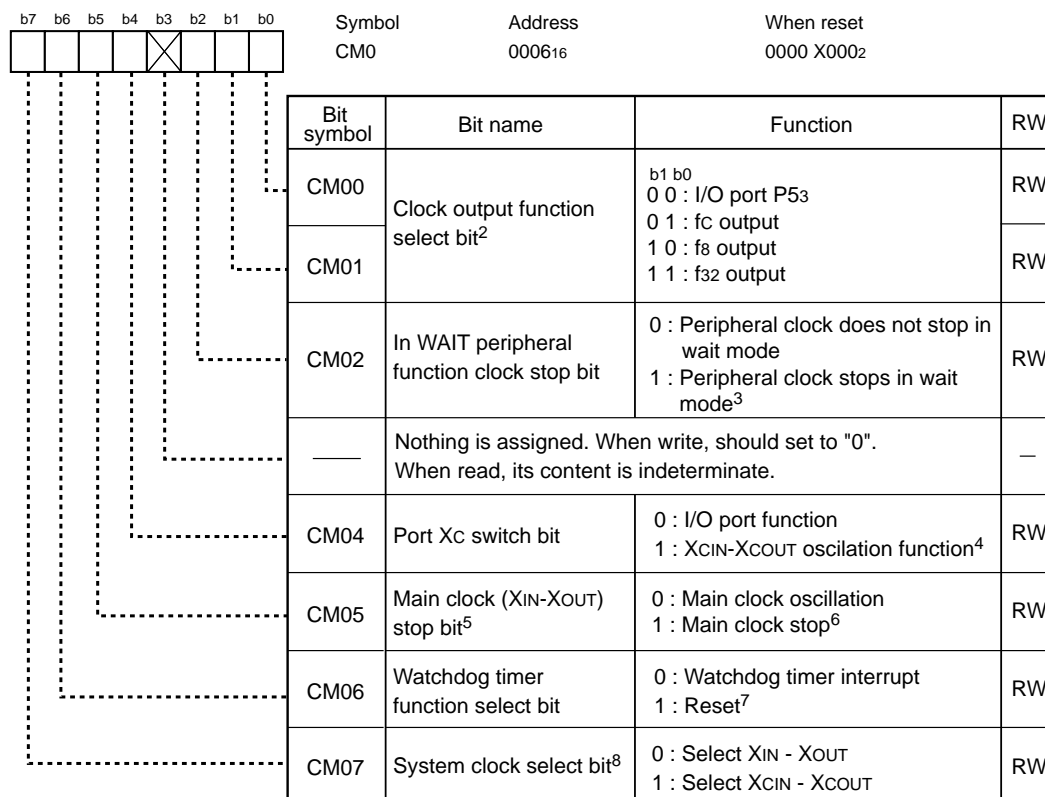


Figure 1.10.2. WDC Register and WDTS Register

Watchdog Timer

System clock control register 0<sup>1</sup>



Notes :

1. The PRC0 bit in the PRCR register should be set to "1" (write enable) before rewriting to the CM0 register.
2. When the PM07 bit in the PM0 register is set to "0" (BCLK output), the CM01 to CM00 bits should be set to "002". When the PM15 to PM14 bits in the PM1 register is set to "012" (ALE output to P5<sub>3</sub>), the CM01 to CM00 bits should be set to "002". When the PM07 bit is set to "1" (function selected in the CM01 to CM00 bits) in microprocessor or memory expansion mode, the CM01 to CM00 bits should be set to "002" to output "L" from port P5<sub>3</sub> (port P5<sub>3</sub> does not function as an I/O port).
3. fc<sub>32</sub> is not stopped. When setting the CM02 bit to "1", the PLL clock cannot be used in wait mode.
4. When setting the CM04 bit to "1" (XCIN-XCOUT oscillation), the PD8\_7 to PD8\_6 bits should be set to "002" (with port P8<sub>7</sub> and P8<sub>6</sub> input mode) and the PU25 bit in the PUR2 register be set to "0" (no pull-up).
5. To enter low-power consumption mode or ring oscillator low power consumption mode, the CM05 bit stops the main clock. The CM05 bit cannot detect whether the main clock stops or not. To stop the main clock, the CM05 bit should be set to "1" after the CM07 bit is set to "1" with a stable sub clock oscillation or after the CM21 bit in the CM2 register is set to "1" (ring oscillator clock). When setting the CM05 bit to "1" (main clock stop), XOUT is set to "H". Also, an internal feedback resistance remains ON. XIN is pulled up to XOUT ("H" level) via feedback resistance.
6. When setting the CM05 bit to "1", the MCD4 to MCD0 bits in the MCD register are set to "010002" (divide-by-8 mode). In ring oscillation mode, the MCD register is not divided by eight even if XIN-Xout is terminated by the CM05 bit.
7. Once the CM06 bit is set to "1", it cannot be set "0" by program.
8. After setting the CM04 bit to "1" with a stable sub clock oscillation, the CM07 bit should be changed "0" to "1".  
After setting the CM05 bit to "0" with a stable main clock oscillation, the CM07 bit should be changed "1" to "0".  
Avoid setting the CM07 bit and CM04 or CM05 bits simultaneously.

Figure 1.10.3. CM0 Register

# DMAC

This microcomputer contains four DMAC (direct memory access controller) channels that allow data to be sent to memory without using the CPU. DMAC transmits a 8- or 16-bit data of a source address to a destination address whenever a transmit request occurs. DMA 0 and DMA1 should have priority to be used if using DMAC. The same registers are simultaneously used in the high-speed interrupt and in DMA2 and in DMA3. In the high-speed interrupt, neither DMA2 nor DMA3 is available. DMA0 and DMA1 are available. The CPU and DMAC use the same data bus, but DMAC has a higher bus access privilege than the CPU. Cycle-steal method employed on DMAC enables high-speed operating between an occurrence of a transfer request and a complete transmission of 16-bit (word) or 8-bit (byte) data. Figure 1.11.1 shows a mapping of registers used for DMAC. Table 1.11.1 lists specifications of DMAC. Figures 1.11.2 to 1.11.5 show registers associated with DMAC.

As the registers shown in Figure 1.11.1 are allocated in the CPU, the LDC instruction to write should be used. To set DCT2, DCT3, DRC2, DRC3, DMA2 and DMA3 registers, the B flag should be set to "1" (register bank 1) and R0 to R3, A0, A1 registers should be set with the MOV instruction.

To set DSA2, DSA3, DRA2 and DRA3 registers, the B flag should be set to "1" and the SB, FB, SVP, VCT registers should be set with the LDC instruction.

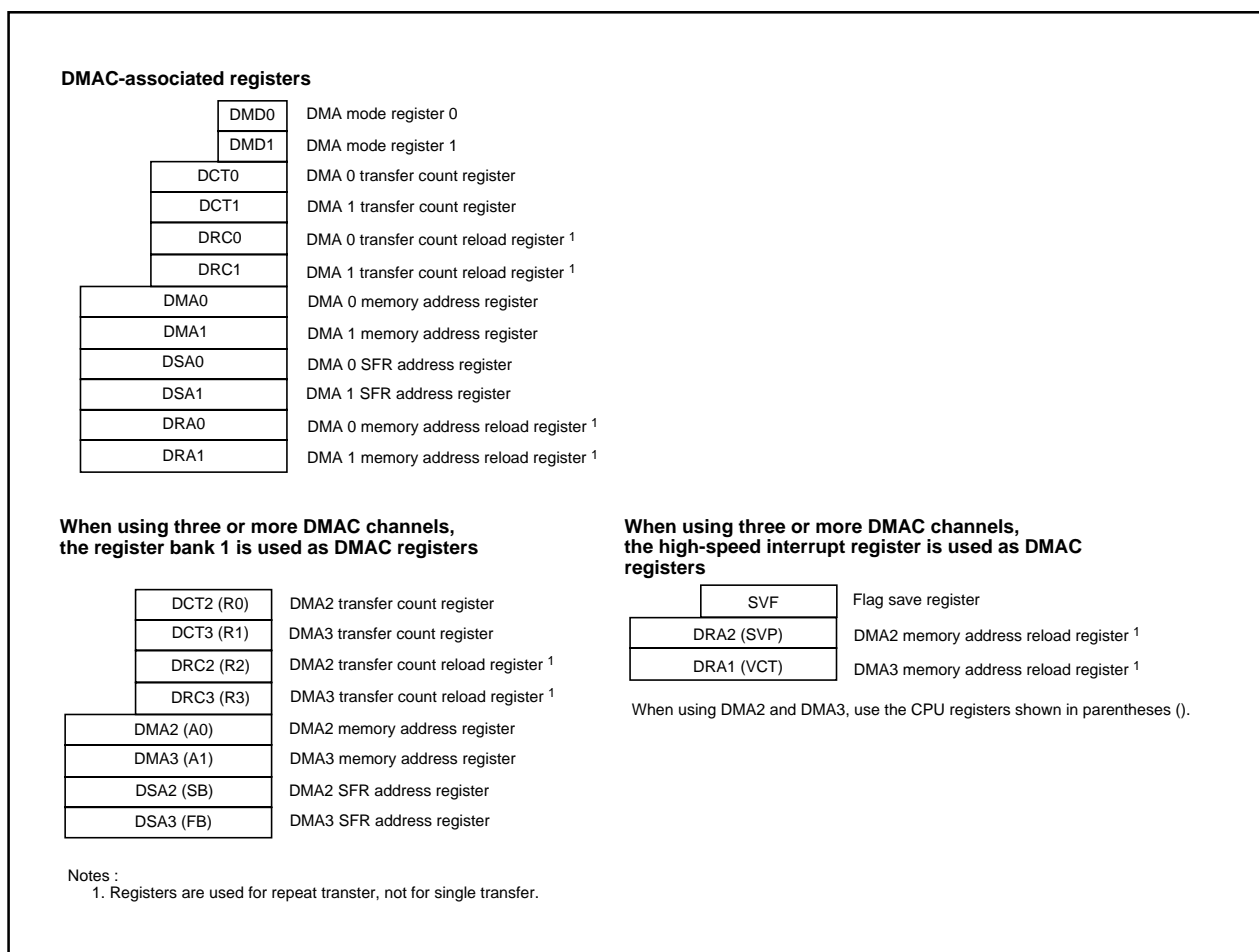


Figure 1.11.1. Register Mapping for DMAC

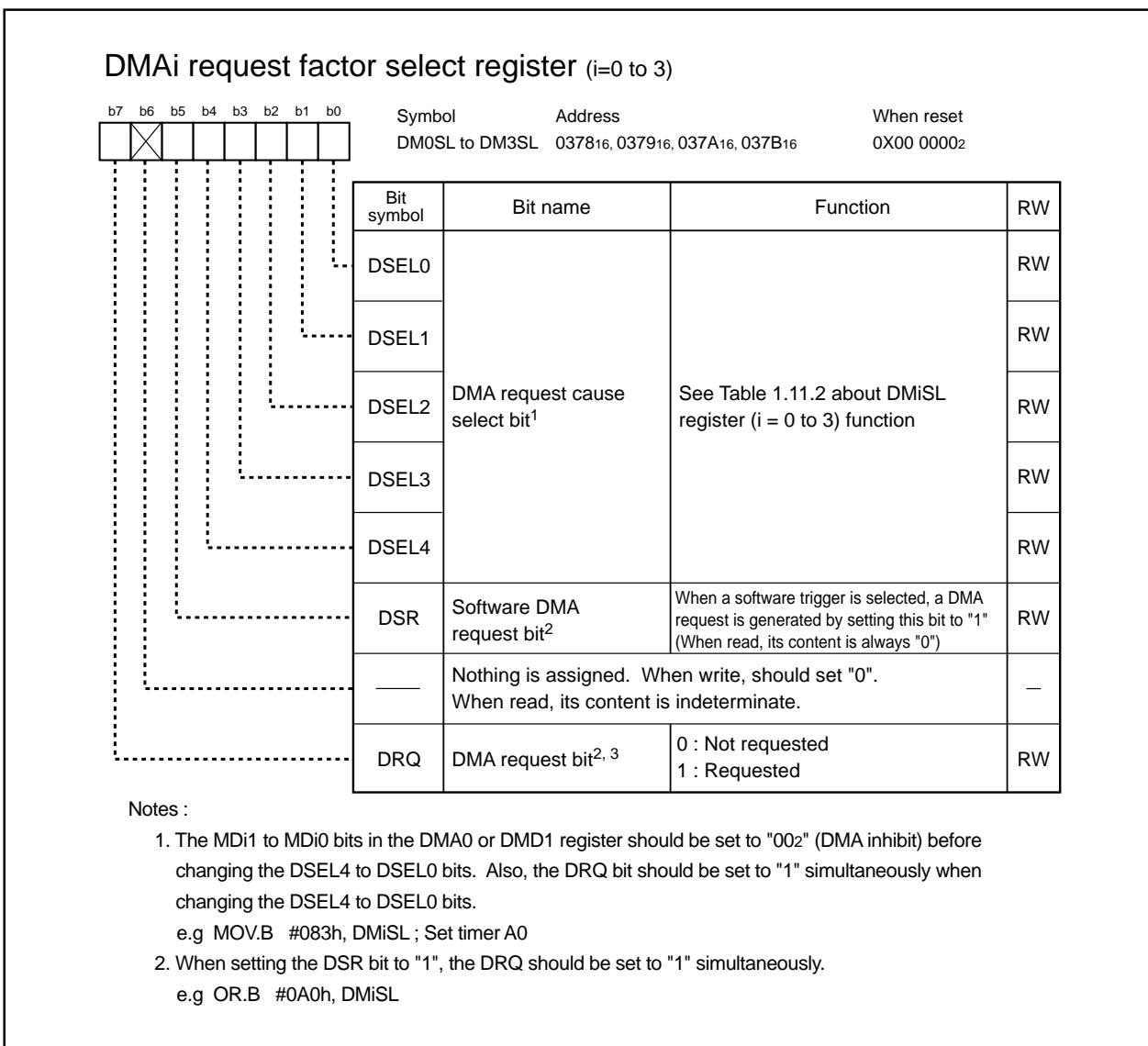
To start a DMAC transfer, in addition to a write into the DSR bits in the DMiSL register (i=0 to 3), interrupt request signals that are output from each functions specified in the DSEL 4 to DSEL0 bits in the DMiSL register are used DMA request. In contrast to an interrupt request, the I flag and interrupt control register do not affect DMA. Therefore a DMA request can be acknowledged even if an interrupt request is rejected or interrupt is restricted. The IR bit in the interrupt control register is never modified in a DMA transfer since no interrupt is affected by DMAC.

**Table 1.11.1. DMAC Specifications**

Item		Specification
Channels		4 channels (cycle-steal method)
Transfer memory space		<ul style="list-style-type: none"> <li>From any address in a 16M-byte space to a fixed address in a 16M-bytes space</li> <li>From a fixed address in a 16M-byte space to any address in a 16M-bytes space</li> </ul>
Maximum bytes transferred		128K-byte (with 16-bit transfer) or 64K-byte (with 8-bit transfer)
DMA request factors <sup>1</sup>		Falling edge or both edge of inputs to INT0 to INT3 pins Timer A0 to timer A4 interrupt requests Timer B0 to timer B5 interrupt requests UART0 to UART4 transmit and receive interrupt requests A-D0 and A-D1 conversion interrupt requests Intelligent I/O interrupt requests Software trigger
Channel priority		DMA0 > DMA1 > DMA2 > DMA3 (DMA0 has the higher priority)
Transfer unit		8 bits, 16 bits
Transfer address direction		forward/fixed (forward and fixed directions cannot be specified when specifying source and destination addresses simultaneously)
Transfer mode	Single transfer	Transfer is completed when the DCTi register (i = 0 to 3) is set to "0000 <sub>16</sub> "
	Repeat transfer	When DCTi register is set to "0000 <sub>16</sub> ", a value of the DRCi register is reloaded into the DCTi register and a DMA transfer is continued
DMA interrupt request generation timing		When the DCTi register is set to "0000 <sub>16</sub> " from "0001 <sub>16</sub> "
DMA startup	Single transfer	Transfer starts when the DCTi register is set to "0001 <sub>16</sub> " or more and the DMA is requested after the MDi1 to MD0 bits in the DMDj register (j = 0 to 1) are set to "012" (single transfer)
	Repeat transfer	Transfer starts when DCTi register is set to "0001 <sub>16</sub> " and more and DMA is requested after "112" (repeat transfer) is written to the MDi1 to MD0 bits
DMA shutdown	Single transfer	When the MDi1 to MD0 bits is set to "002" (DMA inhibited) and DCTi register is set to "0000 <sub>16</sub> " (number of times for DMA transfer 0) by DMA transfer or write
	Repeat transfer	The MDi1 to MD0 bits is set to "002" and the DCTi register is set to "0000 <sub>16</sub> " with the DRCi register set to "0000 <sub>16</sub> "
Reload timing to DCTi or DMAi register		When the DCTi register is set to "0000 <sub>16</sub> " from "0001 <sub>16</sub> " in repeat transfer mode
DMA transfer cycles		Minimum 3 cycles

Notes :

- 1.No DMA transfer affects an interrupt.



**Figure 1.11.2. DM0SL to DM3SL Registers**

**Table 1.11.2. DMiSL Register (i = 0 to 3) Function**

Setting value	Conditions generated a DMA request				
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	Falling edge of INT1	Falling edge of INT2	Falling edge of INT3 <sup>1</sup>	(Note 2)
0 0 0 1 0	Both edges of INT0	Both edges of INT1	Both edges of INT2	Both edges of INT3 <sup>1</sup>	(Note 2)
0 0 0 1 1	Timer A0 interrupt request				
0 0 1 0 0	Timer A1 interrupt request				
0 0 1 0 1	Timer A2 interrupt request				
0 0 1 1 0	Timer A3 interrupt request				
0 0 1 1 1	Timer A4 interrupt request				
0 1 0 0 0	Timer B0 interrupt request				
0 1 0 0 1	Timer B1 interrupt request				
0 1 0 1 0	Timer B2 interrupt request				
0 1 0 1 1	Timer B3 interrupt request				
0 1 1 0 0	Timer B4 interrupt request				
0 1 1 0 1	Timer B5 interrupt request				
0 1 1 1 0	UART0 transmit interrupt request				
0 1 1 1 1	UART0 receive or ACK interrupt request <sup>3</sup>				
1 0 0 0 0	UART1 transmit interrupt request				
1 0 0 0 1	UART1 receive or ACK interrupt request <sup>3</sup>				
1 0 0 1 0	UART2 transmit interrupt request				
1 0 0 1 1	UART2 receive or ACK interrupt request <sup>3</sup>				
1 0 1 0 0	UART3 transmit interrupt request				
1 0 1 0 1	UART3 receive or ACK interrupt request <sup>3</sup>				
1 0 1 1 0	UART4 transmit interrupt request				
1 0 1 1 1	UART4 receive or ACK interrupt request <sup>3</sup>				
1 1 0 0 0	A-D0 interrupt request	A-D1 interrupt request	A-D0 interrupt request	A-D1 interrupt request	
1 1 0 0 1	Intelligent I/O interrupt 0 request	Intelligent I/O interrupt 7 request	Intelligent I/O interrupt 2 request	Intelligent I/O interrupt 9 request <sup>4</sup>	
1 1 0 1 0	Intelligent I/O interrupt 1 request	Intelligent I/O interrupt 8 request	Intelligent I/O interrupt 3 request	Intelligent I/O interrupt 10 request <sup>5</sup>	
1 1 0 1 1	Intelligent I/O interrupt 2 request	Intelligent I/O interrupt 9 request <sup>4</sup>	Intelligent I/O interrupt 4 request	Intelligent I/O interrupt 11 request <sup>6</sup>	
1 1 1 0 0	Intelligent I/O interrupt 3 request	Intelligent I/O interrupt 10 request <sup>5</sup>	Intelligent I/O interrupt 5 request	Intelligent I/O interrupt 0 request	
1 1 1 0 1	Intelligent I/O interrupt 4 request	Intelligent I/O interrupt 11 request <sup>6</sup>	Intelligent I/O interrupt 6 request	Intelligent I/O interrupt 1 request	
1 1 1 1 0	Intelligent I/O interrupt 5 request	Intelligent I/O interrupt 0 request	Intelligent I/O interrupt 7 request	Intelligent I/O interrupt 2 request	
1 1 1 1 1	Intelligent I/O interrupt 6 request	Intelligent I/O interrupt 1 request	Intelligent I/O interrupt 8 request	Intelligent I/O interrupt 3 request	

Notes :

1. When the INT3 pin is data bus in the memory expansion mode or microprocessor mode, DMA3 request cannot be generated by an INT3 pin interrupt.
2. The falling edge and both edge of input to the INT<sub>j</sub> pin (j = 0 to 3) cause a DMA request. An INT interrupt (the POL bit in the INTiC register, the LVS bit, the IFSR register) is not affected, and vice versa.
3. UkSMR register and UkSMR2 register (k = 0 to 4) determines receiving UART<sub>j</sub> and switching ACK.
4. An intelligent I/O interrupt 9 request shares DMA1 and DMA3 requests with a CAN interrupt 0 request.
5. An intelligent I/O interrupt 10 request shares DMA1 and DMA3 requests with a CAN interrupt 1 request.
6. An intelligent I/O interrupt 11 request shares DMA1 and DMA3 requests with a CAN interrupt 2 request.

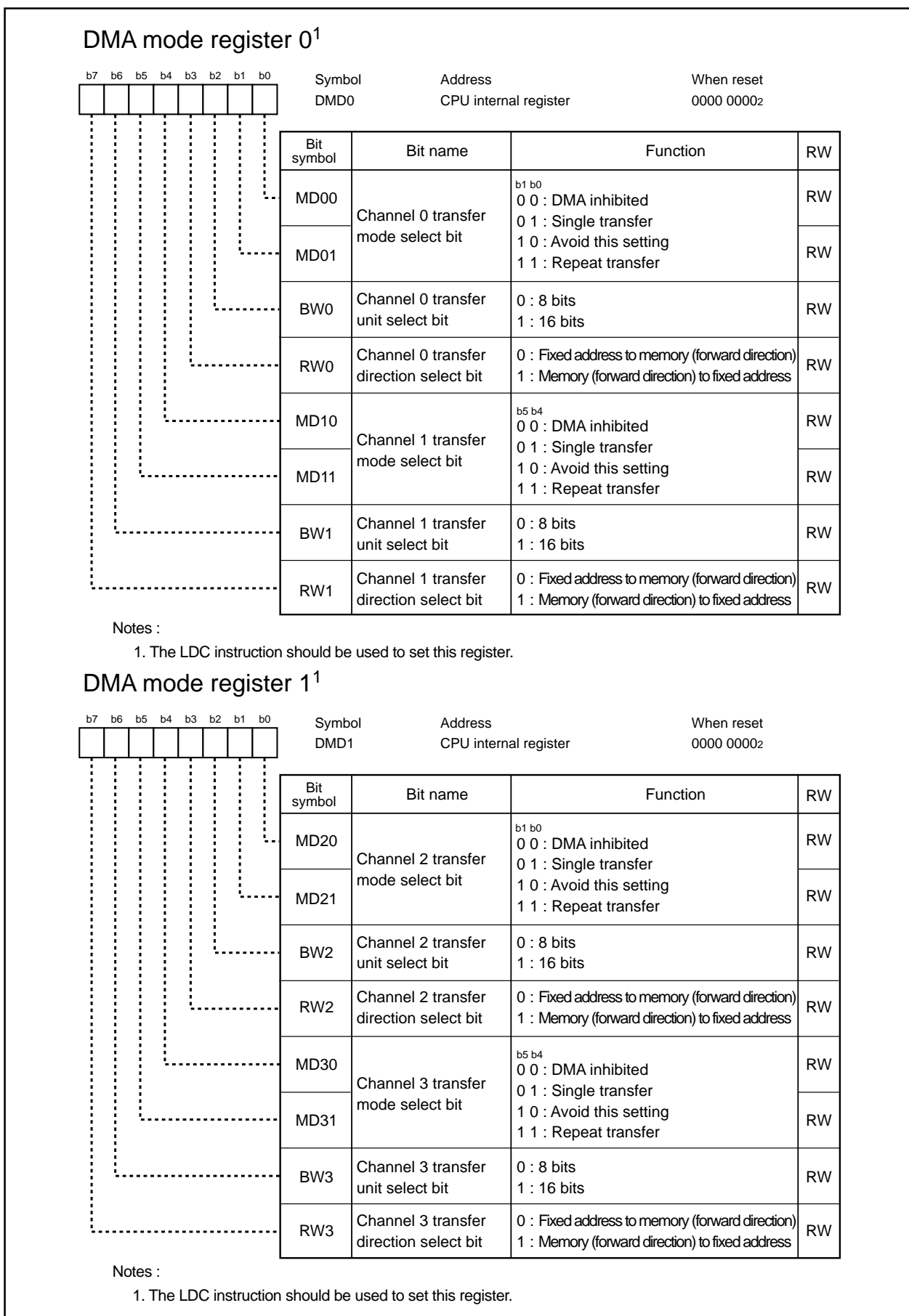


Figure 1.11.3. DMD0 Register, DMD1 Register

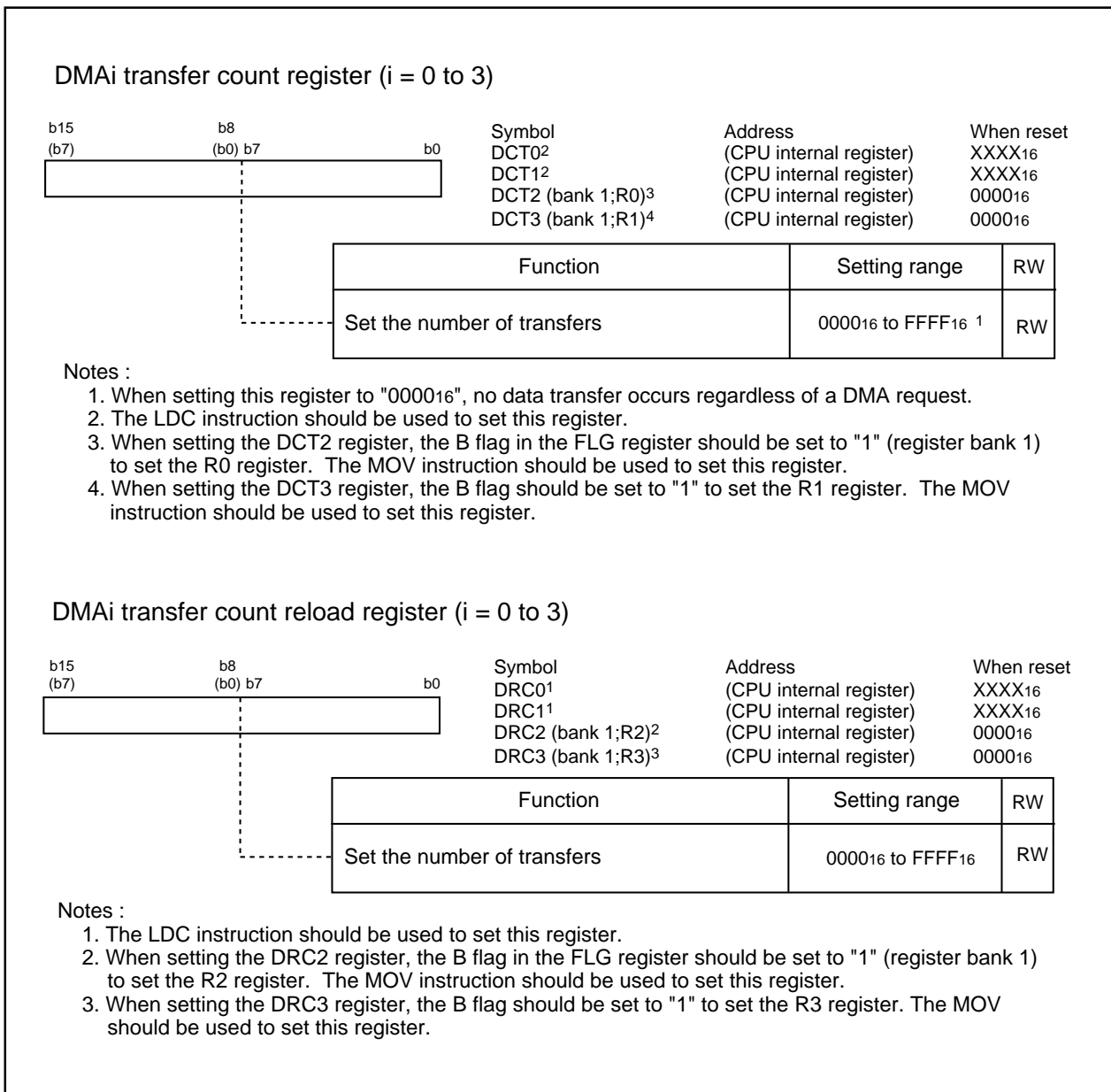


Figure 1.11.4. DCT0 to DCT3 Registers and DRC0 to DRC3 Registers

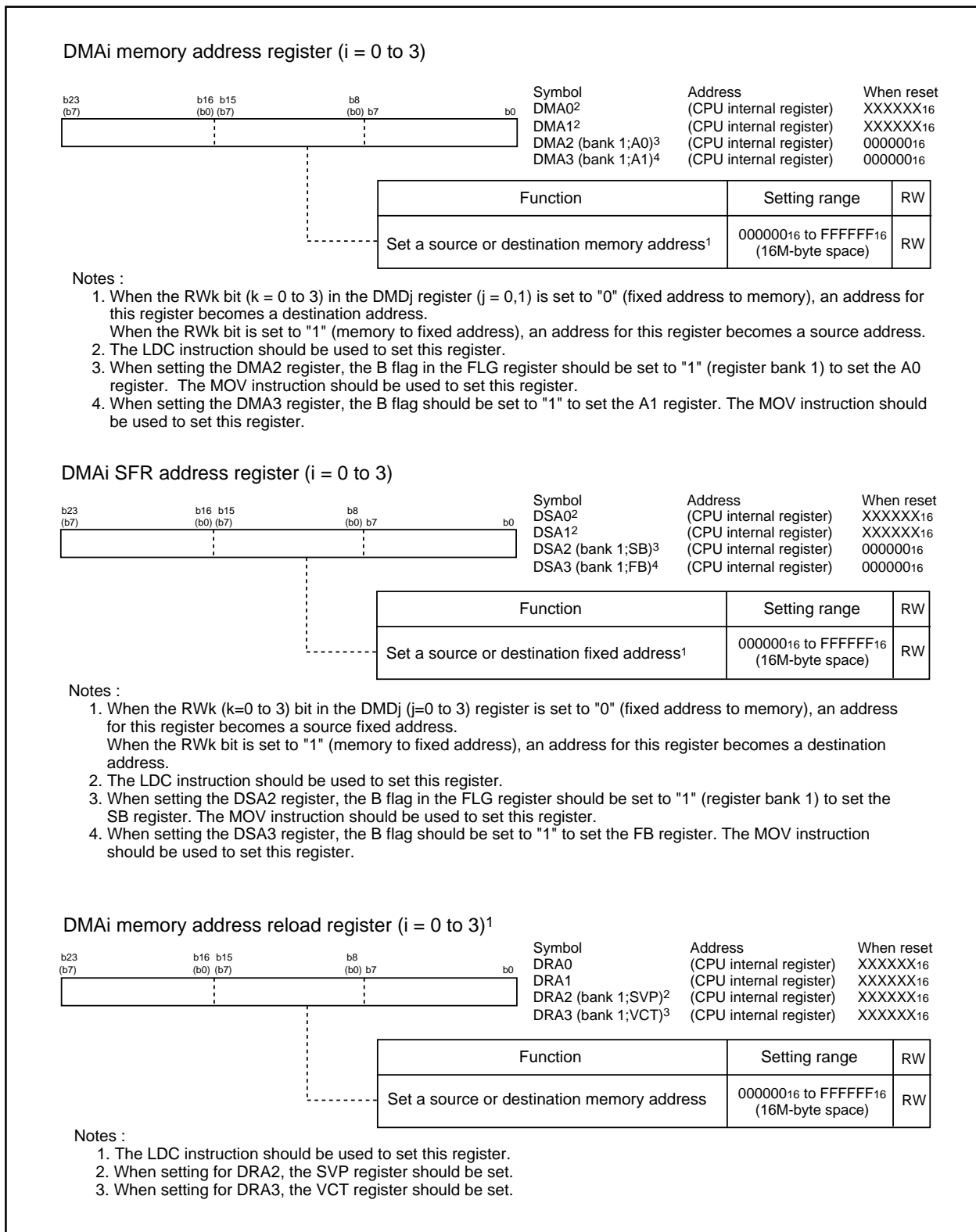


Figure 1.11.5. DMA0 to DMA3 Registers, DSA0 to DSA3 Registers and DRA0 to DRA3 Registers

## Transfer Cycles

Transfer cycle contains a bus cycle to read data from memory or from SFR space (source read) and a bus cycle to write data to a memory space or to a SFR space (destination write). The number of read and write bus cycles depends on source or destination addresses. In memory expansion mode and microprocessor mode, the number of read and write bus cycles also depends on the DS register. A bus cycle is longer when a software wait and the  $\overline{\text{RDY}}$  signal are inserted.

### 1. Effect of Source and Destination Addresses

When source address is an odd address with a 16-bit transfer unit and a 16-bit data bus, source read cycle is incremented by one bus cycle, compared to a source address being an even address.

Likewise, with a 16-bit transfer unit and a 16-bit data bus, a destination address is an odd address, a destination write cycle is incremented by one bus cycle, compared to an destination address being an even address.

### 2. Effect of DS Register

In an external space in memory expansion or microprocessor mode, a transfer cycle varies depending on a data bus used at source and destination address. See Figure 1.7.1 for details about the DS register.

- (1) When an 8-bit data bus each to access both source address and destination address are used to transfer a 16-bit data, an 8-bit data is transferred twice. Therefore, two bus cycles are required for reading and another two bus cycles are for writing.
- (2) When an 8-bit data bus to access a source address and a 16-bit data bus to access a destination address are used to transfer a 16-bit data, an 8-bit data. Therefore, two bus cycles are required for reading and one bus cycle is for writing.
- (3) When a 16-bit data bus to access a source address and an 8-bit data bus to access a destination address are used to transfer a 16-bit data, a 16-bit data is read and an 8-bit data is written twice. Therefore, one bus cycle is required for reading and two bus cycles is for writing.

### 3. Effect of Software Wait

When a SFR space or a memory space with a software wait is accessed, the number of cycles is incremented by software wait(s).

Figure 1.11.6 shows an example of a transfer cycle for a source read. In Figure 1.11.6, the number of source read cycles is illustrated with different conditions, provided that a destination address is in an external space with two bus cycles for a destination write cycle. Indeed, a destination write cycle is affected by each condition as well as a source read cycle and a transfer cycle changes accordingly. When calculating a transfer cycle, apply respective conditions to both destination write cycle and source read cycle. For example (2) in Figure 1.11.6, when an 8-bit data bus each to access both source address and destination address are used to transfer a 16-bit data, two bus cycles are each required as a source read cycle and destination write cycle.

### 4. Effect of $\overline{\text{RDY}}$ Signal

In memory expansion or microprocessor mode, the  $\overline{\text{RDY}}$  signal affect an external space. Refer to the paragraph "Bus control" and "6.  $\overline{\text{RDY}}$  signal" in the section "Bus" for details.

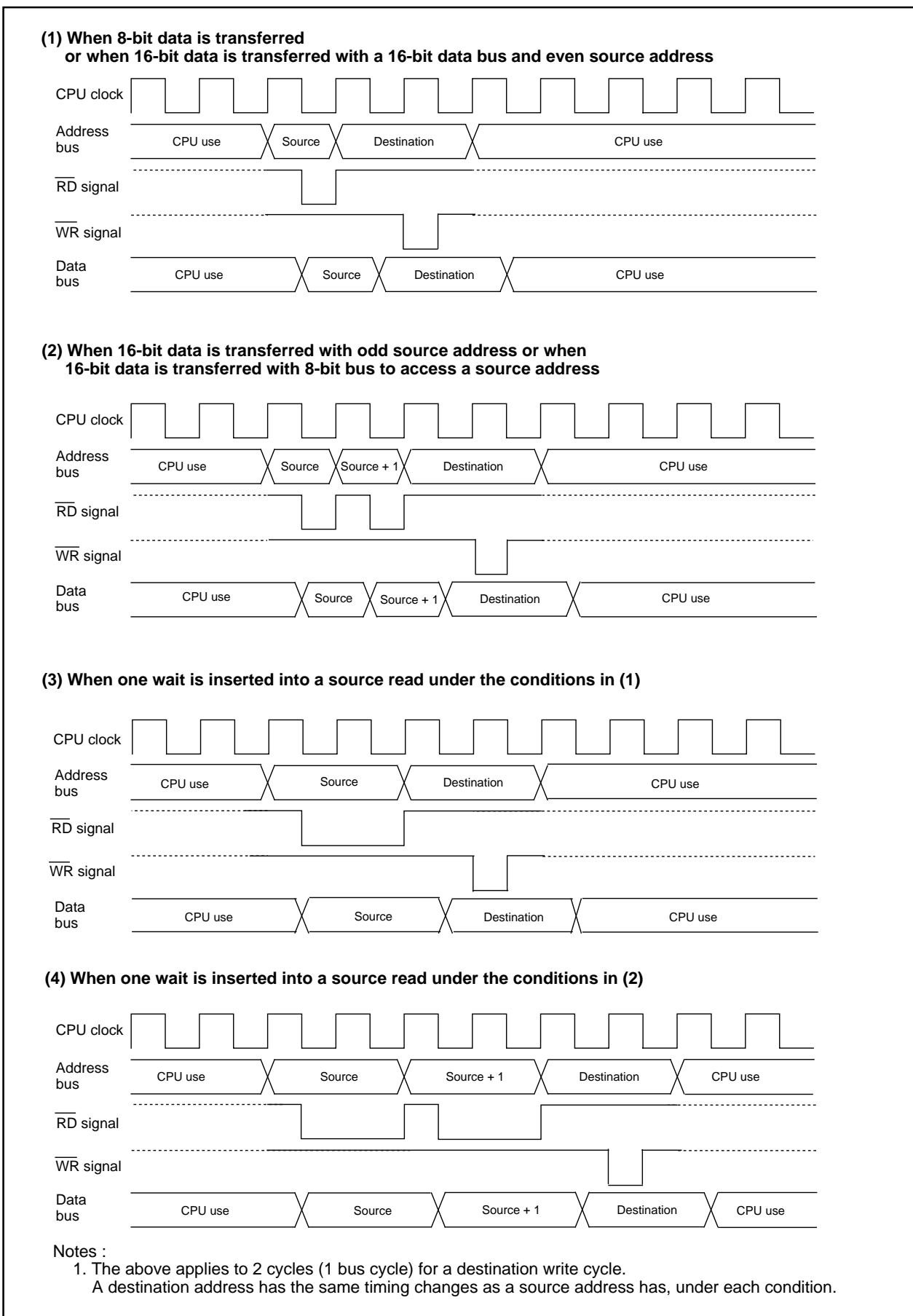


Figure 1.11.6. Transfer Cycles for Source Read

## DMAC Transfer Cycles

The number of DMAC transfer cycle can be calculated as follows.

Any combination of even or odd transfer read and write addresses are available. Table 1.11.3 lists the number of DMAC transfer cycles. Table 1.1.4 lists coefficient j, k.

$$\text{Transfer cycles per transfer unit} = \text{Number of read cycle(s)} \times j + \text{Number of write cycle(s)} \times k$$

**Table 1.11.3. DMAC Transfer Cycles**

Transfer unit	Bus width	Access address	Single-chip mode		Memory expansion mode Microprocessor mode	
			Read cycles	Write cycles	Read cycles	Write cycles
8-bit transfers (BWi bit in the DMDp register = 0)	16-bit	Even	1	1	1	1
		Odd	1	1	1	1
	8-bit	Even	—	—	1	1
		Odd	—	—	1	1
16-bit transfers (BWi bit = 1)	16-bit	Even	1	1	1	1
		Odd	2	2	2	2
	8-bit	Even	—	—	2	2
		Odd	—	—	2	2

i = 0 to 3, p = 0 to 1

**Table 1.11.4. Coefficient j, k**

Internal space			External space					
Internal ROM or internal RAM No wait	Internal ROM or internal RAM Wait	SFR space	Separate bus No wait	Separate bus 1 wait	Separate bus 2 waits	Separate bus 3 waits	Multiplex bus 2 waits	Multiplex bus 3 waits
j=1 k=1	j=2 k=2	j=2 k=2	j=1 k=2	j=2 k=2	j=3 k=3	j=4 k=4	j=3 k=3	j=4 k=4

### Channel Priority and DMA Transfer Timing

When some DMA requests occur in the same sampling period between the falling edge of the CPU clock and the following falling edge, the DRQ bit in the DMiSL register (i = 0 to 3) is set to "1" (with a request) simultaneously. Channel priority in this case is : DMA0 > DMA1 > DMA2 > DMA3.

Figure 1.11.7 shows an example of a DMA transfer by external factors it illustrates. What happens when DMA0 and DMA1 requests occur in the same sampling cycle.

In Figure 1.11.7, a DMA0 request having priority is received first to start transfers when a DMA0 request and DMA1 request occur simultaneously. When one DMA0 transfer unit is completed, a bus privilege is returned to the CPU. When the CPU has completed one bus access, a DMA1 transfer starts. When one DMA1 transfer unit is completed, the privilege is again returned to the CPU.

In addition, DMA requests cannot be counted up since the DRQ bit is 1 bit for each channel. Therefore, when DMA requests, as DMA1 in Figure 1.11.7, occurs more than one time, the DRQ bit is set to "0" as soon as getting the privilege the privilege is returned to the CPU when one transfer unit is completed.

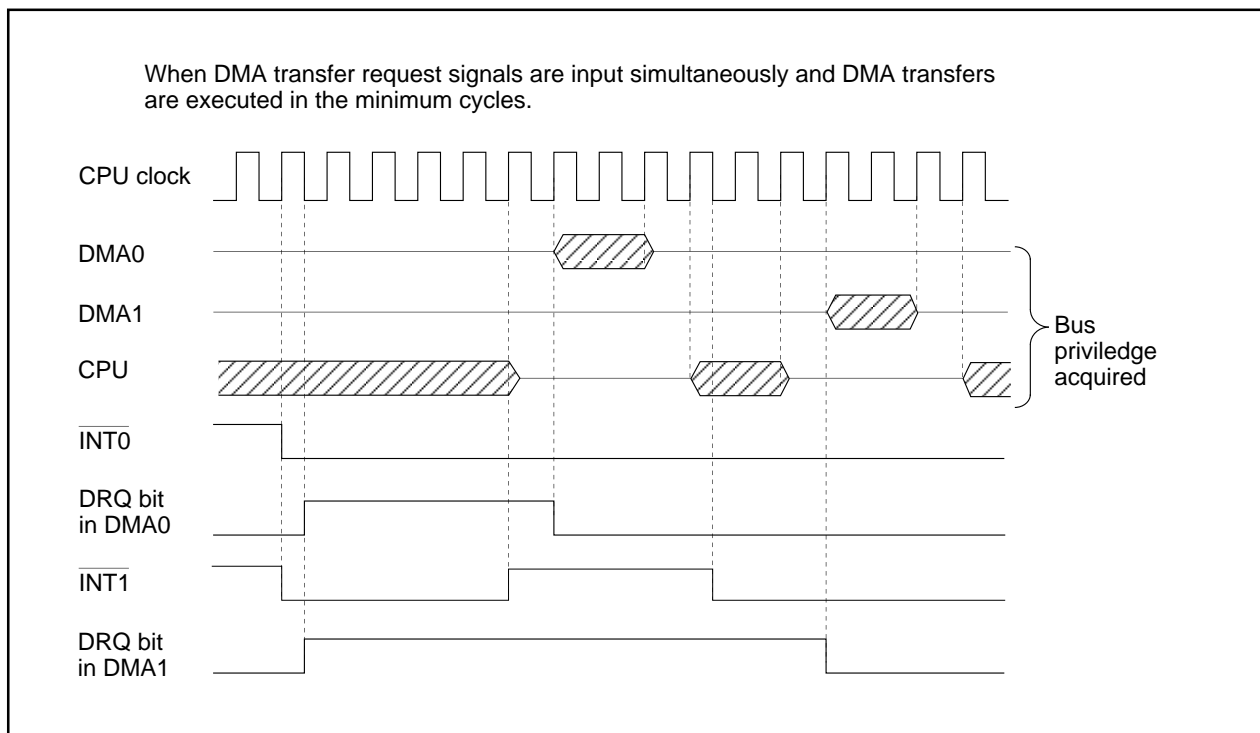


Figure 1.11.7. DMA Transfer by External Factors

## Precautions for DMAC

- (1) When setting registers associated with DMAC, the MDi1 to MDi0 bits ( $i=0$  to 3) in the DMDj register ( $j=0,1$ ) corresponding to channel  $i$  should be set to "002" (DMA disabled). Then the MDi1 to MDi0 bits should be set to "012" (single transfer) or "112" (repeat transfer). This setting allows a DMA request of channels to be received.
- (2) The DRQ bit in the DMiSL register should be avoided setting to "0" (no request).  
When a DMA request is generated with a channel disabled<sup>1</sup>, a DMA transfer is not performed and the DRQ bit is set to "0."  
Notes :  
1. This state means that the MDi1 to MDi0 bits is set to "002" or the DCTi register is set to "000016" (the number of transfer: 0).
- (3) When a DMA transfer is performed by a software trigger, the DSR and DRQ bits in the DMiSL register should be set to "1" simultaneously.  
e.g. OR.B #0A0h, DMiSL ; The DSR and DRQ bits should be set to "1" simultaneously.
- (4) If the DMA interrupt is used including other channels, avoid generating a DMA request of channel  $i$  when setting the DCTi register to "1" and the MDi1 to MDi0 bits of corresponding channel  $i$  to "012" or "112".

A DMA request of channel  $i$  should be generated after setting a DMA-associated register of channel  $i$ . (The peripheral function as DMA request factors should be set after setting a DMA-associated register.) If not fulfilling the above conditions (setting the  $\overline{INT}$  interrupt for DMA request factor), avoid setting the DCTi register to "1".

**DMAC II**

## DMAC II

The DMAC II performs a memory-to-memory transfer, an immediate data transfer or an arithmetic transfer the sum of two data added by an interrupt request from any peripheral functions.

Table 1.12.1 lists specifications of the DMAC II.

**Table 1.12.1. DMAC II Specifications**

Item	Specification
DMAC II request factor	Interrupt request from all peripheral functions I/O the ILVL2 to ILVL0 bits of Interrupt control register is set to "1112" (level 7)
Transfer data	<ul style="list-style-type: none"> <li>• Memory -&gt; memory (memory-to-memory transfer)</li> <li>• Immediate data -&gt; memory (immediate data transfer)</li> <li>• Memory (or immediate data) + memory -&gt; memory (arithmetic transfer)</li> </ul>
Transfer block	8 or 16 bits
Transfer space	64-Kbyte space at addresses 0000 <sub>16</sub> to 0FFFF <sub>16</sub>
Transfer direction	Fixed or forward address Can be selected for either a source address or destination address.
Transfer mode	<ul style="list-style-type: none"> <li>• Single transfer</li> <li>• Burst transfer</li> </ul>
Chained transfer function	Parameters (transfer count, transfer address, and other information) are switched over when a transfer counter reaches zero.
Interrupt at end of transfer	Interrupt is generated when a transfer counter reaches zero.
Multiple transfer function	Multiple data transfers can be performed by one DMA II transfer request generated.

Notes :

1. When transferring a 16-bit data to a destination address as 0FFFF<sub>16</sub>, it is transferred to 0FFFF<sub>16</sub> and 10000<sub>16</sub>. When to a source address as 0FFFF<sub>16</sub>, data is transferred as well.

### DMAC II Settings

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

- RLVL register
- DMAC II Index
- Interrupt control register for the peripheral function causing DMAC II request
- Relocatable vector table for the peripheral function causing DMAC II request
- With the intelligent I/O or CAN interrupt, set the IRLT bit of IIOiE register (i = 0 to 11).

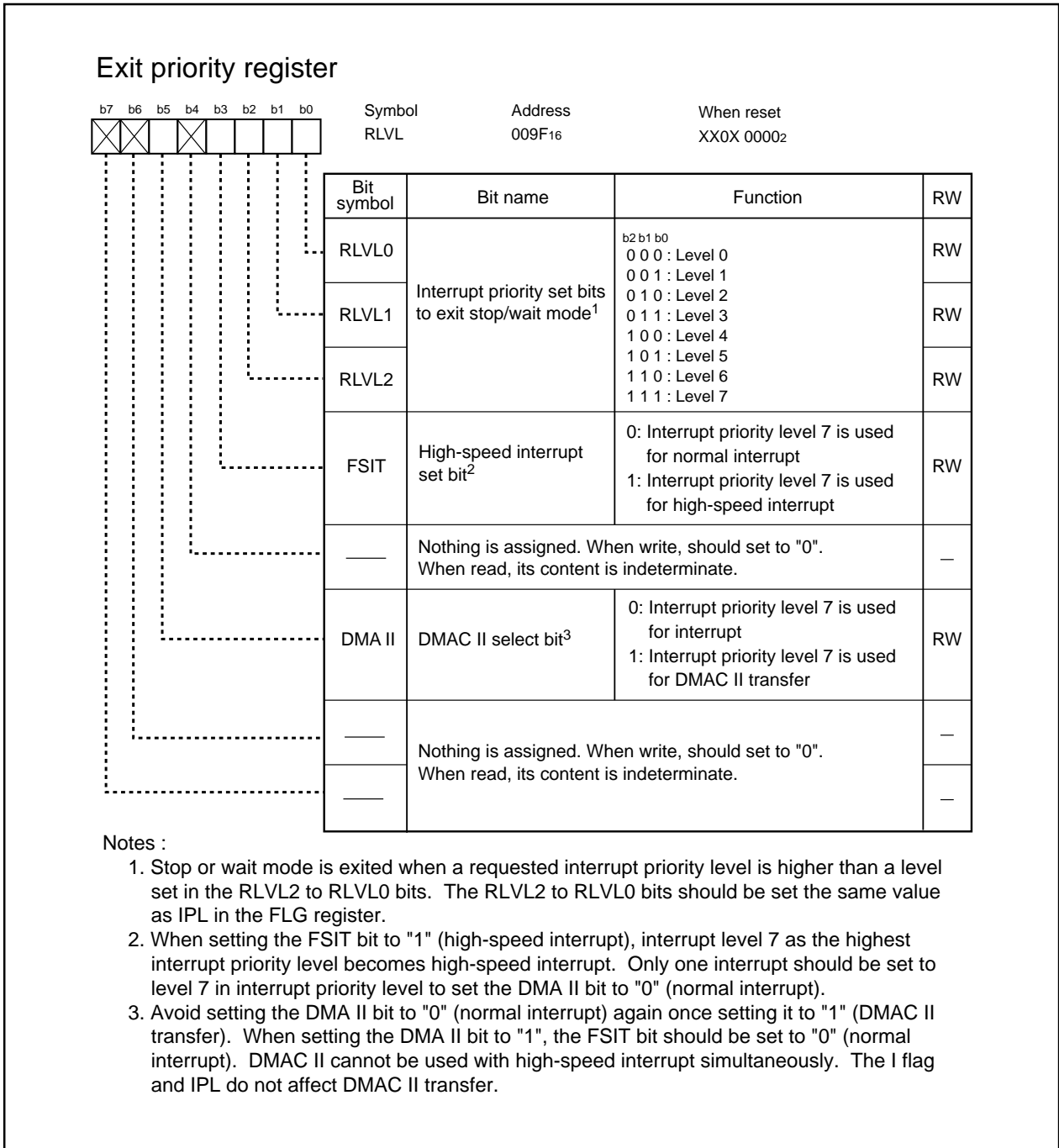
Refer to the section "Interrupt" about the IIOiE register

**DMAC II**

**1. RLVL Register**

When setting the DMA II bit to "1" (DMAC II transfer) and the FSIT bit to "0" (normal interrupt), the DMAC II is activated by an interrupt request from all peripheral functions that is set the ILVL2 to ILVL0 bits in the interrupt control register to "1112" (level 7).

Figure 1.12.1 shows the RLVL register.



**Figure 1.12.1. RLVL Register**

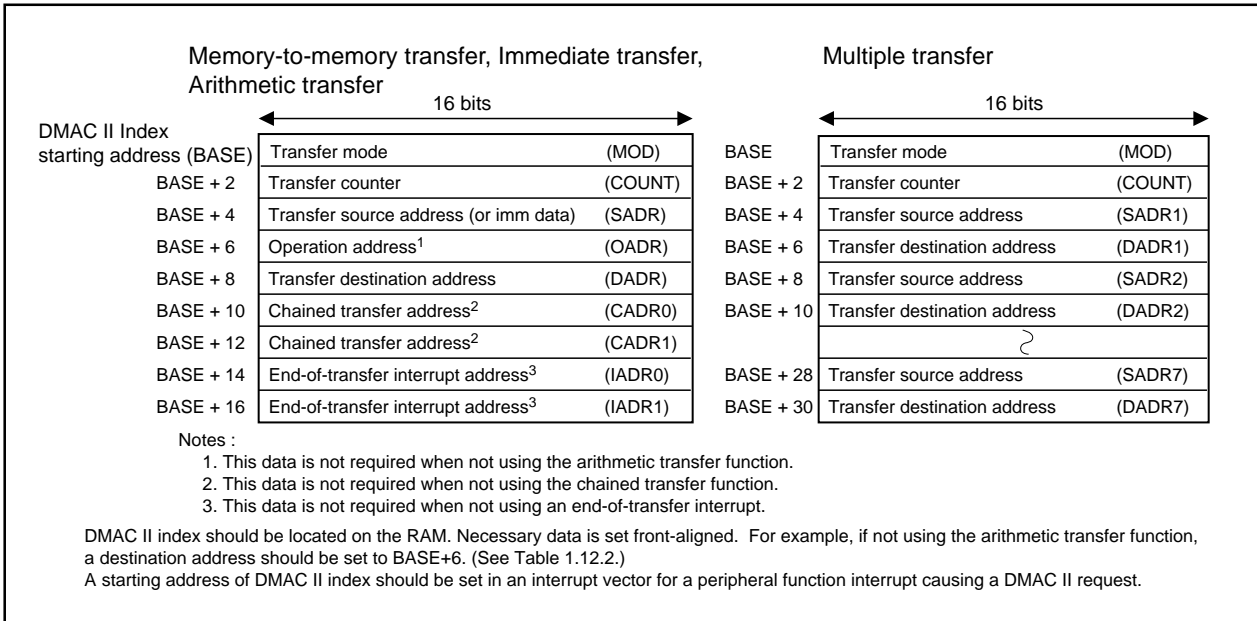
**DMAC II**

**(2) DMAC II Index**

The DMAC II index is a data table, comprised of 8 to 18 bytes (maximum 32 bytes when the multiple transfer function is selected). The DMA II index stores parameters for 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 should be located on the RAM space.

Figure 1.12.2 shows a configuration of the DMAC II index. Table 1.12.2 lists a configuration of the DMAC II index in transfer mode.



**Figure 1.12.2. DMAC II Index**

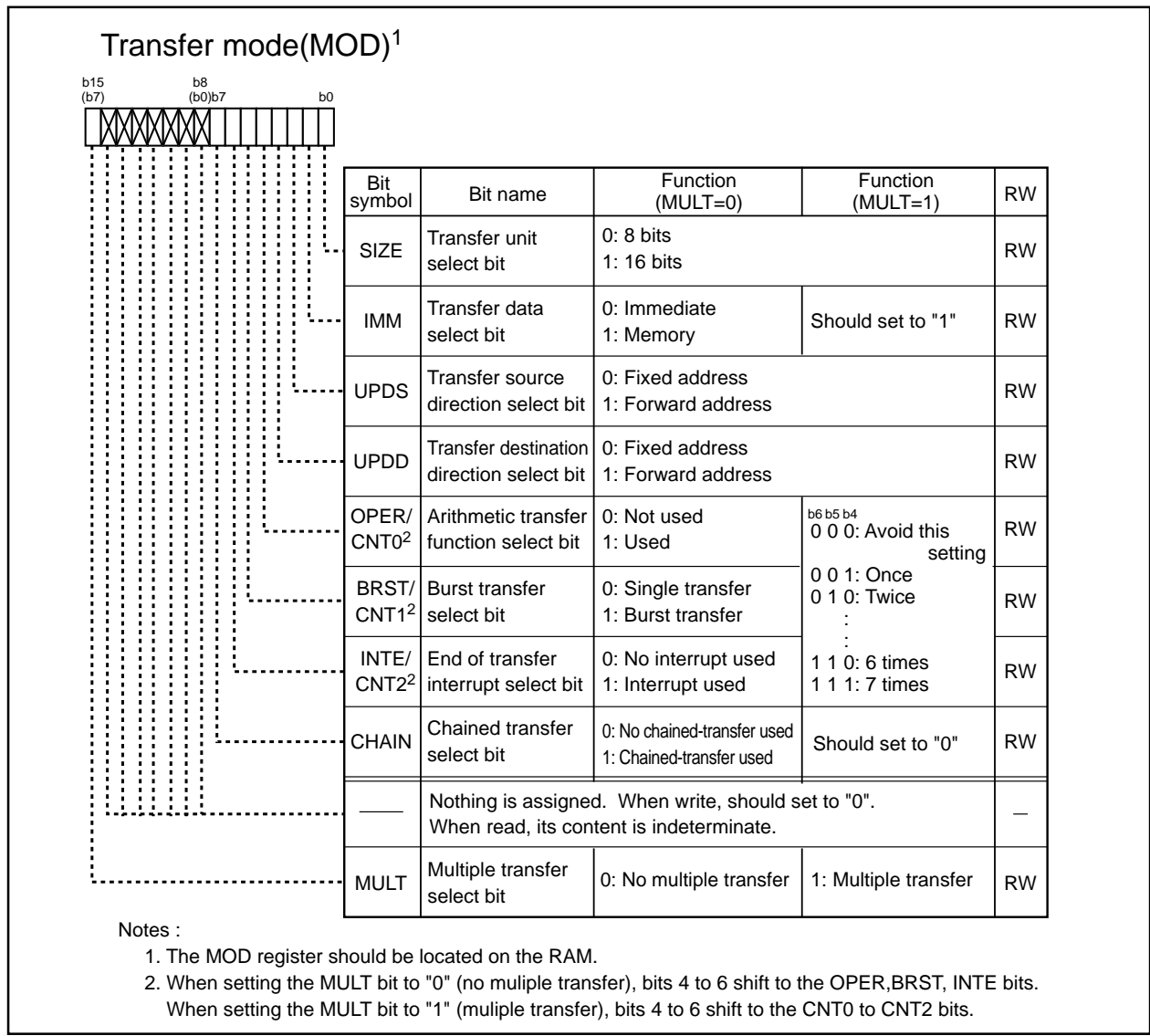
The detail for DMAC II index is below. These parameters should be set in the specified order listed on Table 1.12.2, according to DMAC II transfer mode.

- Transfer mode (MOD)  
 Two-byte data is required to set transfer mode. Figure 1.12.3 shows a configuration for transfer mode.
- Transfer counter (COUNT)  
 Two-byte data is required to set the number of transfer.
- Transfer source address (SADR)  
 Two-byte data is required to set a source memory address or immediate data.
- Operation address (OADR)  
 Two-byte data is required to set a memory address for calculation. This data should be set only with the arithmetic transfer function.
- Transfer destination address (DADR)  
 Two-byte data is required to set a destination memory address.
- Chained transfer address (CADR)  
 Four-byte data is required to set the DMAC II index starting address for the next transfer. This data should be set only with the chained transfer function.
- End-of-transfer interrupt address (IADR)  
 Four-byte data is required to set a jump address for end-of-transfer interrupt processing. This data should be set only with the end-of-transfer interrupt.

**DMAC II**

**Table 1.12.2. DMAC II Index Configuration in Transfer Mode**

Transmit data	Memory-to-memory transfer /immediate data transfer				Arithmetic transfer				Multiple transfer																																																														
	Not use	Use	Not use	Use	Not use	Use	Not use	Use																																																															
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> <tr><td>CADR0</td></tr> <tr><td>CADR1</td></tr> </table> <p>10 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>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>SADR</td></tr> <tr><td>OADR</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>18 bytes</p>	MOD	COUNT	SADR	OADR	DADR	CADR0	CADR1	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 colspan="2">...</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																																																																							
CADR0																																																																							
CADR1																																																																							
MOD																																																																							
COUNT																																																																							
SADR																																																																							
OADR																																																																							
DADR																																																																							
CADR0																																																																							
CADR1																																																																							
MOD																																																																							
COUNT																																																																							
SADR																																																																							
OADR																																																																							
DADR																																																																							
IADR0																																																																							
IADR1																																																																							
MOD																																																																							
COUNT																																																																							
SADR																																																																							
OADR																																																																							
DADR																																																																							
CADR0																																																																							
CADR1																																																																							
IADR0																																																																							
IADR1																																																																							
MOD																																																																							
COUNT																																																																							
SADR1																																																																							
DADR1																																																																							
...																																																																							
SADRi																																																																							
DADRi																																																																							



**Figure 1.12.3. MOD**

## DMAC II

---

### (3) Interrupt Control Register for Peripheral Function

For the peripheral function interrupt used as a DMAC II request factor, the ILVL2 to ILVL0 bits should be set to "1112" (level 7).

### (4) Relocatable Vector Table for Peripheral Function

The DMAC II index starting address should be set in a relocatable vector table for the peripheral function interrupt causing a DMAC II request.

When using chained transfers, a relocatable vector table should be located on the RAM.

### (5) IRLT Bit in the IIOiE Register (i=0 to 11)

When using the intelligent I/O interrupt or CAN interrupt to activate DMAC II, the IRLT bit in the IIOiE register for interrupts causing a request should be set to "0".

## Operation of DMAC II

The DMAC II function is selected by setting the DMA II bit to "1" (DMAC II transfer). All peripheral function interrupt requests that the ILVL2 to ILVL0 bits are set to "1112" (level 7) comprise DMAC II request factors. These function interrupt request signals would change to DMAC II transfer request signals the peripheral function interrupt cannot be used.

When an interrupt request is generated with setting the ILVL2 to ILVL0 bits to "1112" (level 7), the DMAC II is activated no matter which state the I flag and IPL is in.

## Transfer Data

The DMAC II transfers an 8-bit or 16-bit data.

- Memory-to-memory transfer : Data is transferred from any memory location in a 64K-byte space (Addresses 00000<sub>16</sub> to 0FFFF<sub>16</sub>) to any memory location in the same space.
- Immediate data transfer : Data is transferred as an immediate data to any memory location in a 64K-byte space.
- Arithmetic transfer : Two 8-bit or 16-bit data are added together and the result is transferred to any memory location in a 64K-byte space.

When transferring a 16-bit data to a destination address as 0FFFF<sub>16</sub>, it is transferred to 0FFFF<sub>16</sub> and 10000<sub>16</sub>. When to a source address as 16-bit data is transferred as well.

### (1) Memory-to-memory Transfer

Data transfer from any memory location to any memory location can be:

- Transfer from a fixed address to another fixed address
- Transfer from a fixed address to a relocatable address
- Transfer from a locatable address to a fixed address
- Transfer from a locatable address to another relocatable address

When selecting a locatable address, an address is incremented for the next transfer after transfer. In a 8-bit transfer, a transfer address is incremented by one. In a 16-bit transfer, a transfer address is incremented by two.

When a source or destination address exceeds address 0FFFF<sub>16</sub> as a result of address incrementation, a source or destination address is back to address 00000<sub>16</sub> to increment. A source or destination address should be maintained below.

## DMAC II

---

### (2) Immediate Data Transfer

Immediate data is transferred to any memory location. A fixed or locatable address can be selected as a destination address. An immediate data should be stored into the SADR. When transferring a 8-bit immediate data, data should be set in a lower one byte position of the SADR (a high-order byte is ignored).

### (3) Arithmetic Transfer

Calculated results are transferred to any memory after any memory and any memory, or an immediate data and any memory are added together. A memory location address to be operated or immediate data should be set in the SADR and another memory location address to be operated should be set in the OADR. When performing a memory + memory arithmetic transfer, a fixed or relocatable address can be selected for source and destination addresses. When a transfer source address is relocatable, an operation address also becomes relocatable. When performing an immediate data + memory arithmetic transfer, a fixed or locatable address can be selected for a transfer destination address.

## Transfer Modes

In DMAC II, single and burst transfers are available. The BRST bit in MOD determines which transfer method is used single or burst transfer. COUNT determines how many transfer is performed. Transfer is not performed when setting COUNT to "0000<sub>16</sub>".

### 1. Single Transfer

For one request factor, an 8-bit or 16-bit data as one transfer unit is transferred once. When a source or destination address is relocatable, an address is incremented for the following transfer after a transfer.

COUNT is decremented by each transfer performed. With the end-of-transfer interrupt, it is generated when COUNT reaches "0".

### 2. Burst Transfer

For one request factor, data are transferred consecutively only as set in COUNT. COUNT is decremented by each transfer performed. Burst transfer ends when COUNT reaches "0". The end-of-transfer interrupt is generated when burst transfer ends if using the end-of-transfer interrupt. The interrupt is ignored when burst transfer is in progress.

### 3. Multiple Transfer

Multiple transfer can be selected by the MULT bit in MOD. For multiple transfer memory to memory transfer can be also performed. Multiple transfers are performed for one request factor received. The CNT2 to CNT0 bits in MOD determines how many transfer is performed from "001<sub>2</sub>" (once) to "111<sub>2</sub>" (7 times). Avoid setting the CNT2 to CNT0 bits to "000<sub>2</sub>".

The transfer source and transfer destination addresses to be transferred should be allocated alternately in addresses following both MOD and COUNT. When selecting multiple transfer, arithmetic transfer, burst transfer, end-of-transfer interrupt and chained transfer cannot be used.

**DMAC II**

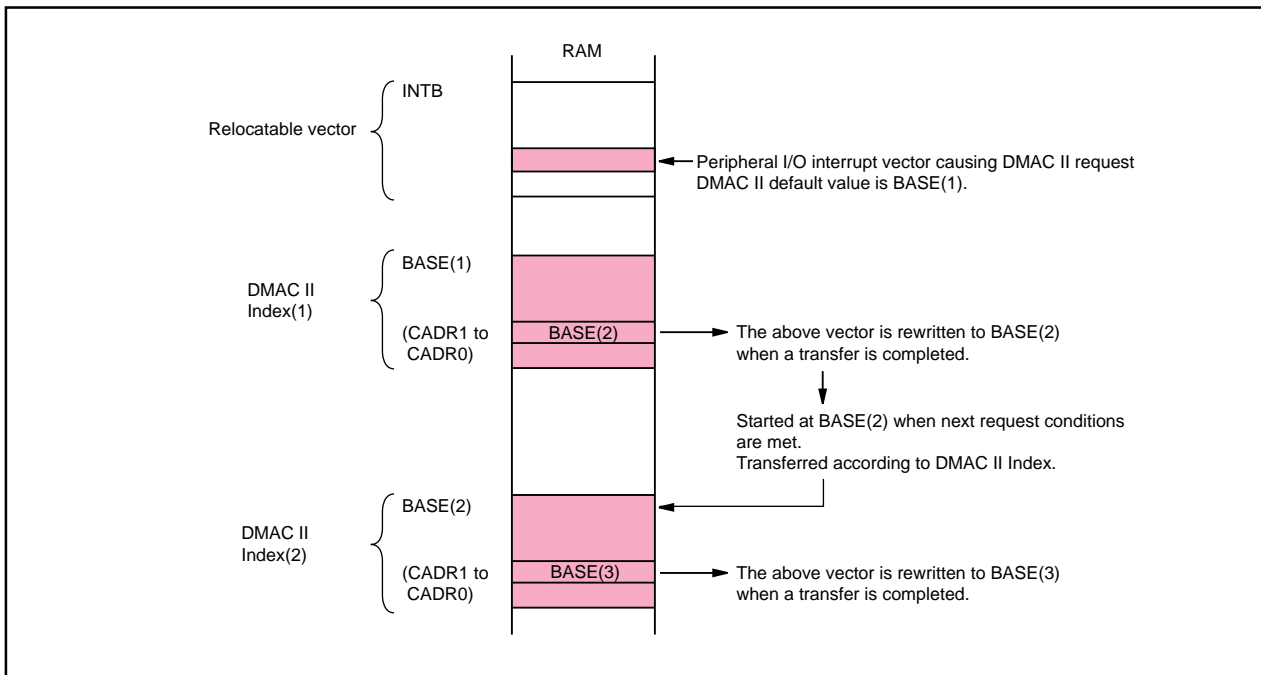
**4. Chained Transfer**

Chained transfer can be selected by the CHAIN bit in MOD.

The following is how chained transfer is performed.

- (1) With a request factor, transfer is performed in accordance with DMAC II index contents provided by vectors of the factor. For one request factor, the BRST bit determines whether single or burst transfer.
- (2) When COUNT reaches "0", contents of CADR1 to CADR0 are rewritten to contents of vector for the peripheral function interrupt causing a DMAC II request. When the INTE bit in the MOD is set to "1," the end-of-transfer interrupt is generated simultaneously.
- (3) When the next DMAC II transfer request conditions are met, transfer occurs in accordance with the DMAC II index provided by a vector for the peripheral function interrupt as rewritten on (2).

Figure 1.12.4 shows a relocatable vector and DMACII index when chained transfer is in progress. With chained transfer, a relocatable vector table must be located in the RAM.



**Figure 1.12.4. Relocatable Vector and DMAC II Index**

**5. End-of-transfer Interrupt**

The INTE bit in MOD selects the End-of-transfer interrupt. A starting address of the end-of-transfer interrupt routine should be set in the IADR1 to IADR0 bits. The end-of-transfer interrupt is generated when COUNT reaches "0."

## DMAC II

---

### Execution Time

DMAC II execution cycle is calculated by the equation below.

Other than multiple transfers,  $t = 6 + (26 + a + b + c + d) \times m + (4 + e) \times n$  (cycles)

Multiple transfers,  $t = 21 + (11 + b + c) \times k$  (cycles)

where :

a: If IMM = 0 (source of transfer is immediate data), a = 0; if IMM = 1 (it is memory), a = -1

b: If UPDS = 0 (source address of transfer is a locatable address), b = 0; if UPDS = 1 (it is a fixed address),  
b = 1

c: If UPDD = 0 (destination address of transfer is a locatable address), c = 0; if UPDD = 1 (it is a fixed  
address), c = 1

d: If OPER = 0 (arithmetic function is not selected), d = 0; if OPER = 1 (arithmetic function is selected) and  
LIPDS = 0 (source of transfer is immediate data or fixed address memory), d = 7; if OPER = 1 (arith-  
metic function is selected) and LIPDS = 1 (source of transfer is locatable address memory), d = 8

e: If CHAIN = 0 (chained transfer function is not selected), e = 0; if CHAIN = 1 (chained transfer function  
is selected), e = 4

m: BRST = 0 (single transfer), m = 1; BRST = 1 (burst transfer), m = the value set by COUNT

n: If COUNT = 1, n = 0; if COUNT = 0, n = 1

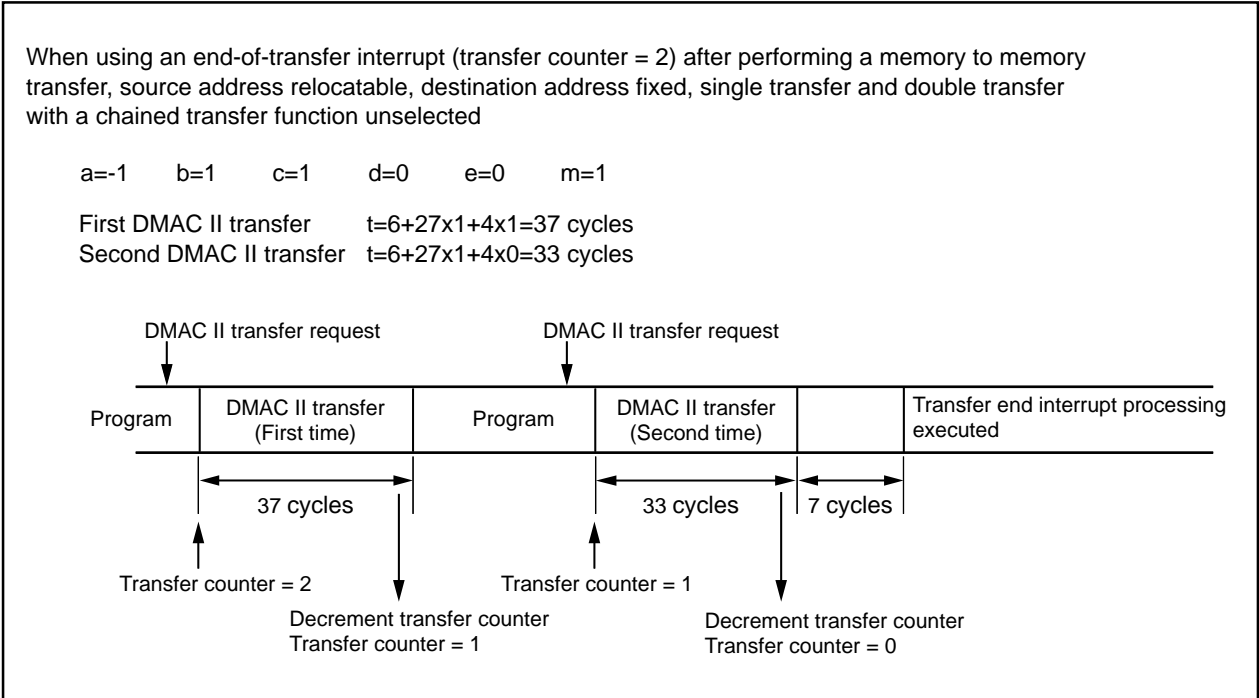
k: Number of transfers set by the CNT2 to CNT0 bits

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

- A bus wait is set to "0".
- DMAC II Index is set to an even address.
- During a word transfer, all transfer source address, transfer destination address, and operation address are set to even addresses.

The first instruction from end-of-transfer interrupt routine is executed in 7 cycles after DMAC II transfers are completed.

**DMAC II**



**Figure 1.12.5. Transfer Cycle**

When an interrupt request causing DMAC II request conditions and interrupt request with higher priority (e.g.  $\overline{NMI}$  or watchdog timer) are generated, this higher priority interrupt takes precedence over the DMAC II transfer to be received. The pending DMAC II transfer starts after an interrupt processing sequence is completed.

# Timer

The microcomputer has eleven 16-bit timers. Five timers A and six timers B have different functions. Each timer functions independently. Count source of each timer is a clock for the timer operation like counting and reloading, etc. Figures 1.13.1 and 1.13.2 show block diagrams of the timer A and timer B configuration.

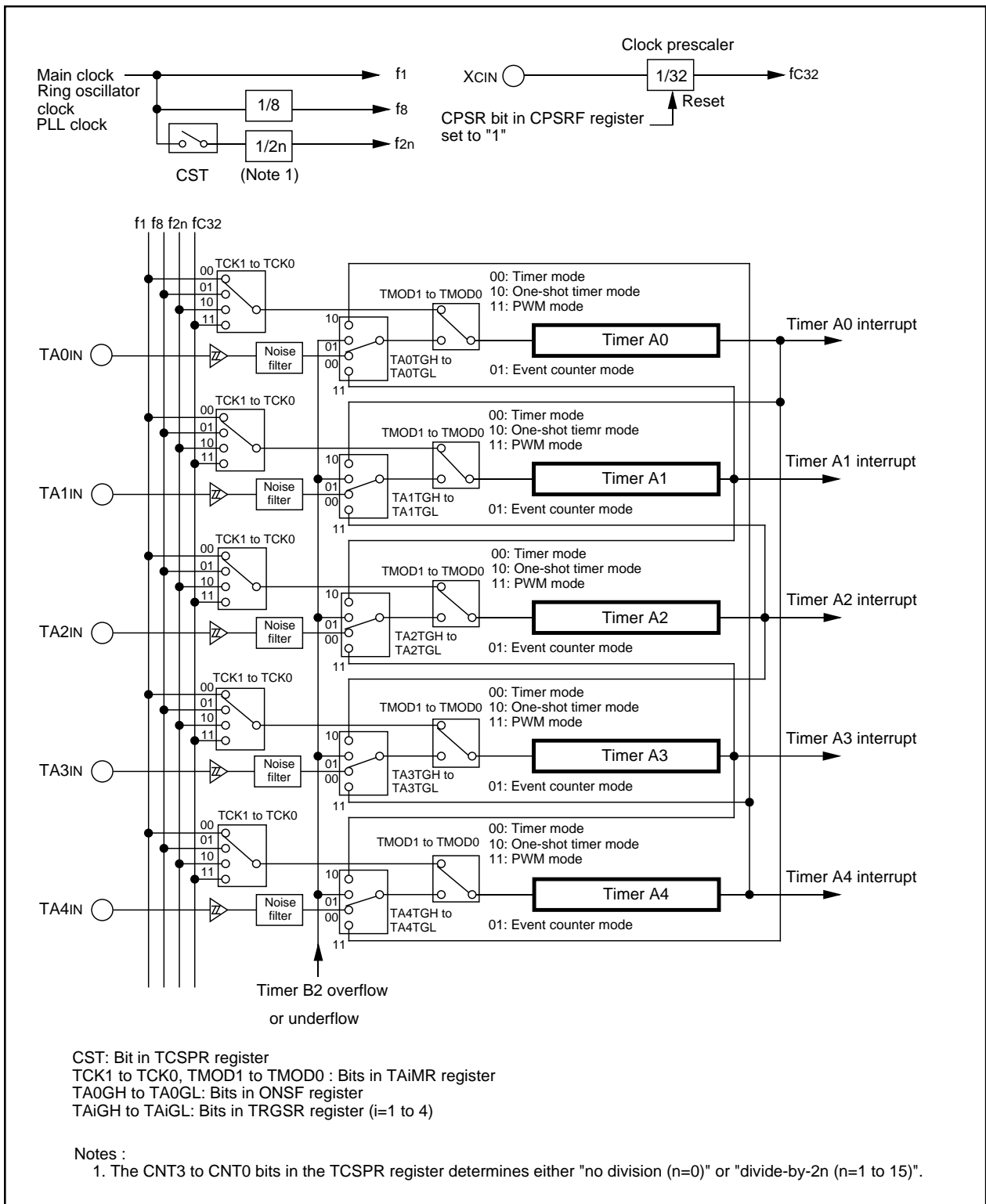


Figure 1.13.1. Timer A Configuration

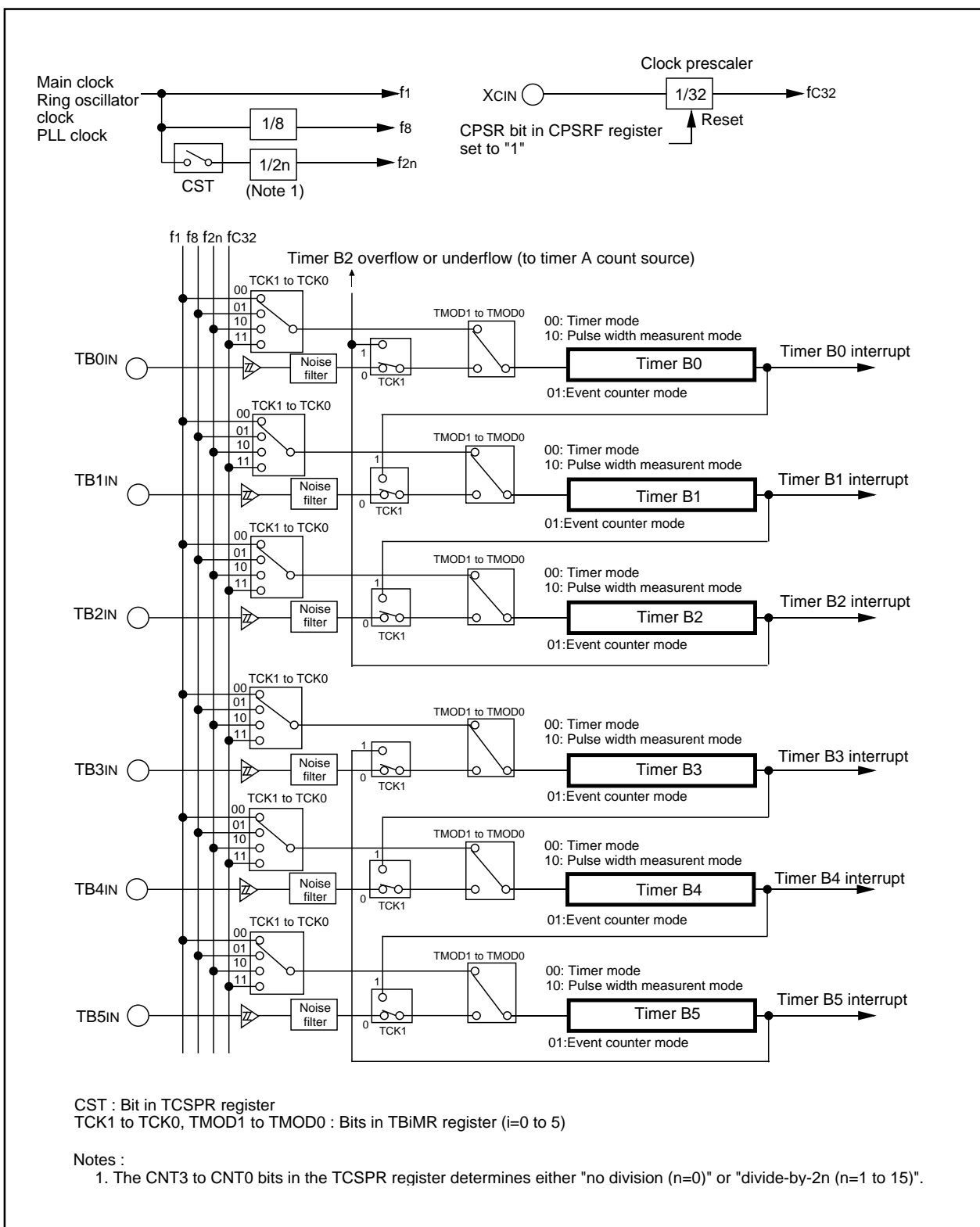


Figure 1.13.2. Timer B Configuration

## Timer (Timer A)

### Timer A

Figure 1.14.1 shows a block diagram of the timer A. Figures 1.14.2 to 1.14.5 show registers associated with the timer A.

The timer A supports the following four modes. Except in event counter mode, all timers A0 to A4 have the same function. The TMOD1 to TMOD0 bits in the TAI<sub>i</sub>MR register (i=0 to 4) determine which mode is used.

- Timer mode: The timer counts an internal count source.
- Event counter mode: The timer counts an external pulse or overflow and underflow of other timers.
- One-shot timer mode: The timer outputs one valid pulse until the counter reaches "000016".
- Pulse width modulation mode: The timer consecutively outputs a given pulse width.

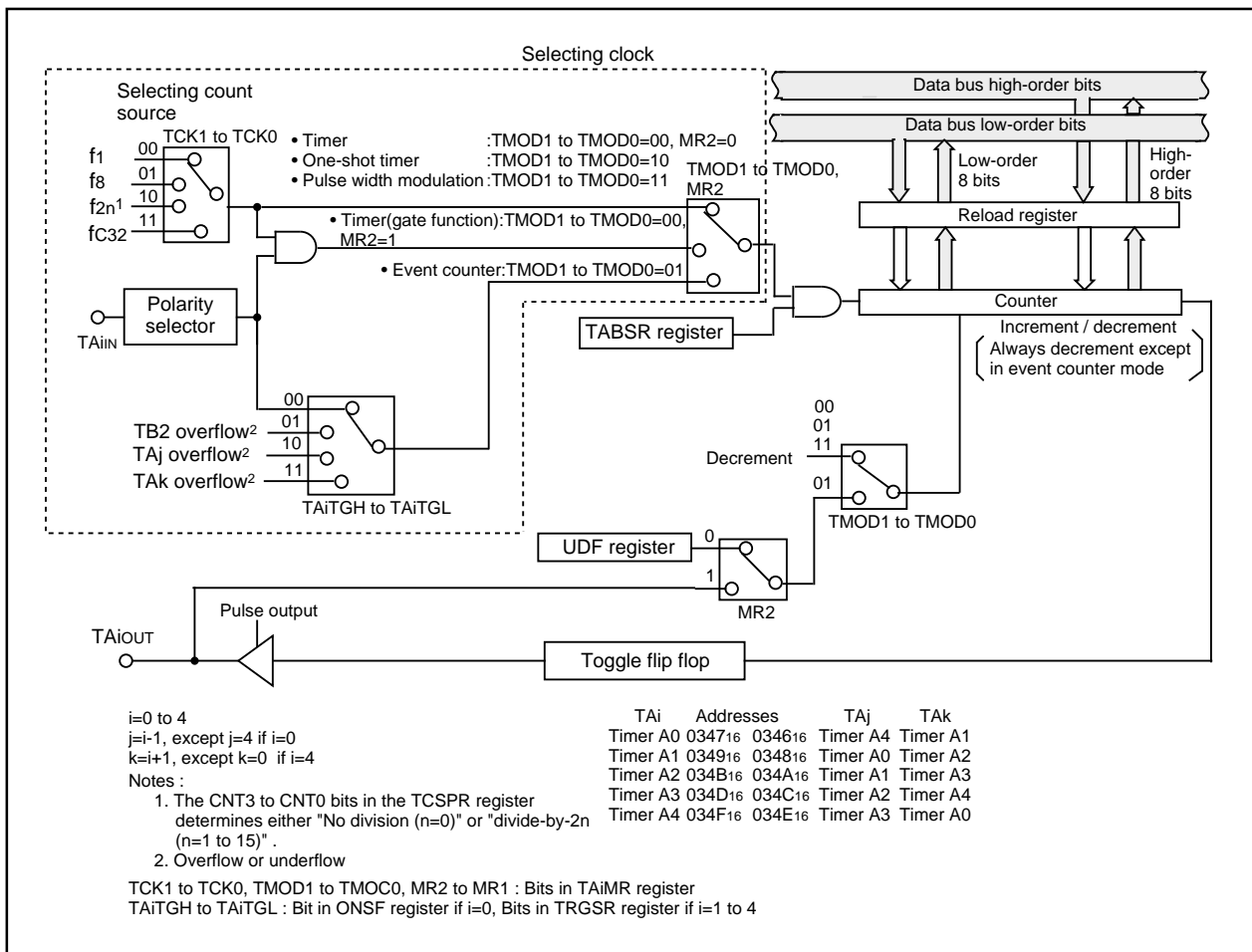
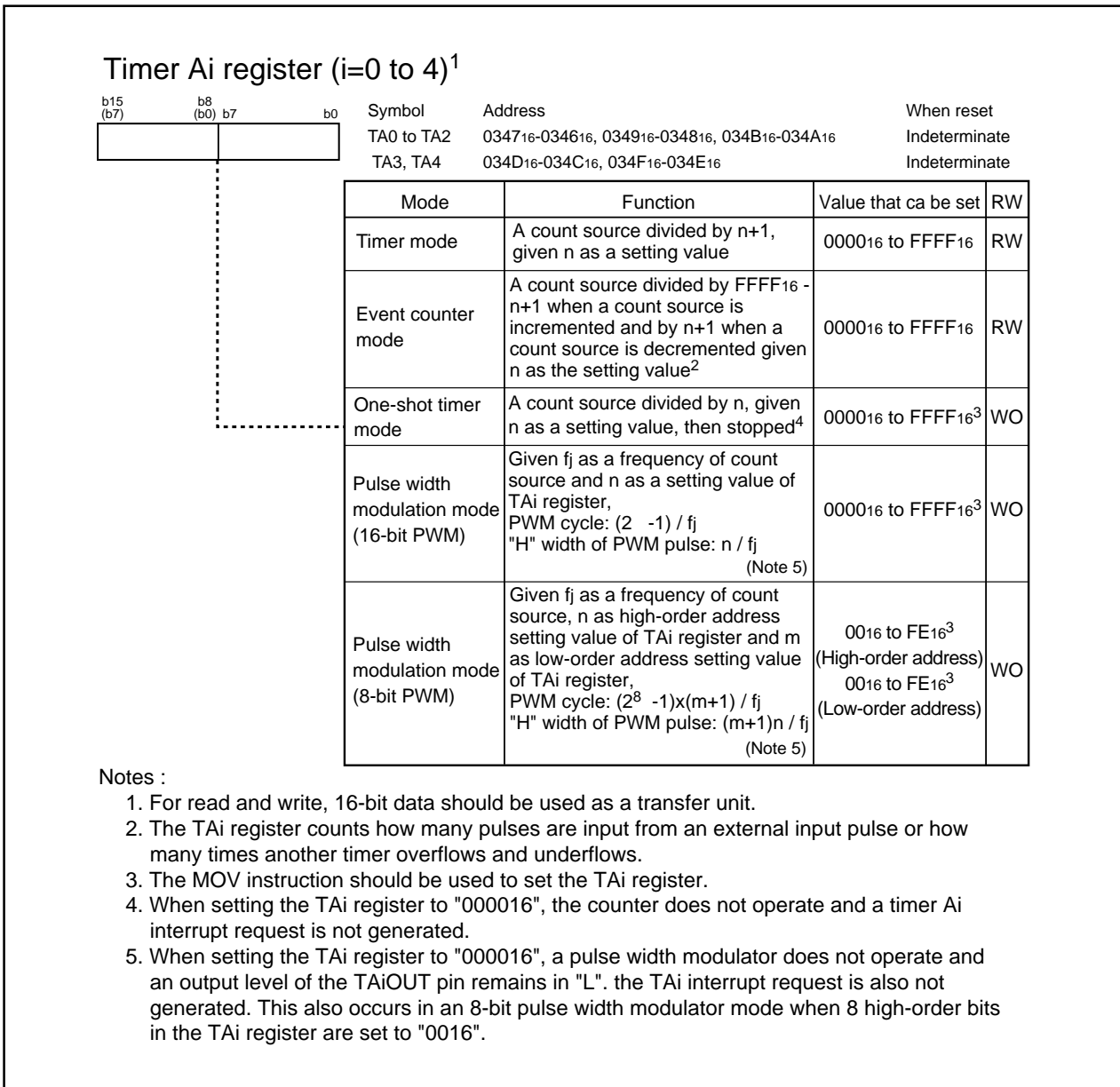


Figure 1.14.1. Timer A Block Diagram

Timer (Timer A)



**Figure 1.14.2. TA0 to TA4 Registers**

Timer (Timer A)

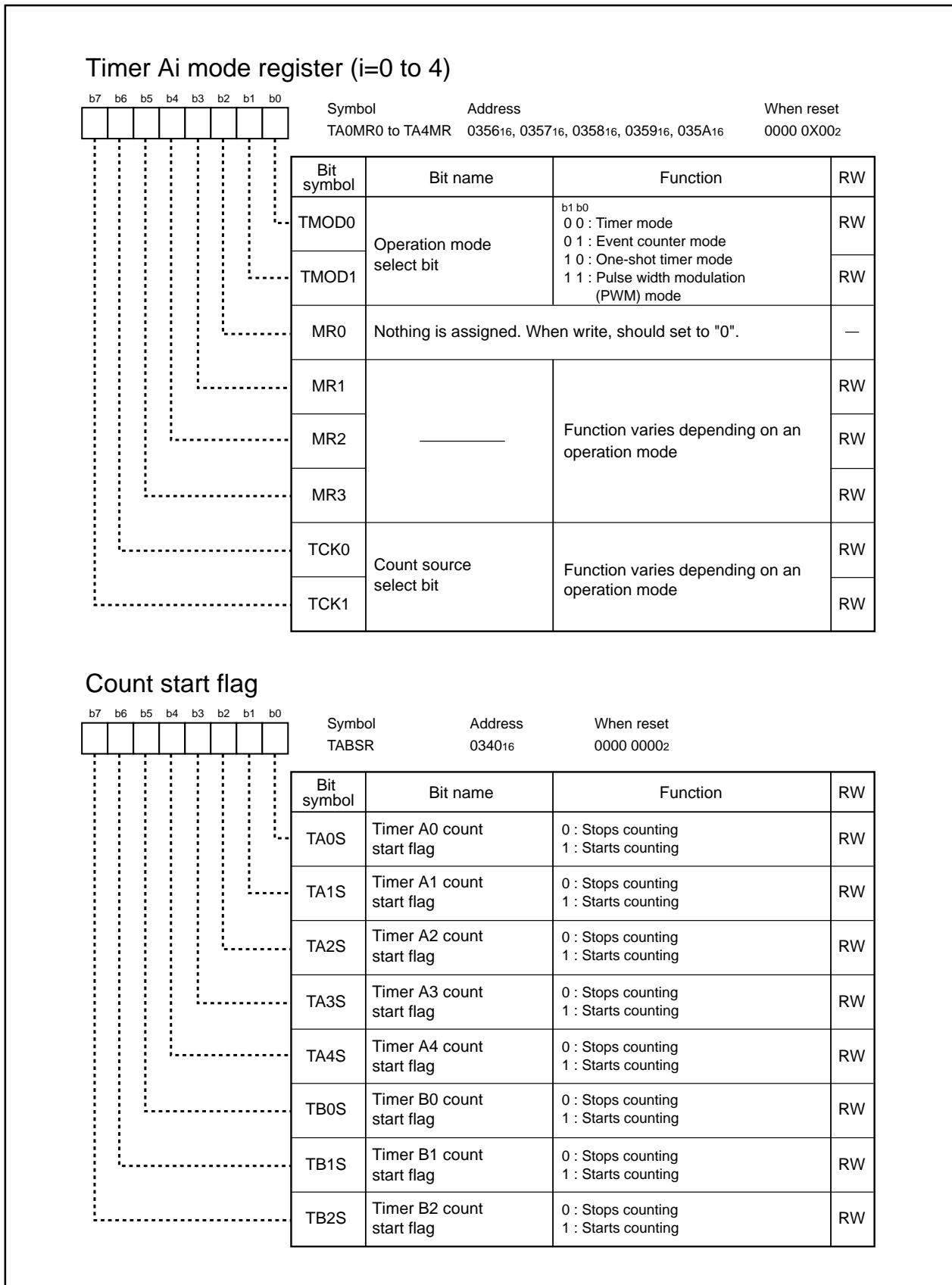


Figure 1.14.3. TA0MR to TA4MR Registers and TABSR Register

Timer (Timer A)

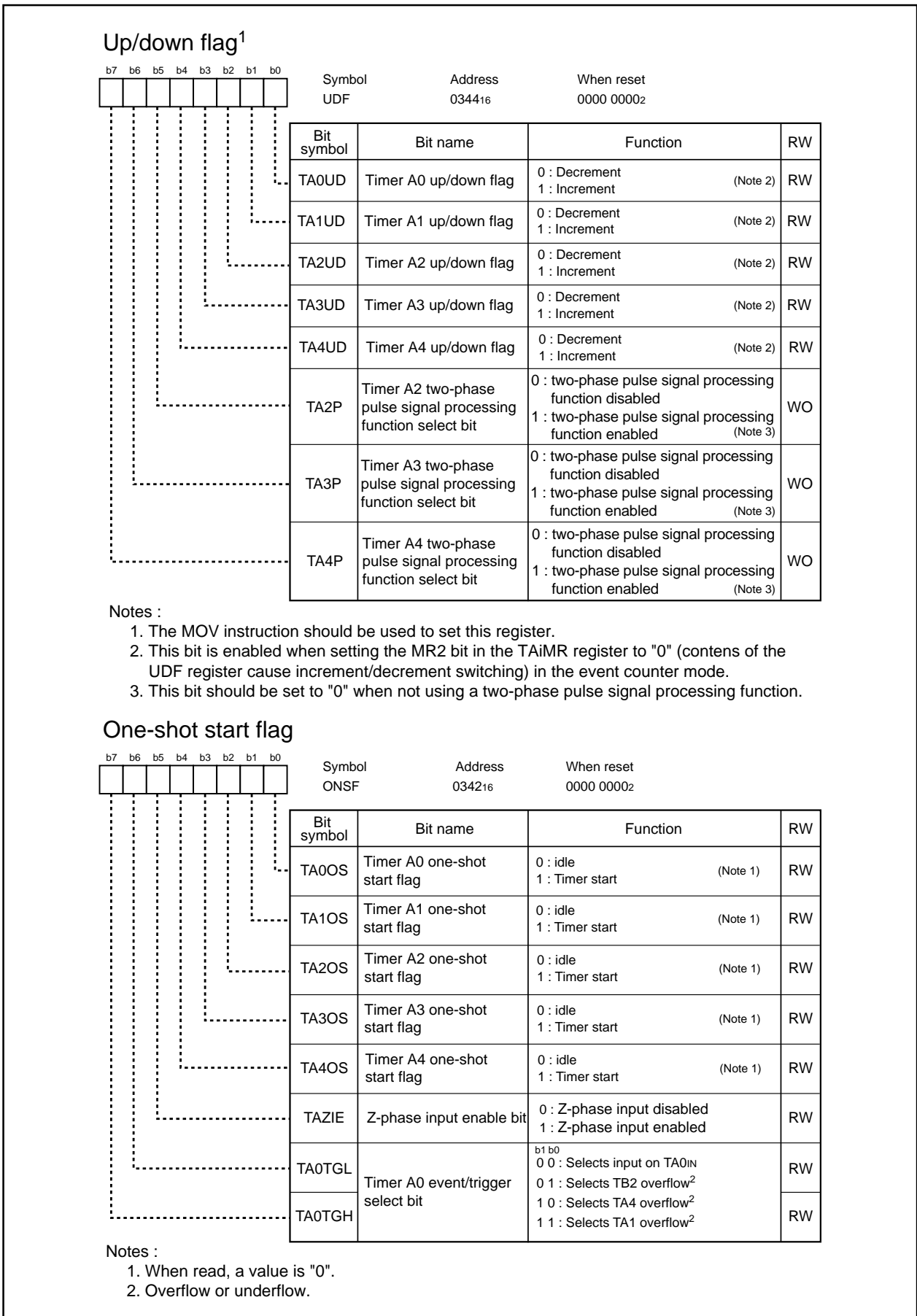


Figure 1.14.4. UDF Register and ONSF Register

Timer (Timer A)

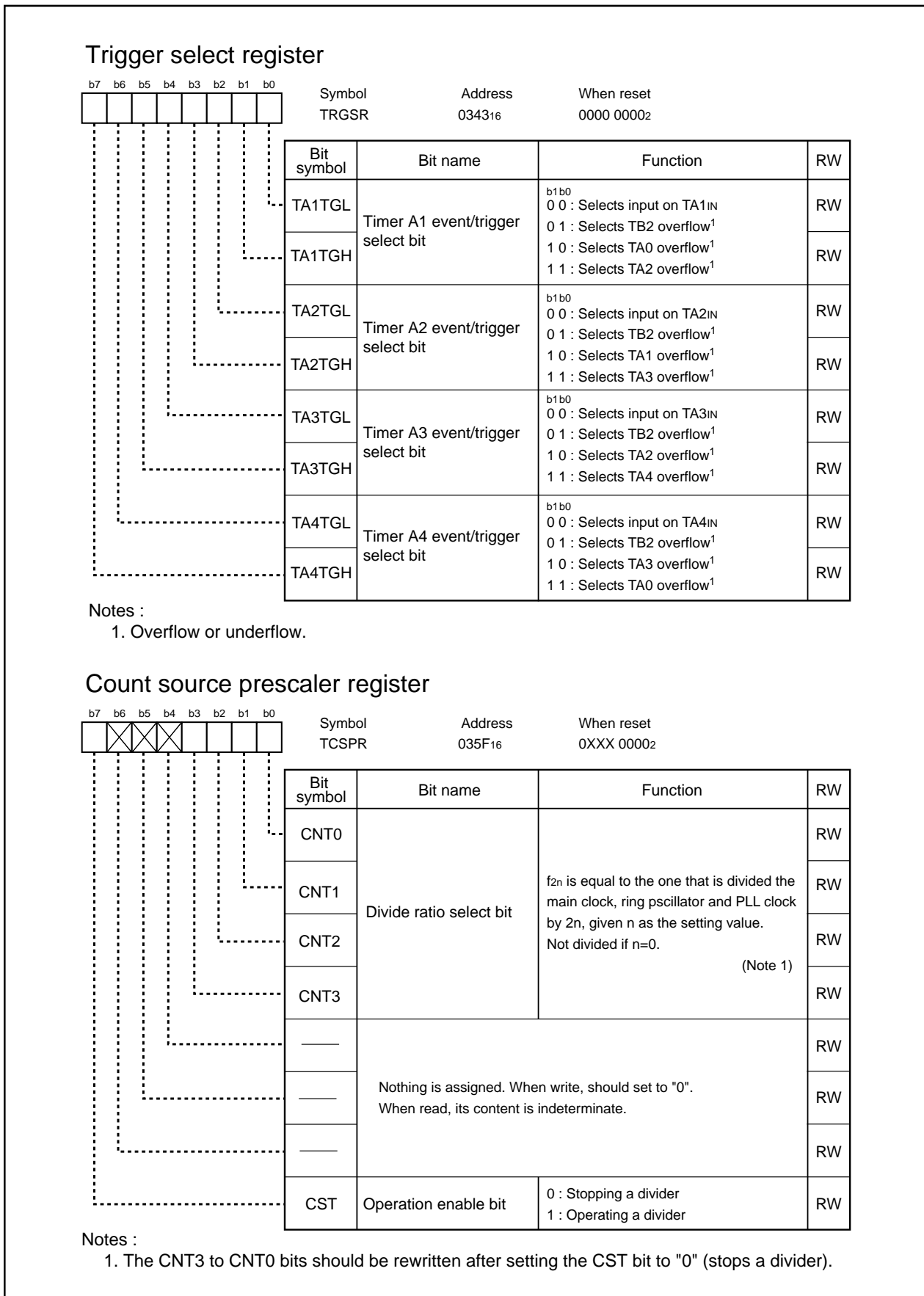


Figure 1.14.5. TRGSR Register and TCSPR Register

## Timer (Timer A)

Table 1.14.1. Pin Settings for Output from TAIOUT Pin

Pin	Bit and setting values		
	PS1, PS2 registers	PSL1, PSL2 registers	PSC register
P70/TA0OUT <sup>1</sup>	PS1_0= 1	PSL1_0=1	-
P72/TA1OUT	PS1_2= 1	PSL1_2=1	-
P74/TA2OUT	PS1_4= 1	PSL1_4=0	PSC_4= 0
P76/TA3OUT	PS1_6= 1	PSL1_6=1	-
P80/TA4OUT	PS2_0= 1	PSL2_0=0	-

Notes :

1. N-channel open drain

Table 1.14.2. Pin Settings for Input from TAIIN and TAIOUT Pins

Pin	Bits and setting values	
	PS1, PS2 registers	PD7, PD8 registers
P70/TA0OUT	PS1_0=0	PD7_0=0
P71/TA0IN	PS1_1=0	PD7_1=0
P72/TA1OUT	PS1_2=0	PD7_2=0
P73/TA1IN	PS1_3=0	PD7_3=0
P74/TA2OUT	PS1_4=0	PD7_4=0
P75/TA2IN	PS1_5=0	PD7_5=0
P76/TA3OUT	PS1_6=0	PD7_6=0
P77/TA3IN	PS1_7=0	PD7_7=0
P80/TA4OUT	PS2_0=0	PD8_0=0
P81/TA4IN	PS2_1=0	PD8_1=0

## Timer (Timer A)

### 1. Timer Mode

In timer mode, the timer counts an internally generated count source (see Table 1.14.3). Figure 1.14.6 shows the TAI<sub>MR</sub> register in timer mode.

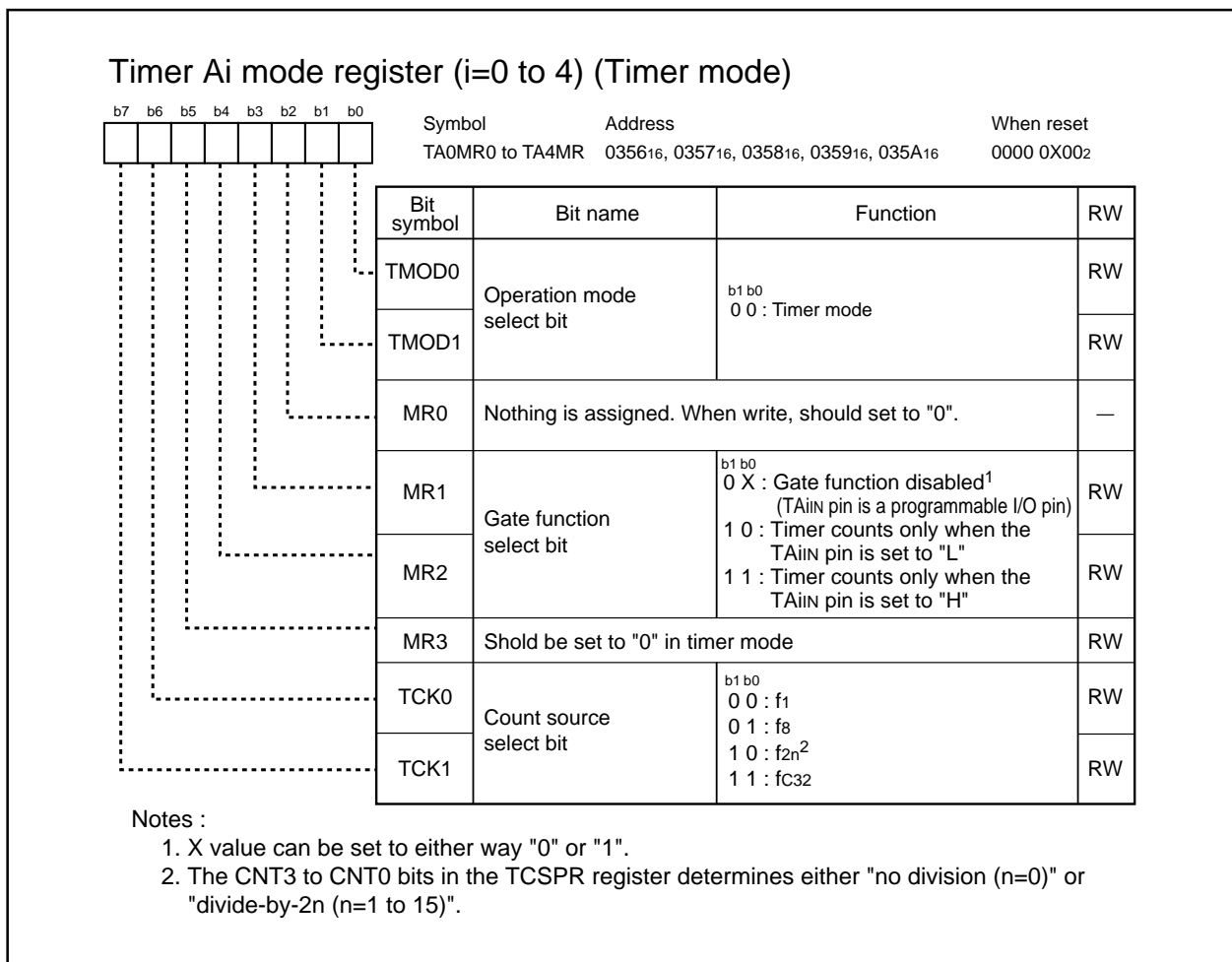
**Table 1.14.3. Specifications in Timer Mode**

Item	Specification
Count source	f <sub>1</sub> , f <sub>8</sub> , f <sub>2<sup>n</sup></sub> <sup>1</sup> , f <sub>C32</sub>
Count operation	<ul style="list-style-type: none"> <li>The timer decrements the counter</li> <li>When the timer underflows, it reloads contents of the reload register into ones of the count register to continue counting.</li> </ul>
Divide ratio	1/(n+1)    n: setting value of TAI register (i=0 to 4) 0000 <sub>16</sub> to FFFF <sub>16</sub>
Count start condition	The TAI <sub>S</sub> bit in the TABSR register is set to "1" (starts counting)
Count stop condition	The TAI <sub>S</sub> bit is set to "0" (stops counting)
Interrupt request generation timing	Timer underflows
TAI <sub>IN</sub> pin function	Programmable I/O port or gate input
TAI <sub>OUT</sub> pin function	Programmable I/O port or pulse output
Read from timer	The Ai register indicates a value of the counter
Write to timer	<ul style="list-style-type: none"> <li>When the counter stops or before the first count source prior to starting of the counter is input A value written to the TAI register is also written to both reload register and counter</li> <li>While counting A value written to the TAI register is written to the reload register only (Transferred a value to the counter at the next reload time)</li> </ul>
Selectable function	<ul style="list-style-type: none"> <li>Gate function Input signal to the TAI<sub>IN</sub> pin determines whether the timer starts or stops counting</li> <li>Pulse output function A polarity of the TAI<sub>OUT</sub> pin is inversed whenever the timer underflows</li> </ul>

Notes :

- The CNT<sub>3</sub> to CNT<sub>0</sub> bits in the TRGSR register determines either "no division (n=0)" or "divide-by-2<sup>n</sup> (n=1 to 15)".

Timer (Timer A)



**Figure 1.14.6. TA0MR to TA4MR Registers**

## Timer (Timer A)

**(2) Event Counter Mode**

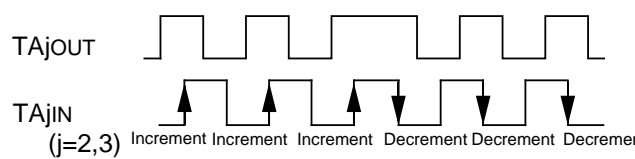
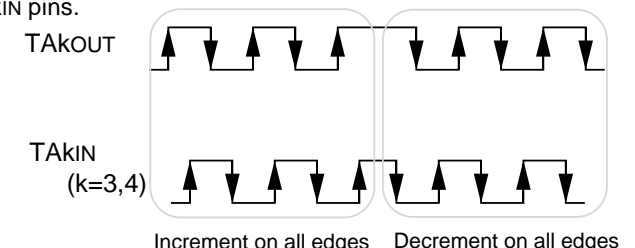
In event counter mode, the timer counts how many external signals are input or how many times another timer overflows and underflows. The timers A2, A3 and A4 can count an external signal two-phase. Table 1.14.4 lists specifications in event counter mode (when not processing two-phase pulse signal). Table 1.14.5 lists specifications in event counter mode (when processing two-phase pulse signal on the timer A2, A3 and A4). Figure 1.14.7 shows the TAI<sub>MR</sub> register in event counter mode.

**Table 1.14.4. Specifications in Event Counter Mode (when not processing two-phase pulse signal)**

Item	Specification
Count source	<ul style="list-style-type: none"> <li>External signal input to the TAI<sub>IN</sub> pin (i = 0 to 4) (valid edge can be selected by program)</li> <li>The timer B2 overflows or underflows, timer A<sub>j</sub> overflows or underflows (j=i-1, except j=4 if i=0) and timer A<sub>k</sub> overflows or underflows (k=i+1, except k=0 if i=4)</li> </ul>
Count operation	<ul style="list-style-type: none"> <li>Determines whether the timer increments or decrements the counter</li> <li>When the timer overflows or underflows, it reloads contents of the reload register into ones of the count register to continue counting. With the free-running count function, the timer continues counting without a reloading operation</li> </ul>
Divide ratio	<ul style="list-style-type: none"> <li>1/(FFFF<sub>16</sub> - n + 1) for a counter increment</li> <li>1/(n + 1) for a counter decrement    n : setting value of the TAI register 0000<sub>16</sub> to FFFF<sub>16</sub></li> </ul>
Count start condition	The TAI <sub>S</sub> bit is set to "1" (starts counting)
Count stop condition	The TAI <sub>S</sub> bit is set to "0" (stops counting)
Interrupt request generation timing	The timer overflows or underflows
TAI <sub>IN</sub> pin function	Programmable I/O port or count source input
TAI <sub>OUT</sub> pin function	Programmable I/O port, pulse output or input to a select the counter increment/decrement
Read from timer	The A <sub>i</sub> register indicates a value of the counter
Write to timer	<ul style="list-style-type: none"> <li>When the counter stops or before the first count source prior to starting of the counter is input A value written to the TAI register is also written to both reload register and counter.</li> <li>While counting A value written to the TAI register is written to reload register only (Transferred a value to the counter at the next reload time).</li> </ul>
Selectable function	<ul style="list-style-type: none"> <li>Free-running count function Contents of the reload register are not reloaded even if the The timer overflows or underflows</li> <li>Pulse output function A polarity of the TAI<sub>OUT</sub> is inversed whenever the timer overflows or underflows</li> </ul>

Timer (Timer A)

**Table 1.14.5. Specifications in Event Counter Mode (when processing two-phase pulse signal on timer A2, A3 and A4)**

Item	Specification
Count source	Two-phase pulse signal input to the TAI <sub>i</sub> N or TAI <sub>i</sub> OUT pin (i = 2 to 4)
Count operation	<ul style="list-style-type: none"> <li>• Determines whether the timer increments or decrements the counter</li> <li>• When the timer overflows or underflows, it reloads contents of the reload register into ones of the count register to continue counting. With the free-running count function, the timer continues counting without a reloading operation</li> </ul>
Divide ratio	<ul style="list-style-type: none"> <li>• <math>1 / (FFFF_{16} - n + 1)</math> for a counter increment</li> <li>• <math>1 / (n + 1)</math> for a counter decrement      n : setting value of the TAI register</li> </ul>
Count start condition	The TAI <sub>S</sub> bit is set to "1" (starts counting)
Count stop condition	The TAI <sub>S</sub> bit is set to "0" (stops counting)
Interrupt request generation timing	The timer overflows or underflows
TAI <sub>i</sub> N pin function	Two-phase pulse input
TAI <sub>i</sub> OUT pin function	Two-phase pulse input
Read from timer	The Ai register indicates a value of the counter
Write to timer	<ul style="list-style-type: none"> <li>• When the counter stops or before the first count source prior to starting of the counter is input                      A value written to the TAI register is also written to both reload register and counter</li> <li>• While counting                      A value written to the TAI register is written to the reload register only (Transferred to counter at next reload time).</li> </ul>
Selectable function <sup>1</sup>	<ul style="list-style-type: none"> <li>• Normal processing operation (the timer A2 and timer A3)                      While input signal to the TAJ<sub>i</sub>OUT pin is set to "H" the timer increments the counter on the rising edge of the TAJ<sub>i</sub>IN pin or decrements the counter on the falling edge</li> </ul>  <ul style="list-style-type: none"> <li>• Multiplied-by-4 processing operation (the timer A3 and timer A4)                      When an input signal to the TAK<sub>i</sub>OUT pin is set to "H" with the rising edge of the TAK<sub>i</sub>IN pin, the timer increments the counter on the rising and falling edges of the TAK<sub>i</sub>OUT and TAK<sub>i</sub>IN pins.                      When an input signal to the TAK<sub>i</sub>OUT pin is set to "H" with the falling edge of the TAK<sub>i</sub>IN pin, the timer decrements the counter on the rising and falling edges of the TAK<sub>i</sub>OUT and TAK<sub>i</sub>IN pins.</li> </ul> 

Notes :

1. Only timer A3 is selectable. The timer A2 is fixed to normal processing operation. The timer A4 is fixed to multiplied-by-4 operation.

Timer (Timer A)

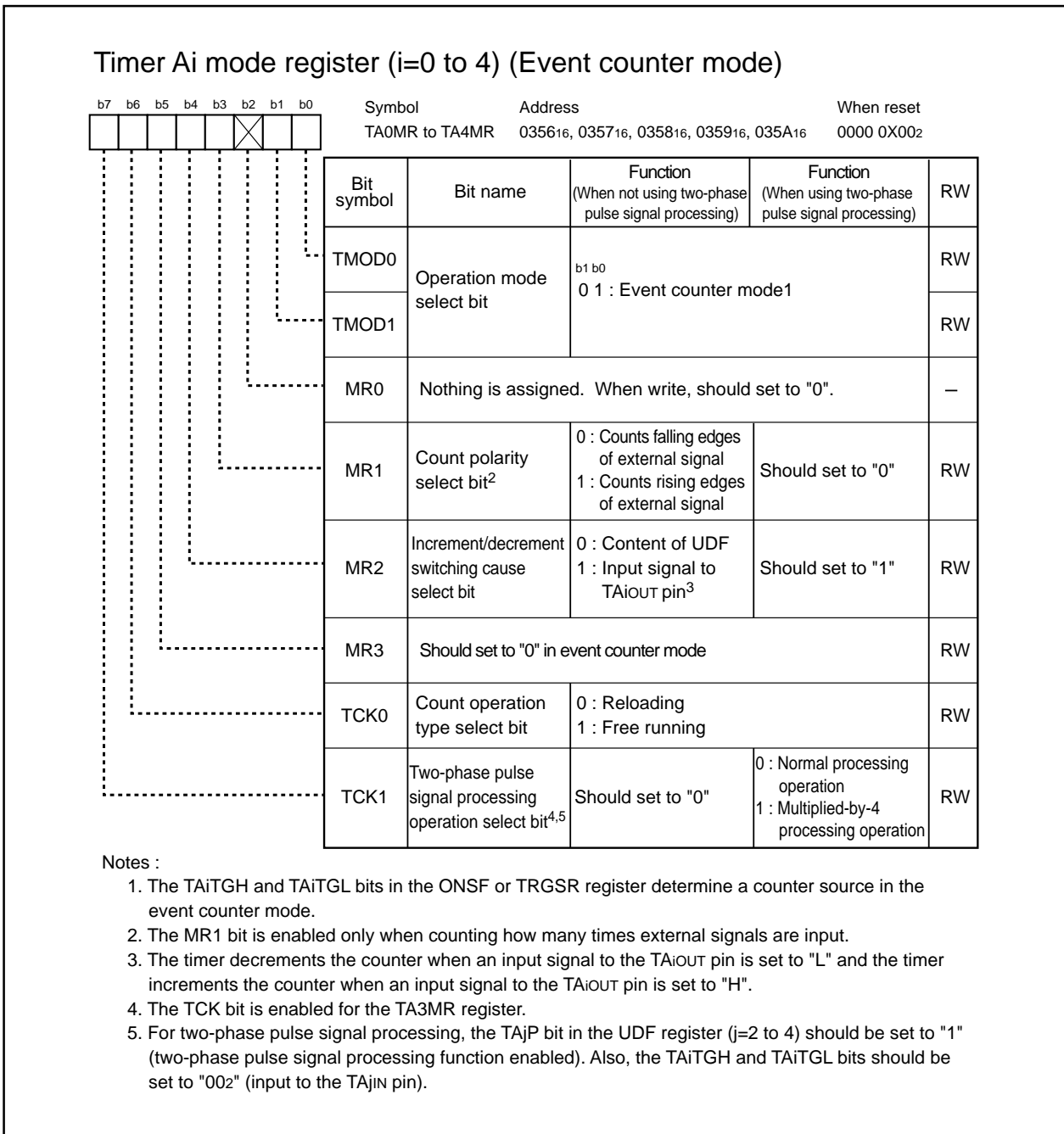


Figure 1.14.7. TA0MR to TA4MR Registers

**Timer (Timer A)**

• **Counter Reset by Two-Phase Pulse Signal Processing**

The timer counter is reset to "0" by the Z-phase (counter reset) input when processing a two-phase pulse signal.

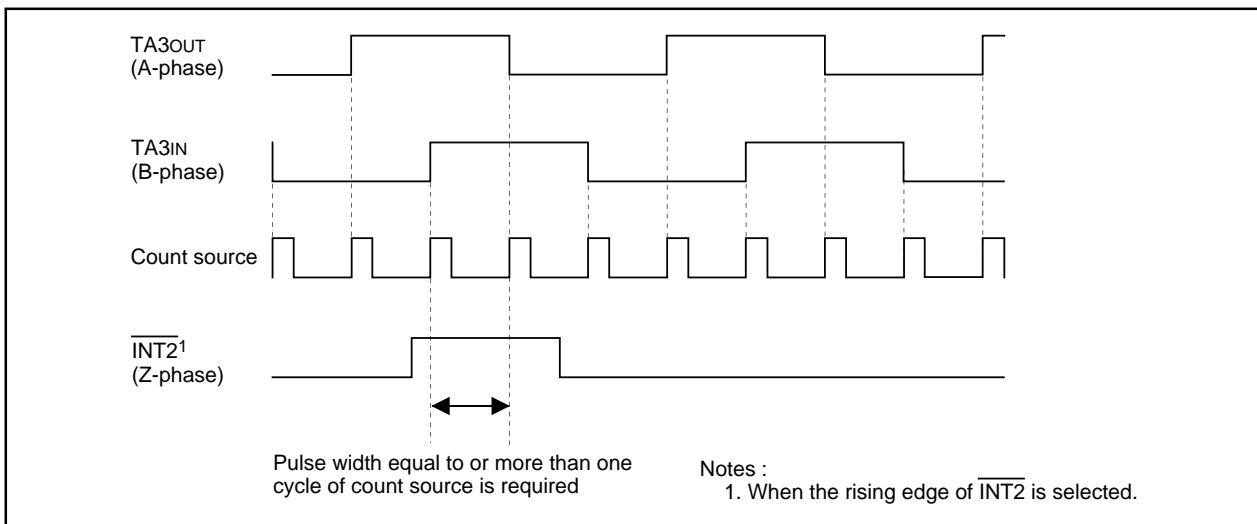
This function can be used in timer A3 event counter mode, two-phase pulse signal processing, free-run type and multiplied-by-4 processing. The Z-phase is input to the  $\overline{\text{INT2}}$  pin.

When the TAZIE bit in the ONSF register is set to "1" (Z-phase input enabled), the counter is enabled by a Z-phase input. When the counter is reset to "0" by Z-phase input, the TA3 register should be set to "0000<sub>16</sub>" initially.

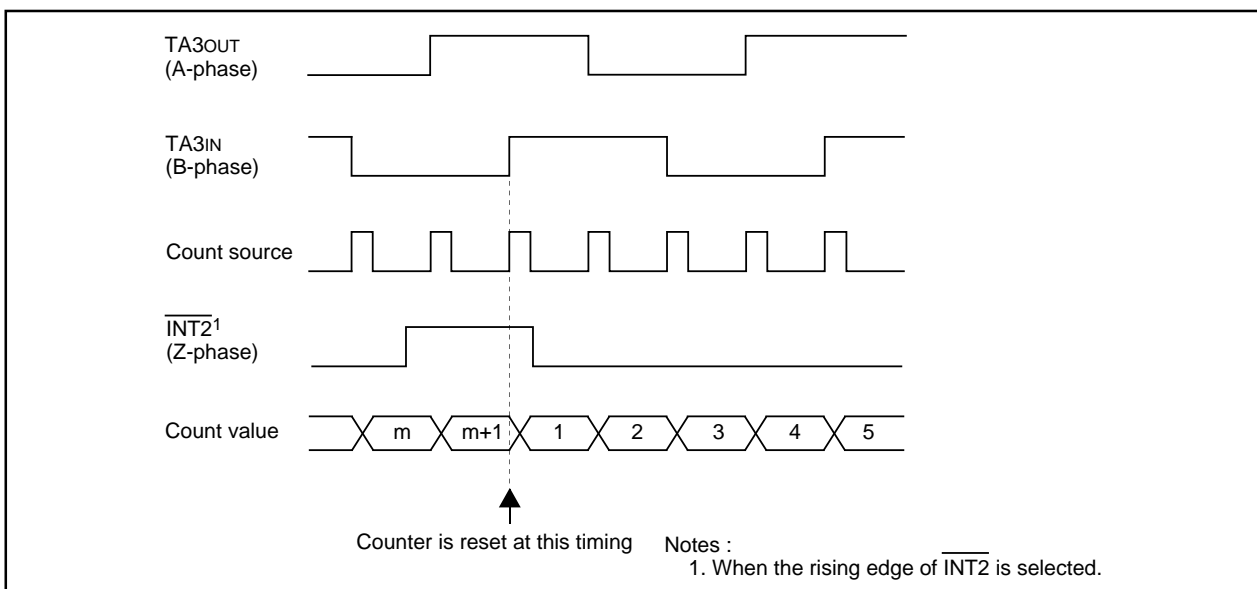
Z-phase input is enabled when an edge of the  $\overline{\text{INT2}}$  input is detected. The POL bit in the INT2IC register determines an edge polarity. The Z-phase should have a pulse width equal to or more than one cycle of timer A3 count source. Figure 1.14.8 shows two-phase pulses (A-phase and B-phase) and the Z-phase.

Z-phase input resets the counter in the next count source following Z-phase input. Figure 1.14.9 shows the counter reset timing.

Timer A3 interrupt request is generated twice when a timer A3 overflow and underflow coincide with counter reset by  $\overline{\text{INT2}}$  input. Avoid using a timer A3 interrupt request when this function is used.



**Figure 1.14.8. Two-phase Pulse (A-phase and B-phase) and the Z-phase**



**Figure 1.14.9. Counter Reset Timing**

## Timer (Timer A)

### 3. One-shot Timer Mode

In one-shot timer mode, the timer operates only once in response to one trigger (see Table 1.14.6). When a trigger occurs, the timer starts up and continues operating for a given period. Figure 1.14.10 shows the TAI<sub>MR</sub> register (i=0 to 4) in one-shot timer mode.

**Table 1.14.6. Specifications in One-shot Timer Mode**

Item	Specification
Count source	f <sub>1</sub> , f <sub>8</sub> , f <sub>2<sup>n</sup></sub> <sup>1</sup> , f <sub>C32</sub>
Count operation	<ul style="list-style-type: none"> <li>The timer decrements the counter</li> <li>When the counter reaches "0000<sub>16</sub>", the timer stops counting after reloading</li> <li>If a trigger occurs while counting, the timer reloads to resume counting</li> </ul>
Divide ratio	1/n    n : setting value of the TAI register (i=0 to 4) 0000 <sub>16</sub> to FFFF <sub>16</sub> , the counter do not run if n=0000 <sub>16</sub>
Count start condition	<ul style="list-style-type: none"> <li>The TAI<sub>S</sub> bit in the TABSR register is set to "1" (starts counting) and following triggers are generated</li> <li>External trigger is input</li> <li>The timer overflows and underflows</li> <li>The TAI<sub>OS</sub> bit in the ONSF register is set to "1" (timer starts)</li> </ul>
Count stop condition	<ul style="list-style-type: none"> <li>Reload after the counter has reached "0000<sub>16</sub>"</li> <li>The TAI<sub>OS</sub> bit is set to "0" (timer stops)</li> </ul>
Interrupt request generation timing	Counter reaches "0000 <sub>16</sub> "
TAI <sub>IN</sub> pin function	Programmable I/O port or trigger input
TAI <sub>OUT</sub> pin function	Programmable I/O port or pulse output
Read from timer	Value is indeterminate by reading the TAI register
Write to timer	<ul style="list-style-type: none"> <li>When the counter stops or before the first count source prior to starting of the counter is input A value written to the TAI register is also written to both reload register and counter.</li> <li>When counting is in progress A value written to the TAI register is written to the reload register only. (Transferred a value to counter at the next reload time.)</li> </ul>

## Notes :

- The CNT3 to CNT0 bits in the TRGSR register determines either "no division (n=0)" or "divide-by-2<sup>n</sup> (n=1 to 15)".

Timer (Timer A)

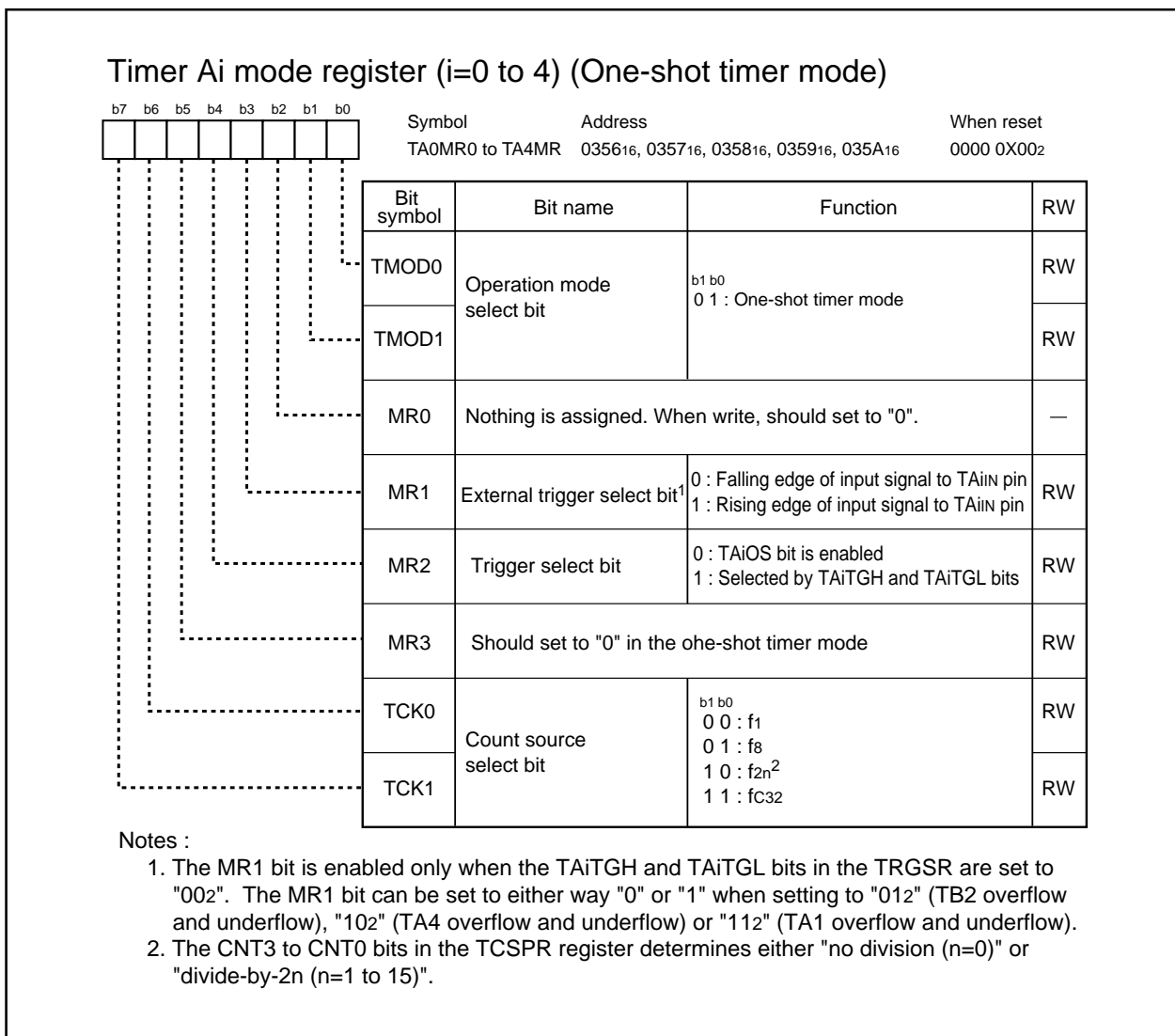


Figure 1.14.10. TA0MR to TA4MR Registers

## Timer (Timer A)

#### 4. Pulse Width Modulation Mode

In pulse width modulation mode, the timer outputs pulses of a given width continuously (see Table 1.14.7). The counter functions as either 16-bit pulse width modulator or 8-bit pulse width modulator. Figure 1.14.11 shows the TAI<sub>MR</sub> register (i=0 to 4) in pulse width modulation mode. Figures 1.14.12 and 1.14.13 show examples of how a 16-bit pulse width modulator operates and of how an 8-bit pulse width modulator operates.

**Table 1.14.7. Specifications in Pulse Width Modulation Mode**

Item	Specification
Count source	f <sub>1</sub> , f <sub>8</sub> , f <sub>2n</sub> <sup>1</sup> , f <sub>C32</sub>
Count operation	<ul style="list-style-type: none"> <li>The timer decrements the counter (operating as an 8-bit or a 16-bit pulse width modulator)</li> <li>The timer reloads on the rising edge of PWM pulse and continues counting</li> <li>The timer is not affected by a trigger that occurs during counting</li> </ul>
16-bit PWM	<ul style="list-style-type: none"> <li>"H" width = <math>n / f_i</math> <ul style="list-style-type: none"> <li>n : setting value of TAI register 0000<sub>16</sub> to FFFF<sub>16</sub></li> <li>f<sub>i</sub> : Count source frequency</li> </ul> </li> <li>Cycle = <math>(2^{16}-1) / f_i</math> fixed</li> </ul>
8-bit PWM	<ul style="list-style-type: none"> <li>"H" width = <math>n \times (m+1) / f_j</math> <ul style="list-style-type: none"> <li>n : setting value of high-order address of the TAI register 00<sub>16</sub> to FF<sub>16</sub></li> </ul> </li> <li>Cycles = <math>(2^8-1) \times (m+1) / f_j</math> <ul style="list-style-type: none"> <li>m : setting value of low-order address of the TAI register 00<sub>16</sub> to FF<sub>16</sub></li> </ul> </li> </ul>
Count start condition	<ul style="list-style-type: none"> <li>External trigger is input</li> <li>The timer overflows and underflows</li> <li>The TAI<sub>S</sub> bit in the TABSR register is set to "1" (start counting)</li> </ul>
Count stop condition	The TAI <sub>S</sub> bit is set to "0" (stop counting)
Interrupt request generation timing	Falling edge of PWM pulse
TAJ <sub>IN</sub> pin function	Programmable I/O port or trigger input
TAJ <sub>OUT</sub> pin function	Pulse output
Read from timer	The Ai register indicates a value of the counter
Write to timer	<ul style="list-style-type: none"> <li>When the counter stops <ul style="list-style-type: none"> <li>A value written to the TAI register is also written to both reload register and counter.</li> </ul> </li> <li>While counting <ul style="list-style-type: none"> <li>A value written to the TAI register is written to the reload register only.</li> <li>(Transferred a value to the counter at the next reload time.)</li> </ul> </li> </ul>

Notes :

- The CNT3 to CNT0 bits in the TRGSR register determines either "no division (n=0)" or "divide-by-2n (n=1 to 15)".

Timer (Timer A)

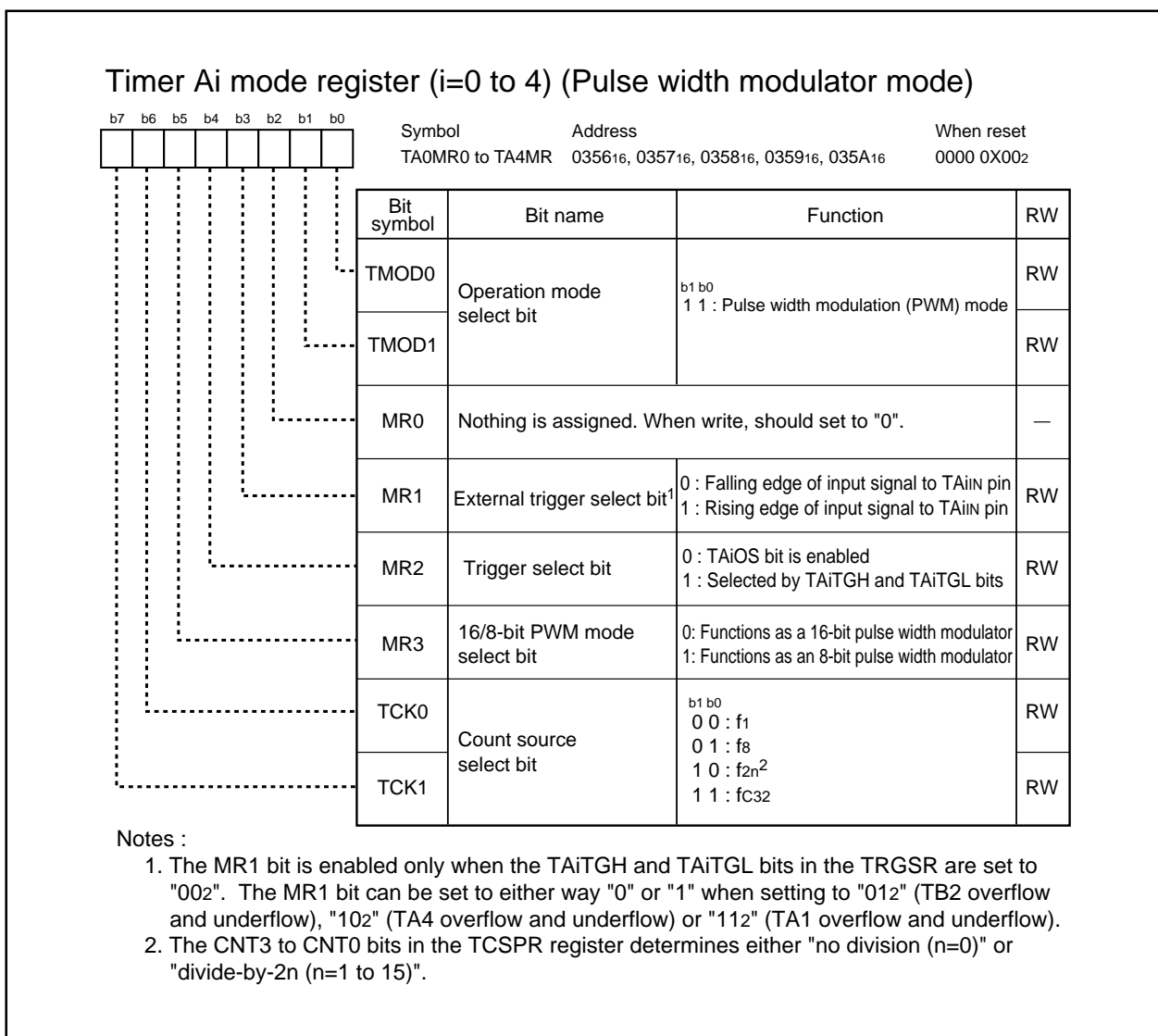


Figure 1.14.11. TA0MR to TA4MR Registers

Timer (Timer A)

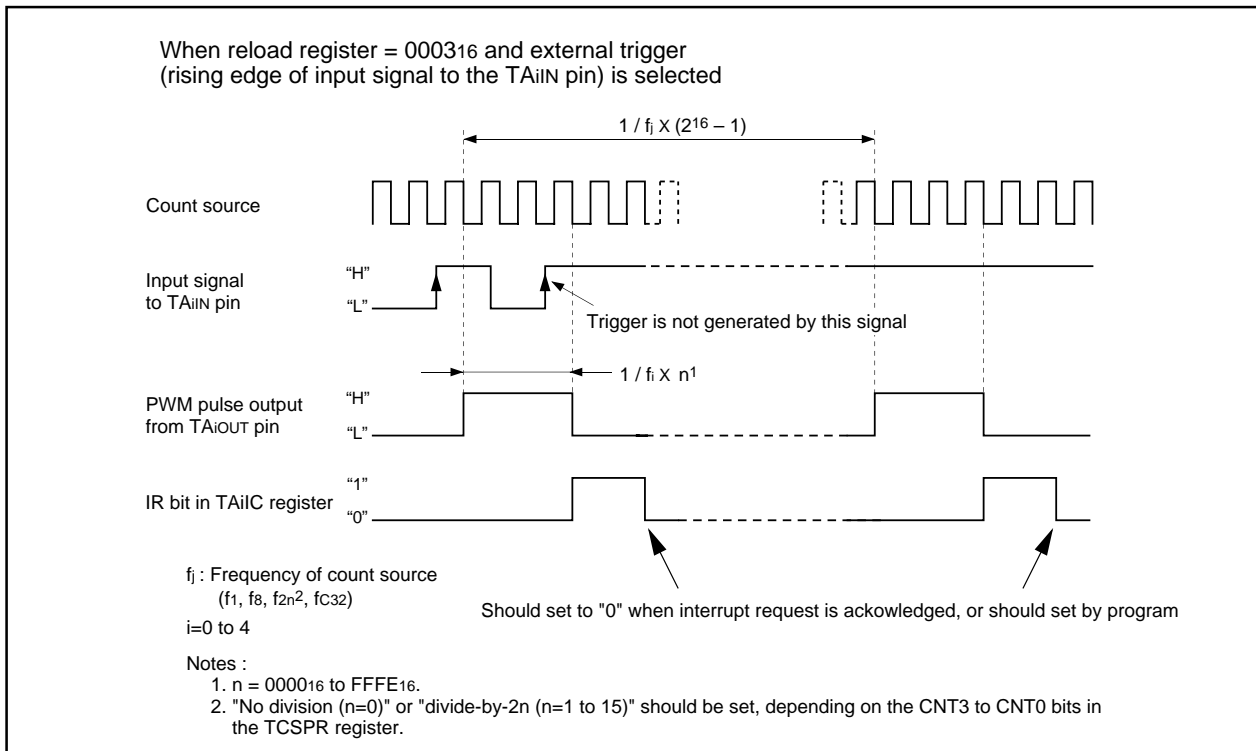


Figure 1.14.12. 16-bit Pulse Width Modulator Operation

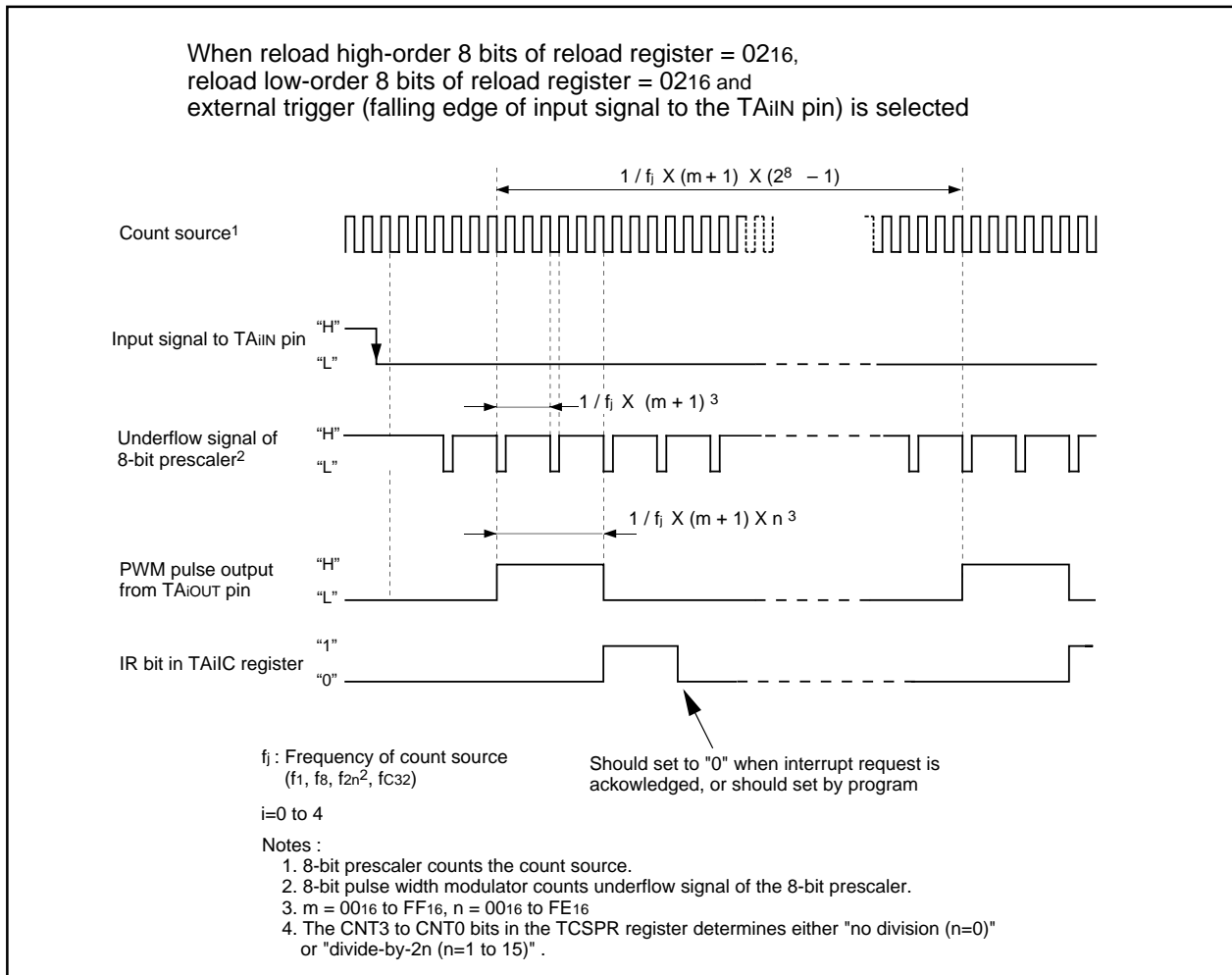


Figure 1.14.13. 8-bit Pulse Width Modulator Operation

# Timer (Timer B)

## Timer B

Figure 1.15.1 shows a block diagram of the timer B. Figures 1.15.2 and 1.15.4 show registers associated with the timer B. The timer B operates in three modes below. The TMOD1 to TMOD0 bits in the TBiMR register ( $i=0$  to 5) determine which mode is used.

- Timer mode : The timer counts an internal count source.
- Event counter mode : The timer counts pulses from an external source or overflow and underflow of another timer.
- Pulse period/pulse width measuring mode : The timer measures pulse period or pulse width of an external signal.

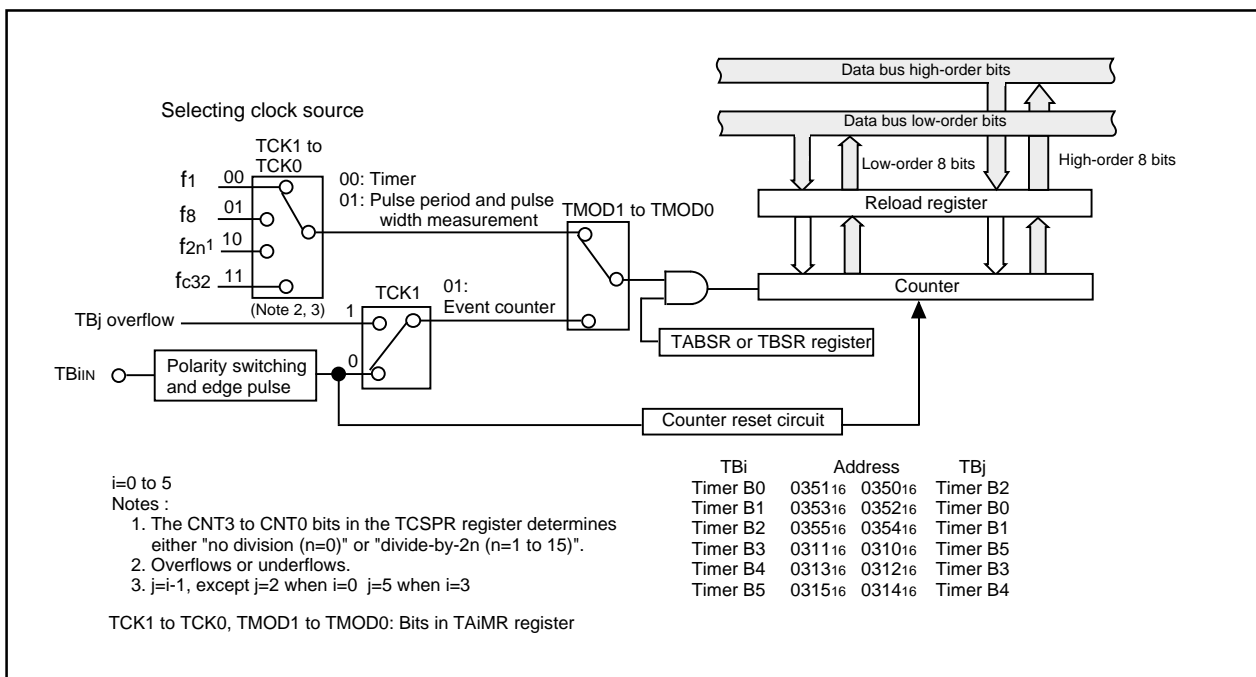


Figure 1.15.1. Timer B Block Diagram

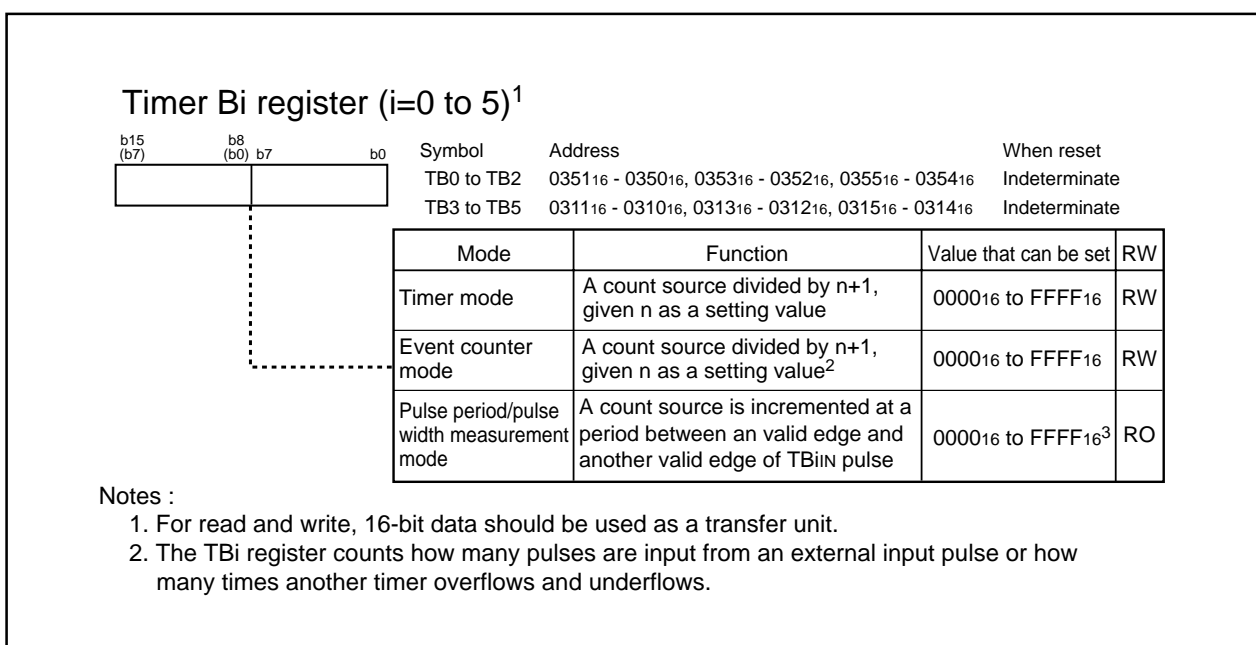


Figure 1.15.2. TB0 to TB5 Registers

Timer (Timer B)

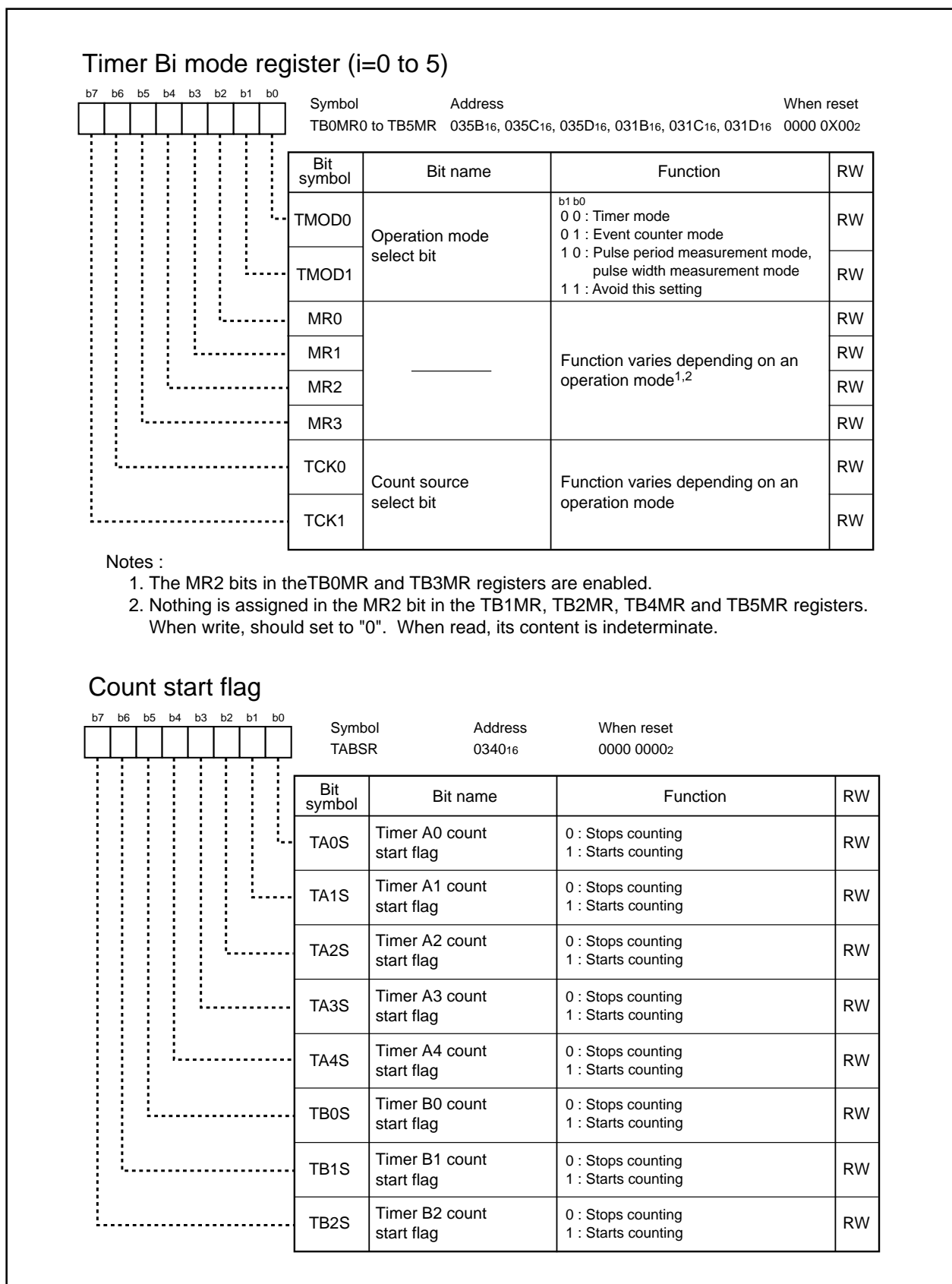


Figure 1.15.3. TB0MR to TB5MR Registers, TABSR Register

Timer (Timer B)

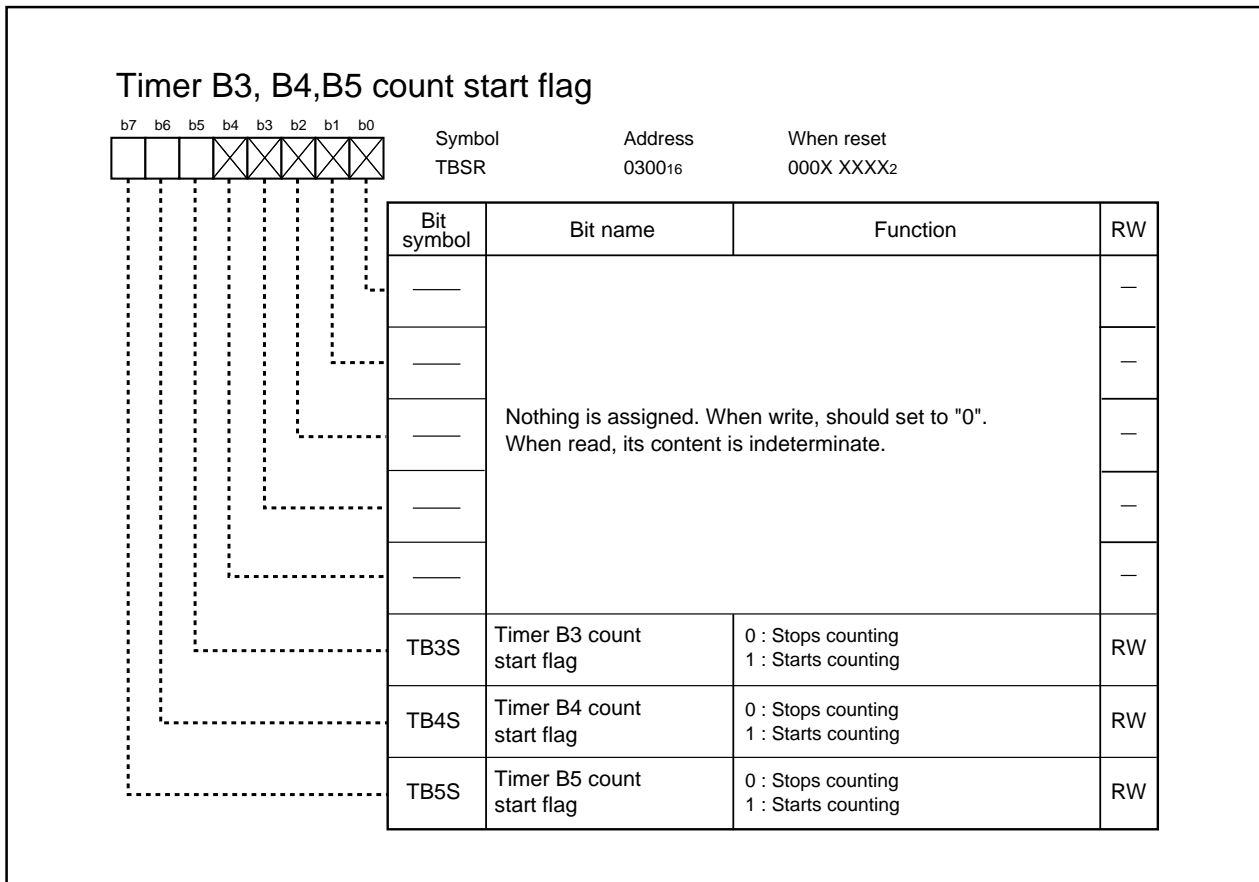


Figure 1.15.4. TBSR Register

Table 1.15.1. Settings for the TB*i*IN Pins (i=0 to 5)

Port name	Function	Bits and Setting values	
		PS1, PS3 <sup>1</sup> registers	PD7, PD9 <sup>1</sup> registers
P90	TB0IN	PS3_0=0	PD9_0=0
P91	TB1IN	PS3_1=0	PD9_1=0
P92	TB2IN	PS3_2=0	PD9_2=0
P93	TB3IN	PS3_3=0	PD9_3=0
P94	TB4IN	PS3_4=0	PD9_4=0
P71	TB5IN	PS1_1=0	PD7_1=0

Notes :

1. The PD9 and PS3 registers should be set with subsequent to the instruction that the PRC2 bit in the PRCR register is set to "1" (write enable). Avoid generating an interrupt and operating a DMA transfer between the instruction to set the PRC2 bit to "1" and the instruction to write to the PD9 and PS3 registers.

Timer (Timer B)

1. Timer Mode

In timer mode, the timer counts an internally generated count source (see Table 1.15.2). Figure 1.15.5 shows the TBiMR register (i=0 to 5) in timer mode.

Table 1.15.2. Specifications in Timer Mode

Item	Specification
Count source	f1, f8, f2n <sup>1</sup> , fc32
Count operation	<ul style="list-style-type: none"> <li>The timer decrements the counter</li> <li>When the timer overflows or underflows, it reloads contents of the reload register into ones of the count register to continue counting with the free-running count function, the timer continue counting without a reloading operation.</li> </ul>
Divide ratio	1/(n+1)    n : setting value of the TBi register (i=0 to 5) 0000 <sub>16</sub> to FFFF <sub>16</sub>
Count start condition	The TBiS bit in the TABSR or TBSR register is set to "1" (start counting)
Count stop condition	The TBiS bit is set to "0" (stop counting)
Interrupt request generation timing	The timer underflows
TBiIn pin function	Programmable I/O port
Read from timer	The Ai register indicates a value of the counter
Write to timer	<ul style="list-style-type: none"> <li>When the counter stops or before the first count source prior to stating of the counter is input A value written to the TBi register is written to both reload register and counter.</li> <li>When counting is in progress A value written to the TBi register is written to the reload register only (Transferred a value to the counter at the next reload time).</li> </ul>

Notes :

- The CNT3 to CNT0 bits in the TRGSR register determines either "no division (n=0)" or "divide-by-2n (n=1 to 15)".

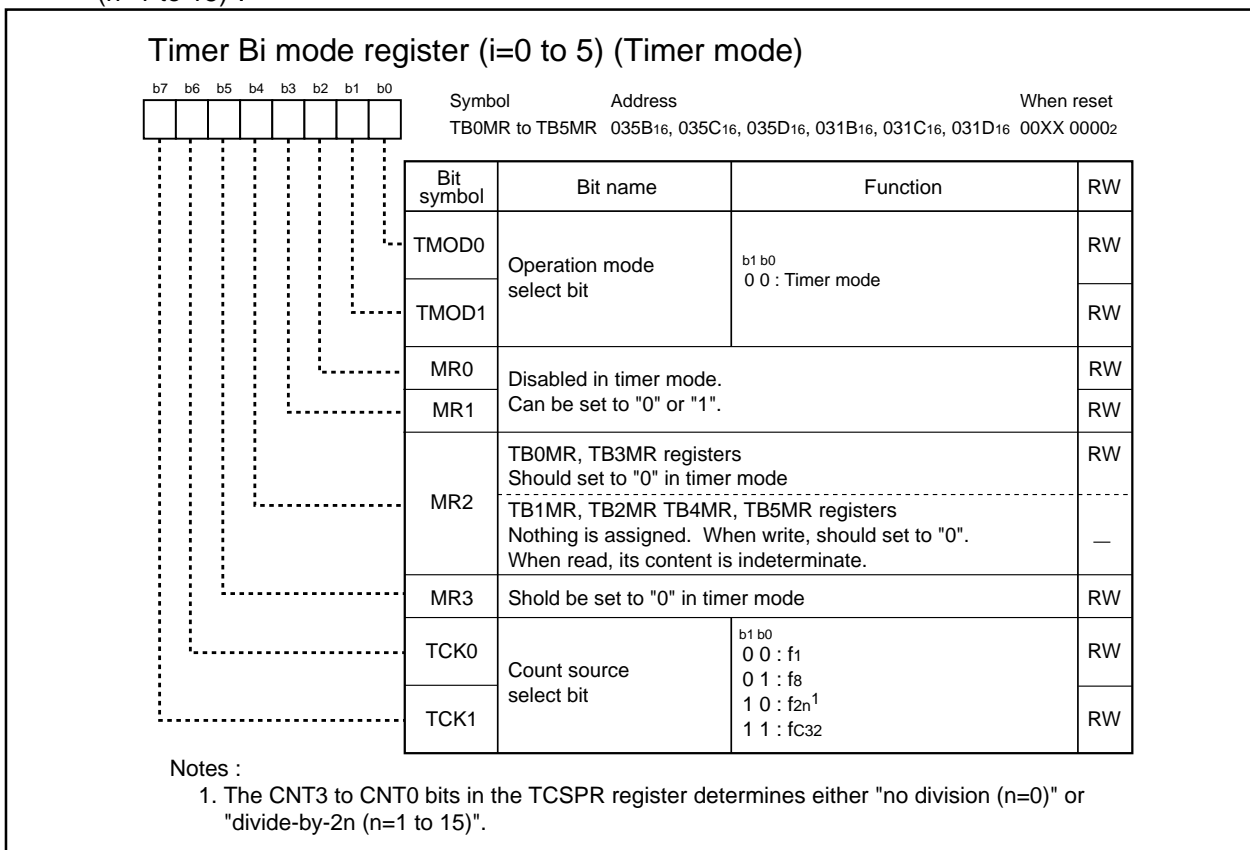


Figure 1.15.5. TB0MR to TB5MR Registers

## Timer (Timer B)

## 2. Event Counter Mode

In event count mode, the timer counts how many external signals are input or how many timers overflows and underflows. (See Table 1.15.3.) Figure 1.15.6 shows the TBiMR register (i=0 to 5) in event counter mode.

**Table 1.15.3. Specifications in event counter mode**

Item	Specification
Count source	<ul style="list-style-type: none"> <li>External signal input to the TBiIN pin (i = 0 to 5) Rising edge and falling edge or falling edge and rising edge can be selectable as a valid edge of a count source by program.</li> <li>TBj overflows or underflows (j=i-1, except j=2 when i=0, j=5 when i=3)</li> </ul>
Count operation	<ul style="list-style-type: none"> <li>The timer decrements the counter</li> <li>When the timer underflows, it reloads contents the reload register into ones of the register to continue counting</li> </ul>
Divide ratio	$1/(n+1)$ n : setting value of the TBi register    0000 <sub>16</sub> to FFFF <sub>16</sub>
Count start condition	The TBiS bit in the TABSR or TBSR register is set to "1" (starts counting)
Count stop condition	The TBiS bit is set to "0" (stops counting)
Interrupt request generation timing	The timer underflows until the first count source, after start counting, is input
TBiIN pin function	Programmable I/O port, count source input
Read from timer	The Ai register indicates a value of the counter
Write to timer	<ul style="list-style-type: none"> <li>When the counter stops or before the first count source prior to starting of a counter is input A value written to the TBi register is also written to both reload register and counter.</li> <li>While counting A value written to the TBi register is written to the reload register only (Transferred to counter at next reload time).</li> </ul>

Timer (Timer B)

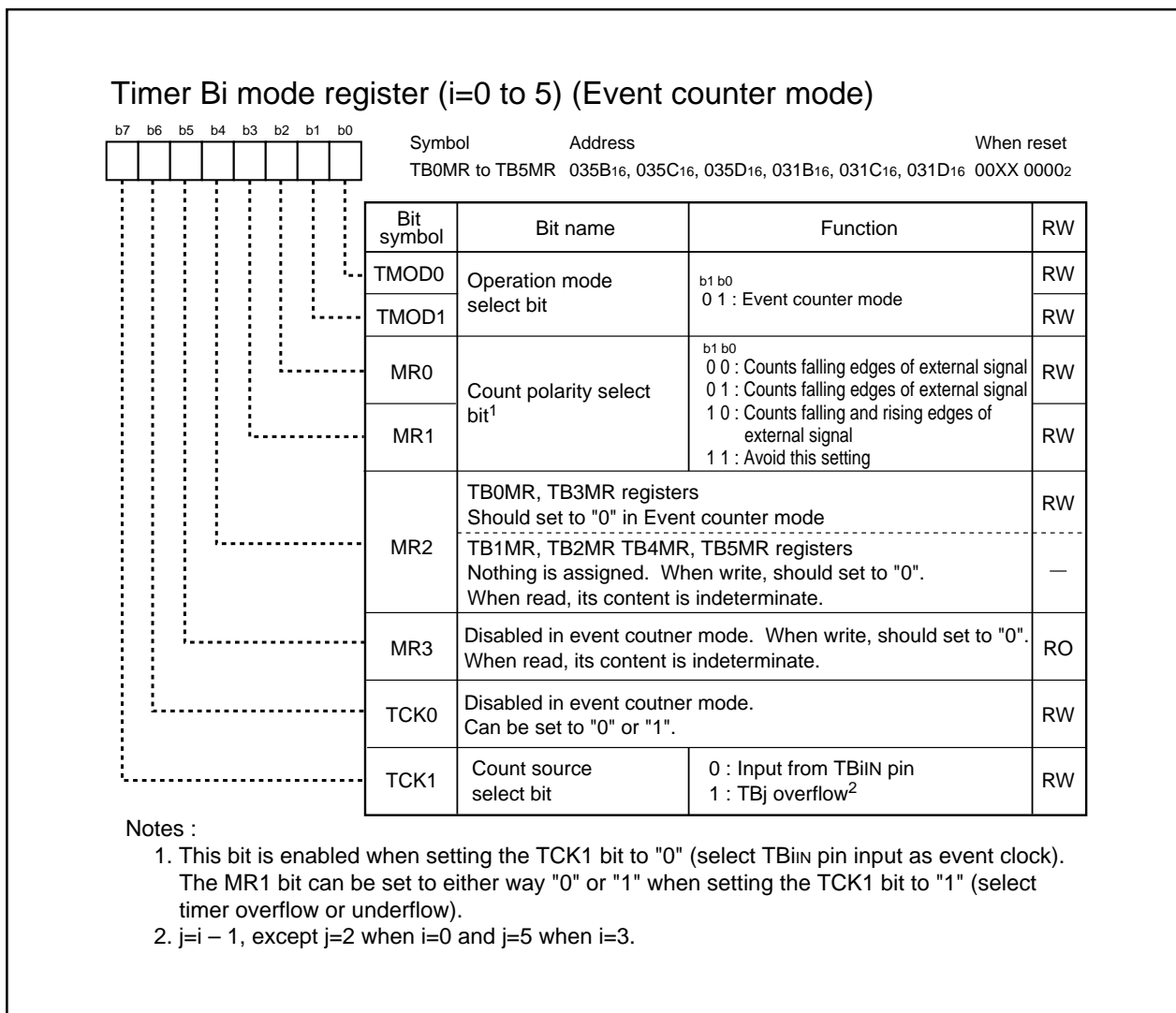


Figure 1.15.6. TB0MR to TB5MR Registers

## Timer (Timer B)

### 3. Pulse Period and Pulse Width Measurement Mode

In pulse period and pulse width measurement mode, the timer measures pulse period or pulse width of an external signal. (See Table 1.15.4.) Figure 1.15.7 shows the TBiMR register (i=0 to 5) in pulse period and pulse width measurement mode. Figure 1.15.8 shows a operation timing when measuring a pulse period. Figure 1.15.9 shows an example of the pulse width measurement.

**Table 1.15.4. Specifications in Pulse Period and Pulse Width Measurement Mode**

Item	Specification
Count source	f1, f8, f2n <sup>3</sup> , fC32
Count operation	<ul style="list-style-type: none"> <li>The timer increments the counter</li> <li>Counter value is transferred to the reload register on a valid edge of measurement pulse</li> <li>The counter value is set to "0000<sub>16</sub>" to continue counting</li> </ul>
Count start condition	The TBiS bit in the TABSR or TBSR register is set to "1" (starts counting)
Count stop condition	The TBiS bit is set to "0" (stops counting)
Interrupt request generation timing	<ul style="list-style-type: none"> <li>When valid edge of measurement pulse is input<sup>1</sup></li> <li>When the timer overflows, the MR3 bit in the TBiMR register is set to "1" (overflow) simultaneously. When the TBiS bit is set to "1" (start counting) and the next count source is counted after setting the MR3 bit to "1" (overflow), the MR3 bit is set to "0" (no overflow) by the TBiMR register again.</li> </ul>
TBiIN pin function	Measurement pulse input
Read from timer	The Ai register indicates a value of the counter <sup>2</sup>
Write to timer	Value written to TBi register can be written to neither reload register nor counter

Notes :

1. No interrupt request is generated when the first valid edge is input after the timer has started counting.
2. Value read from the TBi register is indeterminate until the second valid edge is input after the timer starts counting.
3. The CNT3 to CNT0 bits in the TRGSR register determines either "no division (n=0)" or "divide-by-2n (n=1 to 15)".

Timer (Timer B)

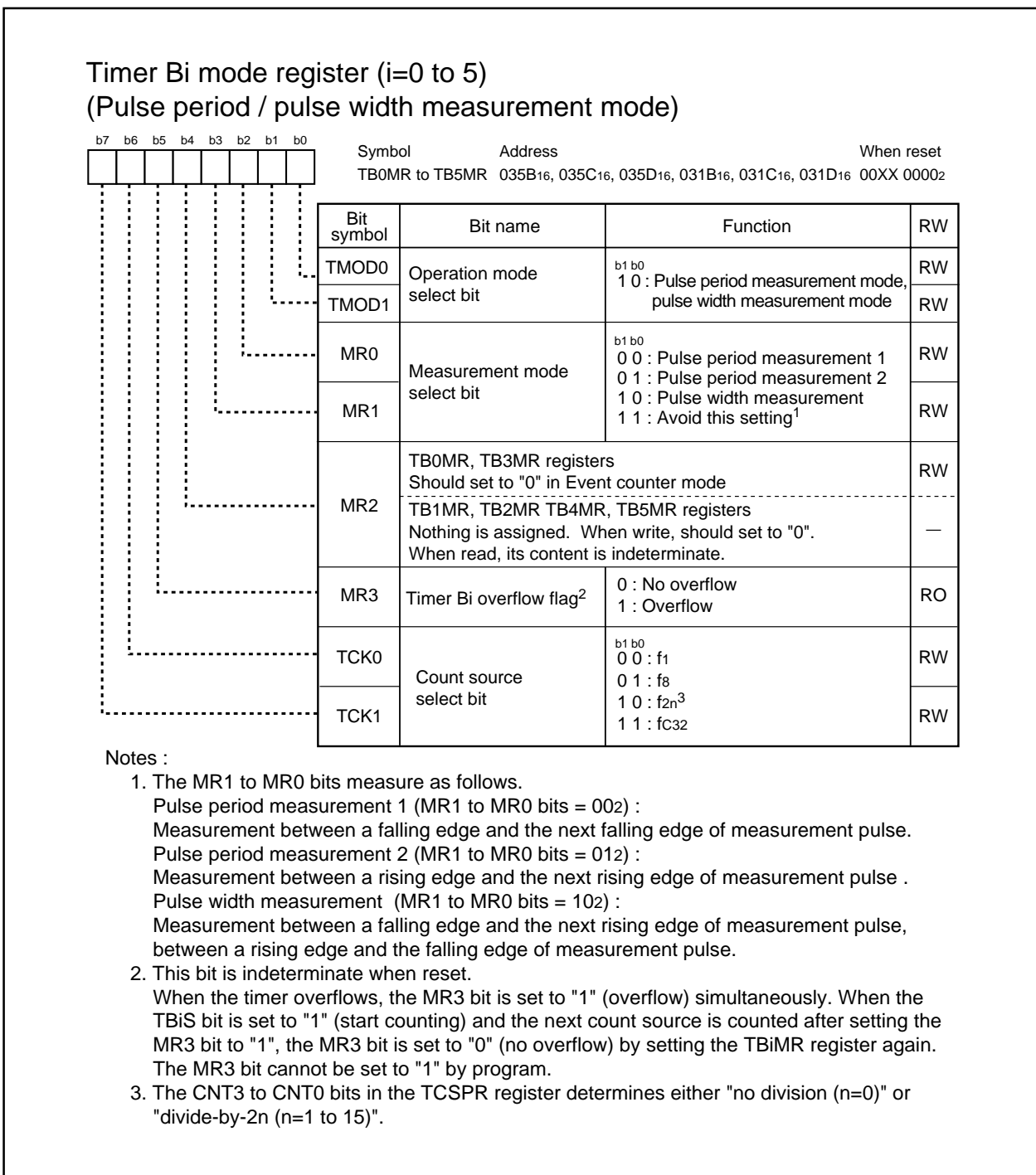


Figure 1.15.7. TB0MR to TB5MR Registers

Timer (Timer B)

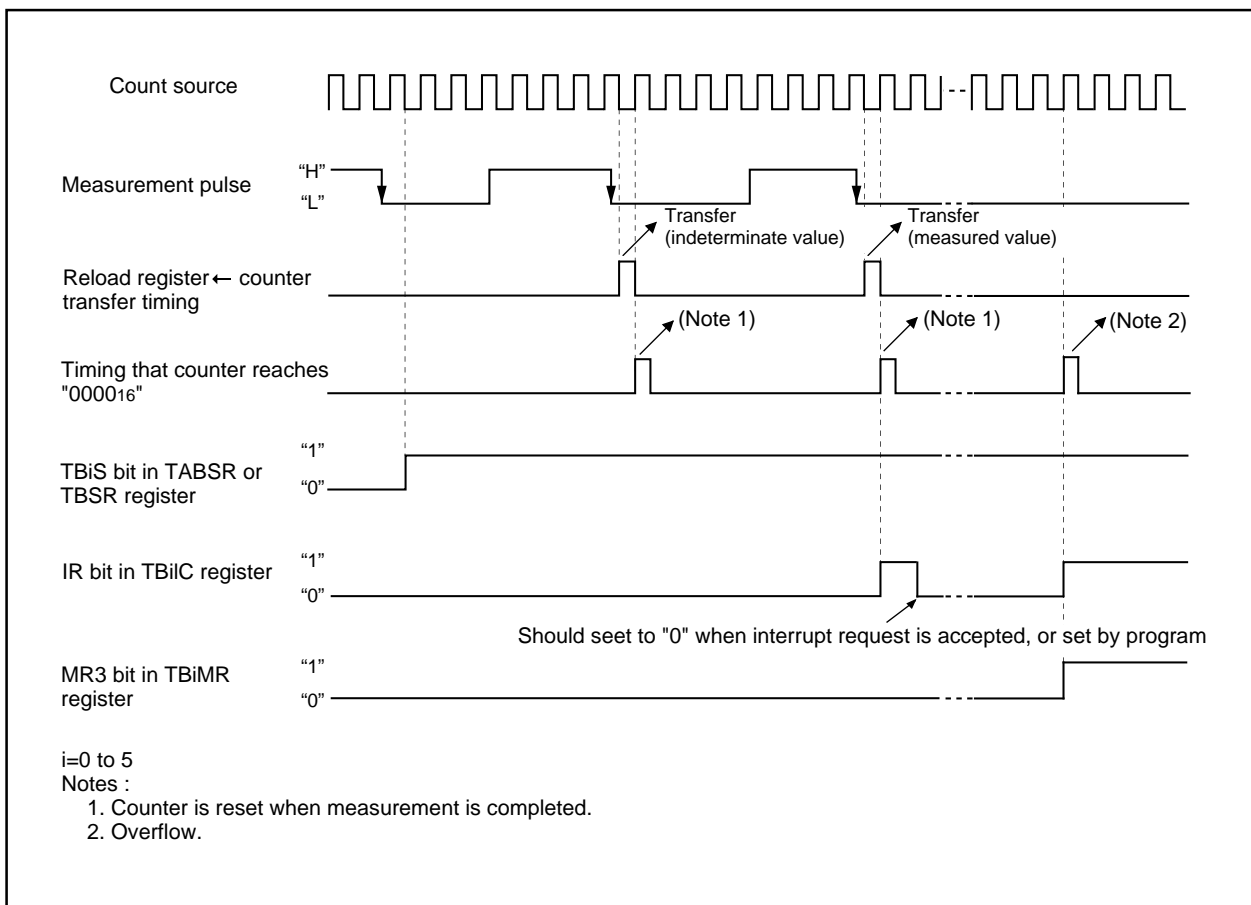


Figure 1.15.8. Pulse Period 1 Measurement

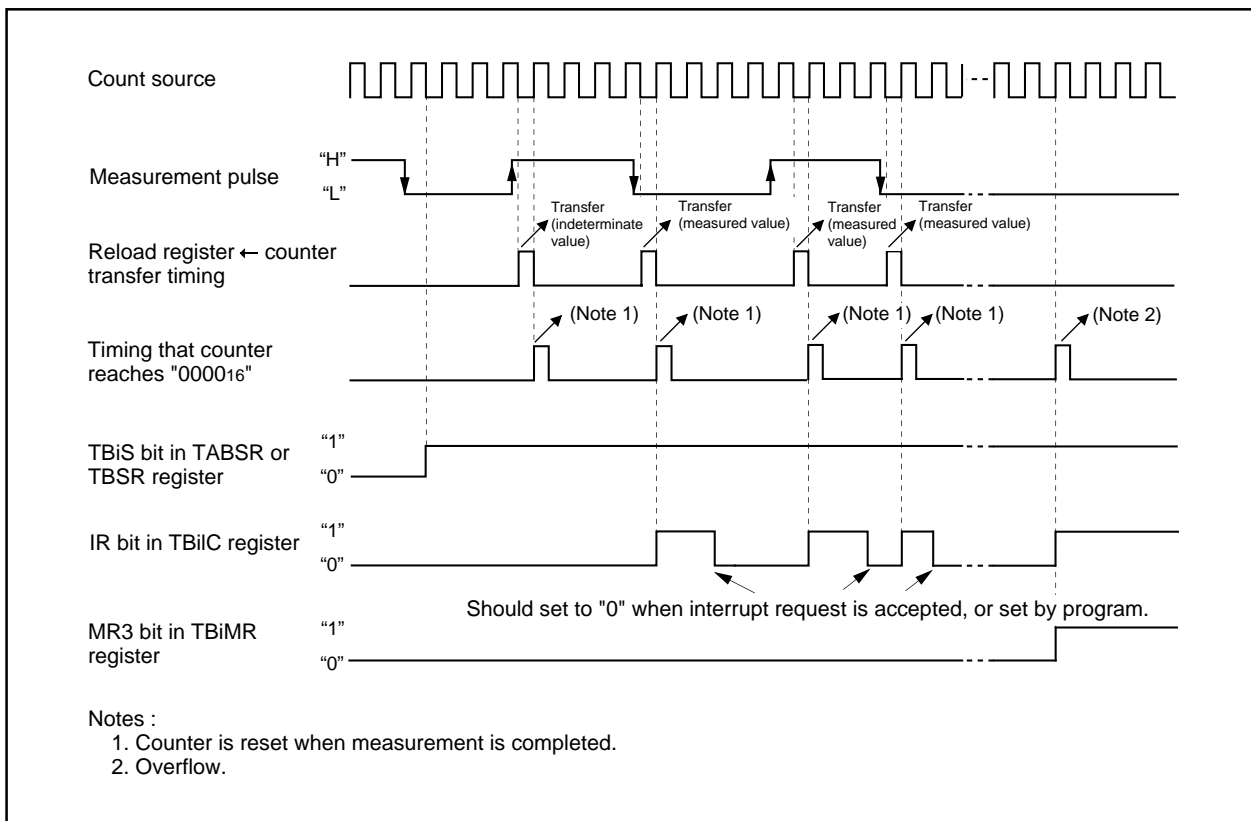


Figure 1.15.9. Pulse Width Measurement

## Three-phase motor control timer functions

# Three-phase Motor Control Timer Functions

Three-phase motor driving waveform can be output by using the timers A1, A2, A4 and B2. Table 1.16.1 lists specifications of the three-phase motor control timer functions. Table 1.16.2 lists pin settings. Figure 1.16.1 shows a block diagram. Figures 1.16.2 to 1.16.7 show registers associated with the three-phase control timer functions.

**Table 1.16.1 Three-phase Motor Control Timer Functions Specification**

Item	Specification
Three-phase waveform output pin	Six pins (U, $\bar{U}$ , V, $\bar{V}$ , W, $\bar{W}$ )
Forced cutoff input <sup>1</sup>	Input "L" to the $\overline{NMI}$ pin
Timers to be used	Timer A4, A1, A2 (used in one-shot timer mode) Timer A4: U- and $\bar{U}$ -phase waveform control Timer A1: V- and $\bar{V}$ -phase waveform control Timer A2: W- and $\bar{W}$ -phase waveform control Timer B2 (used in timer mode) Carrier wave cycle control Dead time timer (three 8-bit timer share reload register) Dead time control
Output waveform	Triangular wave modulation, Sawtooth wave modification Can output "H" or "L" for one cycle Can set in positive-phase level and negative-phase level respectively
Carrier wave cycle	Triangular wave modulation: count source x (m+1) x 2 Sawtooth wave modulation: count source x (m+1) m: setting value of the TB2 register, 0000 <sub>16</sub> to FFFF <sub>16</sub> Count source: f <sub>1</sub> , f <sub>8</sub> , f <sub>2n<sup>2</sup></sub> , f <sub>c32</sub>
Three-phase PWM output width	Triangular wave modulation: count source x n x 2 Sawtooth wave modulation: count source x n n: setting value of the TA4, TA1 and TA2 register (of the TA4, TA41, TA1, TA11, TA2 and TA21 registers when setting the INV11bit to "1"), 0000 <sub>16</sub> to FFFF <sub>16</sub> Count source: f <sub>1</sub> , f <sub>8</sub> , f <sub>2n</sub> , f <sub>c32</sub>
Dead time	Count source x p, or no dead time p: setting value of the DTT register, 00 <sub>16</sub> to FF <sub>16</sub> Count source: f <sub>1</sub> , or f <sub>1</sub> divided by 2
Active level	Selectable from "H" or "L"
Positive and negative-phase concurrent active disable function	Positive and negative-phases concurrent active disable function Positive and negative-phases concurrent active detect function
Interrupt frequency	For the timer B2 interrupt, a carrier wave cycle-to-cycle basis through 15 times carrier wave cycle-to-cycle basis can be selected

Notes :

1. Forced cutoff with  $\overline{NMI}$  input is enabled when setting the INV02 bit to "1" (the three-phase motor control timer functions) and the INV03 bit to "1" (the three-phase motor control timer input enabled).
2. The CNT3 to CNT0 bits in the TRGSR register determines either "no division (n=0)" or "divide-by-2n (n=1 to 15)".

**Three-phase motor control timer functions****Table 1.16.2. Pin Settings**

Pin	Bits and Setting values		
	PS1, PS2 registers <sup>1</sup>	PSL1, PSL2 registers	PSC register
P72/V	PS1_2 bit =1	PSL1_2 bit =0	PSC_2 bit =1
P73/ $\bar{V}$	PS1_3 bit =1	PSL1_3 bit =1	—
P74/W	PS1_4 bit =1	PSL1_4 bit =1	—
P75/ $\bar{W}$	PS1_5 bit =1	PSL1_5 bit =0	—
P80/U	PS2_0 bit =1	PSL2_0 bit =1	—
P81/ $\bar{U}$	PS2_1 bit =1	PSL2_1 bit =0	—

## Notes :

1. The PS1\_2 to PS1\_5 and PS2\_0 to PS2\_1 bits in the PS1 and PS2 registers should be set to "1" after setting the INV02 bit to "1".

Three-phase motor control timer functions

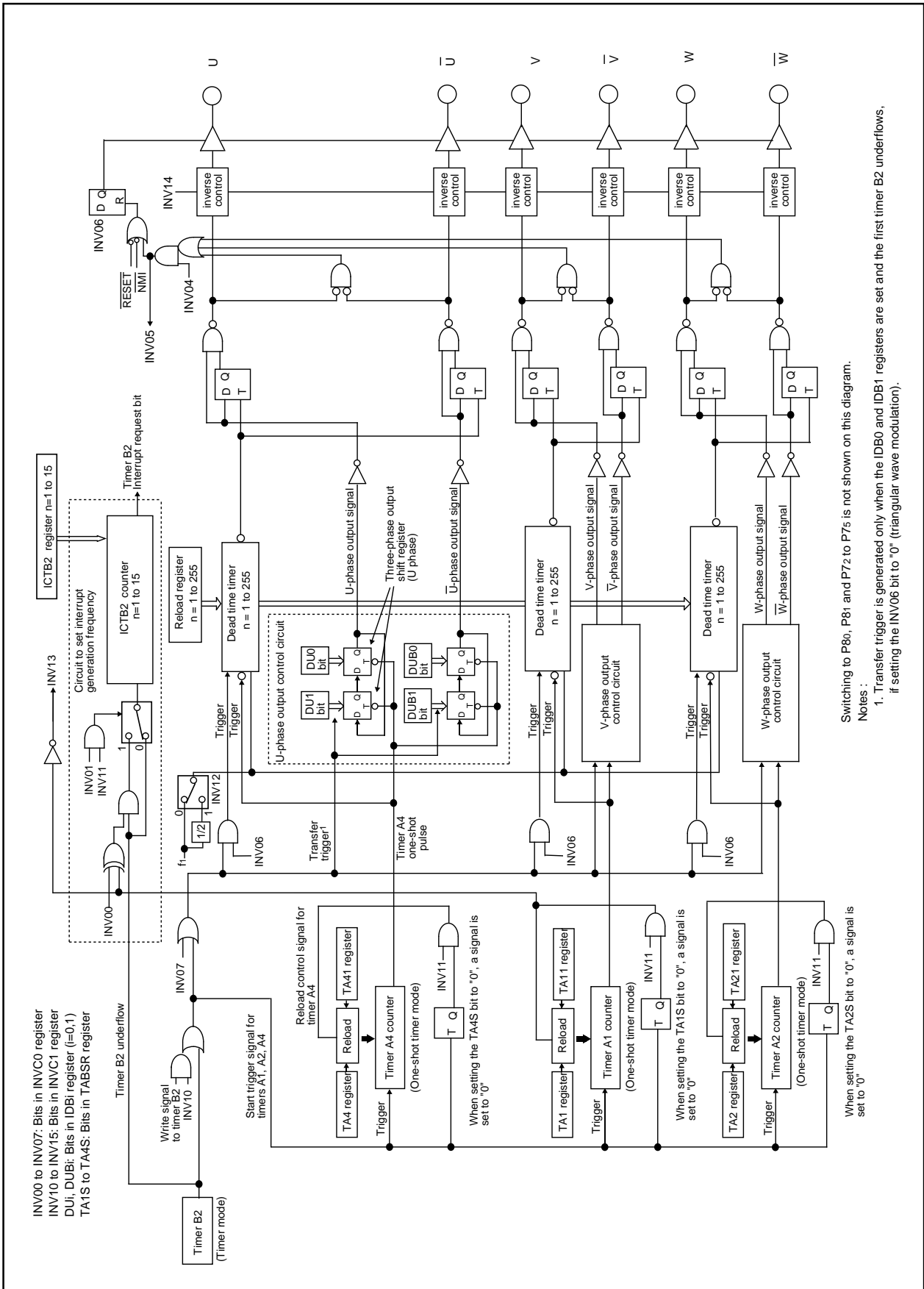


Figure 1.16.1. Three-phase Motor Control Timer Functions Block Diagram

### Three-phase motor control timer functions

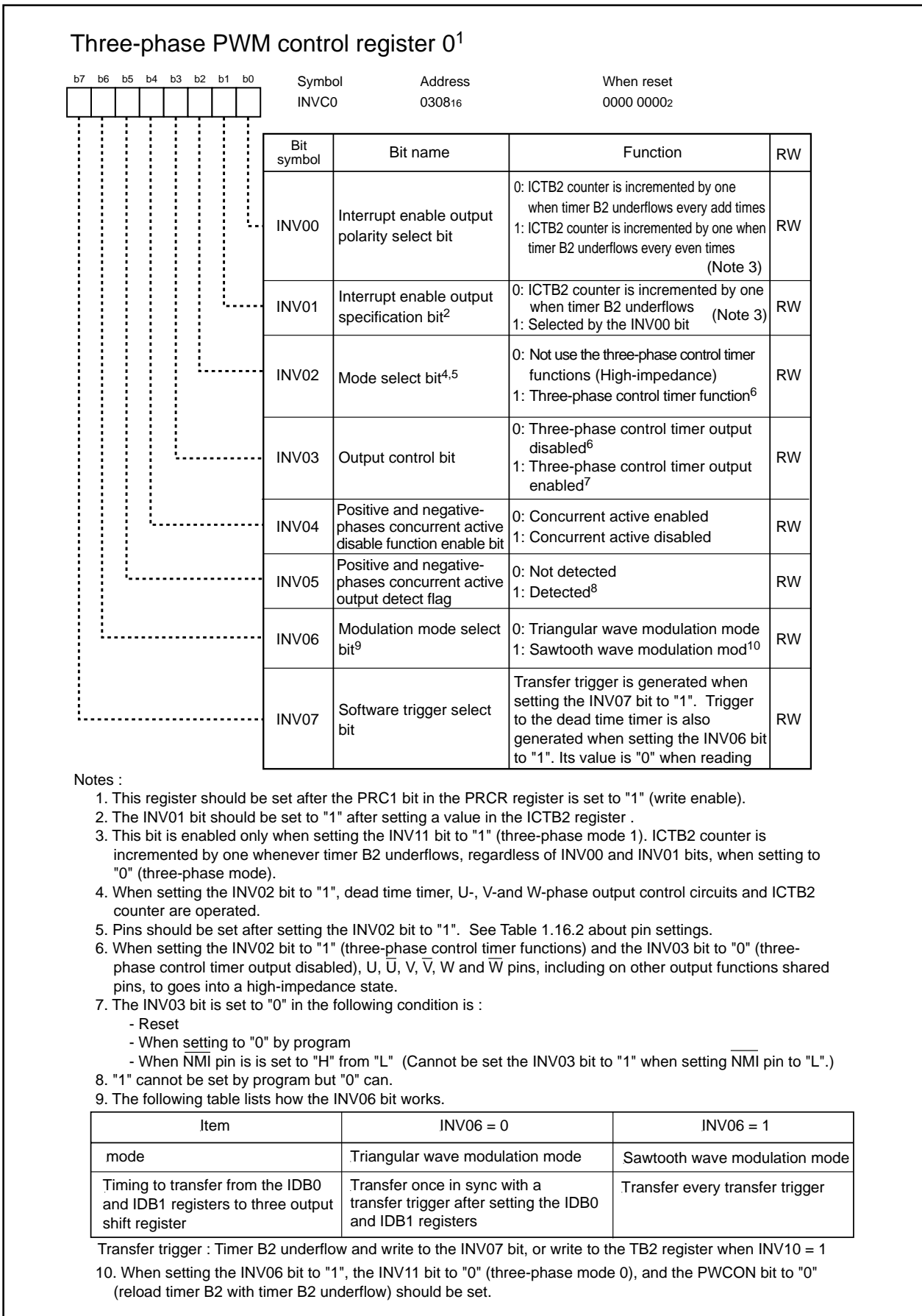


Figure 1.16.2. INVC0 Register

Three-phase motor control timer functions

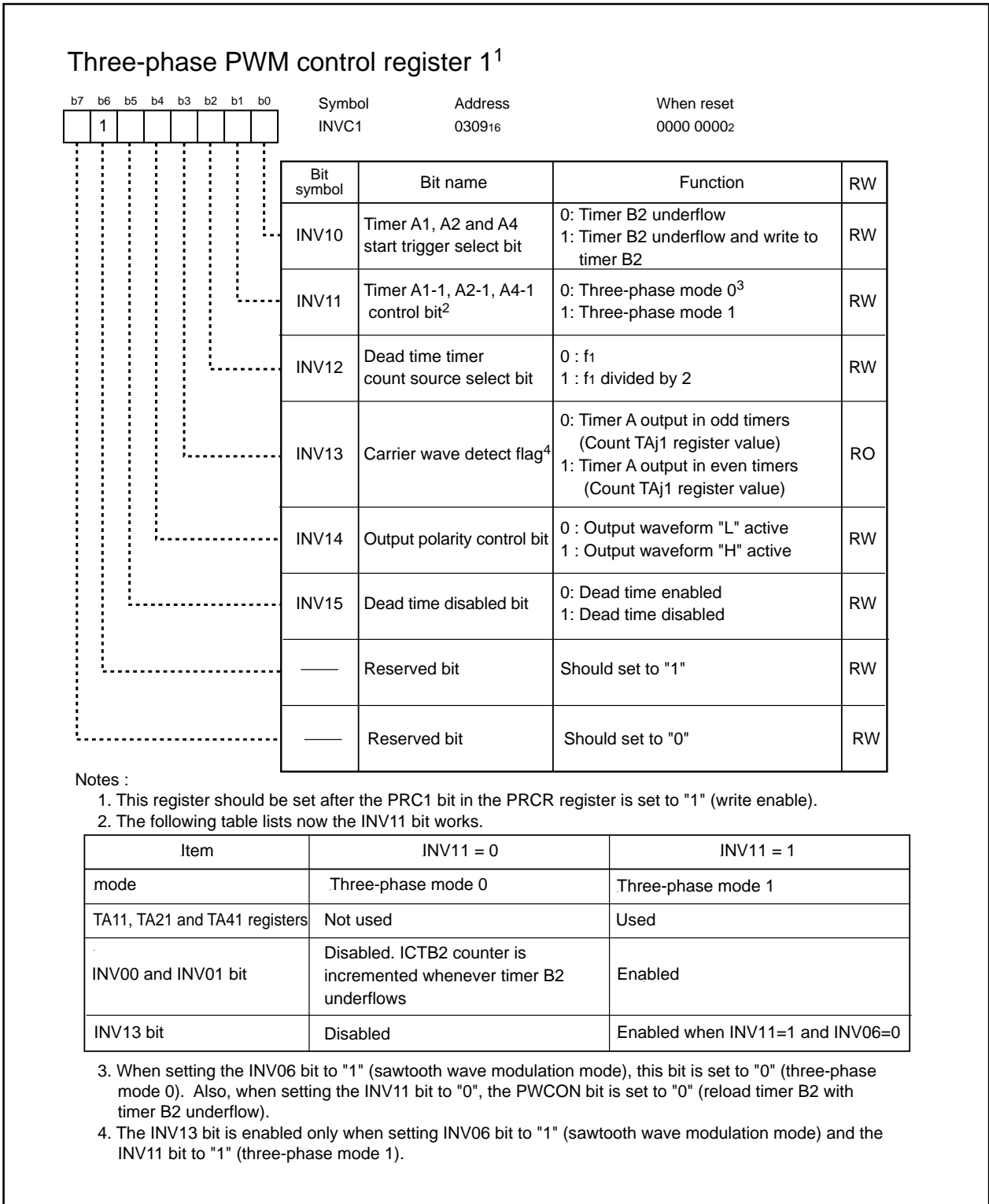


Figure 1.16.3. INVC1 Register

### Three-phase motor control timer functions

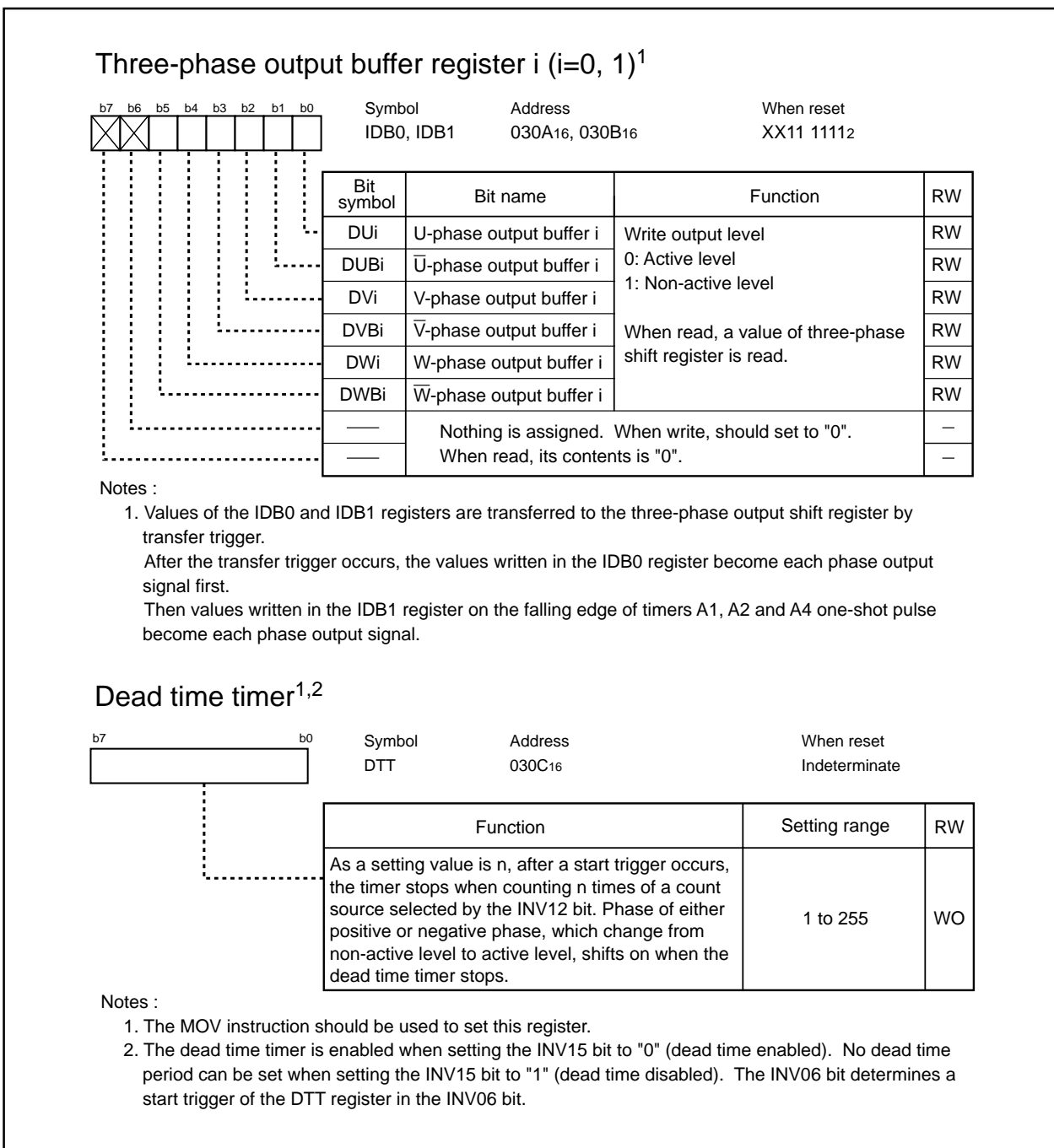


Figure 1.16.4. IDB0, IDB1 and DTT Registers

Three-phase motor control timer functions

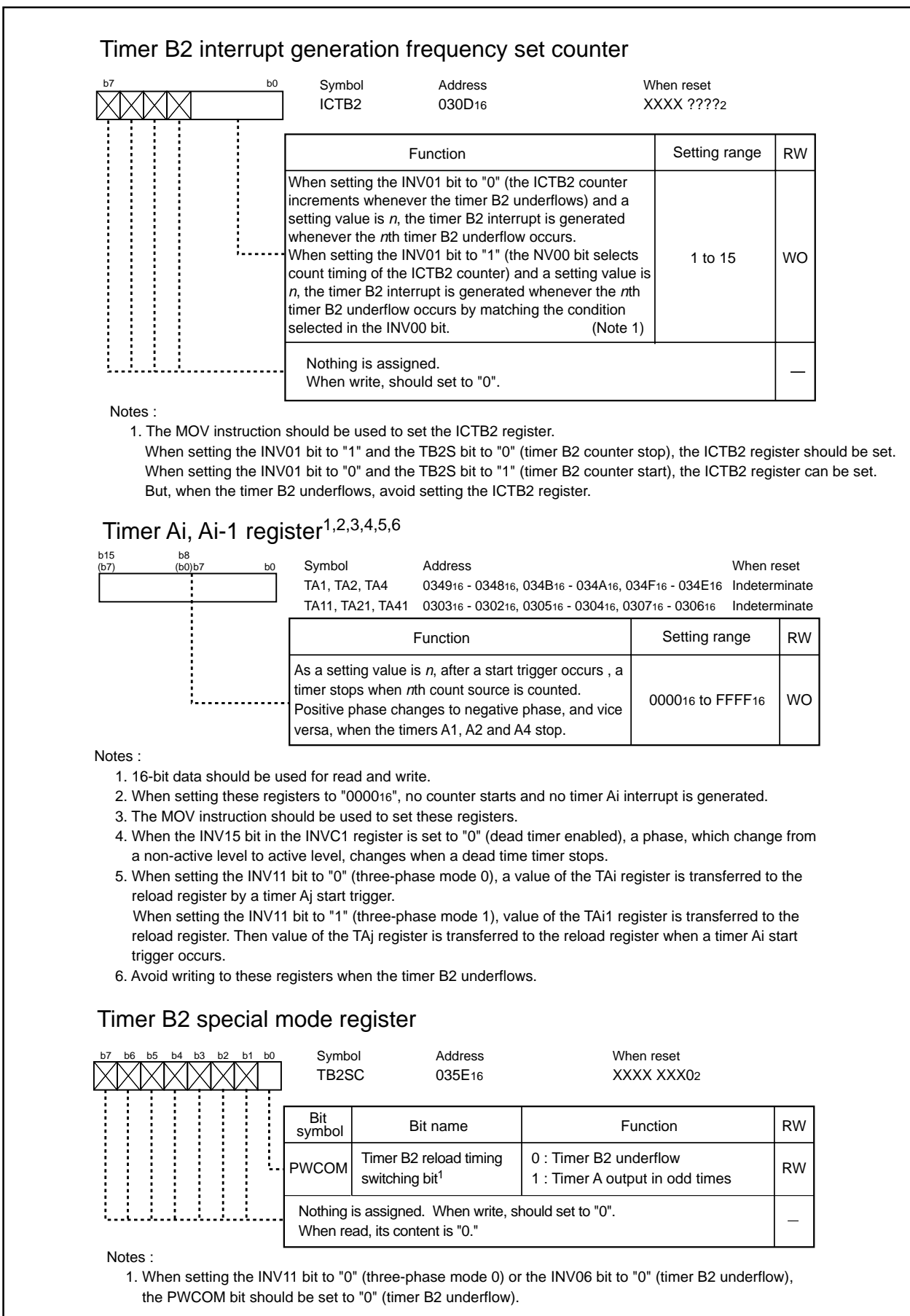


Figure 1.16.5. ICTB Register, TA1, TA2, TA4, TA11, TA21 and TA41 Registers and TB2SC Register

### Three-phase motor control timer functions

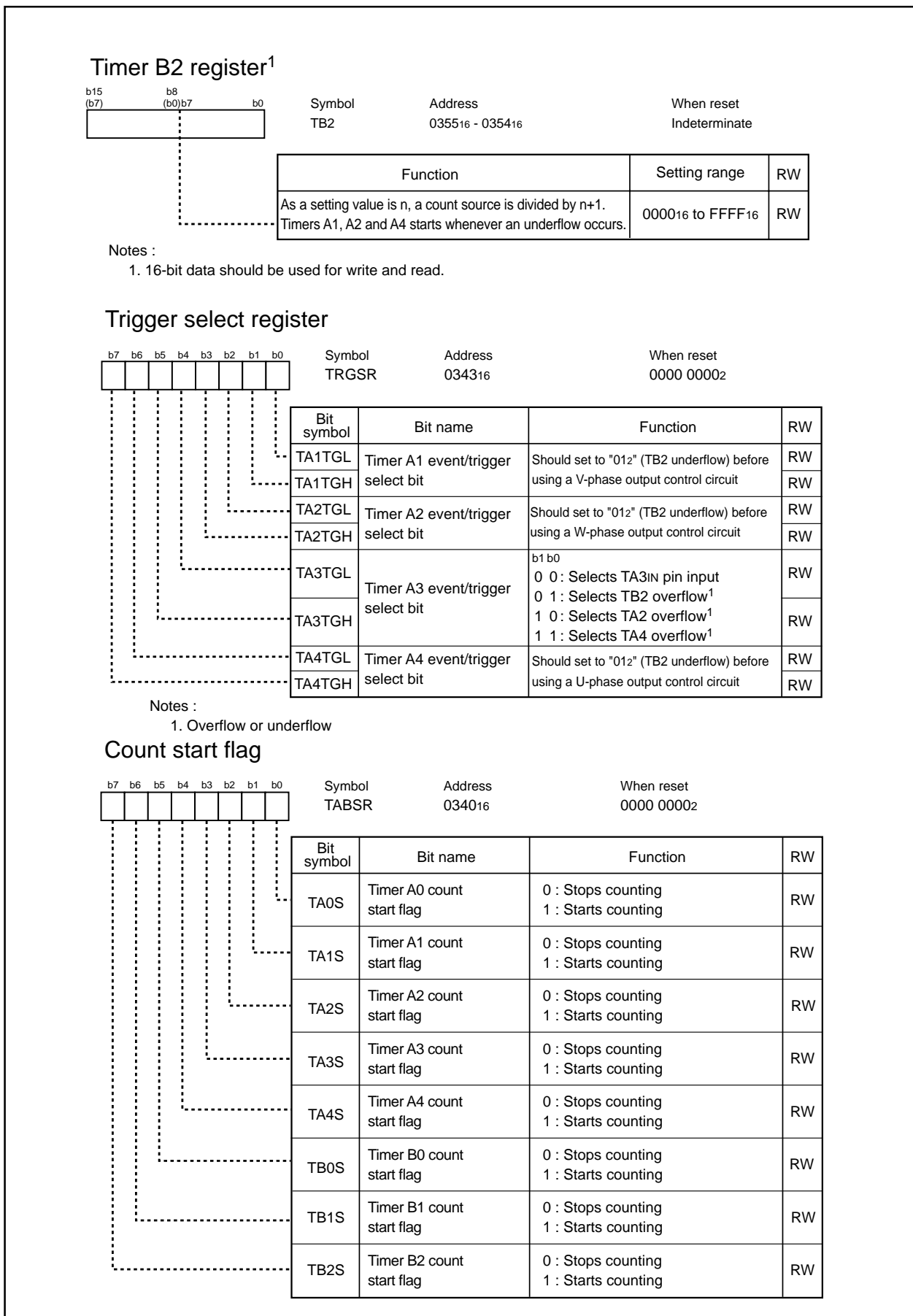


Figure 1.16.6. TB2, TRGSR and TABSR Registers

Three-phase motor control timer functions

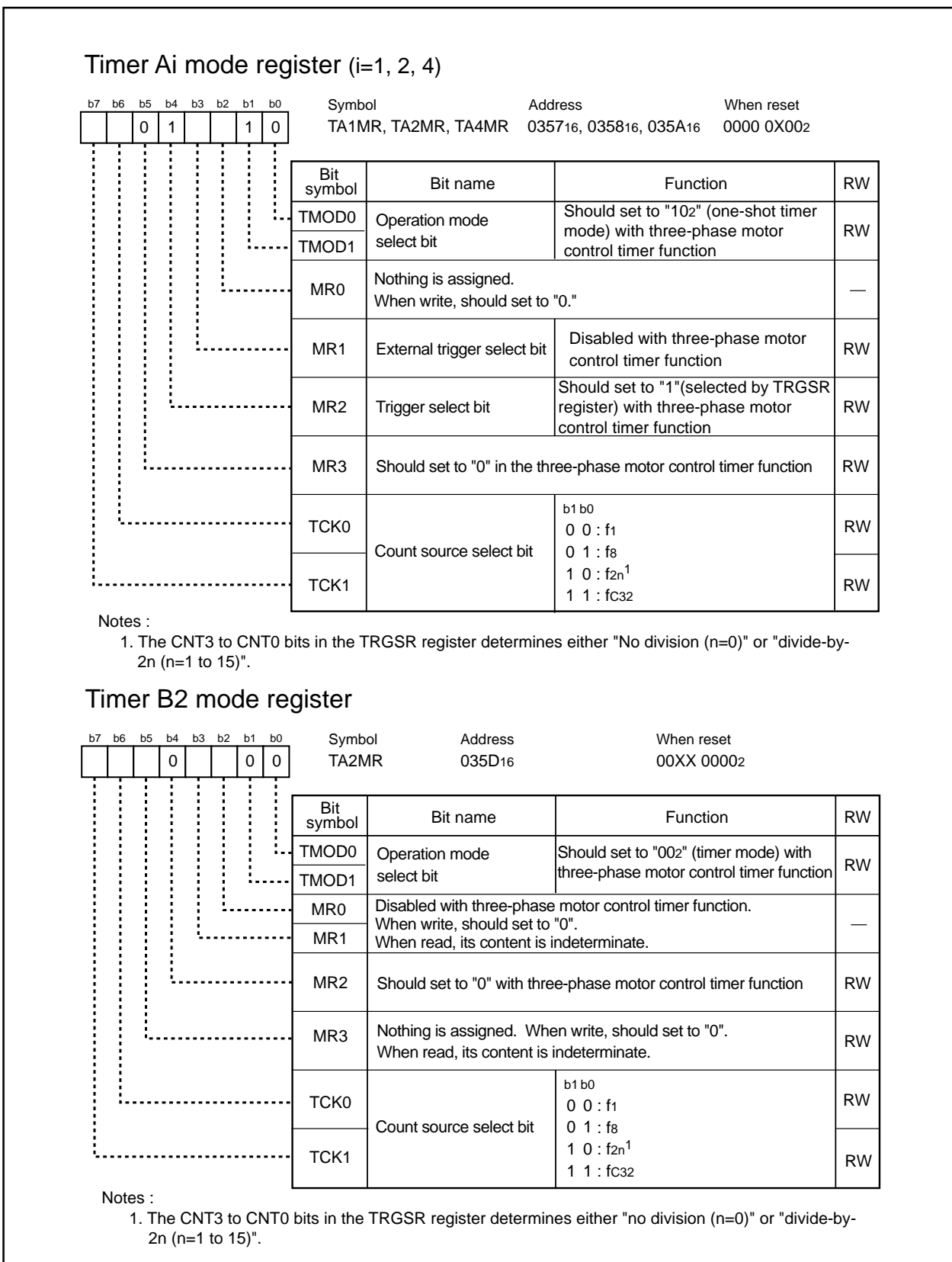


Figure 1.16.7. TA1MR, TA2MR, TA4MR Registers and TB2MR Register

### Three-phase motor control timer functions

When the INV02 bit in the INVC0 register is set to "1", the three-phase control timer function is activated. The timer B2 is used for carrier wave control and timer A1, A2, A4 for three-phase PWM output (U,  $\bar{U}$ , V,  $\bar{V}$ , W,  $\bar{W}$ ) control. The exclusive dead time timer controls dead time. Figure 1.16.8 shows an example of the triangular modulation waveform. Figure 1.16.9 shows an example of the sawtooth modulation waveform.

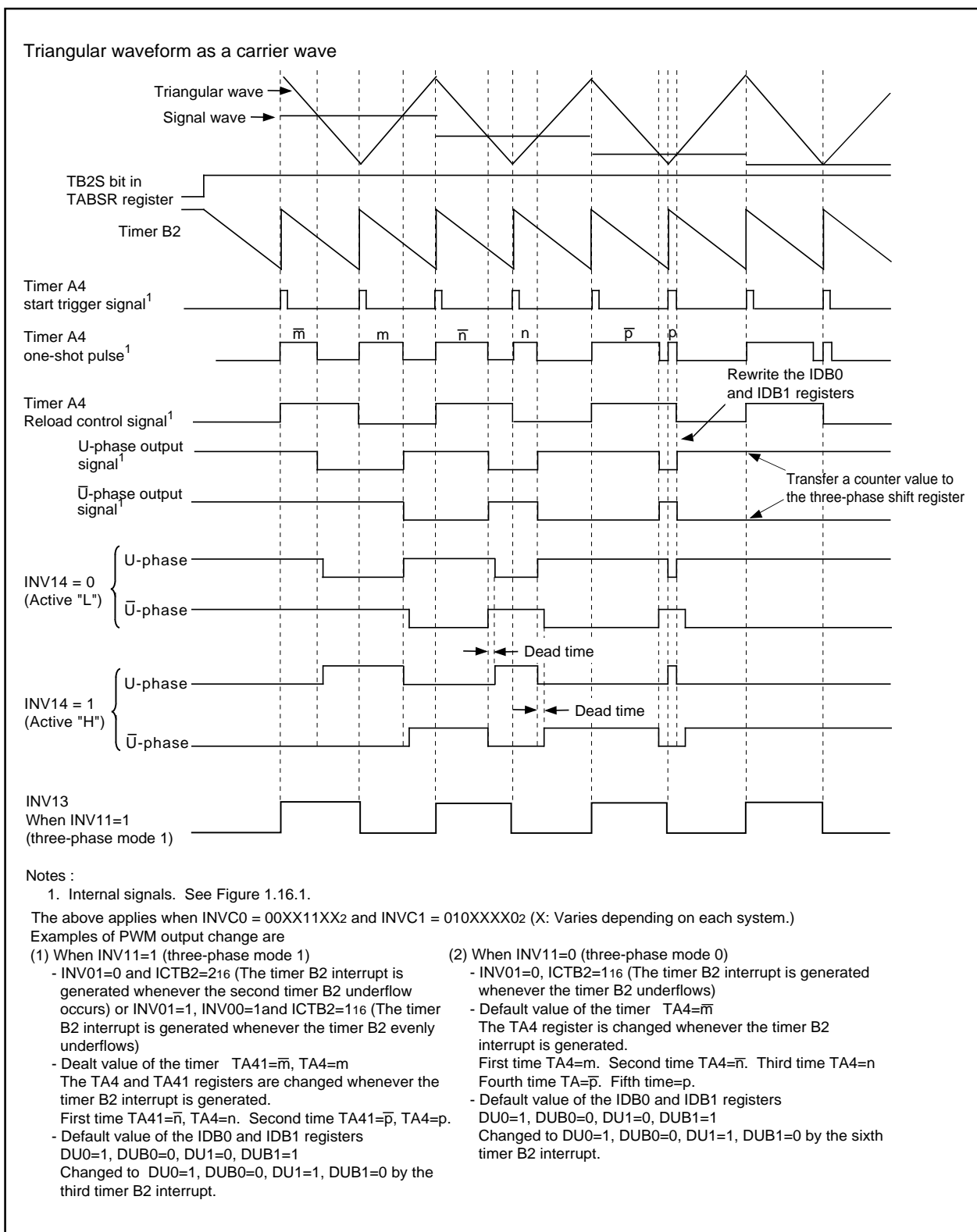
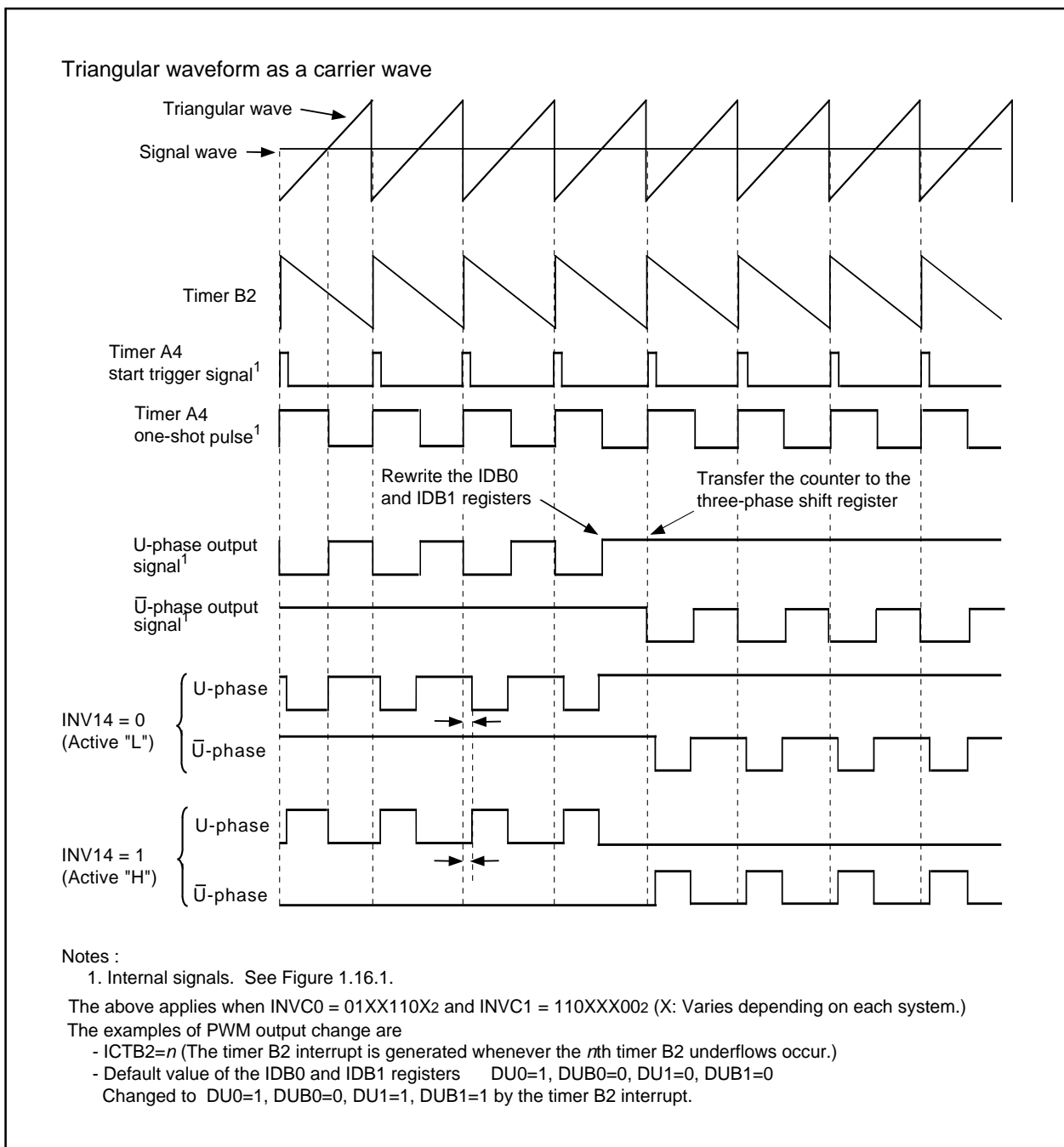


Figure 1.16.8. Triangular Wave Modulation Operation

### Three-phase motor control timer functions



**Figure 1.16.9. Sawtooth Wave Modulation Operation**

## Serial I/O

Serial I/O consists of five channels (UART0 to UART4).

Each UART<sub>i</sub> (i=0 to 4) has an exclusive timer to generate the transfer clock and operates independently.

Figure 1.17.1 shows a UART<sub>i</sub> block diagram.

UART<sub>i</sub> supports the following modes :

- Clock synchronous serial I/O mode
- Clock asynchronous serial I/O mode (UART mode)
- Special mode 1 (IIC mode)
- Special mode 2
- Special mode 3 (Clock-divided synchronous function, GCI mode)
- Special mode 4 (Bus conflict detect function, IE mode)
- Special mode 5 (SIM mode)

Figures 1.17.2 to 1.17.9 show registers associated with UART<sub>i</sub>.

Refer to the tables listed each mode for register and pin settings.

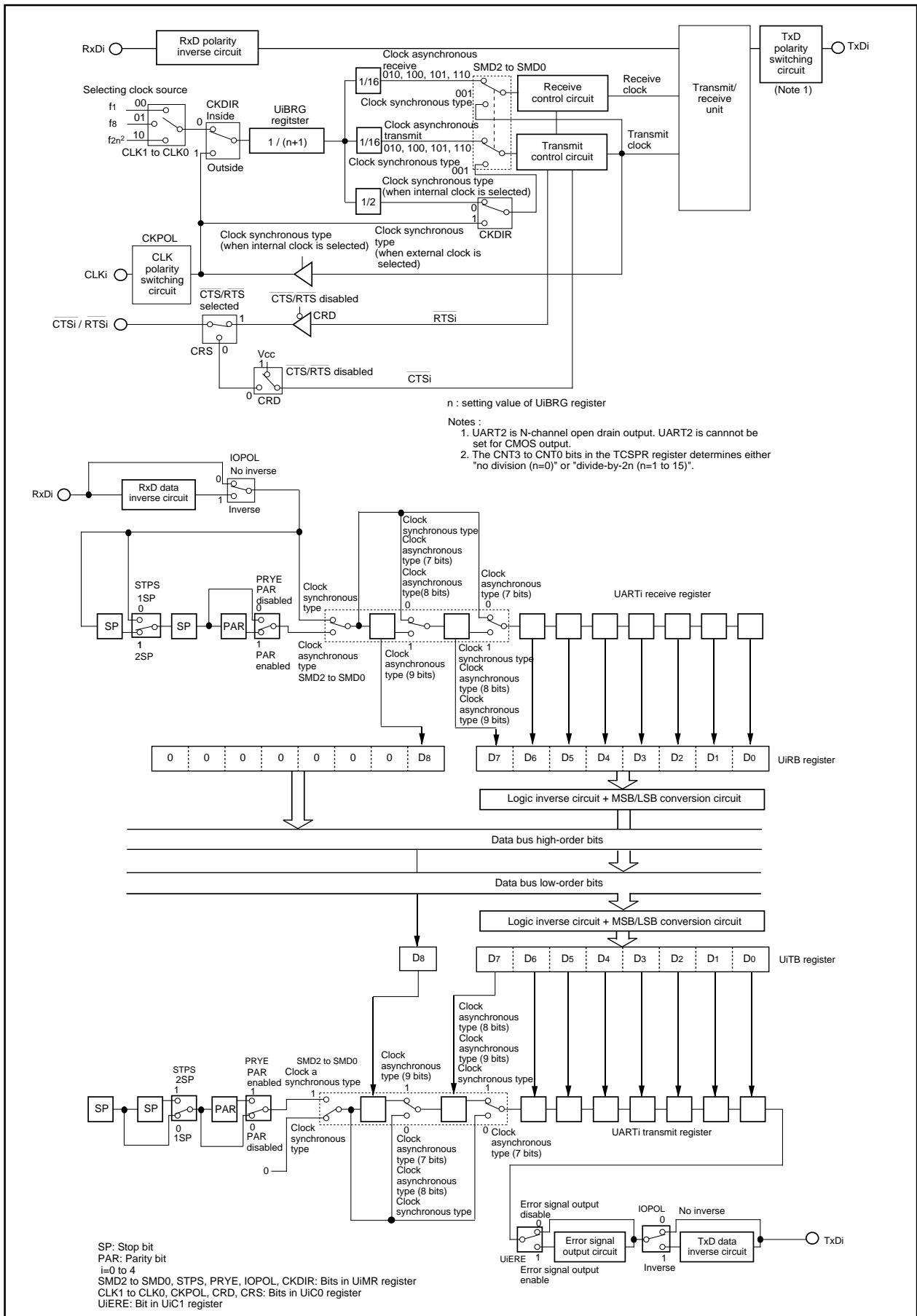
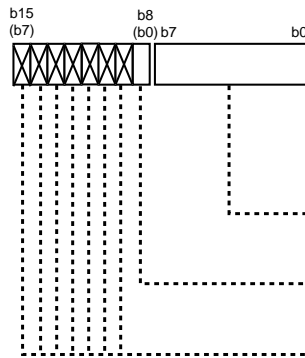


Figure 1.17.1. UARTi Block Diagram

### UART<sub>i</sub> transmit buffer register (i=0 to 4)<sup>1</sup>



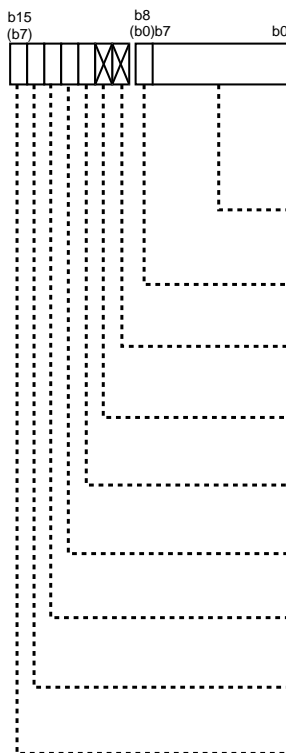
Symbol	Address	When reset
U0TB to U2TB	036B <sub>16</sub> -036A <sub>16</sub> , 02EB <sub>16</sub> -02EA <sub>16</sub> , 033B <sub>16</sub> -033A <sub>16</sub>	Indeterminate
U3TB, U4TB	032B <sub>16</sub> -032A <sub>16</sub> , 02FB <sub>16</sub> -02FA <sub>16</sub>	Indeterminate

Bit symbol	Function	RW
——	Data to be transmitted (D <sub>7</sub> to D <sub>0</sub> )	WO
——	Data to be transmitted (D <sub>8</sub> )	WO
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.	—

Notes :

1. The MOV instruction should be used to set this register.

### UART<sub>i</sub> receive buffer register (i=0 to 4)



Symbol	Address	When reset
U0RB to U2RB	036F <sub>16</sub> - 036E <sub>16</sub> , 02EF <sub>16</sub> - 02EE <sub>16</sub> , 033F <sub>16</sub> - 033E <sub>16</sub>	Indeterminate
U3RB, U4RB	032F <sub>16</sub> - 032E <sub>16</sub> , 02FF <sub>16</sub> - 02FE <sub>16</sub>	Indeterminate

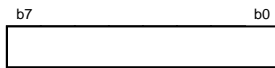
Bit symbol	Bit name	Function	RW
——	——	Received data (D <sub>7</sub> to D <sub>0</sub> )	RO
——	——	Received data (D <sub>8</sub> )	RO
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
ABT	Arbitration lost detect flag <sup>1</sup>	0: Not detected (win) 1: Detected (lose)	RW
OER	Overrun error flag <sup>2</sup>	0: No overrun error 1: Overrun error occurs	RO
FER	Framing error flag <sup>2</sup>	0: No framing error 1: Framing error occurs	RO
PER	Parity error flag <sup>2</sup>	0: No parity error 1: Parity error occurs	RO
SUM	Error sum flag <sup>2</sup>	0: No error 1: Error occurs	RO

Notes :

1. The ABT bit can be set to "0" only.
2. When the SMD<sub>2</sub> to SMD<sub>0</sub> bits in the UiMR register is set to "000<sub>2</sub>" (serial I/O disable) or the RE bit in the UiC1 register be set to "0" (receive disable), the OER, FER, PER and SUM bits are set to "0" (no error).  
When setting all OER, FER and PER bits to "0" (no error), the SUM bit is set to "0" (no error).  
Also, the FER and PER bits is set to "0" when reading low-order bits of the UiRB register.

Figure 1.17.2. U0TB to U4TB Registers and U0RB to U4RB Registers

### UARTi baud rate register (i=0 to 4)<sup>1,2</sup>



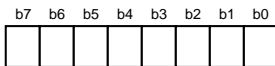
Symbol: U0BRG to U4BRG  
Address: 0369<sub>16</sub>, 02E9<sub>16</sub>, 0339<sub>16</sub>, 0329<sub>16</sub>, 02F9<sub>16</sub>  
When reset: Indeterminate

Function	Setting value	RW
As a setting value is n, the UiBRG register divides a count source by n+1	00 <sub>16</sub> to FF <sub>16</sub>	WO

Notes :

1. The MOV instruction should be used to set this register.
2. The UiBRG register should be set while stopping transmitting and receiving.

### UARTi transmit/receive mode register (i=0 to 4)



Symbol: U0MR to U4MR  
Address: 0368<sub>16</sub>, 02E8<sub>16</sub>, 0338<sub>16</sub>, 0328<sub>16</sub>, 02F8<sub>16</sub>  
When reset: 0000 0000<sub>2</sub>

Bit symbol	Bit name	Function	RW
SMD0	Serial I/O mode select bit	b <sub>2</sub> b <sub>1</sub> b <sub>0</sub> 0 0 0: Serial I/O disabled 0 0 1: Clock synchronous serial I/O mode 0 1 0: IIC mode	RW
SMD1		1 0 0: UART mode, transfer data 7 bits long	RW
SMD2		1 0 1: UART mode, transfer data 8 bits long 1 1 0: UART mode, transfer data 9 bits long Avoid setting other than	RW
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock	RW
STPS	Stop bit length select bit	0 : 1 stop bit 1 : 2 stop bits	RW
PRY	Odd/even parity select bit	Enabled when PRYE = 1 0 : Odd parity 1 : Even parity	RW
PRYE	Parity enable bit	0 : Parity disabled 1 : Parity enabled	RW
IOPOL	TxD,RxD input/output polarity switch bit	0: No inverse 1: Inverse	RW

Figure 1.17.3. U0BRG to U4BRG Registers and U0MR to U4MR Registers

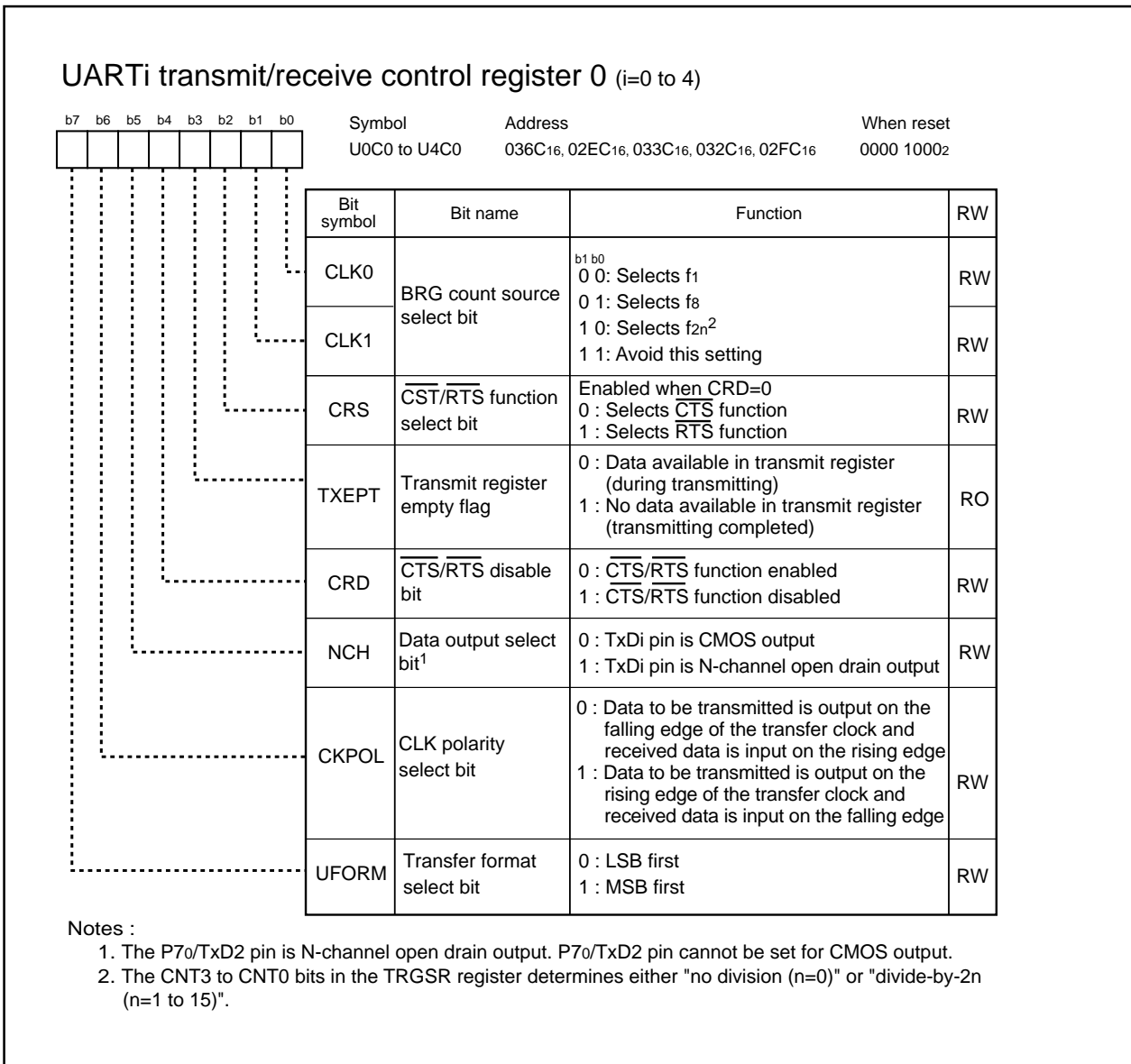


Figure 1.17.4. U0C0 to U4C0 Registers

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

Bit symbol	Bit name	Function	RW
TE	Transmit enable bit	0: Transmit disable 1: Transmit enable	RW
TI	Transmit buffer empty flag	0: Data available in the UiTB register 1: No data available in the UiTB register	RO
RE	Receive enable bit	0: Receive disable 1: Receive enable	RW
RI	Receive complete flag	0: Data available in the UiRB register 1: No data available in the UiRB register	RO
UiIRS	UARTi transmit interrupt cause select bit	0: UiTB register empty (TI = 1) 1: Transmission is completed (TXEPT = 1)	RW
UiRRM	UARTi continuous receive mode enable bit	0: Continuous receive mode disabled 1: Continuous receive mode enabled	RW
UiLCH	Data logic select bit	0: No inverse 1: Inverse	RW
SCLKSTPB /UiERE	Clock-divided synchronous stop bit / Error signal output enable bit	Clock-divided synchronous stop bit (special mode 3) 0: Stop synchronizing 1: Start synchronizing Error signal output enable bit (UART mode) 0: Output disabled 1: Output enabled	RW

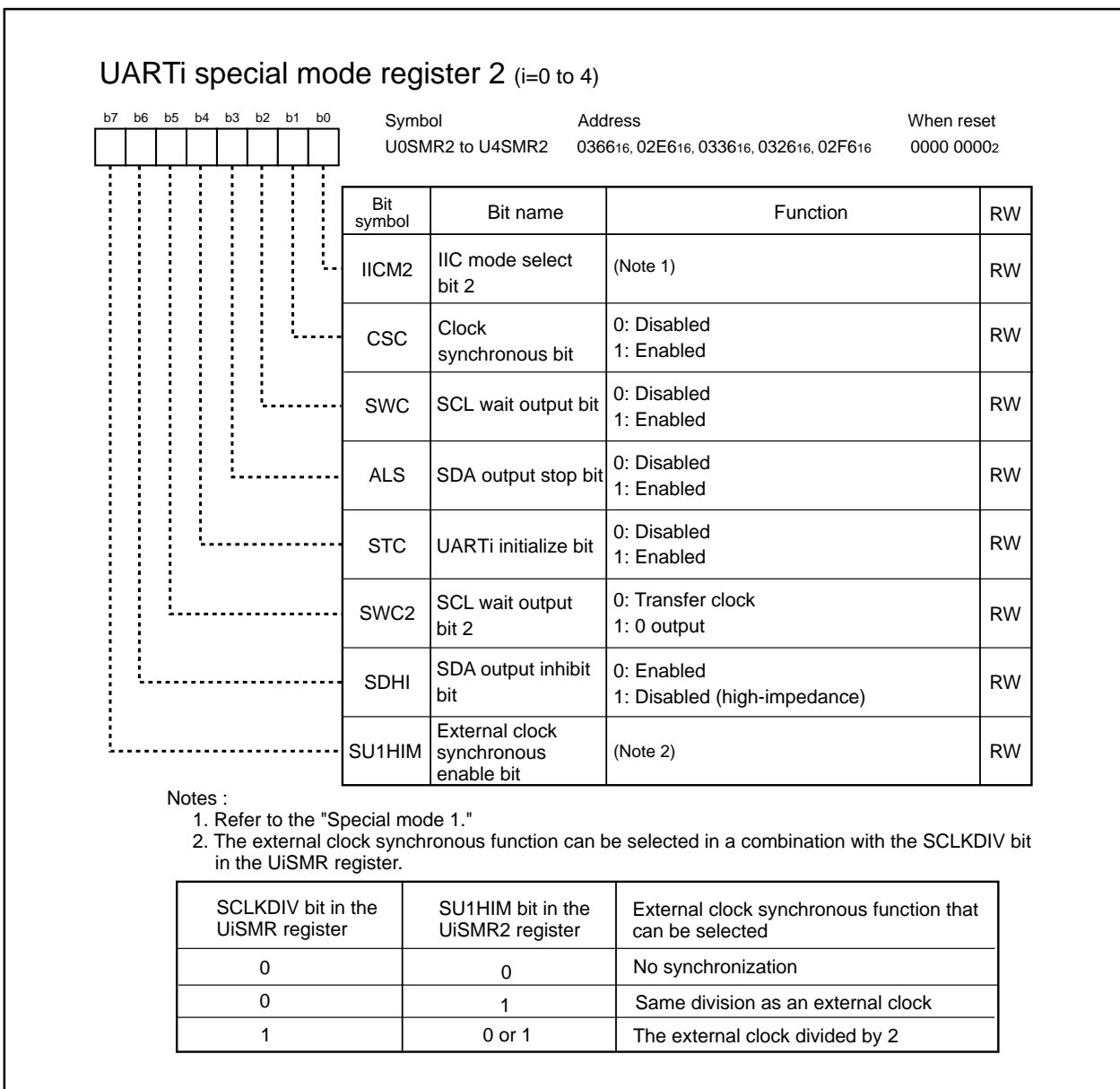
### UARTi special mode register (i=0 to 4)

Bit symbol	Bit name	Function	RW
IICM	IIC mode select bit	0: Except IIC mode 1: IIC mode	RW
ABC	Arbitration lost detecting flag	0: Update per bit 1: Update per byte	RW
BBS	Bus busy flag	0: Stop condition detected 1: Start condition detected (Busy)	RW <sup>1</sup>
LSYN	SCLL sync output enable bit	0: Disabled 1: Enabled	RW
ABSCS	Bus conflict detect sampling clock select	0: Rising edge of transfer clock 1: Timer Aj underflow (Note 2)	RW
ACSE	Auto clear function select bit of transmit enable bit	0: No auto clear function 1: Auto clear at bus conflict	RW
SSS	Transmit start condition select bit	0: Not related to RxDi 1: Synchronized with RxDi	RW
SCLKDIV	Clock divide synchronous bit 0	(Note 3)	RW

Notes :

1. The BBS bit is set to "0" when setting it to "0" by program. It remains unchanged if setting it to "1".
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.
3. Refer to the notes for the SU1HIM bit in the UiSMR2 register.

Figure 1.17.5. U0C1 to U4C1 Registers and U0SMR to U4SMR Registers



**Figure 1.17.6. U0SMR2 to U4SMR2 Registers**

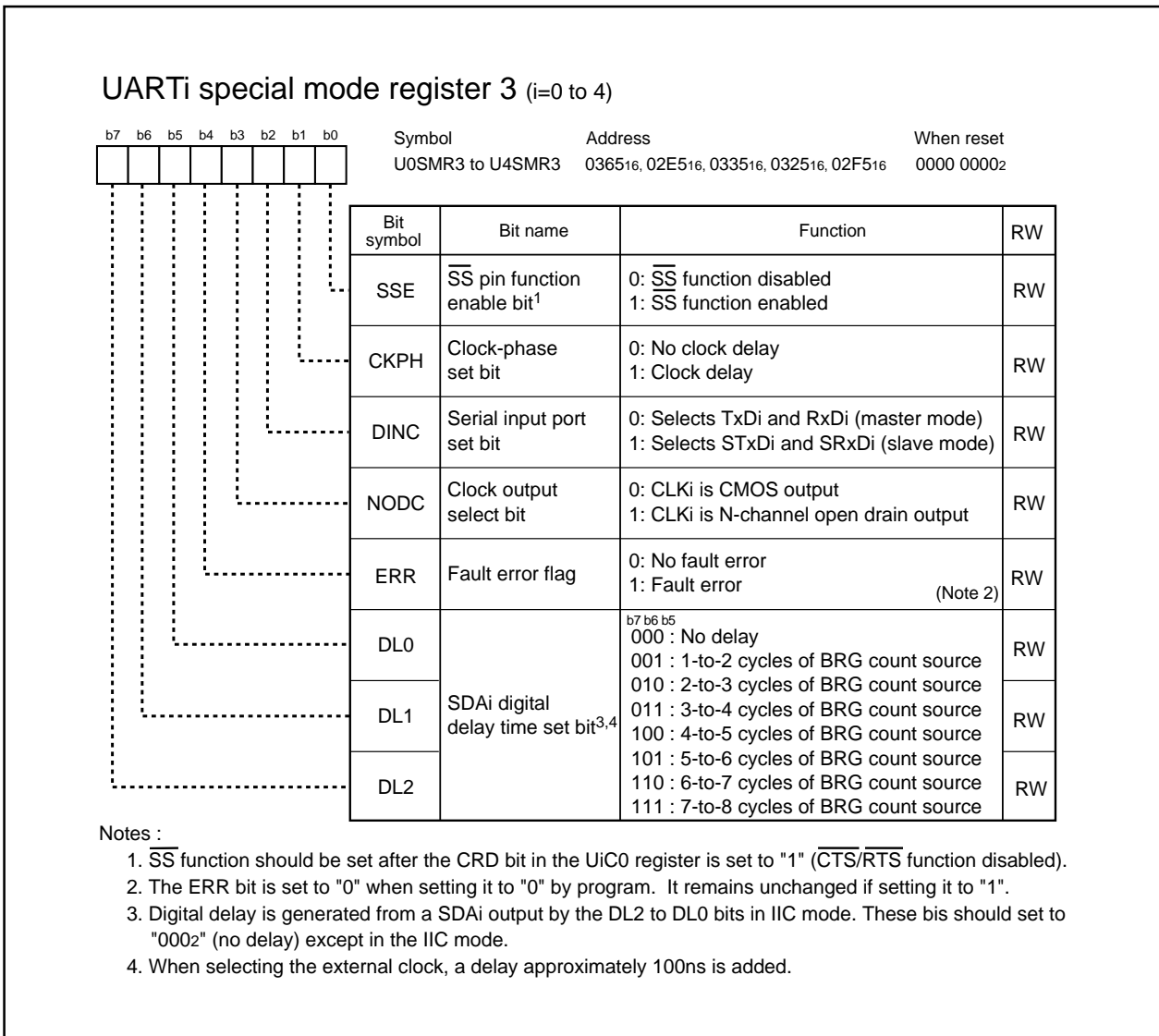


Figure 1.17.7. U0SMR3 to U4SMR3 Registers

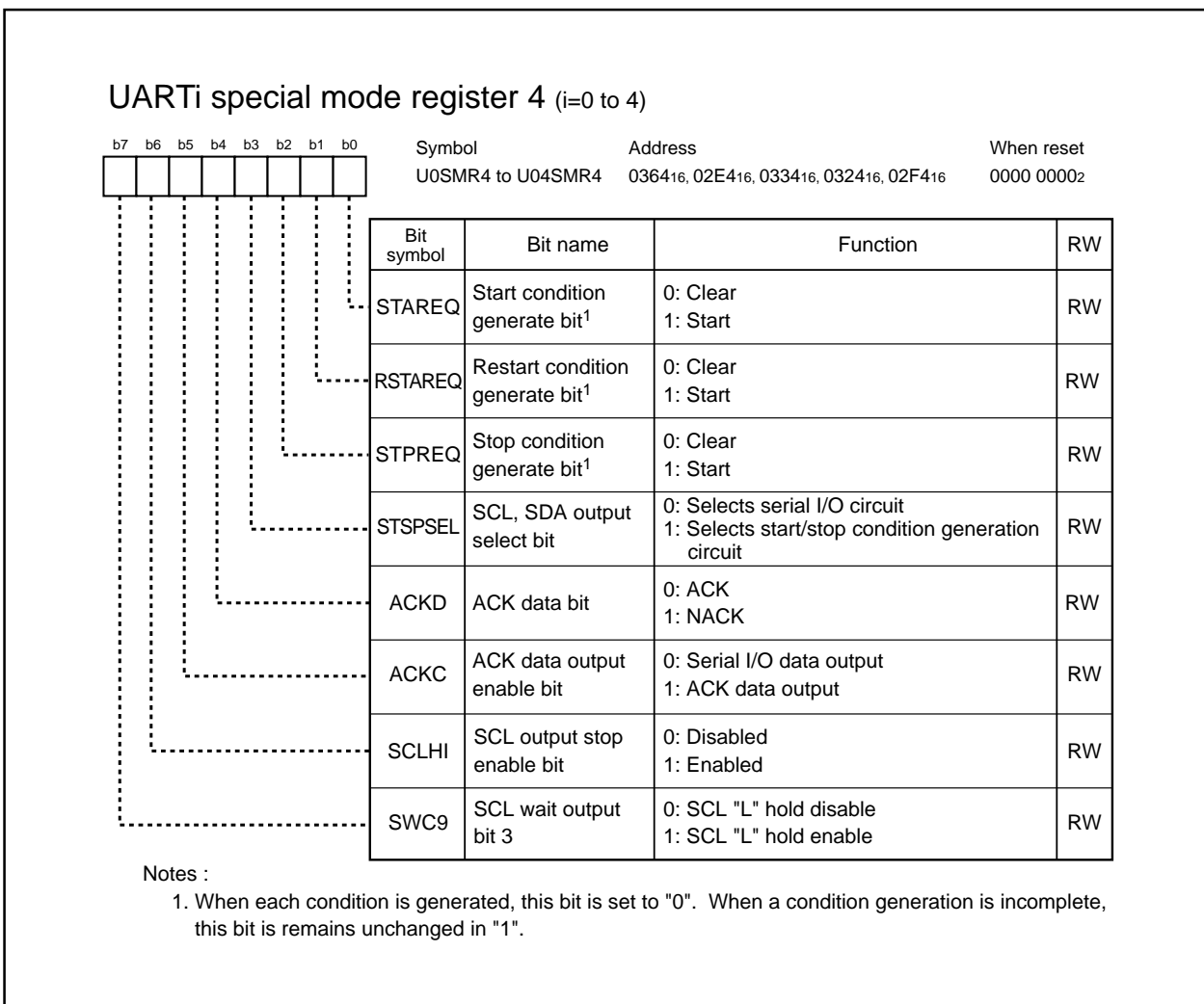


Figure 1.17.8. U0SMR4 to U4SMR4 Registers

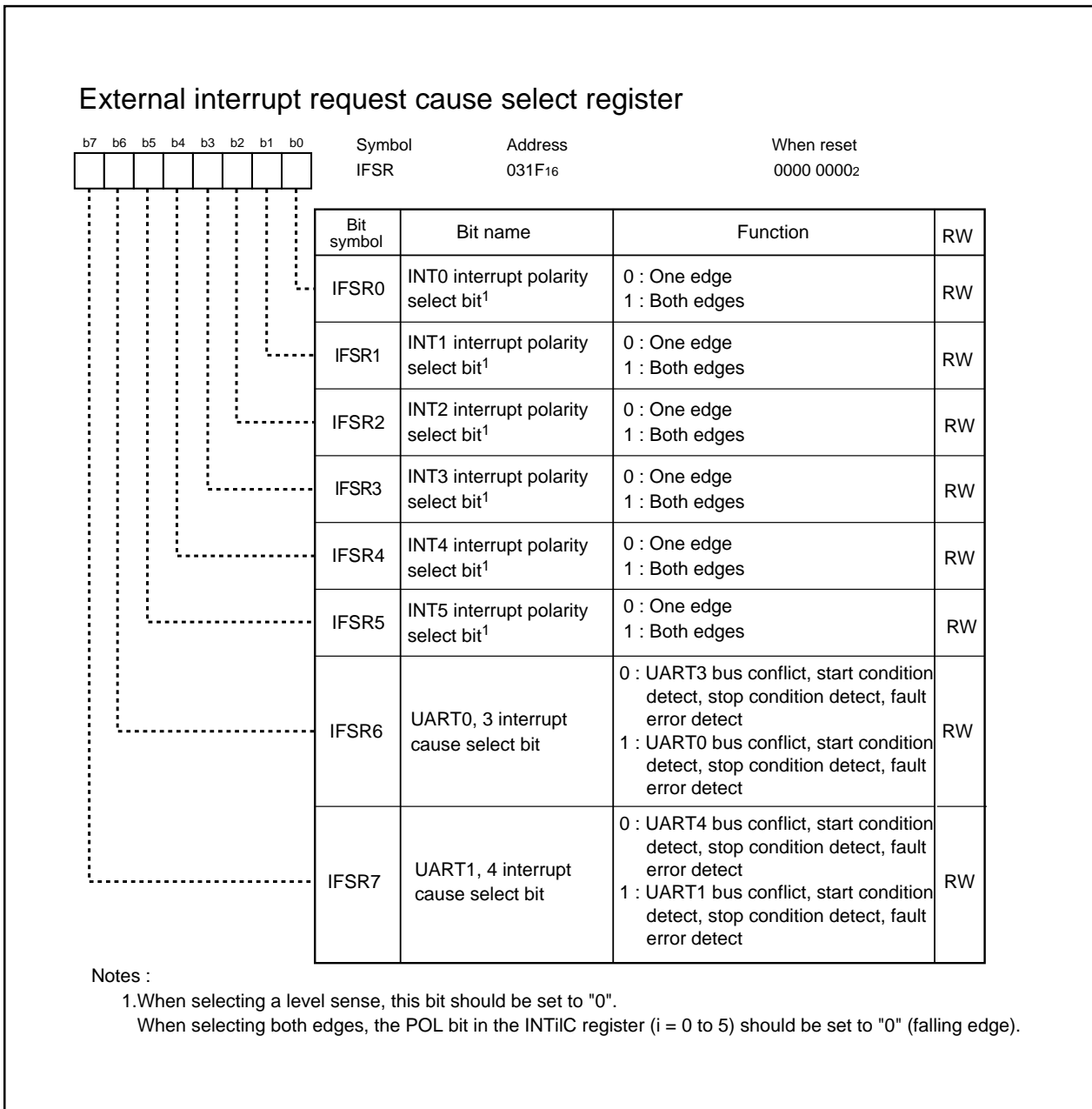


Figure 1.17.9. IFSR Register

## Serial I/O (Clock Synchronous Serial I/O)

## Clock Synchronous Serial I/O Mode

In clock synchronous serial I/O mode, the transfer clock transmits and receives data. Table 1.18.1 lists specifications of clock synchronous serial I/O mode. Table 1.18.2 lists registers to be used and register settings. Tables 1.18.3 to 1.18.5 list pin settings. When selecting UART<sub>i</sub> (i=0 to 4) operation mode, the TxDi pin outputs "H" before a transfer starts (this pin is in high-impedance state when N-channel open drain is selected).

Table 1.18.1. Clock Synchronous Serial I/O Mode Specifications

Item	Specification
Transfer data format	• Transfer data long : 8 bits
Transfer clock	• The CKDIR bit in the UiMR register (i=0 to 4) is set to "0" (internal clock is selected): $\frac{f_j}{2^{(m+1)}} \quad f_j=f_1, f_8, f_{2n^1} \quad m :$ setting value of the UiBRG register. 00 <sub>16</sub> to FF <sub>16</sub> . • CKDIR bit is set to "1" (external clock is selected) : input from CLK <sub>i</sub> pin
Transmit/receive control	• Selectable from the CTS function, RTS function or CTS/RTS function disabled
Transmit start condition	• To start transmitting, the following requirements should be met <sup>2</sup> : - The TE bit in the UiC1 register is set to "1" (transmit enable) - The TI bit in the UiC1 register is set to "0" (data available in the UiTB register) - A CTS <sub>i</sub> input level is in "L" when the CTS function is selected
Receive start condition	• To start receiving, the following requirements must be met <sup>2</sup> : - The RE bit in the UiC1 register is set to "1" (receive enable) - The TE bit is set to "1" (transmit enable) - The TI bit is set to "0" (data available in the UiTB register)
Interrupt request generation timing	• While transmitting, the following condition can be selected: - The UiIRS bit in the UiC1 register is set to "0" (Transmit buffer empty) : when data is transferred from the UiTB register to the UART <sub>i</sub> transmit register (transfer starts) - The UiIRS bit is set to "1" (transmission completes) : when data transfer from the UART <sub>i</sub> transmit register is completed • While receiving When data is transferred from the UART <sub>i</sub> receive register to the UiRB register (reception completes)
Error detection	• Overrun error <sup>3</sup> This error occurs when reading a seventh bit of the next received data before reading the UiRB register
Selectable function	• CLK polarity Either rising edge or falling edge of the transfer clock can be selected when a transferred data is output and input. • LSB first/MSB first Whether a data is transmitted/received in bit0 or in bit7 can be selected • Continuous receive mode Reception is enabled simultaneously by reading the UiRB register • Serial data logic inverse This function inverses transmitted/received data logically

Notes :

1. The CNT3 to CNT0 bits in the TRGSR register determines either "no division (n=0)" or "divide-by-2<sup>n</sup> (n=1 to 15)".
2. These conditions should be met when the internal clock is selected and the CKPOL bit in the UiC0 register is set to "0" (transmitted data is output on the falling edge of the transfer clock and received data is input on the rising edge) with the CLK<sub>i</sub> pin in "H" and the CKPOL bit is set to "1" (transmit data is output on the rising edge of the transfer clock and received data is input on the falling edge) with the CLK<sub>i</sub> pin in "L".
3. If an overrun error occurs, the next data will be set in the UiRB register. The IR bit in the SiRIC register does not change to "1" (interrupt request).

## Serial I/O (Clock Synchronous Serial I/O)

Table 1.18.2. Registers to be Used and Setting Value in Clock Synchronous Serial I/O Mode

Register	Bit	Function
UiTB	0 to 7	Set data to be transmitted
UiRB	0 to 7	Received data can be read
	OER	Overflow error flag
UiBRG	0 to 7	Set a baud rate
UiMR	SMD2 to SMD0	Set to "0012"
	CKDIR	Select the internal clock or external clock
	IOPOL	Set to "0"
UiC0	CLK1 to CLK0	Select a count source of the UiBRG register
	CRS	Select either CTS or RTS when using one of them
	TXEPT	Transmit register empty flag
	CRD	Select the CTS or RTS function enabled or disabled
	NCH	Select an output format of the TxDi pin
	CKPOL	Select transmit clock polarity
	UFORM	Select either LSB first or MSB first
UiC1	TE	When data transmission and reception are enabled set to "1"
	TI	Transmit buffer empty flag
	RE	When data reception is enabled set to "1"
	RI	Receive complete flag
	UiIRS	Select how the UARTi transmit interrupt is generated
	UiRRM	Set to "1" when using continuous receive mode
	UiLCH	Set to "1" when using data logic inverse
	SCLKSTPB	Set to "0"
UiSMR	0 to 7	Set to "0"
UiSMR2	0 to 7	Set to "0"
UiSMR3	0 to 2	Set to "0"
	NODC	Select a clock output format
	4 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"

i=0 to 4

## Serial I/O (Clock Synchronous Serial I/O)

Table 1.18.3. Pin Settings in Clock Synchronous Serial I/O Mode

Port	Function	Setting value		
		PS0 register	PSL0 register	PD6 register
P60	CTS0 input	PS0_0=0	-	PD6_0=0
	RTS0 output	PS0_0=1	-	-
P61	CLK0 input	PS0_1=0	-	PD6_1=0
	CLK0 output	PS0_1=1	-	-
P62	RxD0 input	PS0_2=0	-	PD6_2=0
P63	TxD0 output	PS0_3=1	-	-
P64	CTS1 input	PS0_4=0	-	PD6_3=0
	RTS1 output	PS0_4=1	PSL0_4=0	-
P65	CLK1 input	PS0_5=0	-	PD6_5=0
	CLK1 output	PS0_5=1	-	-
P66	RxD1 input	PS0_6=0	-	PD6_6=0
P67	TxD1 output	PS0_7=1	-	-

Table 1.18.4. Pin Settings (Continued)

Port	Function	Setting value			
		PS1 register	PSL1 register	PSC register	PD7 register
P70 <sup>1</sup>	TxD2 output	PS1_0=1	PSL1_0=0	PSC_0=0	-
P71 <sup>1</sup>	RxD2 input	PS1_1=0	-	-	PD7_1=0
P72	CLK2 input	PS1_2=0	-	-	PD7_2=0
	CLK2 output	PS1_2=1	PSL1_2=0	PSC_2=0	-
P73	CTS2 input	PS1_3=0	-	-	PD7_3=0
	RTS2 output	PS1_3=1	PSL1_3=0	PSC_3=0	-

Notes :

1. N-channel open drain output

Table 1.18.5. Pin Settings (Continued)

Port	Function	Setting value		
		PS3 register <sup>1</sup>	PSL3 register	PD9 register <sup>1</sup>
P90	CLK3 input	PS3_0=0	-	PD9_0=0
	CLK3 output	PS3_0=1	-	-
P91	RxD3 input	PS3_1=0	-	PD9_1=0
P92	TxD3 output	PS3_2=1	PSL3_2=0	-
P93	CTS3 input	PS3_3=0	PSL3_3=0	PD9_3=0
	RTS3 output	PS3_3=1	-	-
P94	CTS4 input	PS3_4=0	PSL3_4=0	PD9_4=0
	RTS4 output	PS3_4=1	-	-
P95	CLK4 input	PS3_4=0	PSL3_5=0	PD9_5=0
	CLK4 output	PS3_5=1	-	-
P96	TxD4 output	PS3_6=1	-	-
P97	RxD4 input	PS3_7=0	-	PD9_7=0

Notes :

1. The PD9 and PS3 registers should be set with subsequent to the instruction that the PRC2 bit in the PRCR register is set to "1" (write enable). Avoid generating an interrupt and operating a DMA transfer between the instruction to set the PRC2 bit to "1" and the instruction to write to the PD9 and PS3 registers.

Serial I/O (Clock Synchronous Serial I/O)

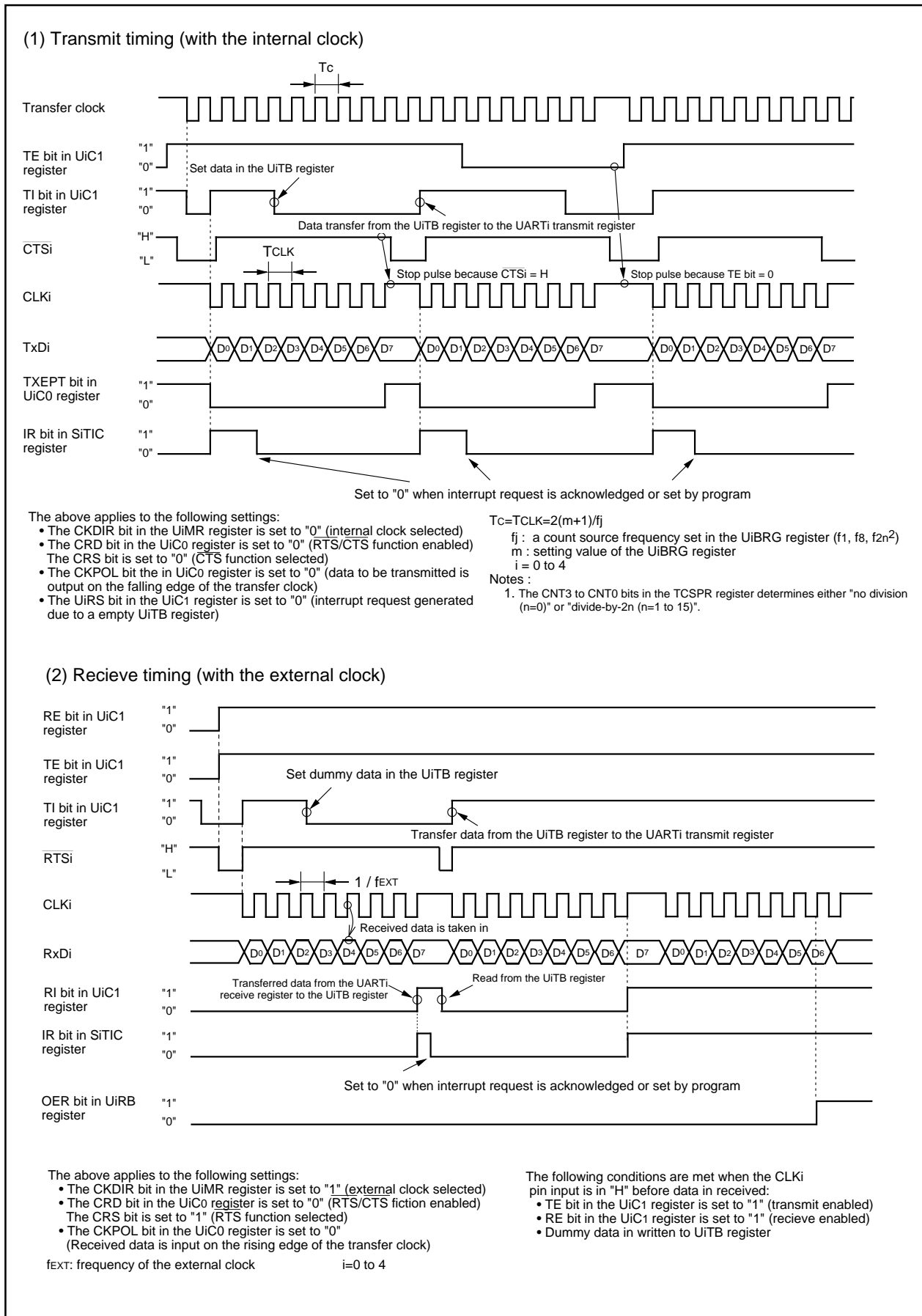


Figure 1.18.1. Transmit and receive Operation

Serial I/O (Clock Synchronous Serial I/O)

### 1. Selecting CLK Polarity

As shown in Figure 1.18.2, the CKPOL bit in the UiC0 register (i=0 to 4) determines polarity of the transfer clock.

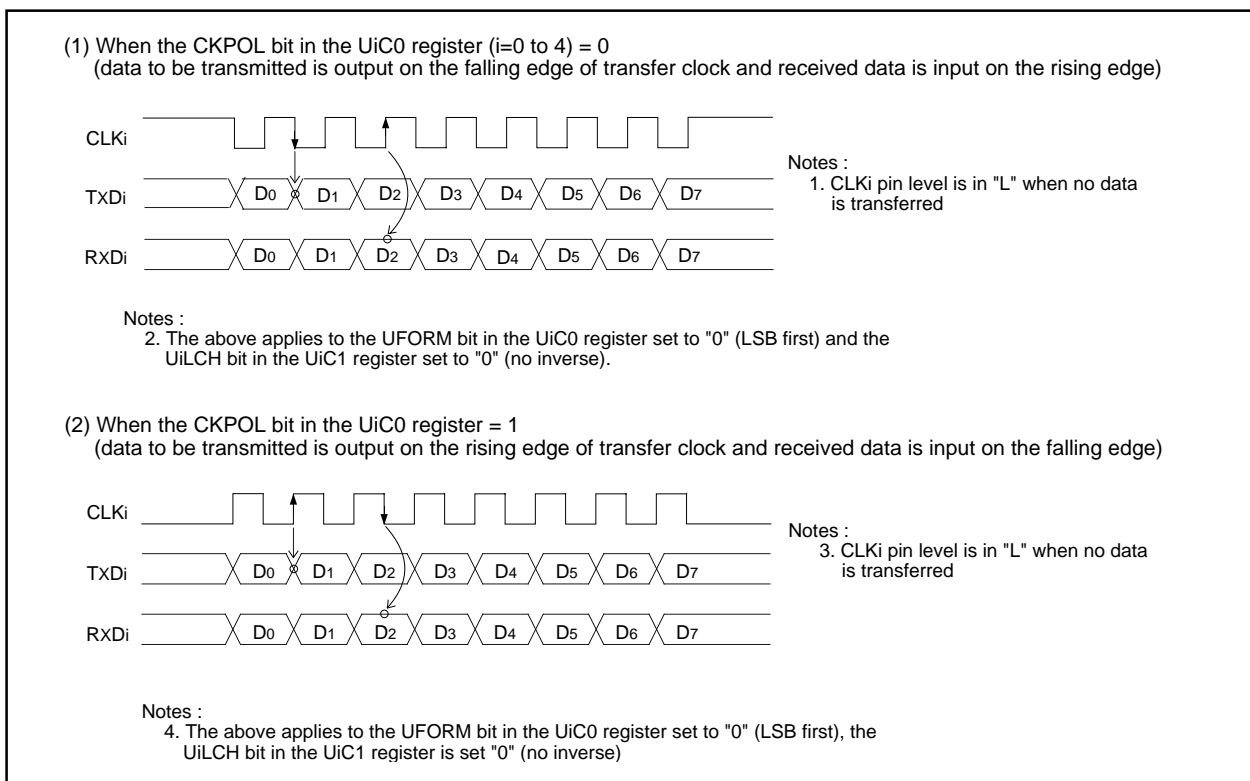


Figure 1.18.2. Transfer Clock Polarity

### 2. Selecting LSB First/MSB First

As shown in Figure 1.18.3, the UFORM bit in the UiC0 register (i=0 to 4) determines data transfer format.

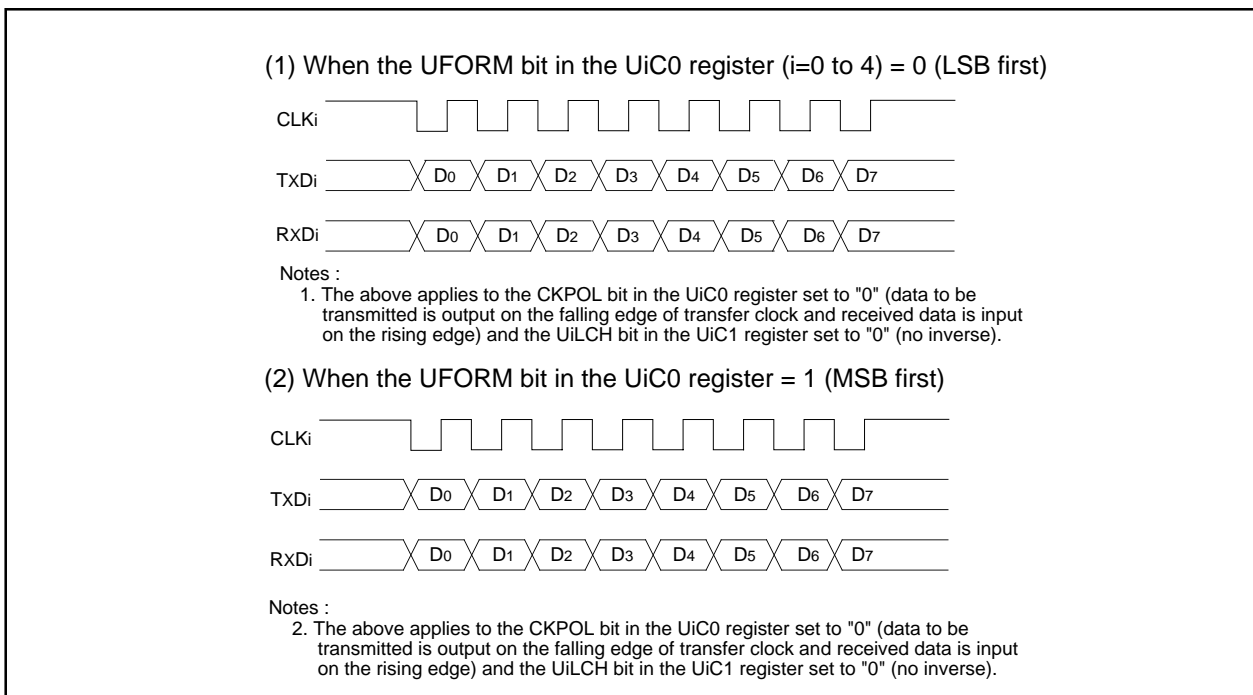


Figure 1.18.3. Transfer Format

## Serial I/O (Clock Synchronous Serial I/O)

### 3. Continuous Receive Mode

When the UiRRM bit in the UiC1 register (i=0 to 4) is set to "1" (continuous receive mode), the TI bit is set to "0" (data available in the UiTB register) by reading the UiRB register. When the UiRRM bit is set to "1", avoid setting a dummy data in the UiTB register by program.

### 4. Serial Data Logic Inverse Function

When the UiLCH bit in the UiC1 register is set to "1" (inverse), while transmitting, data logic written in the UiTB register is inverted to transmit. When reading the UiRB register, the inverted receive data logic can be read. Figure 1.18.4 shows a serial data logic.

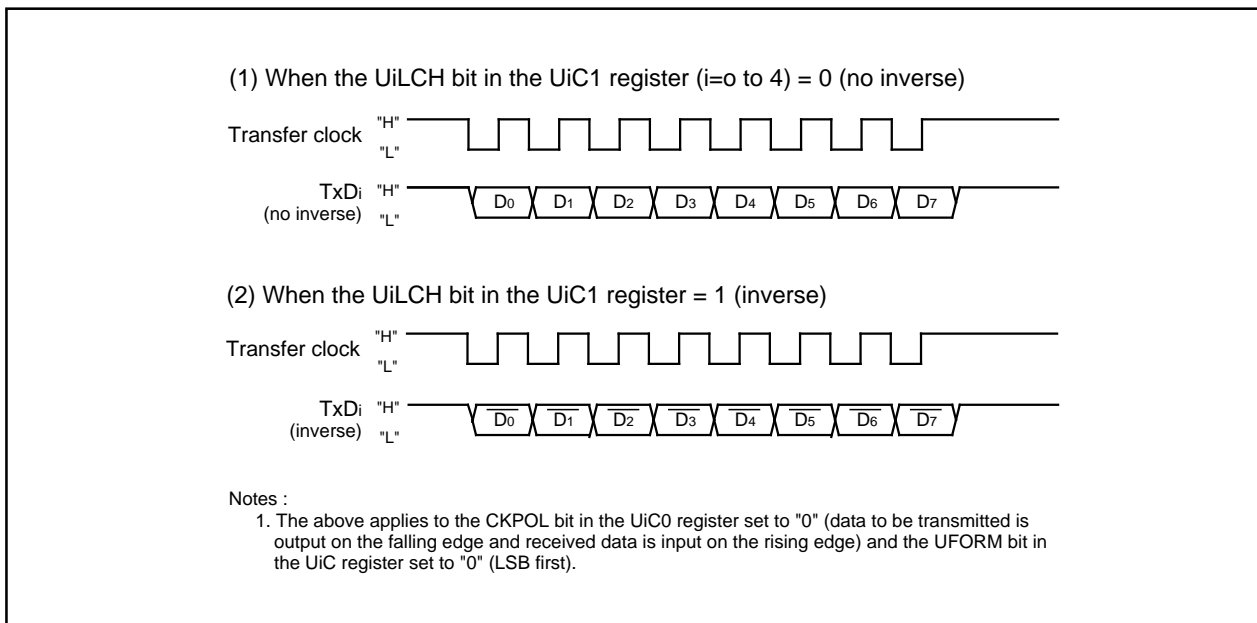


Figure 1.18.4. Switching Serial Data Logic Switching

## Serial I/O (UART)

**Clock Asynchronous serial I/O (UART) mode**

In UART mode, data is transmitted and received after setting the desired baud rate and data transfer format. Table 1.19.1 lists specifications of UART mode.

**Table 1.19.1. UART Mode Specifications**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>• character bit (transfer data) : selectable from 7 bits, 8 bits, or 9 bits</li> <li>• Start bit: 1 bit</li> <li>• Parity bit: selectable from Odd, even, or nothing</li> <li>• Stop bit: selectable from 1 bit or 2 bits</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>• When the CKDIR bit in the UiMR register is set to "0" (internal clock is selected) :  <math>f_j/16(m+1)</math> <math>f_j = f_1, f_8, f_{2n}^1</math> <math>m</math>: setting value of the UiBRG register 00<sub>16</sub> to FF<sub>16</sub></li> <li>• When the CKDIR bit is set to "1" (external clock is selected) :  <math>f_{EXT}/16(m+1)</math> <math>f_{EXT}</math>: input clock from the CLKi pin</li> </ul>
Transmit/receive control	<ul style="list-style-type: none"> <li>• Selectable from CTS function, RTS function or CTS/RTS function disabled</li> </ul>
Transmit start condition	<ul style="list-style-type: none"> <li>• To start transmitting, the following requirements should be met: <ul style="list-style-type: none"> <li>- The TE bit in the UiC1 register is set to "1" (transmit enable)</li> <li>- The TI bit in the UiC1 register is set to "0" (data available in transmit buffer register)</li> <li>- <math>\overline{CTS}</math> input level is in "L" when <math>\overline{CTS}</math> function is selected</li> </ul> </li> </ul>
Receive start condition	<ul style="list-style-type: none"> <li>• To start receiving, the following requirements should be met: <ul style="list-style-type: none"> <li>- The RE bit in the UiC1 register is set to "1" (receive enable)</li> <li>- The start bit is detected</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• While transmitting, the following condition can be selected: <ul style="list-style-type: none"> <li>- The UiIRS bit in the UiC1 register is set to "0" (transmit buffer empty) : when data is transferred from the UiTB register to the UARTi transmit register (starts transfer)</li> <li>- The UiIRS bit is set to "1" (transmission completes) : when data transmission from the UARTi transfer register is completed</li> </ul> </li> <li>• While receiving <ul style="list-style-type: none"> <li>- When data is transferred from UARTi receive register to UiRB register (reception completes)</li> </ul> </li> </ul>
Error detection	<ul style="list-style-type: none"> <li>• Overrun error<sup>2</sup> This error occurs when the next data is started to receive and a bit immediately before last stop bit is received (first stop bit when selecting 2 stop bits)</li> <li>• Framing error This error occurs when the number of stop bits set is not detected</li> <li>• Parity error When parity is enabled, this error occurs when the number of "1" in parity and character bits does not match the number of "1" set</li> <li>• Error sum flag This flag is set to "1" when any of an overrun, framing or parity errors occur</li> </ul>
Selectable function	<ul style="list-style-type: none"> <li>• LSB first or MSB first Whether a data is transmitted/received in bit 0 or in bit 7 can be selected</li> <li>• Serial data logic inverse Logic values of data to be transmitted and received data are inverted. The start bit and stop bit are not inverted</li> <li>• TxD, RxD I/O polarity switching TxD pin output and RxD pin input are inverted. All I/O data levels are also inverted.</li> </ul>

Notes :

1. The CNT3 to CNT0 bits in the TRGSR register determines either "no division (n=0)" or "divide-by-2n (n=1 to 15)".
2. If an overrun error occurs, the UiRB register becomes indeterminate. The IR bit in the SiRIC register remains unchanged in "0" (interrupt request).

## Serial I/O (UART)

Table 1.19.2 lists registers to be used and register settings. Tables 1.19.3 to 1.19.5 list pin settings. When UARTi operation mode is selected, the TxDi pin outputs "H" before transfer is started (this pin is in high-impedance state when N-channel open drain is selected).

**Table 1.19.2. Registers to be Used and Settings in UART**

Register	Bit	Function
UiTB	0 to 8	Set data to be transmitted <sup>1</sup>
UiRB	0 to 8	Received data can be read <sup>1</sup>
	OER, FER, PER, SUM	Error flags
UiBRG	0 to 7	Set a baud rate
UiMR	SMD2 to SMD0	When transfer data is 7 bits long, set to "1002" When transfer data is 8 bits long, set to "1012" When transfer data is 9 bits long, set to "1102"
	CKDIR	Select the internal clock or external clock
	STPS	Select a stop bit length
	PRY, PRYE	Select parity enable or disable, odd or even
	IOPOL	Select TxD / RxD I/O polarity
UiC0	CLK0, CLK1	Select a count source of the UiBRG register
	CRS	Select either CTS or RTS when using one of them
	TXEPT	Transfer register empty flag
	CRD	Select the CTS or RTS function enabled or disabled
	NCH	Select an output format of the TxDi pin
	CKPOL	Set to "0"
	UFORM	When transfer data is 8 bits long, LSB first or MSB first can be selected. When transfer data is 7 bits or 9 bits long, set to "0".
UiC1	TE	When transfer is enabled, set to "1"
	TI	Transfer buffer empty flag
	RE	Set to "1" when data reception is enabled
	RI	Receive complete flag
	UiIRS	Select how the UARTi transmit interrupt is generated
	UiRRM	Set to "0"
	UiLCH	When using continuous receive mode, set to "1"
	UiERE	Set to "0"
UiSMR	0 to 7	Set to "0"
UiSMR2	0 to 7	Set to "0"
UiSMR3	0 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"

## Notes :

1. Bits to be used are bits 0 to 6 when transfer data is 7 bits long, bits 0 to 7 when 8 bits long, bits 0 to 8 when 9 bits long.

## Serial I/O (UART)

Table 1.19.3. Pin Settings in UART

Port	Function	Bit and setting value		
		PS0 register	PSL0 register	PD6 register
P60	CTS0 input	PS0_0=0	–	PD6_0=0
	RTS0 output	PS0_0=1	–	–
P61	CLK0 input	PS0_1=0	–	PD6_1=0
P62	RxD0 input	PS0_2=0	–	PD6_2=0
P63	TxD0 output	PS0_3=1	–	–
P64	CTS1 input	PS0_4=0	–	PD6_3=0
	RTS1 output	PS0_4=1	PSL0_4=0	–
P65	CLK1 input	PS0_5=0	–	PD6_5=0
P66	RxD1 input	PS0_6=0	–	PD6_6=0
P67	TxD1 output	PS0_7=1	–	–

Table 1.19.4. Pin Settings (Continued)

Port	Function	Bit and setting value			
		PS1 register	PSL1 register	PSC register	PD7 register
P70 <sup>1</sup>	TxD2 output	PS1_0=1	PSL1_0=0	PSC_0=0	–
P71 <sup>1</sup>	RxD2 input	PS1_1=0	–	–	PD7_1=0
P72	CLK2 input	PS1_2=0	–	–	PD7_2=0
P73	CTS2 input	PS1_3=0	–	–	PD7_3=0
	RTS2 output	PS1_3=1	PSL1_3=0	PSC_3=0	–

Notes :

1. N-channel open drain output

Table 1.19.5. Pin Settings (Continued)

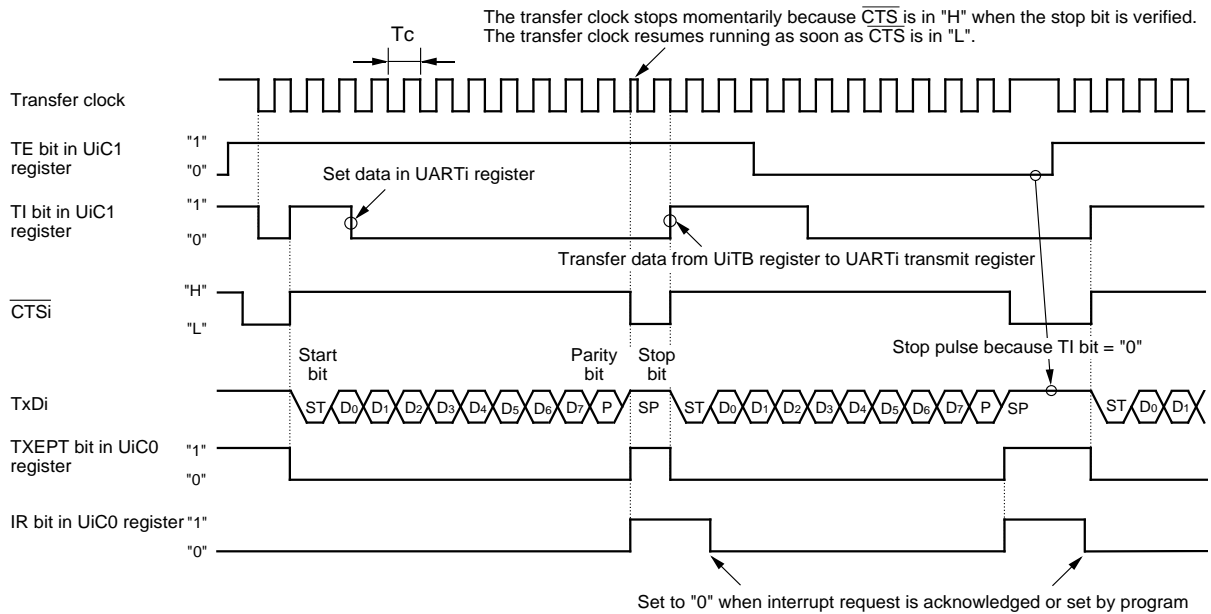
Port	Function	Bit and setting value		
		PS3 register <sup>1</sup>	PSL3 register	PD9 register <sup>1</sup>
P90	CLK3 input	PS3_0=0	–	PD9_0=0
P91	RxD3 input	PS3_1=0	–	PD9_1=0
P92	TxD3 output	PS3_2=1	PSL3_2=0	–
P93	CTS3 input	PS3_3=0	PSL3_3=0	PD9_3=0
	RTS3 output	PS3_3=1	–	–
P94	CTS4 input	PS3_4=0	PSL3_4=0	PD9_4=0
	RTS4 output	PS3_4=1	–	–
P95	CLK4 input	PS3_5=0	PSL3_5=0	PD9_5=0
P96	TxD4 output	PS3_6=1	–	–
P97	RxD4 input	PS3_7=0	–	PD9_7=0

Notes :

1. The PD9 and PS3 registers should be set with subsequent to the instruction that the PRC2 bit in the PRCR register is set to "1" (write enable). Avoid generating an interrupt and operating a DMA transfer between the instruction to set the PRC2 bit to "1" and the instruction to write to the PD9 and PS3 registers.

Serial I/O (UART)

(1) 8-bit data transmit timing (with parity enabled and 1 stop bit)



i=0 to 4

The above timing applies to the following settings :

- The PRYE bit in the UiMR register is set to "1" (parity enabled)
- The SRPS bit in the UiMR register is set to "0" (1-stop bit)
- The CRD bit in the UiC0 register is set to "0" and the CRS bit is set to "0" (CTS function selected)
- The UiIRS bit in the UiC1 register is set to "1" (interrupt request is generated due to the UiTB register empty)

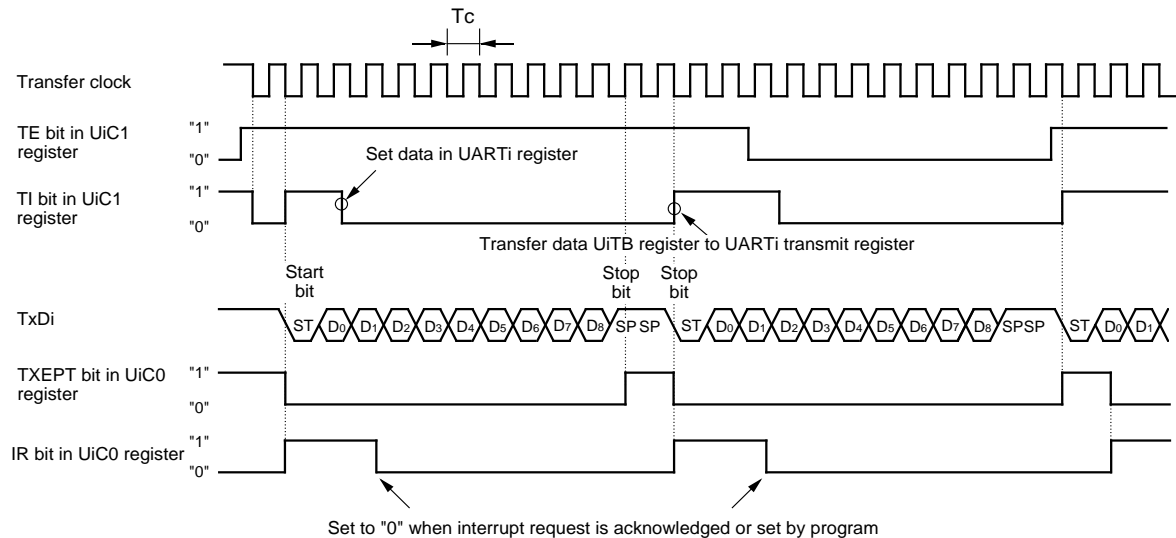
$$T_c = 16(m + 1) / f_j \text{ or } 16(m + 1) / f_{EXT}$$

- $f_j$  : a count source frequency set in the UiBRG register ( $f_1, f_8, f_{2n^1}$ )
- $f_{EXT}$  : a count source frequency set in the UiBRG register (extern: clock)
- $m$  : setting value of the UiBRG register

Notes :

1. The CNT3 to CNT0 bits in the TCSPR register determines either "1 division ( $n=0$ )" or "divide-by-2n ( $n=1$  to 15)".

(2) 9-bit data transmit timing (with parity disabled and 2-stop bit)



i=0 to 4

The above timing applies to the following settings :

- The PRYE bit in the UiMR register is set to "0" (parity disabled)
- The SRPS bit in the UiMR register is set to "1" (2-stop bit)
- The CRD bit in the UiC0 register is set to "1" (CTS function disabled)
- The UiIRS bit in the UiC1 register is set to "1" (interrupt request is generated due to the UiTB register empty)

$$T_c = 16(m + 1) / f_j \text{ or } 16(m + 1) / f_{EXT}$$

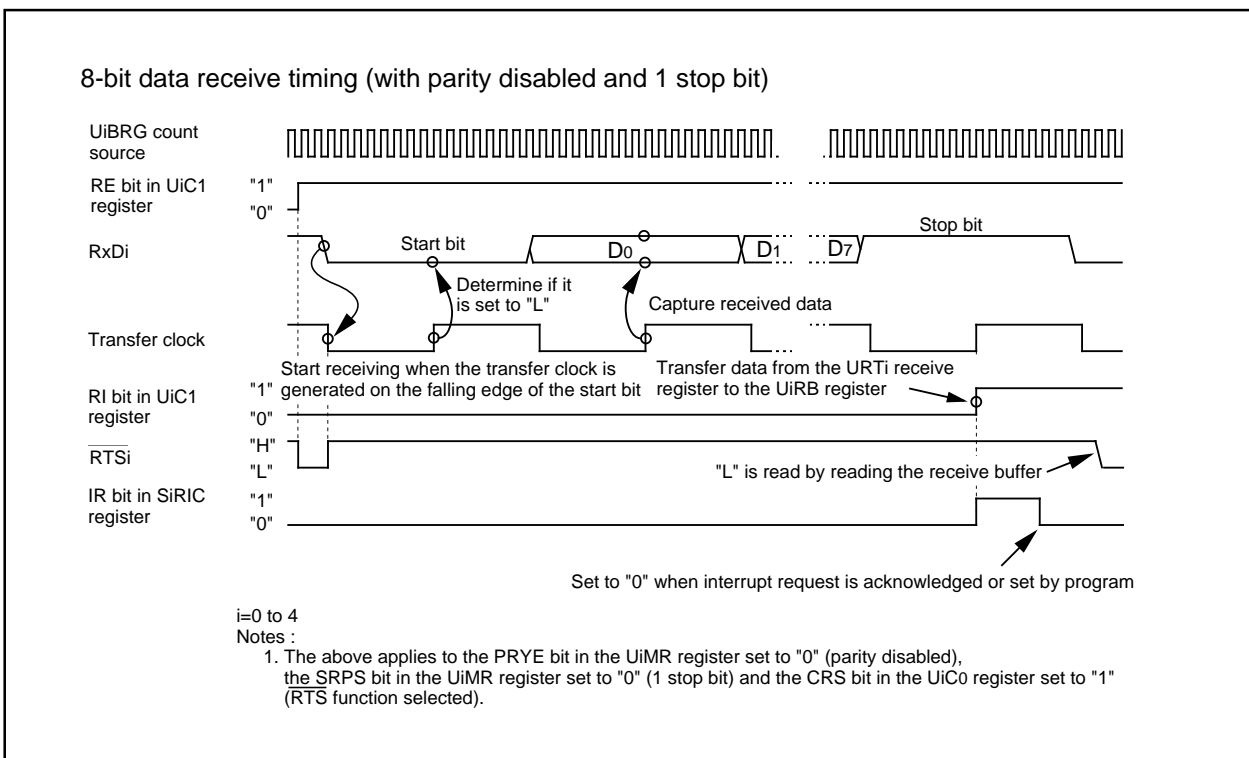
- $f_j$  : a count source frequency set in the UiBRG register ( $f_1, f_8, f_{2n^1}$ )
- $f_{EXT}$  : a count source frequency set in the UiBRG register (extern: clock)
- $m$  : setting value of the UiBRG register

Notes :

1. The CNT3 to CNT0 bits in the TCSPR register determines either "1 division ( $n=0$ )" or "divide-by-2n ( $n=1$  to 15)".

Figure 1.19.1. Transmit Operation

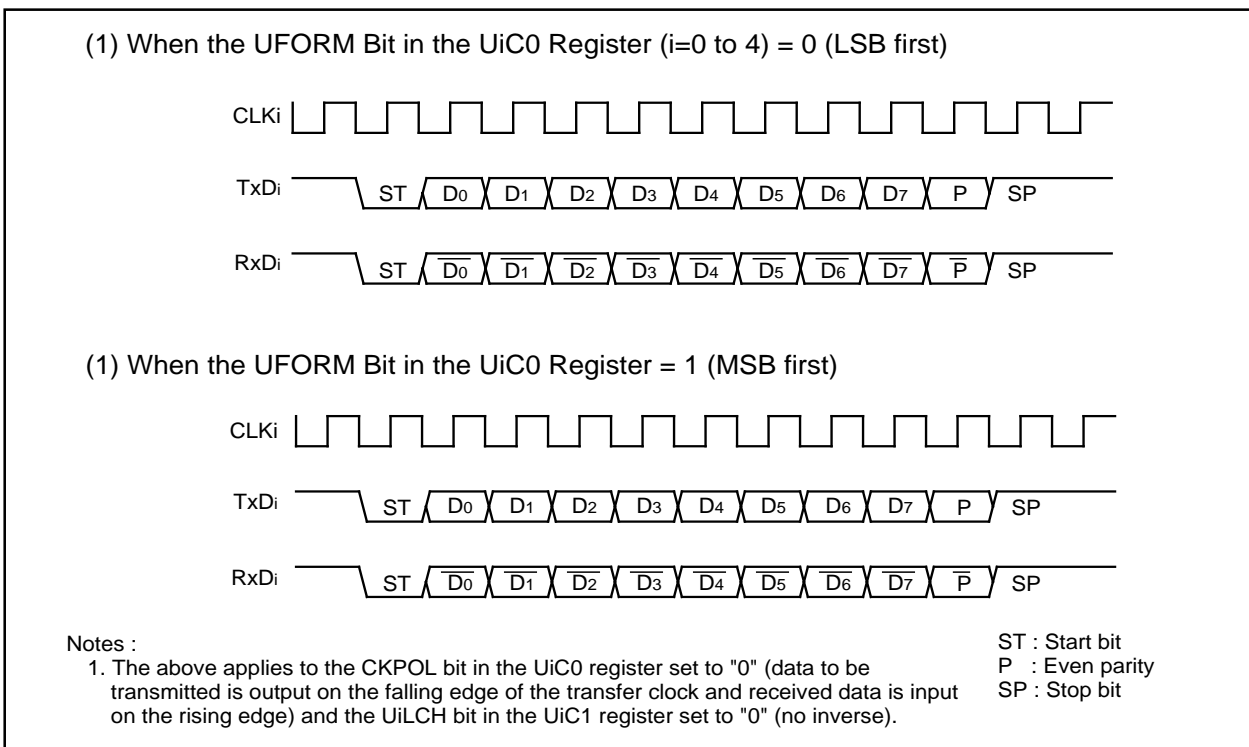
**Serial I/O (UART)**



**Figure 1.19.2. Receive Operation**

**1. LSB First / MSB First Select Function**

As shown in Figure 1.19.3, the UFORM bit in the UiC0 register determines data transfer format. This function is available for 8-bit transfer data.



**Figure 1.19.3. Transfer Format**

## Serial I/O (UART)

### 2. Serial Data Logic Inverse Function

When writing to and reading from the UiTB register, data logic is inverted by setting the UiLCH bit in the UiC1 register. Figure 1.19.4 shows a serial data logic.

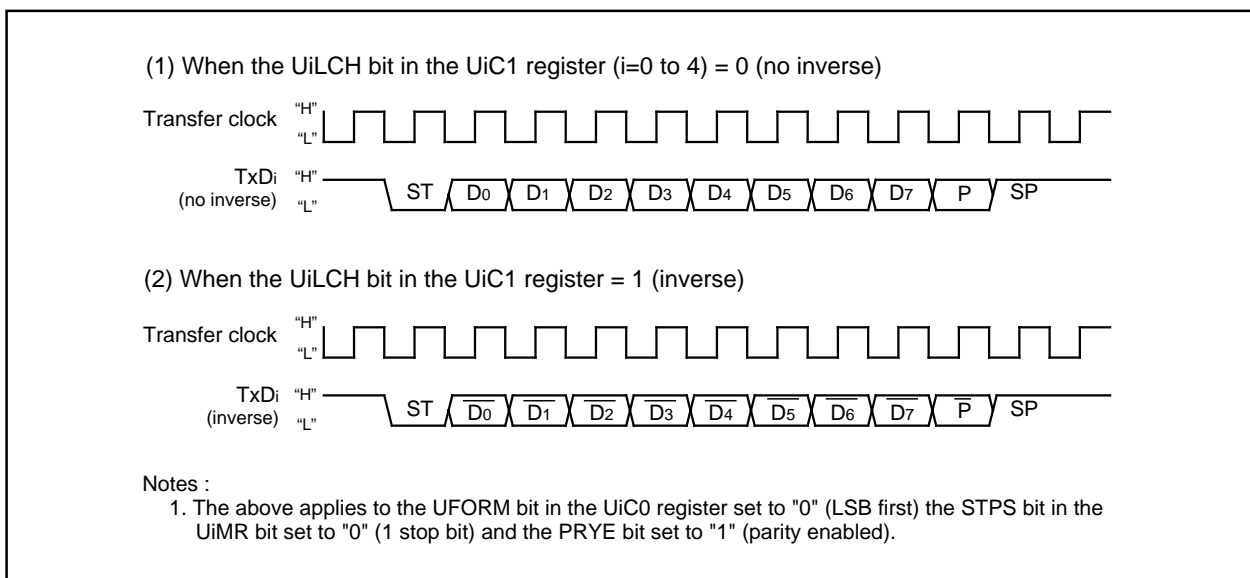


Figure 1.19.4. Serial Data Logic Inverse

### 3. TxD and RxD I/O Polarity Inverse Function

TxD pin output and RxD pin input are inverted. All I/O data level, including the start bit, stop bit and parity bit, are inverted. Figure 1.19.5 shows TxD and RxD I/O polarity inverse.

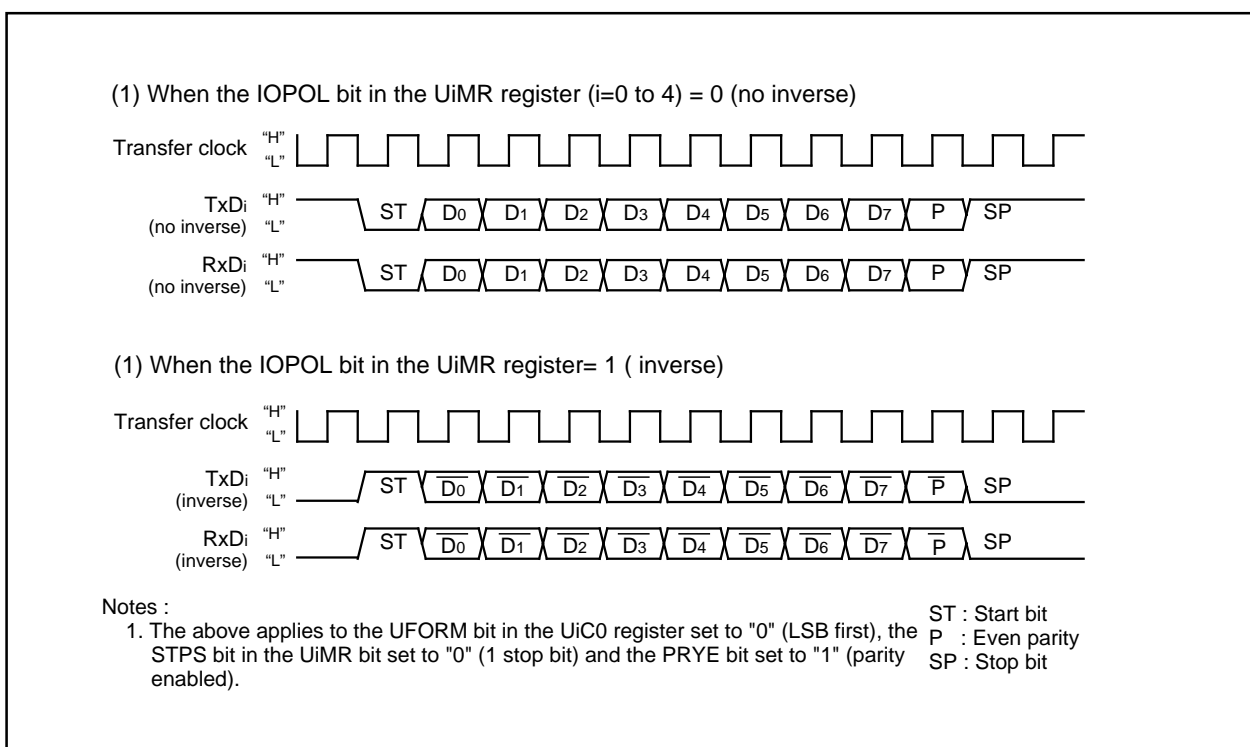


Figure 1.19.5. TxD, RxD I/O Polarity reverse function

## Serial I/O (Special Function)

**Special Function**

Serial I/O has the following modes.

- Special mode 1 (IIC mode)
- Special mode 2
- Special mode 3 (Clock-divided synchronous function, GCI mode)
- Special mode 4 (Bus conflict detect function, IE mode)
- Special mode 5 (SIM mode)

**1. Special Mode 1 (IIC Mode)**

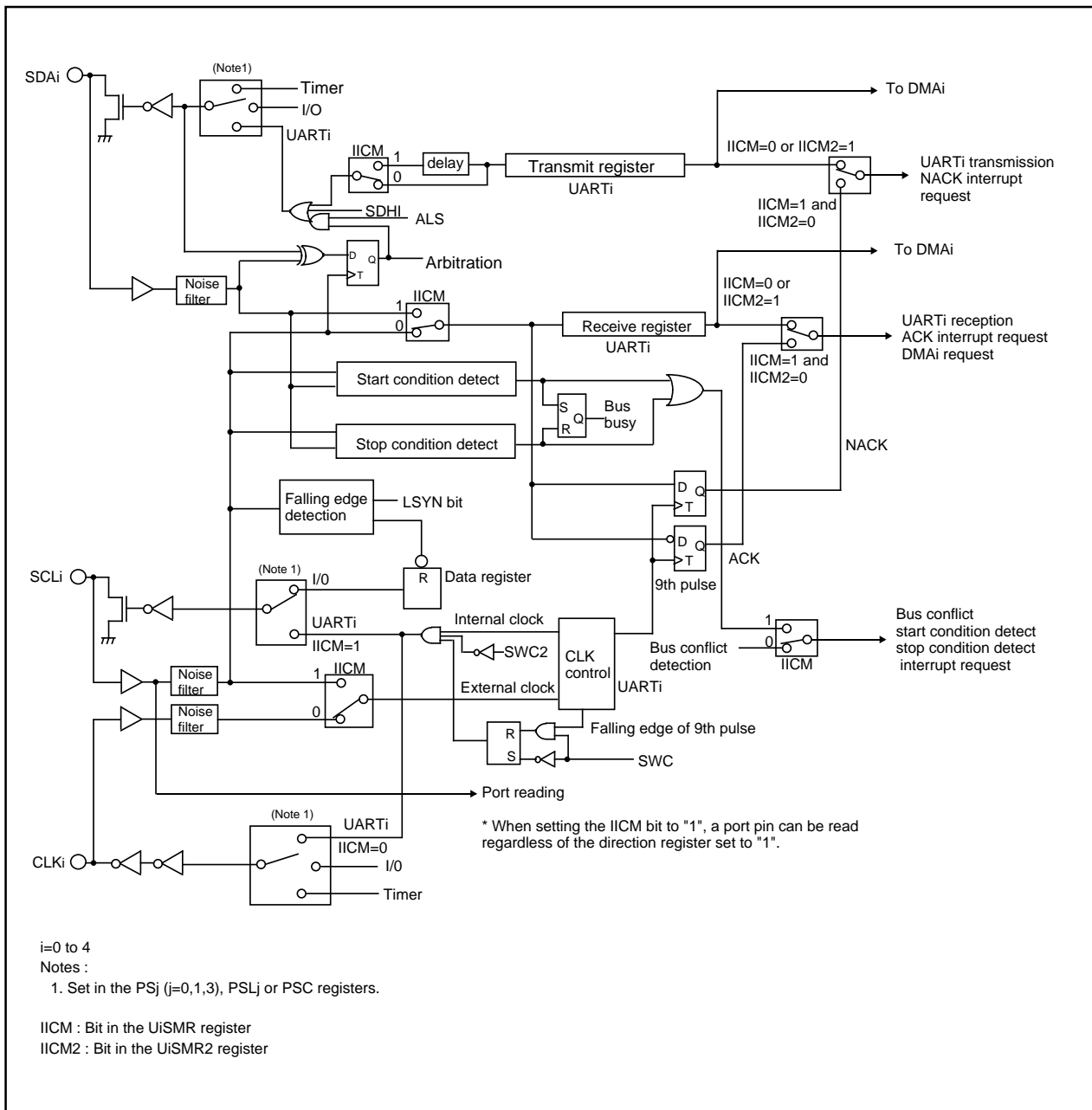
IIC mode is a mode to communicate with external devices with a simplified I<sup>2</sup>C. Table 1.20.1 lists specifications of IIC mode. Table 1.20.2 lists registers to be used and settings, Table 1.20.3 lists each function. Figure 1.20.1 shows a block diagram of IIC mode. Figure 1.20.2 shows SCLi timing (i=0 to 4). Tables 1.20.4 to 1.20.6 list pin settings.

As shown in Table 1.20.3, IIC mode is entered when the SMD2 to SMD0 bits in the UiMR register is set to "0102" and the IICM bit in the UiMR register is set to "1". SDAi output changes after SCLi is stable in "L" due to a SDAi transmitted output via the delay circuit.

**Table 1.20.1. IIC Mode Specifications**

Item	Specifications
Interrupt	Start condition detect, stop condition detect, no acknowledgment detect, acknowledgment detect
Selectable function	<ul style="list-style-type: none"> <li>• Arbitration lost An update timing of the ABT bit in the UiRB register can be selected Refer to the paragraph "• Arbitration"</li> <li>• SDAi digital delay Selectable from no digital delay or 2 to 8 cycle delay of a count source of BRG Refer to the paragraph "• SDAi"</li> <li>• Clock phase setting Selectable from clock delay or no clock delay Refer to the paragraph "• Transfer clock"</li> </ul>

**Serial I/O (Special Function)**



**Figure 1.20.1. IIC Mode Block Diagram**

Serial I/O (Special Function)

**Table 1.20.2. Registers to Be Used and Settings (IIC Mode)**

Register	Bit	Function	
		Master	Slave
UiTB	0 to 7	Set data to be transmitted	
UiRB	0 to 7	Received data can be read	
	8	R/W and ACK bits can be read	
	ABT	Arbitration lost detect flag	Disabled
	OER	Overrun error flag	
UiBRG	0 to 7	Set baud rate	Disabled
UiMR	SMD2 to SMD0	Set to "0102"	
	CKDIR	Set to "1" with a slave, set to "0" with a master	Set to "1"
	IOPOL	Set to "0"	
UiC0	CLK1 to CLK0	Select a count source of the UiBRG register	Disabled
	CRS	Disabled since CRD = 1	
	TXEPT	Transfer register empty flag	
	CRD, NCH	Set to "1"	
	CKPOL	Set to "0"	
	UFORM	Set to "1"	
UiC1	TE	When data transmission is enabled, set to "1"	
	TI	Transfer buffer empty flag	
	RE	When data reception is enabled, set to "1"	
	RI	Receive complete flag	
	UiIRS	Select how the UARTi transmit interrupt is generated	
	UiRRM, UiLCH, SCLKSTPB	Set to "0"	
UiSMR	IICM	Set to "1"	
	ABC	Select arbitration lost detect timing	Disabled
	BBS	Bus busy flag	
	LSYN	Set to "0"	
	3 to 7	Set to "0"	
UiSMR2	IICM2	See Table 1.20.3	
	CSC	When clock synchronization is enabled, set to "1"	Set to "1"
	SWC	Set to "1" when SCLi output is fixed to output "L" on the falling edge of ninth bit of the transfer clock	
	ALS	Set to "1" when SDAi output is terminated by detecting the arbitration lost	Not used. Set to "0".
	STC	Not used. Set to "0".	Set to "1" when UARTi is reset by detecting the start condition
	SWC2	When SCL output is forcibly set to "L", set to "1"	
	SDHI	When SDA output is disabled, set to "1"	
	SU1HIM	Set to "0"	
UiSMR3	SSE	Set to "0"	
	CKPH	See Table 1.20.3.	
	DINC, NODC, ERR	Set to "0"	
	DL2 to DL0	Set a digital delay value	
UiSMR4	STAREQ	When generating a start condition, set to "1"	Not used. Set to "0".
	RSTAREQ	When generating a restart condition, set to "1"	
	STPREQ	When using condition generating function, set to "1"	
	STSPSEL	When using a condition generating function, set to "1"	
	ACKD	Select ACK or NACK	
	ACKC	When ACK data is output, set to "1"	
	SCLHI	Set to "1" when SCL output stop is enabled by detecting an arbitration lost	Not used. Set to "0".
	SWC9	Not used. Set to "0".	Set to "1" when SCLi output is fixed to output "L" on the falling edge of ninth bit of the transfer clock
IFSR	IFSR6, IFSR7	Select how the start condition and stop condition detect interrupts are generated	

i=0 to 4

## Serial I/O (Special Function)

Table 1.20.3. IIC Mode Functions

Function	Clock synchronous serial I/O mode (SMD2 to SMD0=0012, IICM=0)	IIC mode (SMD2 to SMD0=0102, IICM=1)			
		IICM2=0 (NACK/ACK interrupt)		IICM2=1 (UART transmit / UART receive interrupt)	
		CKPH=0 (No clock delay)	CKPH=1 (clock delay)	CKPH=0 (No clock delay)	CKPH=1 (clock delay)
Intrrupt numbers 39 to 41 generated by <sup>1</sup> (See Figure 1.20.2)	-	Start condition detect, stop condition detect (See Table 1.20.7)			
Intrrupt number 17, 19, 33, 35, 37 generated by <sup>1</sup> (See Figure 1.20.2)	UARTi transmission Transmission start or transmission completed (selected by UiRS register)	No acknowledge detected (NACK)	UARTi transmission Rising edge of 9th bit of SCLi	UARTi transmission Next falling edge of 9th bit of SCLi	
Intrrupt numbers 18, 20, 34, 36, 38 generated by	UARTi reception receiving at 8th bit CKPOL=0(rising edge) CKPOL=1(falling edge)	Acknowledge detect (ACK) Rising edge of 9th bit of SCLi	UARTi reception Falling edge of 9th bit of SCLi		
Data transfer timing from UART receive shift register to UiRB register	CKPOL=0(rising edge) CKPOL=1(falling edge)	Rising edge of 9th bit of SCLi	Rising edge of 9th bit of SCLi	Falling edge and rising edge of 9th bit of SCLi	
UARTi transmit output delay	No delay	Selects delay time			
P63, P67, P70, P92, P96 pin functions	TxDi output	SDAi I/O			
P62, P66, P71, P91, P97 pin functions	RxDi input	SCLi I/O			
P61, P65, P72, P90, P95 pin functions	Select CLKi input or output	- (Not used in IIC mode)			
Noise filter width	15ns	50ns			
Read RxDi, SCLi pin level	When the direction register is set to "0"	Can be read regardless of the direction register			
TxDi, SDAi output initial value	CKPOL=0 (H) CKPOL=1 (L)	Values set in the port register before entering IIC mode			
SCLi initial value	-	H	L	H	L
DMA generated by (See Figure 1.20.2)	UARTi reception	Acknowledge detected (ACK)	UARTi reception Rising edge of 9 bit of SCLi		
Storing received data	Store first to eighth bit into bits 7 to 0 in the UiRB register	Store first to eighth bit into bits 7 to 0 in the UiRB register	Store first to seventh bits into bits 6 to 0 in UiRB register. Store eighth bit into bit 8 in UiRB register.		
			Store first to eighth bits into bits 7 to 0 in UiRB register <sup>3</sup>		
Reading received data	Read out UiRB register state	Read out bits 6 to 0 in UiRB register as bits 7 to 1. Read our bit 8 in UiRB register as bit 0. (Note 4)			

## Notes :

- The following procedures should be taken if changing how an interrupt is generated.
  - Disable an interrupt of a corresponding interrupt number.
  - Change how an interrupt is generated.
  - Set the IR bit of a corresponding interrupt number to "0" (interrupt disabled).
  - Set the IPL2 to IPL0 bits of a corresponding interrupt number.
- An default value of a SDAi output should be set when the SMD2 to SMD0 bits in the UiMR register are set to "0002" (serial I/O disabled).
- Second data transfer to the UiRB register (on the rising edge of the ninth bit of SCLi).
- First data transfer to the UiRB register (on the rising edge of the ninth bit of SCLi).

Serial I/O (Special Function)

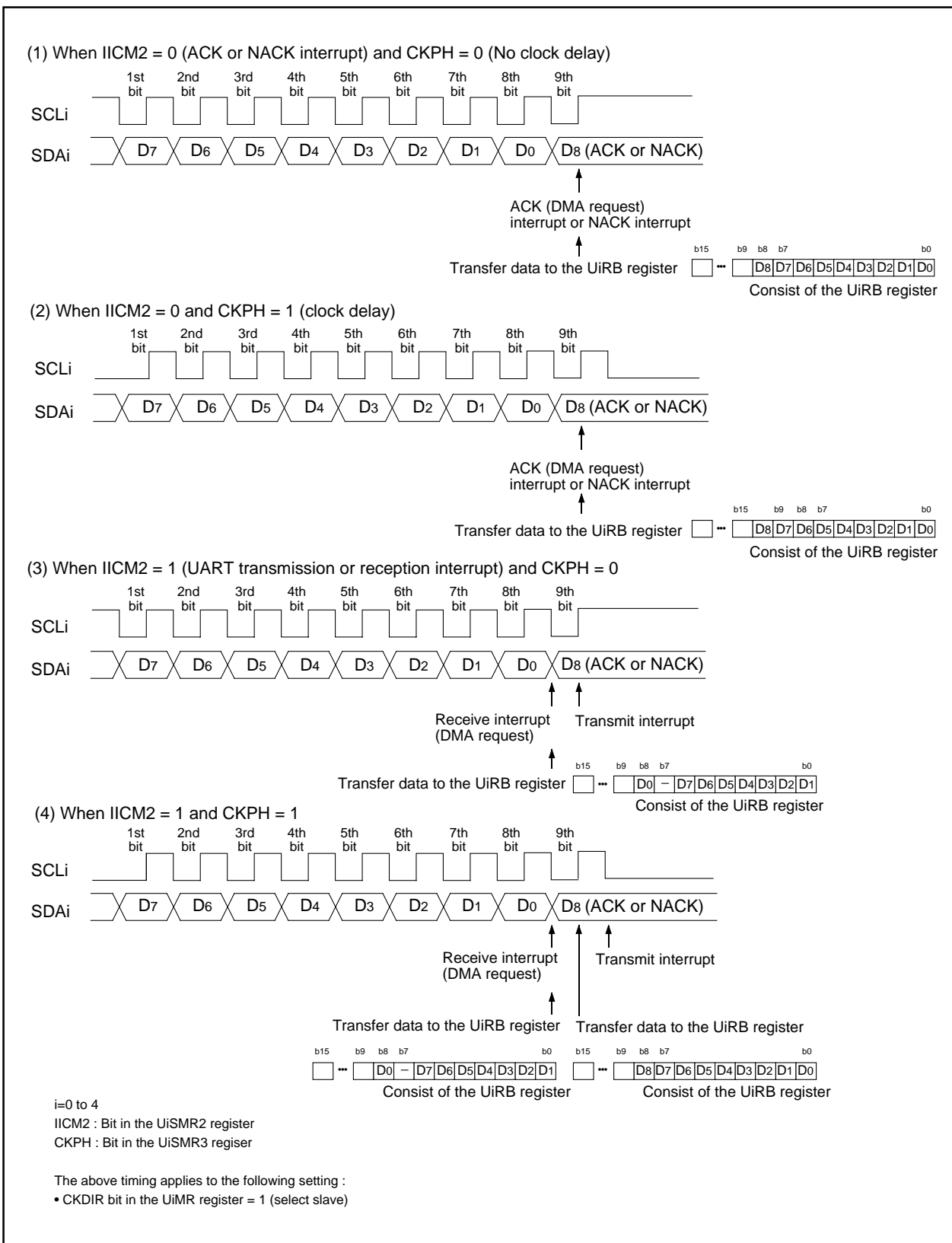


Figure 1.20.2. SCLi Timing

**Serial I/O (Special Function)**

**Table 1.20.4. Pin Settings in IIC Mode**

Port	Function	Bit and setting value		
		PS0 register	PSL0 register	PD6 register
P62	SCL0 output	PS0_2=1	PSL0_2=0	PD6_2=0
P63	SDA0 output	PS0_3=1	-	PD6_3=0
P66	SCL1 output	PS0_6=1	PSL0_6=0	PD6_6=0
P67	SDA1 output	PS0_7=1	-	PD6_7=0

**Table 1.20.5. Pin Settings (Continued)**

Port	Function	Bit and setting value			
		PS1 register	PSL1 register	PSC register	PD7 register
P70 <sup>1</sup>	SDA2 output	PS1_0=1	PSL1_0=0	PSC_0=0	PD7_0=0
P71 <sup>1</sup>	SCL2 output	PS1_1=1	PSL1_1=0	PSC_1=0	PD7_1=0

Notes :

- 1. N-channel open drain output.

**Table 1.20.6. Pin Settings (Continued)**

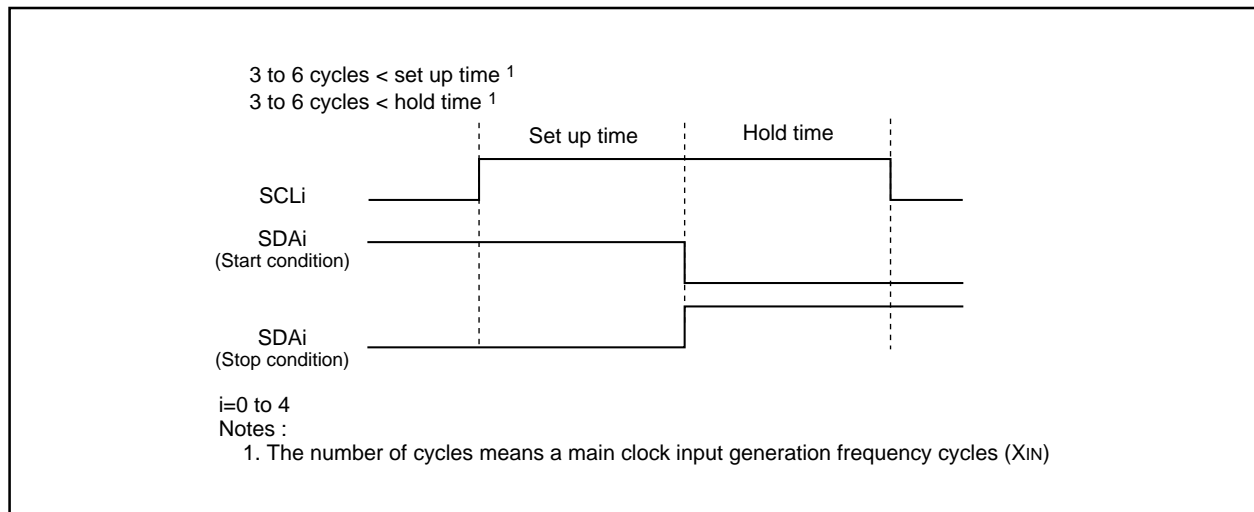
Port	Function	Bit and setting value		
		PS3 register <sup>1</sup>	PSL3 register	PD9 register <sup>1</sup>
P91	SCL3 output	PS3_1=1	PSL3_1=0	PD9_1=0
P92	SDA3 output	PS3_2=1	PSL3_2=0	PD9_2=0
P96	SCL4 output	PS3_6=1	-	PD9_6=0
P97	SDA4 output	PS3_7=1	PSL3_7=0	PD9_7=0

Notes :

- 1. The PD9 and PS3 registers should be set with subsequent to the instruction that the PRC2 bit in the PRCR register is set to "1" (write enable). Avoid generating an interrupt and operating a DMA transfer between the instruction to set the PRC2 bit to "1" and the instruction to write to the PD9 and PS3 registers.

**• Start Condition and Stop Condition**

The microcomputer detects either start condition or stop condition. The start condition detect interrupt is generated when the SCL<sub>i</sub> (i=0 to 4) pin is set to "H" and also the SDA<sub>i</sub> pin changes "H" to "L". The stop condition detect interrupt is generated when the SCL<sub>i</sub> pin is set to "H" and also the SDA<sub>i</sub> pin changes "L" to "H". The start condition detect interrupt and stop condition detect interrupt need the same interrupt control registers and vectors. The BBS bit in the UiSMR register determines which interrupt is requested.



**Figure 1.20.3. Start Condition or Stop Condition Detection**

**Serial I/O (Special Function)**

• **Start Condition or Stop Condition Output**

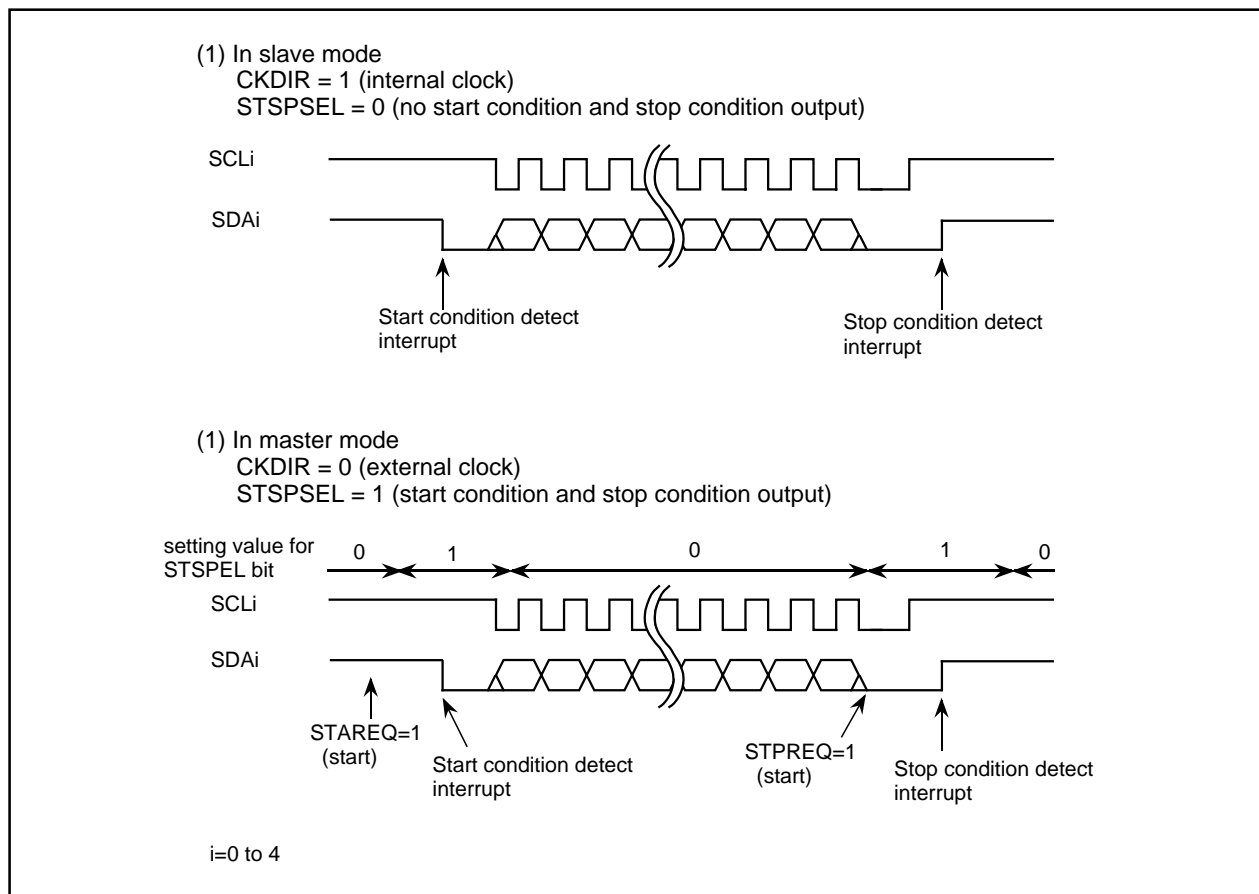
The start condition is generated when the STAREQ bit in the UiSMR4 register (i=0 to 4) is set to "1" (start). The restart condition is generated when the RSTAREQ bit in the UiSMR4 register is set to "1" (start). The stop condition is generated when the STPREQ bit in the UiSMR4 is set to "1" (start). The start condition is output when the STAREQ bit is set to "1" and the STSPSEL bit in the UiSMR4 register is set to "1" (start or stop condition output). The restart condition is output when setting the RSTAREQ bit and STSPSEL bit to "1". The stop condition is output when setting the STPREQ bit and the STSPSEL bit to "1".

When the start condition, stop condition or restart condition is output, avoid generating an interrupt between the instruction to set the STAREQ bit, STPREQ bit or RSTAREQ bit to "1" and the instruction to set the STSPSEL bit to "1". When the start condition is output, the STAREQ bit should be set to "1" before setting the STSPSEL bit to "1".

Table 1.20.7 lists function of the STSPSEL bit. Figure 1.20.4 shows function of the STSPSEL bit.

**Table 1.20.7 STSPSEL Bit Function**

Function	STSPSEL = 0	STSPSEL = 1
Start condition and stop condition output	Program with a port determines how the start condition or stop condition is output	The STAREQ bit, RSTAREQ bit and STPREQ bit determine how the start condition or stop condition is output
Timing to generate a start condition and stop condition interrupt request	The start condition and stop condition are detected	Start condition and stop condition generation are completed



**Figure 1.20.4. STSPSEL Bit Function**

## Serial I/O (Special Function)

---

### • Arbitration

The ABC bit in the UiSMR register (i=0 to 4) determines update timing for the ABT bit in the UiRB register. On the rising edge of SCLi, the microcomputer determines whether data being transmitted matches data input to the SDAi pin.

When the ABC bit is set to "0" (update per bit), the ABT bit is set to "1" as soon as detecting a data discrepancy. The ABT bit is set to "1" (detected-arbitration is lost) on the falling edge of the ninth bit of the transfer clock if once detecting the discrepancy. When the ABT bit is updated per byte, the ABT bit should be set to "0" (not detected-won) between an ACK detection in the first byte data and the next byte data to be transferred. When the ALS bit is set to "1" (SDA output stop enable), the arbitration lost occurs. The ABT bit is set to "1" as soon as the SDAi pin is placed in a high-impedance state.

### • Transfer Clock

The transfer clock transmits and receives data shown in Figure 1.20.4.

The CSC bit in the UiSMR2 register (i=0 to 4) synchronizes an internally generated clock (internal SCLi) with the external clock that is input into the SCLi pin. When setting the CSC bit to "1" (clock synchronous enabled) and internal SCLi to "H", internal SCL is set to "L" if the SCLi pin is on the falling edge. The UiBRG register is reloaded to start counting for the "L" leg. When setting the SCLi pin to "L", internal SCL is set to "H" from "L" to stop counting. Counting is resumed when setting the SCLi pin to "H". The transfer clock of UARTi is equivalent to the AND for an internal SCLi and signals from the SCLi pin.

The transfer clock is synchronized between a half cycle before the falling edge of first bit of an internal SCLi and the rising edge of the ninth bit. The internal clock should be selected as the transfer clock while the CSC bit is set to "1".

The SWC bit in the UiSMR2 register determines whether the SCLi pin that is set to "L" output is fixed on the falling edge of ninth cycle of the transfer clock or not.

When the SCLHI bit in the UiSMR4 register is set to "1" (enabled), SCLi output is halted when detecting a stop condition (high-impedance).

When the SWC2 bit in the UiSMR2 register is set to "1" (0 output), the SCLi pin can output "L" forcibly while transmitting and receiving. When setting the SWC2 bit to "0" (transfer clock), the SCLi pin stops "L" output to input and output the transfer clock.

When the CKPH bit in the UiSMR3 register is set to "1" and the SWC9 bit in the UiSMR4 register is set to "1" (SCL "L" hold enabled), the SCLi pin is fixed to output "L" on the next falling edge after ninth bit of the clock. When setting the SWC9 bit to "0" (SCL "L" hold disabled), the SCLi pin is not fixed to output "L".

### • SDA Output

Values in bits 7 to 0 (D7 to D0) in the UiTB register (i=0 to 4) are output in descending order from D7. The ninth bit (D8) is ACK or NACK.

A default value of SDAi output for transmission should be set when the IICM bit is set to "1" (IIC mode) the SMD2 to SMD0 bits in the UiMR register are set to "0002" (serial I/O disabled).

The DL2 to DL0 bits in the UiSMR3 register determine a SDAi output delay status or a count source of the UiBRG register with 2 to 8 cycle delay.

When the SDHI bit in the UiSMR2 register is set to "1" (SDA output disabled), the SDAi pin is placed in a high-impedance state forcibly. Avoid setting in the SDHI bit on the rising edge of the URTi transfer clock. The ABT bit in the UiRB register may be set to "1" (detected).

### • SDA Input

When the IICM2 bit in the UiSMR2 register (i=0 to 4) is set to "0", store first to eighth bits of received data are stored into bits 7 to 0 (D7 to D0) in the UiRB register. The ninth bit (D8) is ACK or NACK.

When setting the IICM2 bit to "1", first to seventh bits (D7 to D1) of received data are stored into bits 6 to 0 in the UiRB register. Eighth bit (D8) should be stored into bit 8 in the UiRB register.

If the IICM bit in the UiSMR register is set to "1" and the CKPH bit is set to "1", the same data can be read as data when setting bit to "0". To read the data, the UiRB register should be read after the rising edge of ninth bit of the transfer clock.

### • ACK, NACK

When the STSPSEL bit in the UiSMR4 register (i=0 to 4) is set to "0" (not output start or stop condition) and the ACKC bit in the UiSMR4 register is set to "1" (ACK data output), the value of the ACKD bit is output from the SDAi pin.

If setting the IICM2 bit to "0", NACK interrupt request is generated when the SDAi pin is set to "H" on the rising edge of ninth bit of the transfer clock. ACK interrupt request is generated when the SDAi pin is set to "L" on the rising edge of ninth bit of the transfer clock.

When selecting ACK how a DMAi request is generated, DMA transfer is activated by an ACK detection.

### • Transmit and Receive Reset

When the STC bit in the UiSMR2 register is set to "1" (UARTi initialization enabled) and a start condition is detected.

- the transmit shift register is reset and content of the UiTB register is transferred to the transmit shift register. The next clock becomes a first bit to start transmitting. UARTi output value remains unchanged between a clock provided and data of first bit output. The value is the same as a value when a start condition is detected.
- the receive shift register is reset and the next clock provided becomes first bit to start transmitting.
- the SWC bit is set to "1" (SCL wait output enabled). The SCLi pin is set to "L" on the falling edge of ninth bit of the transfer clock.

When UARTi transmission and reception are started with this function, the TI bit remains unchanged. The external clock should be selected as the transfer clock when using this function.

## Serial I/O (Special Function)

## 2. Special Mode 2

Special mode 2 is a mode that serial communication between one or multiple master(s) and multiple slaves is available. The  $\overline{SSi}$  input pin ( $i=0$  to 4) controls serial bus communication. Table 1.20.8 lists specifications of special mode 2. Table 1.20.9 lists registers to be used register and settings. Tables 1.20.10 to 1.20.12 list pin settings.

**Table1.20.8. Special Mode 2 Specifications**

Item	Specification
Transfer data format	Transfer data : 8 bits long
Transfer clock	When setting the CKDIR bit in the UiMR register ( $i=0$ to 4) to "0" (internal clock selected) : $f_j/2(m+1)$ $f_j = f_1, f_8, f_{2n^1}$ $m$ : setting value of the UiBRG register 0016 to FF16 When setting the CKDIR bit to "1" (external clock selected) : input clock from the CLKi pin
Transmit/receive control	Selectable from $\overline{CTS}$ function, $\overline{RTS}$ function or $\overline{CTS/RTS}$ function disabled
Transmit start condition	<ul style="list-style-type: none"> <li>To start transmitting, the following requirement should be met<sup>2</sup> :               <ul style="list-style-type: none"> <li>The TE bit in the UiC1 register is set to "1" (transmit enable)</li> <li>The TI bit in the UiC1 register to "0" (data available in the UiTB register)</li> <li><math>\overline{CTSi}</math> input level is "L" when <math>\overline{CTS}</math> function is selected</li> </ul> </li> </ul>
Receive start condition	<ul style="list-style-type: none"> <li>To start receiving, the following requirement should be met<sup>2</sup> :               <ul style="list-style-type: none"> <li>The RE bit in the UiC1 register is set to "1" (receive enable)</li> <li>The TE bit is set to "1" (receive enable)</li> <li>TI bit is set to "0" (data available in the UiTB register)</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>While transmitting, the following condition can be selected:               <ul style="list-style-type: none"> <li>The UiIRS bit in the UiC1 register is set to "0" (transmit buffer empty) : when data is transferred from the UiTB register to the UARTi transmit register (transmit starting)</li> <li>The UiRS register is set to "1" (transmit completed): when data transmission from UARTi transfer register is completed</li> </ul> </li> <li>While receiving When data is transferred from UARTi receive register to the UiRB register (receive completed)</li> </ul>
Error detection	<ul style="list-style-type: none"> <li>Overrun error<sup>3</sup> This error occurs when reading a seventh bit of the next received data before reading the UiRB register</li> </ul>
Selectable function	<ul style="list-style-type: none"> <li>CLK polarity Either rising edge or falling edge of a the transfer clock can be selected when a transferred data is output and input</li> <li>LSB first or MSB first Whether a data is transmitted/received in bit0 or in bit7 can be selected</li> <li>Continuous receive mode Reception is enabled simultaneously by reading the UiRB register</li> <li>Serial data logic inverse This function inverses transmitted/received data logically</li> <li>TxD, RxD I/O polarity switching TxD pin output and RxD pin input are inversed. All I/O data levels are also inversed</li> <li>Clock phase One of 4 combinations of transfer data polarity and phases can be read</li> <li><math>\overline{SSi}</math> input pin function Output pin is placed in a high-impedance state to avoid data conflict between a master and another masters or slave</li> </ul>

Notes :

- The CNT3 to CNT0 bits in the TRGSR register determine either "no division ( $n=0$ )" or "divide-by- $2n$  ( $n=1$  to 15)".
- When selecting the external clock, the external clock is in "H" when the CKPOL bit in the UiC0 register is set to "0" and is in "L" when the CKPOL bit is set to "1".
- If an overrun error occurs, the UiRB register is indeterminate. The IR bit of SiRIC register does not change to "1" (interrupt request).

**Serial I/O (Special Function)**

**Table 1.20.9. Registers to Be Used and Settings in Special Mode 2**

Register	Bit	Function
UiTB	0 to 7	Set data to be transmitted
UiRB	0 to 7	Received data can be read
	OER	Overrun error flag
UiBRG	0 to 7	Set baud rate
UiMR	SMD2 to SMD0	Set to "0012"
	CKDIR	Set to "0" in master mode, "1" in slave mode
	IOPOL	Set to "0"
UiC0	CLK0, CLK1	Select a count source of the UiBRG register
	CRS	Disabled since CRD = 1
	TXEPT	Transfer register empty flag
	CRD	Set to "1"
	NCH	Select output format of the TxDi pin
	CKPOL	Clock phase can be set by a combination of the CKPOL bit and the CKPH bit in the UiSMR3 register
	UFORM	Set to "0"
UiC1	TE	When data transfer and reception is enabled, set to "0"
	TI	Transfer buffer empty flag
	RE	When data reception is enabled, set to "1"
	RI	Receive complete flag
	UiIRS	Select how the UARTi transmit interrupt is generated
	UiLCH, SCLKSTPB	Set to "0"
	UiRRM	When continuous receive mode is enabled, set to "1"
UiSMR	0 to 7	Set to "0"
UiSMR2	0 to 7	Set to "0"
UiSMR3	SSE	Set to "1"
	CKPH	Clock phase can be set by a combination of the CKPH bit and the CKPOL bit in the UiC0 register
	DINC	Set to "0" in master mode, "1" in slave mode
	NODC	Set to "0"
	ERR	Fault error flag
	5 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"
IFSR	IFSR6, IFSR7	Select how a fault error occurs

i=0 to 4

## Serial I/O (Special Function)

Table 1.20.10. Pin settings in Special Mode 2

Port	Function	Bit and setting value		
		PS0 register	PSL0 register	PD6 register
P60	SS0 input	PS0_0=0	–	PD6_0=0
P61	CLK0 input (slave)	PS0_1=0	–	PD6_1=0
	CLK0 output (master)	PS0_1=1	–	–
P62	RxD0 input (master)	PS0_2=0	–	PD6_2=0
	STxD0 input (slave)	PS0_2=1	PSL0_2=1	–
P63	TxD0 output (master)	PS0_3=1	–	–
	SRxD0 output (slave)	PS0_3=0	–	PD6_3=0
P64	SS1 input	PS0_4=0	–	PD6_4=0
P65	CLK1 output (slave)	PS0_5=0	–	PD6_5=0
	CLK1 output (master)	PS0_5=1	–	–
P66	RxD1 input (master)	PS0_6=0	–	PD6_6=0
	STxD1 input (slave)	PS0_6=1	PSL0_6=1	–
P67	TxD1 output (master)	PS0_7=1	–	–
	SRxD1 input (slave)	PS0_7=0	–	PD6_7=0

Table 1.20.11. Pin settings (Continued)

Port	Function	Bit and setting value			
		PS1 register	PSL1 register	PSC register	PD7 register
P70 <sup>1</sup>	TxD2 output (master)	PS1_0=1	PSL1_0=0	PSC_0=0	–
	SRxD2 input (slave)	PS1_0=0	–	–	PD7_0=0
P71 <sup>1</sup>	RxD2 input (master)	PS1_1=0	–	–	PD7_1=0
	STxD2 output (slave)	PS1_1=1	PSL1_1=1	–	–
P72	CLK2 input (slave)	PS1_2=0	–	–	PD7_2=0
	CLK2 output (master)	PS1_2=1	PSL1_2=0	PSC_2=0	–
P73	SS2 input	PS1_3=0	–	–	PD7_3=0

Notes :

1. N-channel open drain output

Table 1.20.12. Pin settings (Continued)

Port	Function	Bit and setting value		
		PS3 register <sup>1</sup>	PSL3 register	PD9 register <sup>1</sup>
P90	CLK3 input (slave)	PS3_0=0	–	PD9_0=0
	CLK3 output (master)	PS3_0=1	–	–
P91	RxD3 input (master)	PS3_1=0	–	PD9_1=0
	STxD3 output (slave)	PS3_1=1	PSL3_1=1	–
P92	TxD3 output (master)	PS3_2=1	PSL3_2=0	–
	SRxD3 input (slave)	PS3_2=0	–	PD9_2=0
P93	SS3 input	PS3_3=0	PSL3_3=0	PD9_3=0
P94	SS4 input	PS3_4=0	PSL3_4=0	PD9_4=0
P95	CLK4 input (slave)	PS3_5=0	PSL3_5=0	PD9_5=0
	CLK4 output (master)	PS3_5=1	–	–
P96	TxD4 output (master)	PS3_6=1	–	–
	SRxD4 input (slave)	PS3_6=0	PSL3_6=0	PD9_6=0
P97	RxD4 input (master)	PS3_7=0	–	PD9_7=0
	STxD4 input (slave)	PS3_7=1	PSL3_7=1	–

Notes :

1. The PD9 and PS3 registers should be set with subsequent to the instruction that the PRC2 bit in the PRCR register is set to "1" (write enable). Avoid generating an interrupt and operating a DMA transfer between the instruction to set the PRC2 bit to "1" and the instruction to write to the PD9 and PS3 registers.

• **SSi Input Pin Function (i=0 to 4)**

When the SSE bit in the UiSMR3 register to "1" ( $\overline{SS}$  function enabled), the SSi input pin function is selected to activate pin functions.

The DINC bit in the UiSMR3 register determines which microcomputers perform as master(s) or slave(s). When multiple microcomputers as a master (multi-master system), the  $\overline{SSi}$  pin state determines at each activation of this mode.

**(1) When setting the DINC Bit to "1" (Slave Mode)**

When an "H" signal is input to the  $\overline{SSi}$  pin, the STxDi and SRxDi pins are placed in a high-impedance state and an input to the transfer clock is ignored. When an "L" signal is input to the  $\overline{SSi}$  input pin, the transfer clock is enabled to input and serial communications are available.

**(2) When setting the DINC Bit to "0" (Master Mode)**

When an "H" signal is input to the  $\overline{SSi}$  pin, serial communication is available due to a transmission. The master outputs the transfer clock. When a "L" signal is input to  $\overline{SSi}$  pin, it indicates that another master already works and all TxDi, RxDi and CLKi pins are placed in a high-impedance state. Moreover, a fault error occurs and the IR bit in the BCNiC register is set to "1" (interrupt request). The ERR bit in the UiSMR3 register identifies whether a fault error occurs.

In master mode, software interrupt numbers 39, 40 and 41 become fault error interrupt. The fault error interrupt is generated when the ERR bit changes "0" to "1". The fault error interrupt of UART0 shares an interrupt vector with the one of UART3. The fault error interrupt of UART1 shares an interrupt vector with the one of UART4. The IFSR6 and IFSR7 bits in the IFSR register determine which fault error interrupt is used.

Communications is not terminated even when a fault error is generated during communications. To stop communications, the SMD 2 to SMD0 bit in the UiMR register is set to "0002" (serial I/O disabled).

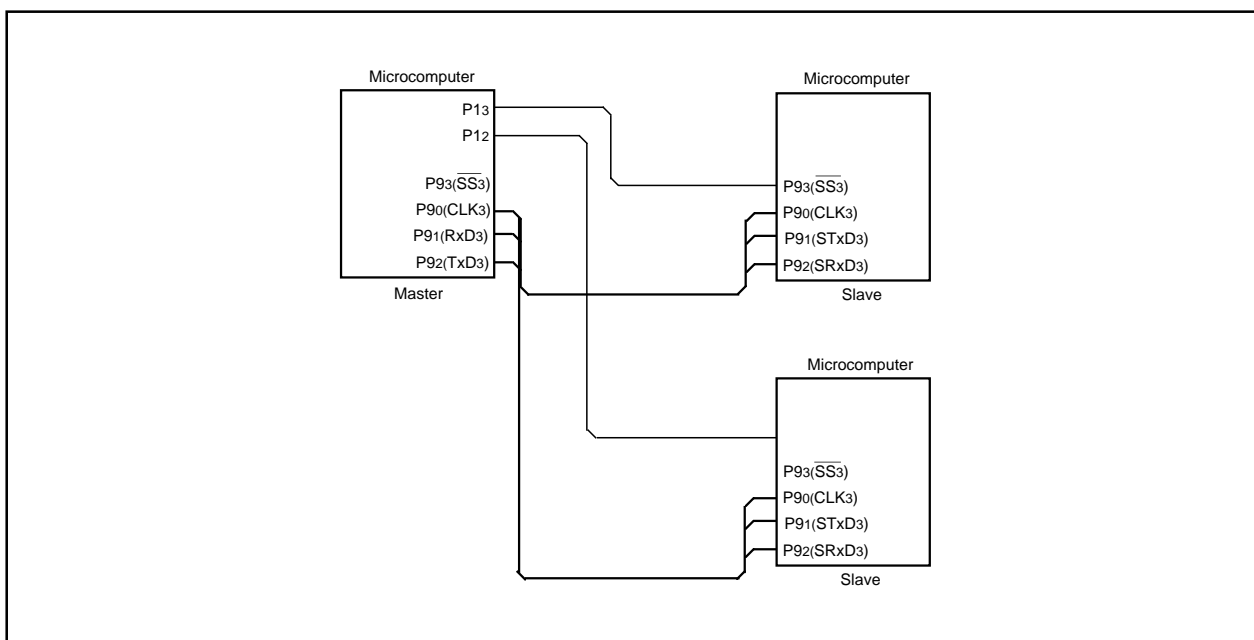


Figure 1.20.5. Serial Bus Communication Control with  $\overline{SS}$  Pin

## Serial I/O (Special Function)

### • Clock Phase Function

The CKPH bit in the UiSMR3 register (i=0 to 4) and the CKPOL bit in the UiC0 register determine of four combinations of transfer clock polarity and phases.

The transfer clock phase and polarity should be the same between a master and a slave involved in the transfer.

#### (1) When setting the DINC Bit to "0" (Master (Internal Clock))

Figure 1.20.6 shows a transmit and receive timing.

#### (2) When setting the DINC Bit to "1" (Slave (External Clock))

When setting the CKPH bit to "0" (no clock delay) and the  $\overline{SS}_i$  input pin to "H", output data is placed in a high-impedance state. When  $\overline{SS}_i$  input pin is set to "L", an serial transmit start conditions are met though an output is indeterminate. Serial transmission is synchronized with the transfer clock. Figure 1.20.7 shows the transmit and receive timing.

When setting the CKPH bit to "1" (clock delay) and the  $\overline{SS}_i$  input pin to "H", output data is placed in a high-impedance state. When the  $\overline{SS}_i$  pin is set to "L", the first data is output. Serial transmission is synchronized with the transfer clock. Figure 1.20.8 shows the transmit and receive timing.

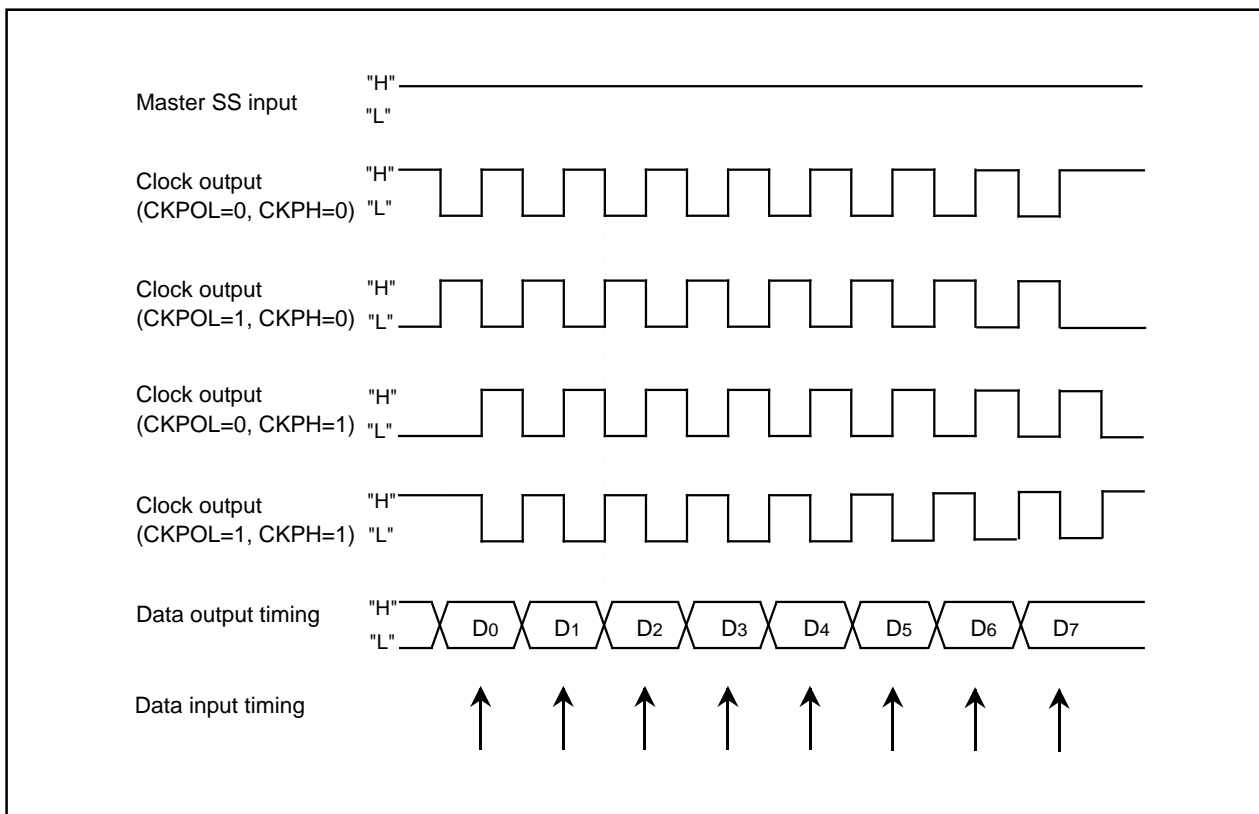


Figure 1.20.6. Transmit and Receive Timing in Master Mode (Internal Clock)

Serial I/O (Special Function)

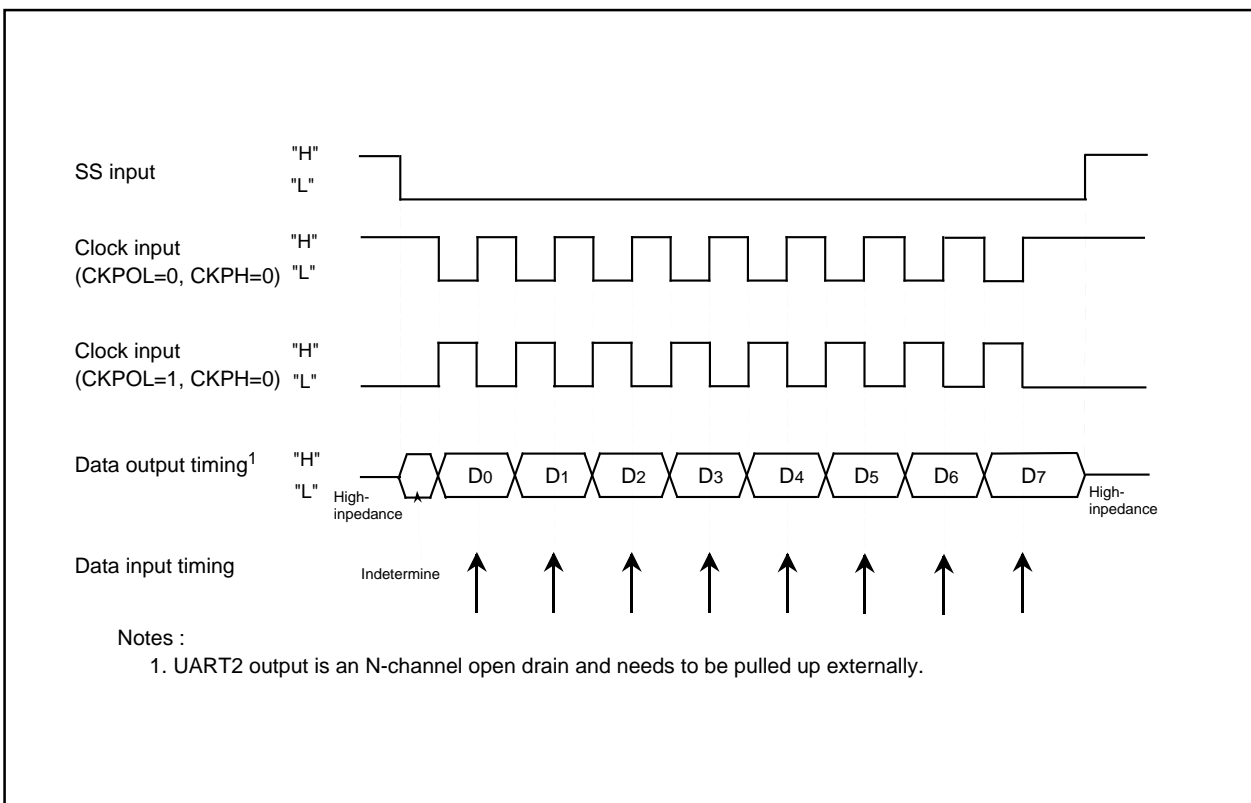


Figure 1.20.7. Transmit and Receive Timing in Slave Mode (External Clock) (CKPH=0)

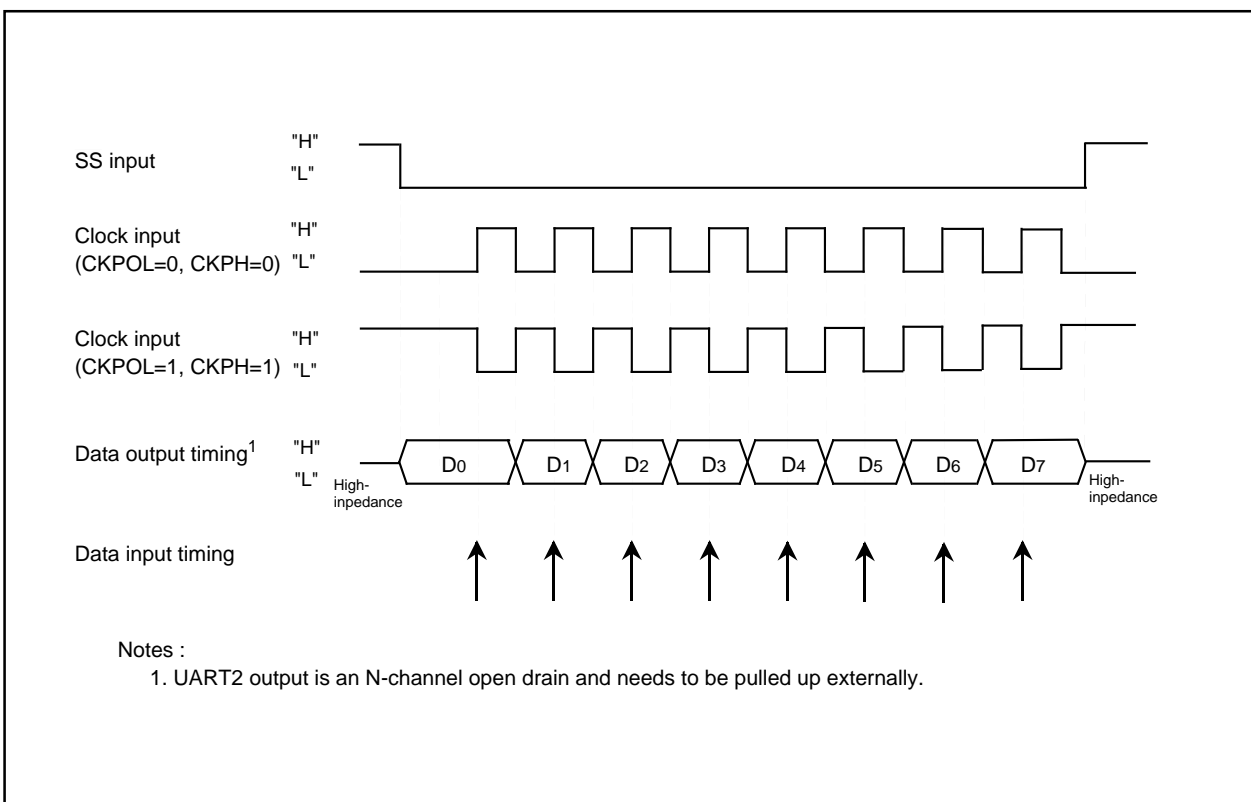


Figure 1.21.9. Transmit and Receive Timing in Slave Mode (External Clock) (CKPH=1)

## Serial I/O (Special Function)

### 3. Special Mode 3 (GCI Mode)

GCI mode is a mode to synchronize the external clock with the transfer clock used in the clock synchronous serial I/O mode.

Table 1.20.13 lists specifications of GCI mode. Table 1.20.14 lists registers to be used and register settings. Tables 1.20.15 to 1.20.17 list pin settings.

**Table 1.20.13. GCI Mode Specifications**

Item	Specification
Transfer data format	Transfer data : 8 bits long
Transfer clock	When the CKDIR bit in the UiMR register (i=0 to 4) is set to "1" (external clock selected) : input from the CLKi pin
Clock synchronization function	Input trigger from the $\overline{\text{CTS}}_i$ pin
Transmit/receive start condition	<ul style="list-style-type: none"> <li>• When the external clock is set to "H", the following conditions should be met : <ul style="list-style-type: none"> <li>- The TE bit in the UiC1 register is set to "1" (transmit enable)</li> <li>- The RE bit in the UiC1 register to "1" (receive enable)</li> <li>- The TI bit in the UiC1 register to "0" (Data in UiTB register)</li> <li>- <math>\overline{\text{CTS}}_i</math> input level is "L"</li> </ul> </li> </ul>
Interrupt request generation timing	<ul style="list-style-type: none"> <li>• While transmitting, the following condition can be selected: <ul style="list-style-type: none"> <li>- The UiIRS bit in the UiC1 register is set to "0" (Transmit buffer empty) : when data is transferred from the UiTB register to the UAR<i>T</i><sub>i</sub> transmit register (starting transmit)</li> <li>- The UiIRS bit is set to "1" (Transmit completed): when data transmission from the UAR<i>T</i><sub>i</sub> transfer register is completed</li> </ul> </li> <li>• While receiving <ul style="list-style-type: none"> <li>- When data is transferred from the UAR<i>T</i><sub>i</sub> receive register to the UiRB register (receive completed)</li> </ul> </li> </ul>
Error detection	<p>Overflow error<sup>1</sup></p> <p>This error occurs when reading a seventh bit of the next received data before reading the UiRB register</p>

Notes :

1. If an overrun error occurs, the UiRB register is indeterminate. The IR bit in the SiRIC register does not change to "1" (interrupt request).

**Serial I/O (Special Function)**

**Table 1.20.14. Registers to be Used and Settings in CGI Mode**

Register	Bit	Function
UiTB	0 to 7	Set data to be transmitted
UiRB	0 to 7	Received data
	OER	Overrun error flag
UiBRG	0 to 7	Set to "0016"
UiMR	SMD2 to SMD0	Set to "0012"
	CKDIR	Set to "1"
	IOPOL	Set to "0"
UiC0	CLK1 to CLK0	Set to "002"
	CRS	Disabled since CRD = 1
	TXEPT	Transfer register empty flag
	CRD	Set to "1"
	NCH	Select an output format of the TxDi pin
	CKPOL	Set to "0"
	UFORM	Set to "0"
UiC1	TE	When data transfer is enabled, set to "1"
	TI	Transfer buffer empty flag
	RE	When data reception is enabled, set to "1"
	RI	Receive complete flag
	UiIRS	Select how the UARTi transmit interrupt is generated
	UiRRM, UiLCH,	Set to "0"
	SCLKSTPB	Set to "0"
UiSMR	0 to 6	Set to "0"
	SCLKDIV	See Table 1.20.18.
UiSMR2	0 to 6	Set to "0"
	SU1HIM	See Table 1.20.18.
UiSMR3	0 to 2	Set to "0"
	NODC	Fix to "0"
	4 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"

i=0 to 4

## Serial I/O (Special Function)

**Table 1.20.15. Pin settings in CGI Mode**

Port	Function	Bit and setting value		
		PS0 register	PSL0 register	PD6 register
P60	CTS0 input <sup>1</sup>	PS0_0=0	–	PD6_0=0
P61	CLK0 input	PS0_1=0	–	PD6_1=0
P62	RxD0 input	PS0_2=0	–	PD6_2=0
P63	TxD0 output	PS0_3=1	–	–
P64	CTS1 input	PS0_4=0	–	PD6_4=0
P65	CLK1 input	PS0_5=0	–	PD6_5=0
P66	RxD1 input	PS0_6=0	–	PD6_6=0
P67	TxD1 output	PS0_7=1	–	–

Notes :

1. CTS input is used for trigger input.

**Table 1.20.16. Pin settings (Continued)**

Port	Function	Bit and setting value			
		PS1 register	PSL1 register	PSC register	PD7 register
P70 <sup>1</sup>	TxD2 output	PS1_0=1	PSL1_0=0	PSC_0=0	–
P71 <sup>1</sup>	RxD2 input	PS1_1=0	–	–	PD7_1=0
P72	CLK2 input	PS1_2=0	–	–	PD7_2=0
P73	RTS2 input <sup>2</sup>	PS1_3=0	–	–	PD7_3=0

Notes :

1. N-channel open drain output
2. CTS input is used for trigger input.

**Table 1.20.17. Pin settings (Continued)**

Port	Function	Bit and setting value		
		PS3 register <sup>1</sup>	PSL3 register	PD9 register <sup>1</sup>
P90	CLK3 input	PS3_0=0	–	PD9_0=0
P91	RxD3 input	PS3_1=0	–	PD9_1=0
P92	TxD3 output	PS3_2=1	PSL3_2=0	–
P93	CTS3 input <sup>2</sup>	PS3_3=0	PSL3_3=0	PD9_3=0
P94	CTS4 input <sup>2</sup>	PS3_4=0	PSL3_4=0	PD9_4=0
P95	CLK4 input	PS3_5=0	PSL3_5=0	PD9_5=0
P96	TxD4 output	PS3_6=1	–	–
P97	RxD4 input	PS3_7=0	–	PD9_7=0

Notes :

1. The PD9 and PS3 registers should be set with subsequent to the instruction that the PRC2 bit in the PRCR register is set to "1" (write enable). Avoid generating an interrupt and operating a DMA transfer between the instruction to set the PRC2 bit to "1" and the instruction to write to the PD9 and PS3 registers.
2. CTS input is used for a trigger input.

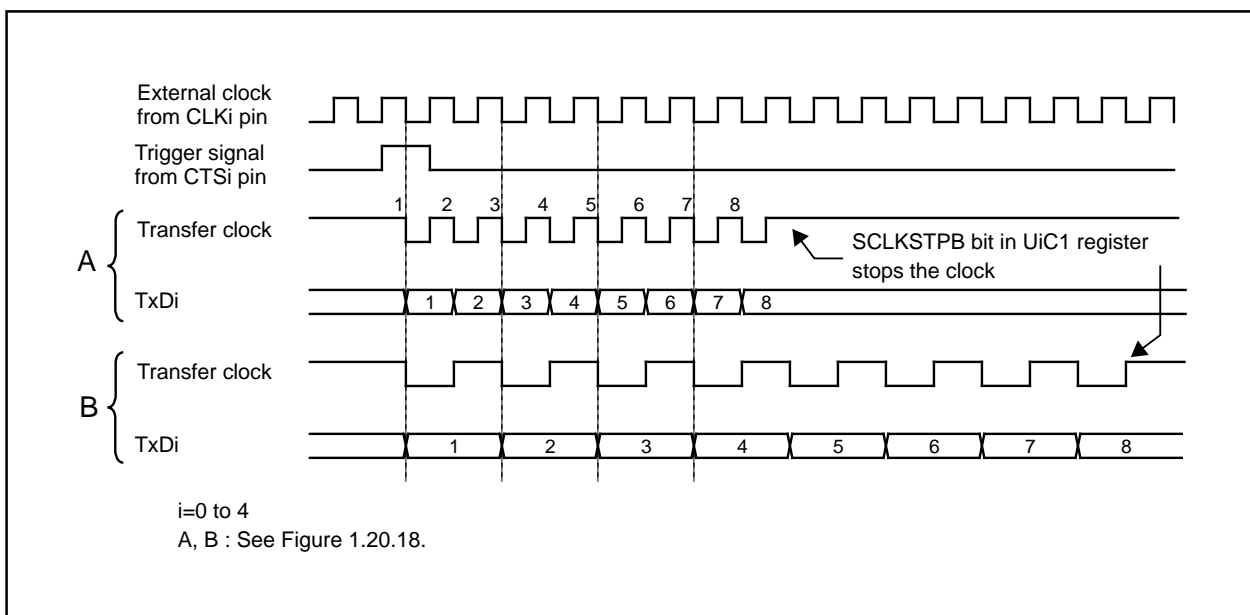
**Serial I/O (Special Function)**

To generate the internal clock synchronized with the external clock, the SU1HIM bit in the UiSMR2 register (i=0 to 4) and the SCLKDIV bit in the UiSMR register should be set to values shown in Table 1.20.18 first. A trigger signal should be input to the CTSi pin. Either the same clock cycle as the external clock or external clock divided by two can be selected as the transfer clock. To start or stop the transfer clock during an external clock operation, the SCLKSTPB bit in the UiC1 register controls the transfer clock. Figure 1.20.9 shows an example of the clock-divided synchronous function.

**Table 1.20.18. Clock-Divided Synchronous Function Select**

SCLKDIV bit in UiSMR register	SU1HIM bit in UiSMR register	Clock-divided synchronous function	Example of waveform
0	0	No synchronized	-
0	0	The same division as the external clock	A in Figure 1.20.9
0	0 or 1	The same division as the external clock divided by 2	B in Figure 1.20.9

i=0 to 4



**Figure 1.20.9. Clock-Divided Synchronous Function**

**4. Special Mode 4 (IE Mode)**

IE mode is a mode to communicate in UART mode among devices connecting with the IEBus. Table 1.20.20 lists registers to be used and register settings. Tables 1.20.21 to 1.20.23 list pin settings.

## Serial I/O (Special Function)

Table 1.20.20. Registers to Be Used and settings in IE Mode

Register	Bit	Function
UiTB	0 to 8	Set data to be transmitted
UiRB	0 to 8	Received data can be read
	OER, FER, PER, SUM	Error flags
UiBRG	0 to 7	Set baud rate
UiMR	SMD2 to SMD0	Set to "1102"
	CKDIR	Select the internal clock or external clock
	STPS	Set to "0"
	PRY	Disabled since PRYE=0
	PRYE	Set to "0"
	IOPOL	Select TxD and RxD I/O polarity
UiC0	CLK1 to CLK0	Select a count source of the UiBRG register
	CRS	Disabled since CRD=1
	TXEPT	Transfer register empty flag
	CRD	Set to "1"
	NCH	Select output format of the TxDi pin
	CKPOL	Set to "0"
	UFORM	Set to "0"
UiC1	TE	When data transfer is enabled, set to "1"
	TI	Transfer buffer empty flag
	RE	When data reception is enabled, set to "1"
	RI	Receive complete flag
	UiIRS	Select how the UARTi transmit interrupt is generated
	UiRRM, UiLCH, SCLKSTPB	Set to "0"
UiSMR	0 to 3	Set to "0"
	ABSCS	Select bus conflict detect sampling timing
	ACSE	When the transmit enable bit is automatically clear, set to "1"
	SSS	Select the transmit start condition
	SCLKDIV	Set to "0"
UiSMR2	0 to 7	Set to "0"
UiSMR3	0 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"
IFSR	IFSR6, IFSR7	Select how the bus conflict interrupt occurs

i=0 to 4

**Serial I/O (Special Function)**

**Table 1.20.21. Pin Settings in IE Mode**

Port	Function	Bit and setting value		
		PS0 register	PSL0 register	PD6 register
P61	CLK0 input	PS0_1=0	–	PD6_1=0
	CLK0 output	PS0_1=1	–	–
P62	RxD0 input	PS0_2=0	–	PD6_2=0
P63	TxD0 output	PS0_3=1	–	–
P65	CLK1 input	PS0_5=0	–	PD6_5=0
	CLK1 output	PS0_5=1	–	–
P66	RxD1 input	PS0_6=0	–	PD6_6=0
P67	TxD1 output	PS0_7=1	–	–

**Table 1.20.22. Pin Settings (Continued)**

Port	Function	Bit and setting value			
		PS1 register	PSL1 register	PSC register	PD7 register
P70 <sup>1</sup>	TxD2 output	PS1_0=1	PSL1_0=0	PSC_0=0	–
P71 <sup>1</sup>	RxD2 input	PS1_1=0	–	–	PD7_1=0
P72	CLK2 input	PS1_2=0	–	–	PD7_2=0
	CLK2 output	PS1_2=1	PSL1_2=0	PSC_2=0	–

Notes :

1. N-channel open drain output

**Table 1.20.23. Pin Settings (Continued)**

Port	Function	Bit and setting value		
		PS3 register <sup>1</sup>	PSL3 register	PD9 register <sup>1</sup>
P90	CLK3 input	PS3_0=0	–	PD9_0=0
	CLK3 output	PS3_0=1	–	–
P91	RxD3 input	PS3_1=0	–	PD9_1=0
P92	TxD3 output	PS3_2=1	PSL3_2=0	–
P95	CLK4 input	PS3_5=0	PSL3_5=0	PD9_5=0
	CLK4 output	PS3_5=1	–	–
P96	TxD4 output	PS3_6=1	–	–
P97	RxD4 input	PS3_7=0	–	PD9_7=0

Notes :

1. The PD9 and PS3 registers should be set with subsequent to the instruction that the PRC2 bit in the PRCR register is set to "1" (write enable). Avoid generating an interrupt and operating a DMA transfer between the instruction to set the PRC2 bit to "1" and the instruction to write to the PD9 and PS3 registers.

## Serial I/O (Special Function)

---

When an output level of the TxDi pin (i=0 to 4) differs from an input level of the RxDi pin, an interrupt request is generated.

UART0 and UART3 are assigned in software interrupt number 40, UART1 and UART4 are assigned in the number 41. When using the bus conflict detect function of UART0 or UART3, of UART1 or UART4, the IFSR6 bit and the IFSR7 bit in the IFSR register select software interrupt number 40 or 41.

When the ABSCS bit in the UiSMR register is set to "0" (rising edge of the transfer clock), on the rising edge of the transfer clock, an output level of the TxD pin is determined whether it matches input level of the RxD pin. When setting the ABSCS bit to "1" (timer Aj underflow), it is determined when an overflow of the timer Aj (timer A3 in UART0, timer A4 in UART1, timer A0 in UART2, timer A3 in UART3, the timer A4 in UART4) occurs. The timer Aj should be used in one-shot timer mode.

When the ACSE bit in the UiSMR register is set to "1" (automatic clear at bus conflict) and the IR bit in the BCNiIC register to "1" (discrepancy detected), the TE bit is set to "0" (transmit disable).

When the SSS bit in the UiSMR register is set to "1" (falling edge of RxDi), transmission from the TxDi pin is started on the rising edge of the RxDi pin. Figure 1.20.10 shows bits associated with the bus conflict detect function.

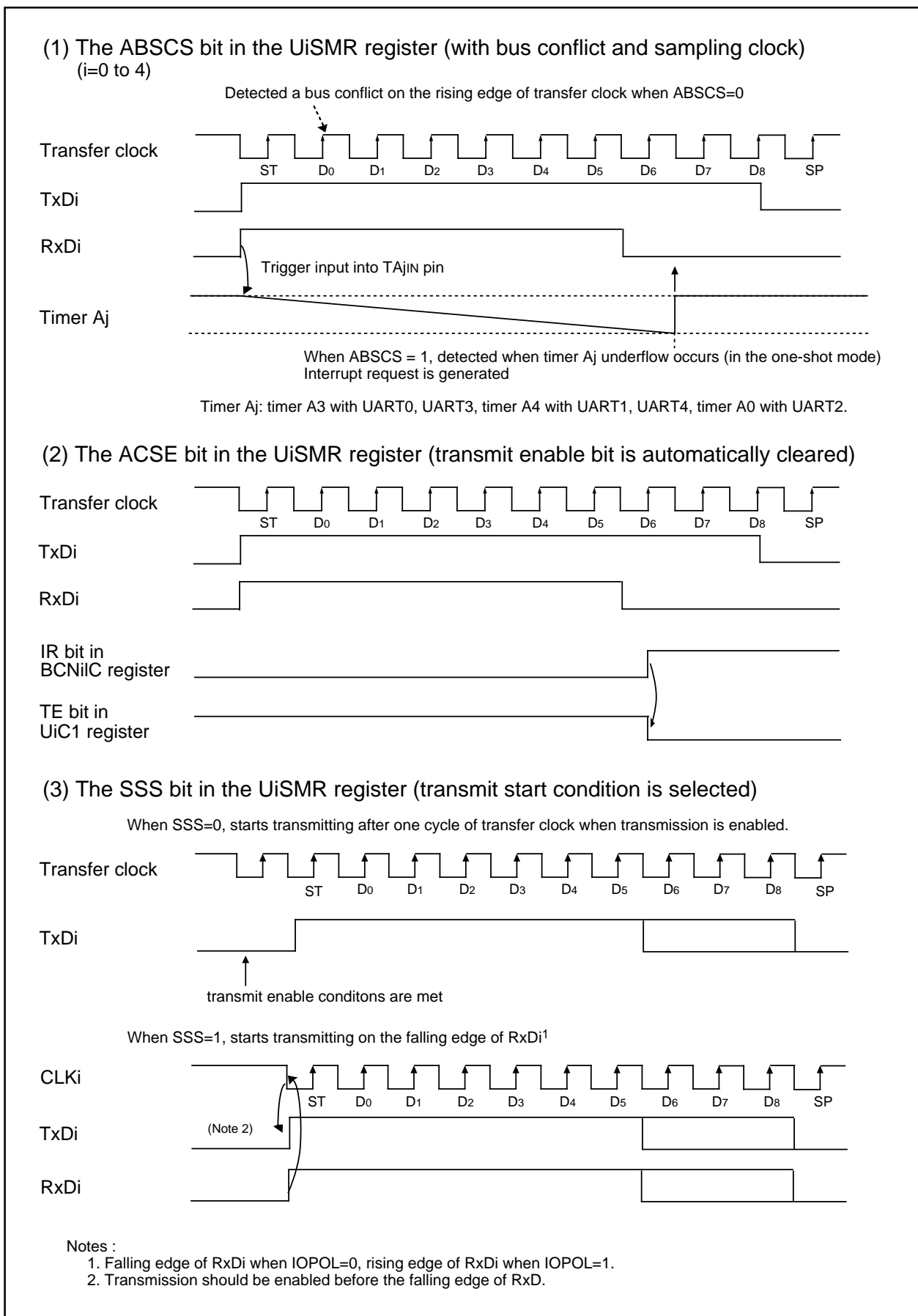


Figure 1.20.10. Bit Function Related Bus Conflict Detection

## Serial I/O (Special Function)

## 5. Special Mode 5 (SIM Mode)

SIM mode is a mode to communicate in UART mode with SIM interface devices. Direct format and inverse format can be achieved and The TxDi pin (i=0 to 4) output "L" when a parity error is detected.

Table 1.20.24 lists specifications of SIM mode. Table 1.20.25 lists registers to be used and register settings. Tables 1.20.26 to 1.20.28 list pin settings.

**Table 1.20.24. SIM Mode Specifications**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>• Transfer data: 8-bit UART mode</li> <li>• In direct format</li> <li>Parity Even</li> <li>Data logic Direct</li> <li>Transfer format LSB first</li> </ul> <ul style="list-style-type: none"> <li>• One stop bit</li> <li>• In inverse format</li> <li>Parity Odd</li> <li>Data logic Inverse</li> <li>Transfer format MSB first</li> </ul>
Transfer clock	When the CKDIR bit in the UiMR register (i=0 to 4) to "0" (internal clock selected) : $f_j/16(m+1)^1$ $f_j = f_1, f_8, f_{2n^2}$ m : setting value of the UiBRG register 00 <sub>16</sub> to FF <sub>16</sub> Avoid setting the CKDIR bit to "1" (external clock selected)
Transmit / receive control	The CRD bit in the UiC0 register is set to "1" (CTS, RTS function disable)
Other setting item	The UiIRS bit in the UiC1 register is set to "1" (transmit interrupt caused by transmit completed)
Transmit start condition	To start transmitting, the following requirements should be met <ul style="list-style-type: none"> <li>• The TE bit in the UiC1 register is set to "1" (transmit enable)</li> <li>• The TI bit in the UiC1 register is set to "0" (data available in the UiTB register)</li> </ul>
Receive start condition	To start receiving, the following requirements should be met <ul style="list-style-type: none"> <li>• The RE bit in the UiC1 register is set to "1" (receive enable)</li> <li>• The start bit is detected</li> </ul>
Interrupt request generation timing	While transmitting: <ul style="list-style-type: none"> <li>• The UiRS bit is set to "1" (transmit is completed): when data transmission from the UARTi transfer register is completed</li> </ul> While receiving <ul style="list-style-type: none"> <li>• When data is transferred from the UARTi receive register to the UiRB register (receive completed)</li> </ul>
Error detection	<ul style="list-style-type: none"> <li>• Overrun error<sup>1</sup> This error occurs when an eighth bit of the next data before reading the UiRB bit is received</li> <li>• Flaming error This error occurs when the number of the stop bit set is not detected</li> <li>• Parity error This error occurs when the number of "1" in parity bit and character bits is different from the number set.</li> <li>• Error sum flag The SUM bit is set to "1" when an overrun error, flaming error or parity error occurs.</li> </ul>

Notes :

1. If an overrun error occurs, the UiRB register is indeterminate. The IR bit in the SiRIC register does not change to "1" (interrupt request).
2. The CNT3 to CNT0 bits in the TRGSR register determine either "no division (n=0)" or "divide-by-2<sup>n</sup> (n=1 to 15)".

## Serial I/O (Special Function)

Table 1.20.25. Registers to Be Used and Settings

Register	Bit	Function
UiTB	0 to 7	Set data to be transmitted
UiRB	0 to 7	Received data can be read
	OER, FER, PER, SUM	Error flags
UiBRG	0 to 7	Set a baud rate
UiMR	SMD2 to SMD0	Set to "1012"
	CKDIR	Set to "0"
	STPS	Set to "0"
	PRY	Set to "1" for direct format or to "0" for inverse format
	PRYE	Set to "1"
	IOPOL	Set to "0"
UiC0	CLK1 to CLK0	Select a count source of the UiBRG register
	CRS	Disabled since CRD=1
	TXEPT	Transfer register empty flag
	CRD	Set to "1"
	NCH	Set to "1"
	CKPOL	Set to "0"
	UFORM	Set to "0" for direct format or to "1" for inverse format
UiC1	TE	When transfer is enabled, set to "1"
	TI	Transfer buffer empty flag
	RE	When reception is enabled, set to "1"
	RI	Receive complete flag
	UiIRS	Set to "1"
	UiRRM	Set to "0"
	UiLCH	Set to "0" for direct format or to "1" for inverse format
	UiERE	Set to "1"
UiSMR	0 to 3	Set to "0"
UiSMR2	0 to 7	Set to "0"
UiSMR3	0 to 7	Set to "0"
UiSMR4	0 to 7	Set to "0"

i=0 to 4

## Serial I/O (Special Function)

**Table 1.20.26. Pin Settings in SIM Mode**

Port	Function	Bit and setting value		
		PS0 register	PSL0 register	PD6 register
P62	RxD0 input	PS0_2=0	–	PD6_2=0
P63	TxD0 output	PS0_3=1	–	–
P66	RxD1 input	PS0_6=0	–	PD6_6=0
P67	TxD1 output	PS0_7=1	–	–

**Table 1.20.27. Pin Settings (Continued)**

Port	Function	Bit and setting value			
		PS1 register	PSL1 register	PSC register	PD7 register
P70 <sup>1</sup>	TxD2 output	PS1_0=1	PSL1_0=0	PSC_0=0	–
P71 <sup>1</sup>	RxD2 input	PS1_1=0	–	–	PD7_1=0

Notes :

1. N-channel open drain output

**Table 1.20.28. Pin Settings (Continued)**

Port	Function	Bit and setting value		
		PS3 register <sup>1</sup>	PSL3 register	PD9 register <sup>1</sup>
P91	RxD3 input	PS3_1=0	–	PD9_1=0
P92	TxD3 output	PS3_2=1	PSL3_2=0	–
P96	TxD4 output	PS3_6=1	–	–
P97	RxD4 input	PS3_7=0	–	PD9_7=0

Notes :

1. The PD9 and PS3 registers should be set with subsequent to the instruction that the PRC2 bit in the PRCR register is set to "1" (write enable). Avoid generating an interrupt and operating a DMA transfer between the instruction to set the PRC2 bit to "1" and the instruction to write to the PD9 and PS3 registers.

Figure 1.20.11 shows an example of a SIM interface operation. Figure 1.20.12 shows an example of a SIM interface connection. TxDi should be connected to RxDi for a pull-up.

Serial I/O (Special Function)

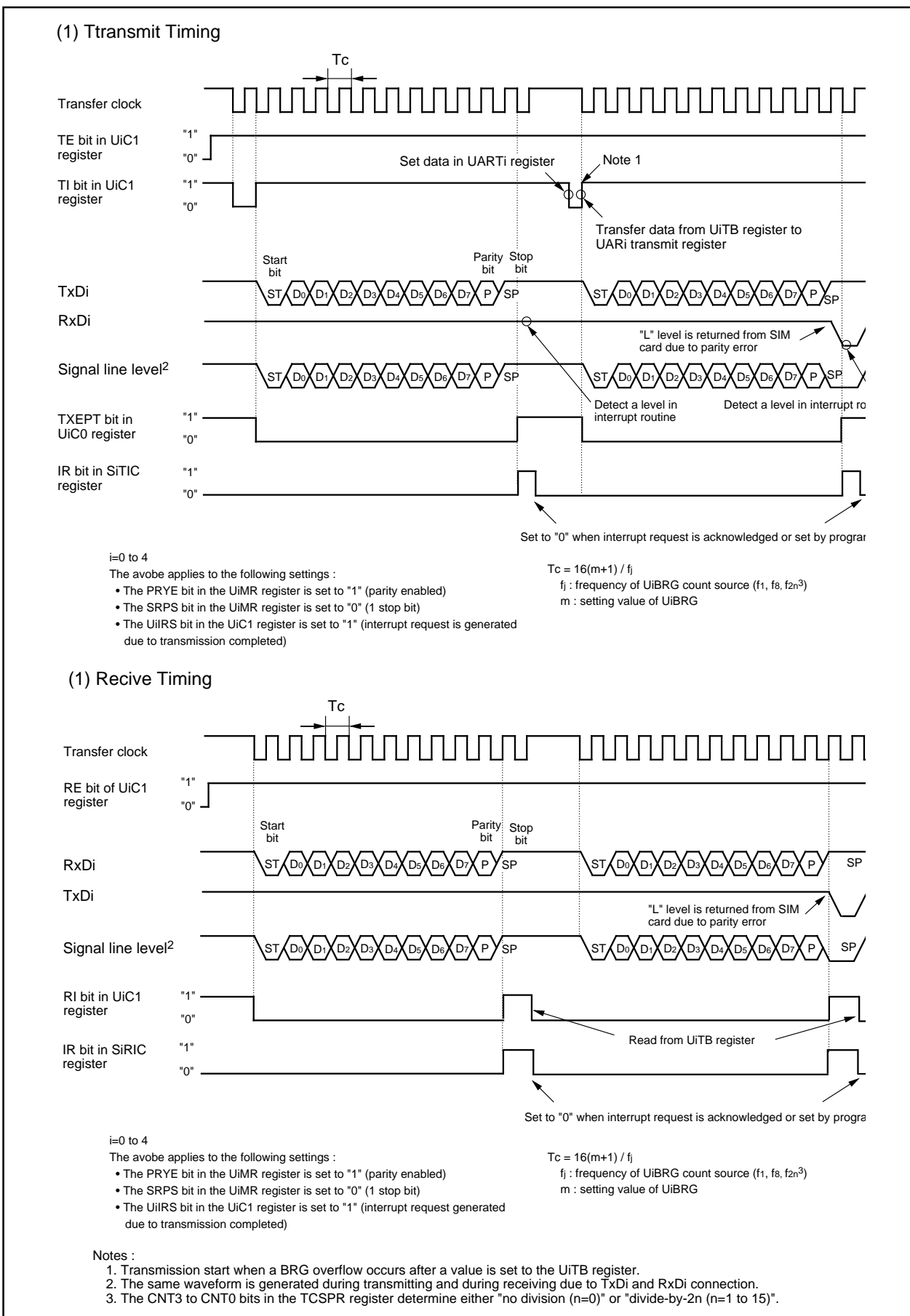
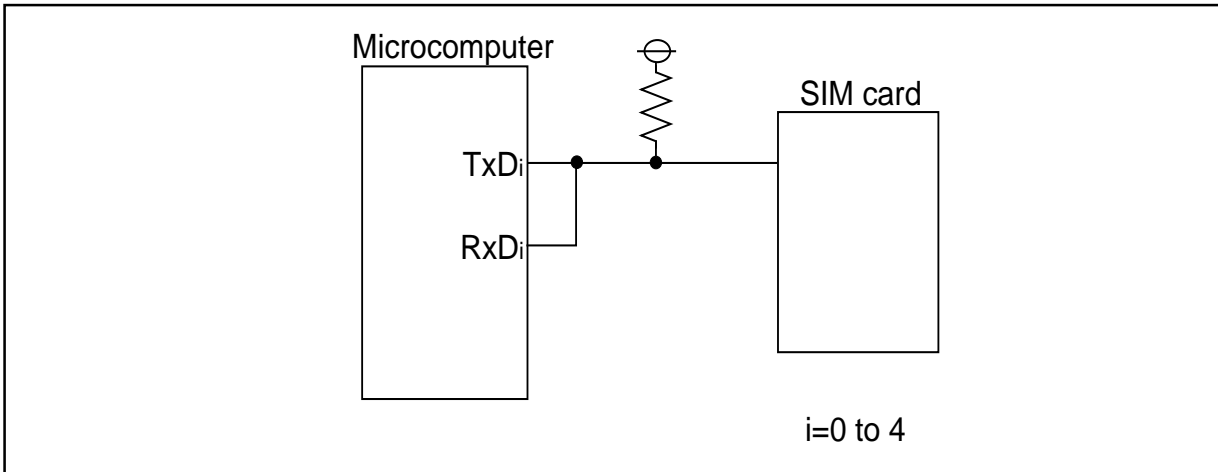


Figure 1.20.11. SIM Interface Operation

**Serial I/O (Special Function)**



**Figure 1.20.12. SIM Interface Connection**

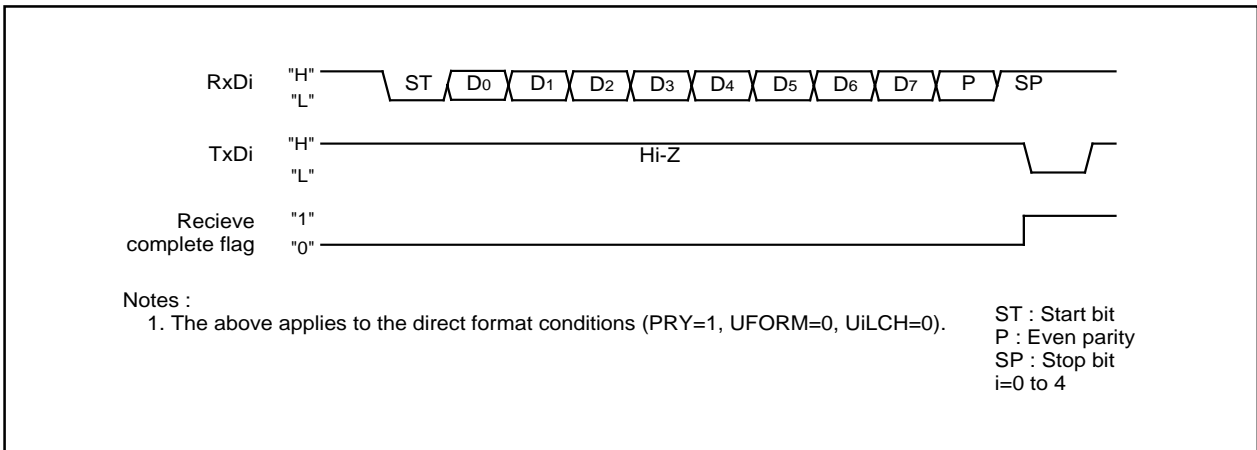
**• Parity Error Signal**

- Parity error signal output function

When the UiERE bit in the UiC1 register (i=0 to 4) is set to "1", the parity error signal can be output. The parity error signal is output with detecting a parity error when data is received. TxDi output is set to "L" in timing shown Figure 1.20.11. When reading the UiRB register during a parity error output, the PER bit in the UiRB register is set to "0" and a TxDi output returns to "H" simultaneously.

- Parity error signal

With a transmit complete interrupt routine, port that shares pins with RxDi indicates whether the parity error signal is returned.



**Figure 1.20.13. Parity Error Signal Output Timing (LSB First)**

## Serial I/O (Special Function)

### • Format

#### - Direct format

The PRY bit in the UiMR register (i=0 to 4) should be set to "1", the UFORM bit in the UiC0 register be set to "0" and the UiLCH bit in the UiC1 register be set to "0". When transmitting data, data set in UiTB register are transmitted with even parity, starting from D0. When receiving data, received data are stored in the UiRB register, starting from D0. Even parity determines whether a parity error occurs.

#### - Inverter format

The PRY bit should be set to "0", the UFORM bit be set to "1" and the UiLCH bit be set to "1".

When receiving data, a value set in the UiTB register is logically inverted and is transmitted with odd parity, starting from D7. When receiving data, received data is logically inverted to be stored in the UiRB register, starting from D7. Odd parity determines whether a parity error occurs.

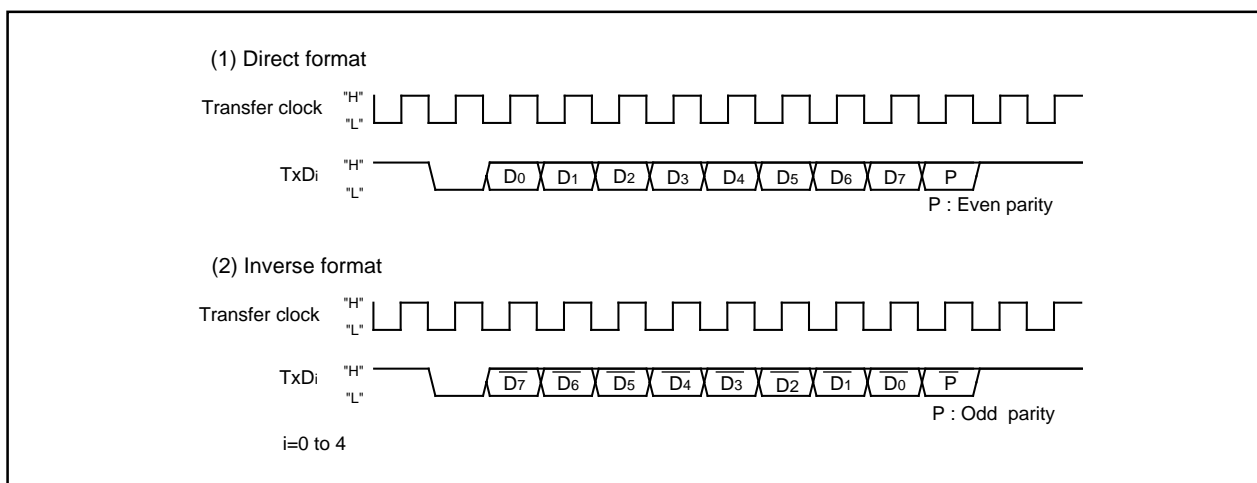


Figure 1.20.14. Parity Error Signal Output Timing (LSB First)

CAN Module

## CAN Module

The CAN (Controller Area Network) module incorporated in the M32C/83 group is a Full CAN module, totally compatible with CAN Specification 2.0 Part B. Table 1.21.1 lists specifications of the CAN module.

**Table 1.21.1. CAN Module Specifications**

Item	Specification
Protocol	CAN Specification 2.0 Part B
Number of message slots	16 slots
Polarity	Dominant: "L" Recessive: "H"
Acceptance filter	Global mask: 1 mask (for CAN0 message slots 0 to13) Local mask: 2 masks (for CAN0 message slots 14 and 15 each)
Baud rate	Baud rate = $\frac{1}{Tq \text{ clock cycle} \times Tq \text{ per bit}}$ --- Max 1 Mbps $Tq \text{ clock cycle} = \frac{BRP + 1}{f_1}$ $Tq \text{ per bit} = SS + PTS + PBS1 + PBS2$ Tq: Time quantum/quanta BRP: Setting value of the C0BRP register, 1-255 SS: Synchronization Segment, 1 Tq PTS: Propagation Time Segment, 1 to 8 Tq PBS1: Phase Buffer Segment 1, 2 to 8 Tq PBS2: Phase Buffer Segment 2, 2 to 8 Tq
Remote frame automatic answering function	A message slot that is received a remote frame transmits a data frame automatically.
Time stamp function	Time stamp function with a 16-bit counter. A count source can be selectable CAN bus bit clock divided by 1, 2, 3, or 4.
BasicCAN mode	BasicCAN function is operated by using CAN0 message slots 14 and 15.
Transmit abort function	A transmission request is aborted.
Loopback function	A frame that the CAN module transmitted is received by the same CAN module.
Forcible error active clear function	Forcibly goes into an error active state.

Notes :

1. An oscillator satisfied 1.58% as maximum oscillator tolerance should be used.

Figure 1.21.1 shows a block diagram of the CAN module. Figure 1.21.2 shows CAN0 message slot buffers (the message slot buffers) and CAN0 message slots (the message slots). Table 1.21.2 lists pin settings of the CAN module.

The message slot i (i=0 to 15) cannot be accessed directly from the CPU. The message slot i to be used should be allocated to the message slot buffer 0 or 1. It can be accessed via an allocated address to the message slot buffer 0 or 1. The C0SBS register selects the message slot i. Figure 1.21.2 shows the 16 byte message slot buffer and message slot.

CAN Module

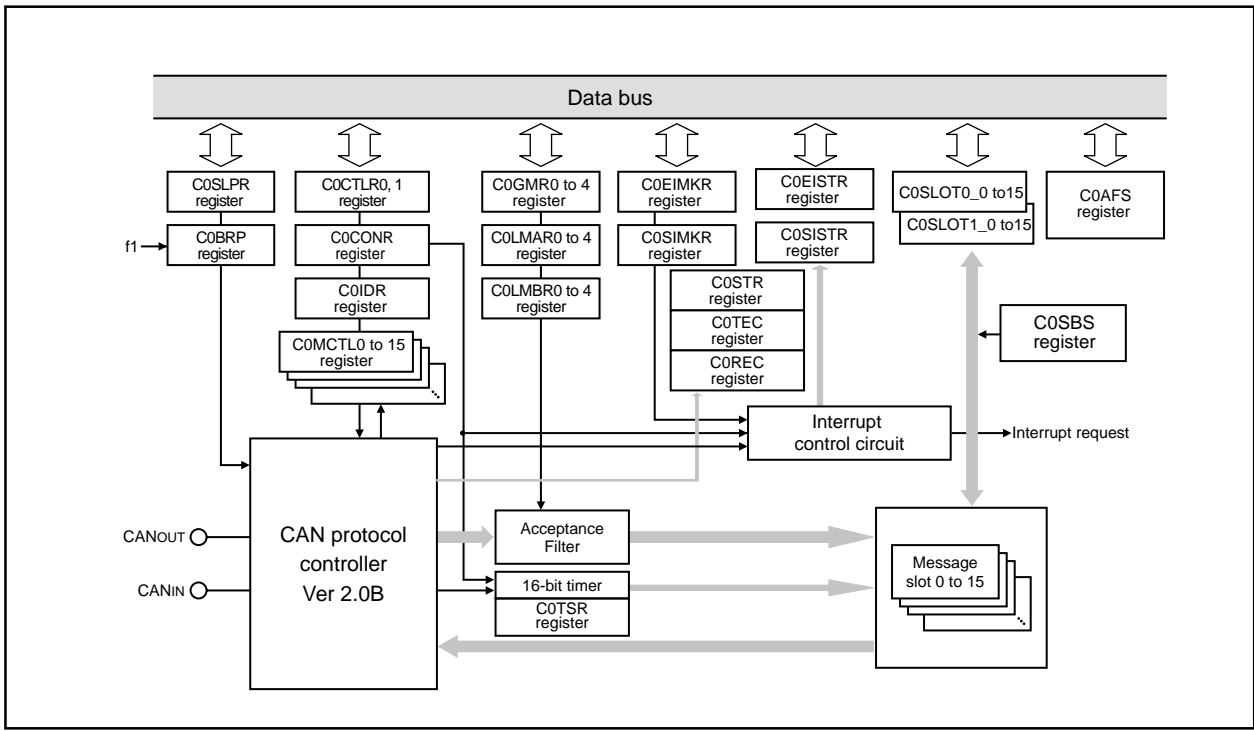


Figure 1.21.1. CAN Module Block Diagram

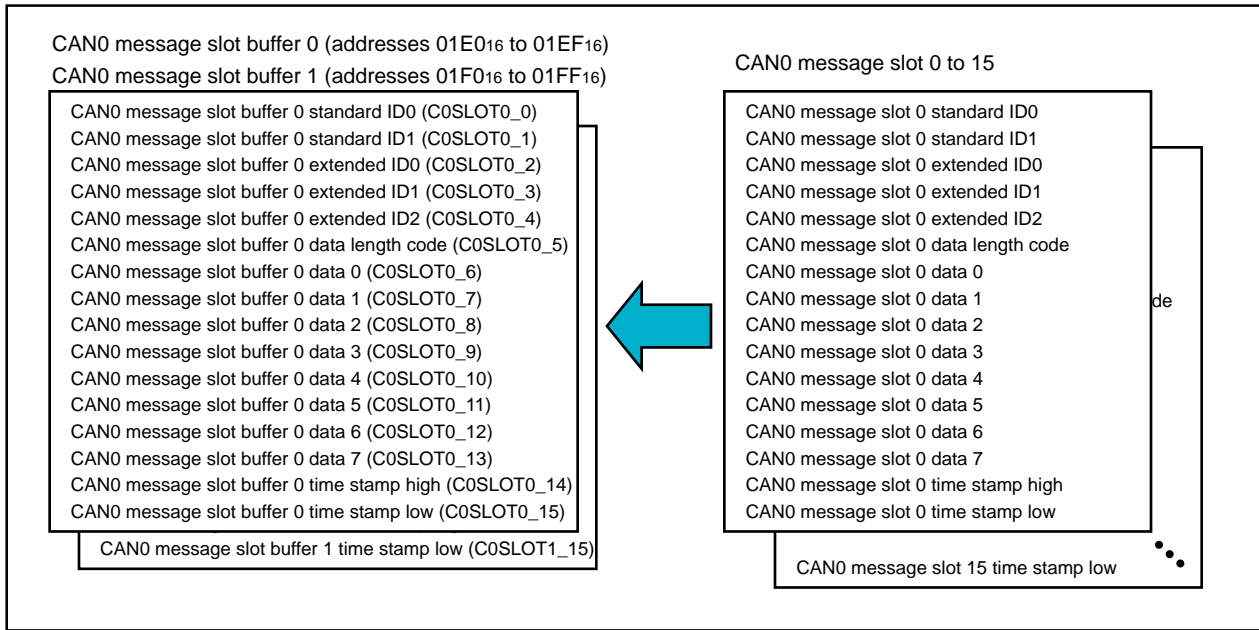


Figure 1.21.2. Message Slot Buffer and Message Slot

Table 1.21.2. Pin Settings

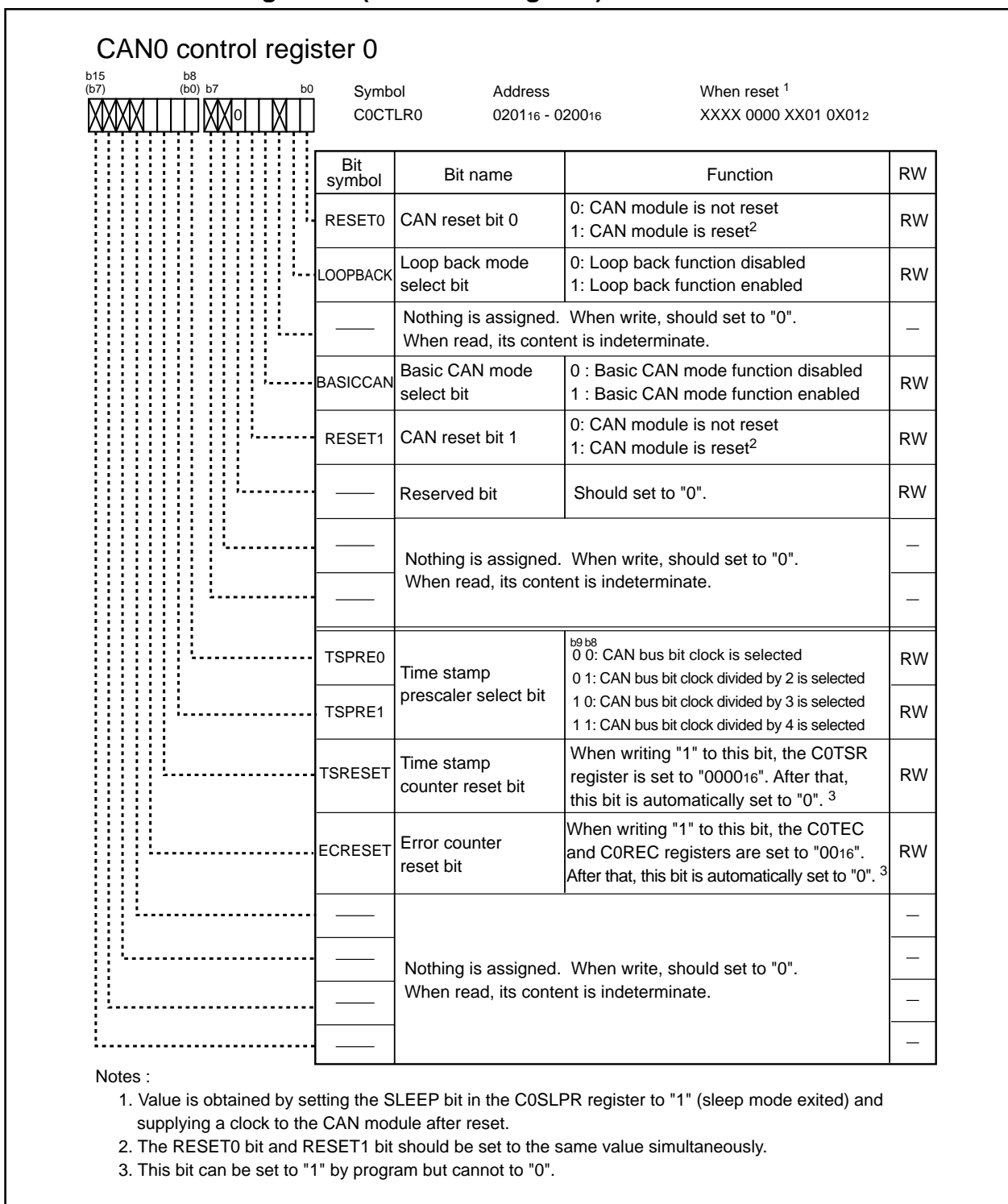
Port	Function	Bit and Setting				
		PS1, PS2 registers	PSL1, PSL2 registers	PSC register	IPS register	PD7, PD8 registers
P76	CANOUT	PS1_6=1	PSL1_6=0	PSC_6=1	—	—
P77	CANIN	PS1_7=0	—	—	IPS3=0	PD7_7=0
P82	CANOUT	PS2_2=1	PSL2_2=1	—	—	—
P83	CANIN	—	—	—	IPS3=1	PD8_3=0

**CAN Module**

**CAN-Associated Registers**

Figures 1.21.3 to 1.21.28 show registers associated with CAN. When accessing the associated registers, the MCD4 to MCD0 bits in the MCD register should be set to "100102" (no division of CPU clock), the PM13 bit in the PM1 register be set to "1" (two waits) and the CM07 bit in the CM0 register be set to "0" (selectable from XIN-XOUT).

**1. CAN0 Control Register 0 (C0CTRL0 Register)**



**Figure 1.21.3 C0CTRL0 Register**

**• RESET0 Bit and RESET1 Bit**

When setting both RESET0 and RESET1 bits to "1", the CAN module is immediately reset regardless of ongoing CAN communication.

When the CAN module reset is completed, the C0TSR register is set to "0000<sub>16</sub>". Also, the C0TEC and C0REC registers are set to "00<sub>16</sub>" and the STATE\_ERRPAS and STATE\_BUSOFF bits in the C0STR register are set to "0".

When both RESET0 and RESET1 bits are changed "1" to "0", the C0TSR register starts counting. CAN communication is available after detecting 11 contiguous recessive bits.

Notes :

1. Both RESET0 and RESET1 bits should be set to the same value simultaneously.
2. After setting the RESET0 and RESET1 bits to "1", confirm that set the STATE\_RESET bit in the C0STR register to "1" (CAN module reset is completed) for CAN configuration.
3. When setting the RESET0 and RESET1 bits to "1", the CANOUT pin outputs "H". CAN bus error may occur by setting the RESET0 and RESET1 bits while CAN frame is transmitting.
4. For CAN communication, the PS1, PS2, PSL1, PSL2, PSC and IPS registers should be set with setting the STATE\_RESET bit to "1" (CAN module reset is completed).

**LOOPBACK Bit**

When the LOOPBACK bit is set to "1" (loopback function enabled) and the receive message slot has a matching identifier and frame format with a transmitted frame, the transmitted frame is stored to the receive message slot.

Notes :

1. No ACK is returned to a transmitted frame.
2. The LOOPBACK bit should be changed only when setting the STATE\_RESET bit to "1" (CAN module reset is completed).

**BASICCAN Bit**

When setting the BASICCAN bit to "1", message slots 14 and 15 enter BasicCAN mode.

**• Operation in BasicCAN Mode**

In BasicCAN mode, the message slots 14 and 15 are used as dual-structured buffers. Received frames with a matching identifier are alternately stored into the message slots 14 and 15 by acceptance filter. When the message slot 14 is active (the next received frame is to be stored in the message slot 14), an identifier in the message slot 14 and the C0LMAR0 to C0LMAR4 registers are used as the acceptance filter. When the message slot 15 is active, an identifier in the message slot 15 and the C0LMBR0 to C0LMBR4 registers are used as the acceptance filter as well. Both data frame and remote frame can be received.

When entering BasicCAN mode, the same identifier should be set in two message slots and the same value be set in the C0LMAR0 to C0LMAR4 registers and in the C0LMBR0 to C0LMBR4 registers.

**• How to Enter BasicCAN Mode**

- (1) Set the BASICCAN bit to "1".
- (2) Set an identifier in the message slots 14 and 15. Set the C0LMAR0 to C0LMAR4 registers and C0LMBR0 to C0LMBR4 registers. (Set the same value.)
- (3) Select a frame format (standard or extended) to be handled with the message slots 14 and 15 in the IDE14 and 15 bits in the C0IDR register. (Set the same format.)
- (4) Set the REMACTIVE bit in the C0MCTL 14 and C0MCTL15 registers for the message slots 14 and 15 to "0" (data frame is received) and the RECREQ bit to "1" (receive requested).

## CAN Module

---

### Notes :

1. The BASICCAN bit should be changed only when the STATE\_RESET bit is set to "1" (CAN module reset is completed).
2. The message slot 14 is the first slot to become active after setting the RESET0 and RESET1 bits to "0".
3. The message slots 0 to 13 are not affected by entering BasicCAN mode.

### TSPRE1, TSPRE0 Bits

The TSPRE1 and TSPRE0 bits determines which count source is used for a time stamp counter.

#### Notes :

1. The TSPRE1 to TSPRE0 bits should be changed only when setting the STATE\_RESET bit to "1" (CAN module reset is completed).

### TSRESET Bit

When setting the TSRESET bit to "1" (counter reset), the C0TSR register is set to "0000<sub>16</sub>". The TSRESET bit is automatically set to "0" after setting the C0TSR register to "0000<sub>16</sub>".

### ECRESET Bit

When setting the ECRESET bit to "1", the C0TEC and C0REC registers are set to "00<sub>16</sub>". The CAN module forcibly enters an error active state.

The ECRESET bit is automatically set to "0" after entering an error active state.

#### Notes :

1. In an error active state, the CAN module is ready to communicate when detecting 11 continuous recessive bits on a CAN bus.

## 2. CAN0 Control Register 1 (C0CTRL1 Register)

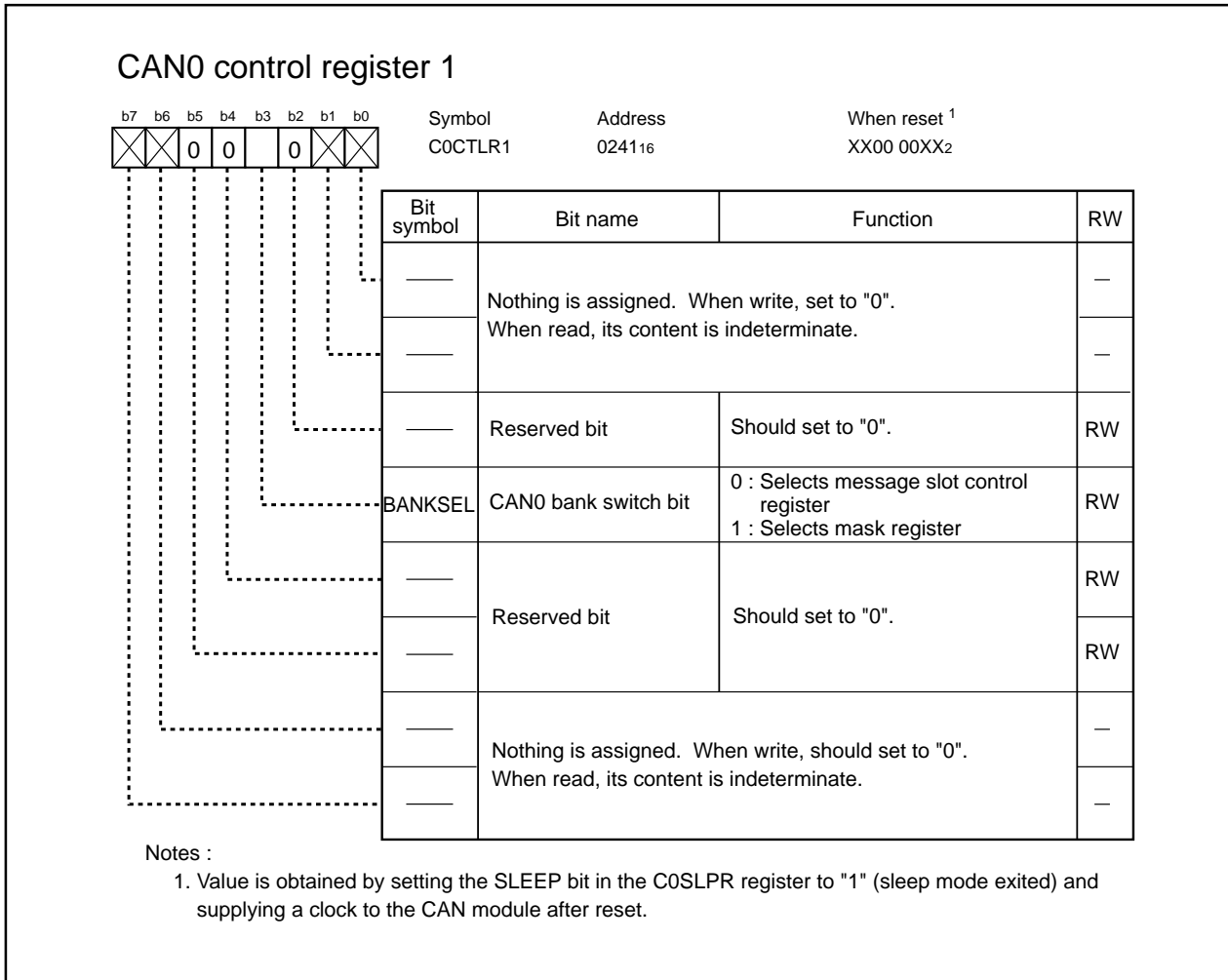


Figure 1.21.4. C0CTRL1 Register

### BANKSEL Bit

The BANKSEL bit selects registers allocated to addresses 0220<sub>16</sub> to 023F<sub>16</sub>.

When setting the BANKSEL bit to "0", the C0MCTL0 to C0MCTL15 registers can be accessed. When setting the BANKSEL bit to "1", the C0GMR0 to C0GMR4 registers, C0LMAR0 to C0LMAR4 registers and C0LMBR0 to C0LMBR4 registers can be accessed.

### 3. CAN0 Sleep Control Register (C0SLPR Register)

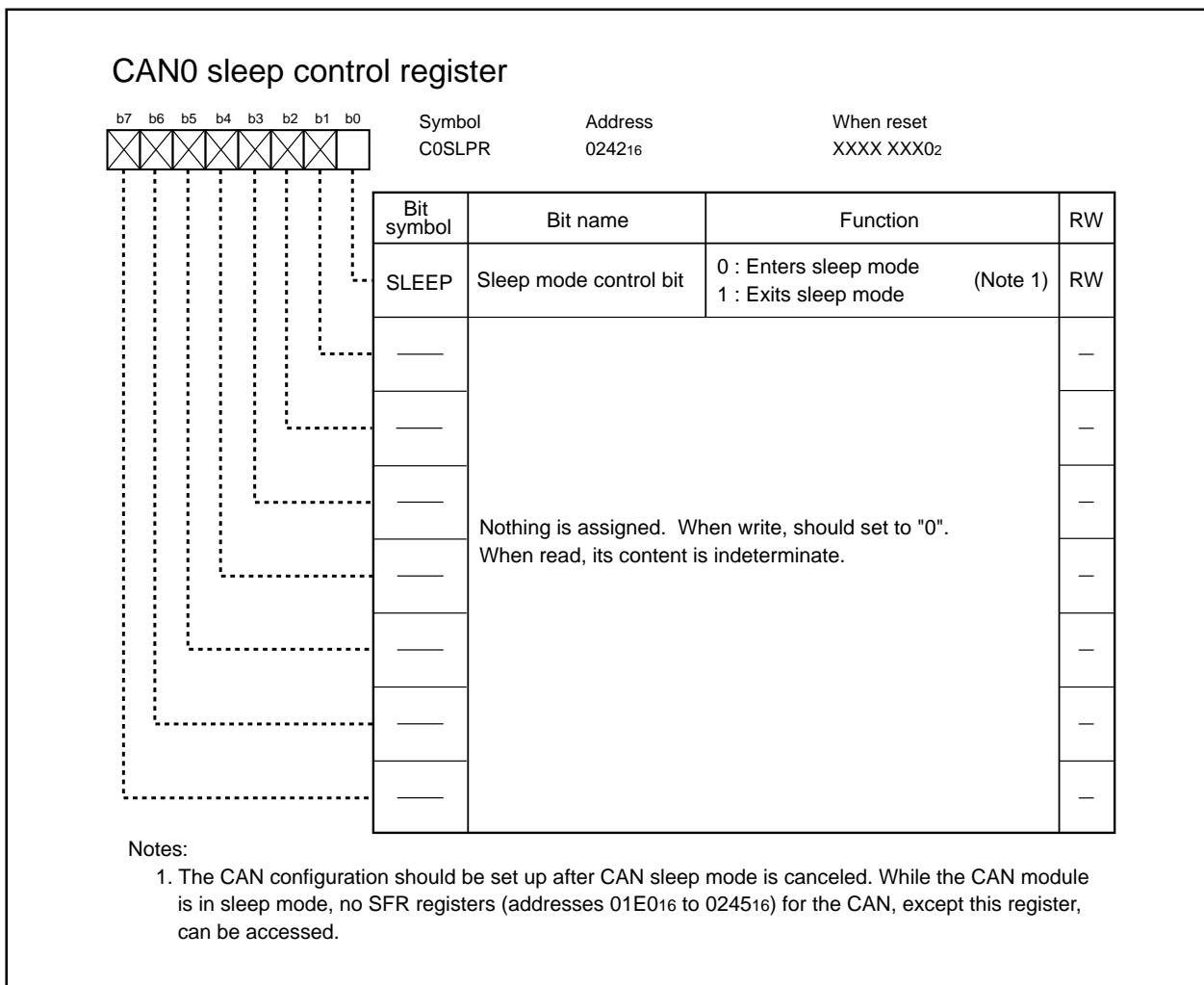


Figure 1.21.5. C0SLPR Register

#### SLEEP Bit

When setting the SLEEP bit to "0", a clock stops running to enter sleep mode.

When setting the SLEEP bit to "1", a clock starts running to exit sleep mode.

Notes :

- Sleep mode should be entered after setting the STATE\_RESET bit to "1" (CAN module reset is completed).

### 4. CAN0 Status Register (C0STR Register)

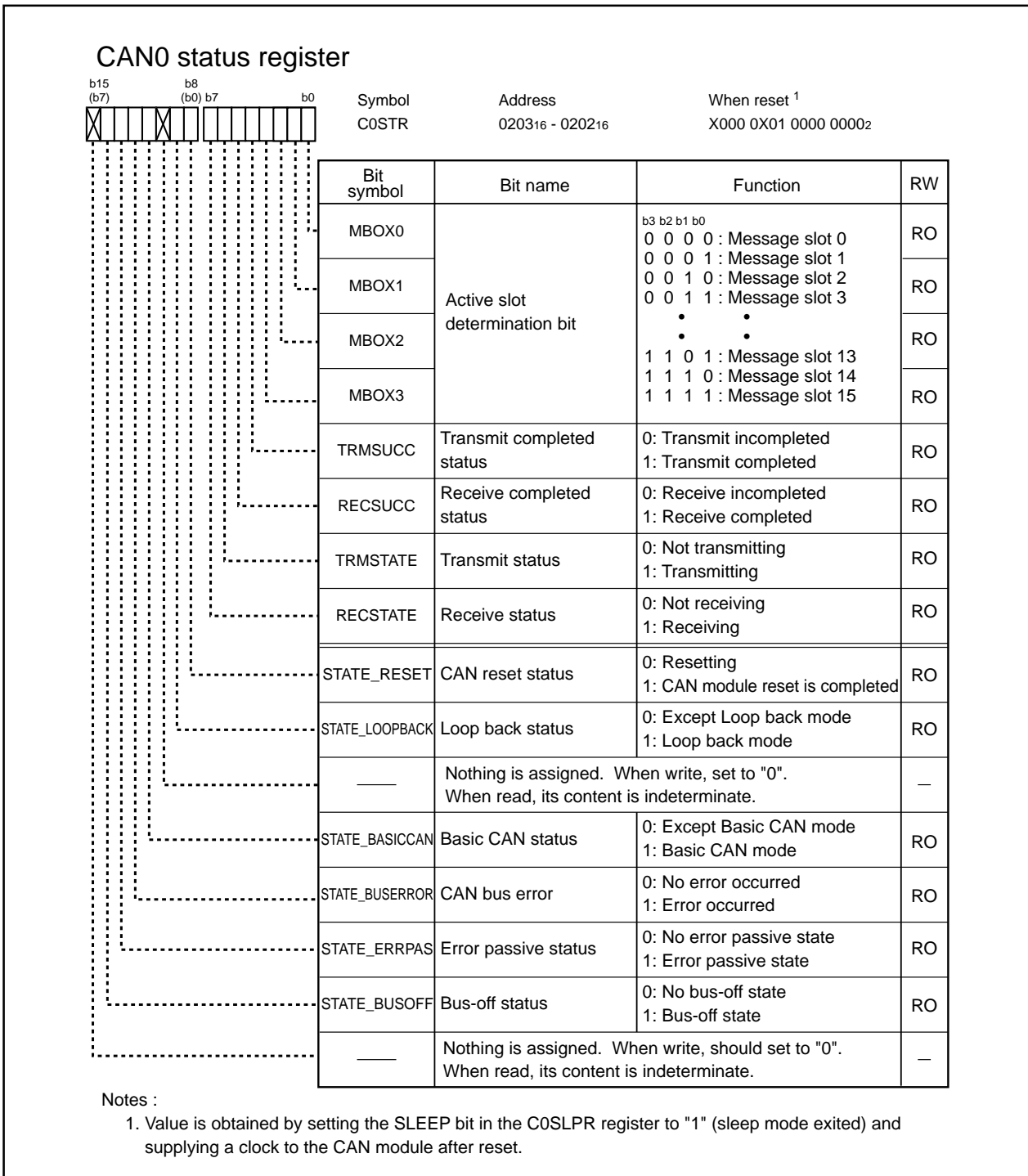


Figure 1.21.6. C0STR Register

#### MBOX3 to MBOX0 Bits

When the CAN module has transmitted data or stored received data, relevant slot number is stored into the MBOX3 to MBOX0 bits.

#### TRMSUCC Bit

The TRMSUCC bit is set to "1" when the CAN module has transmitted data normally.

The TRMSUCC bit is set to "0" when the CAN module has received data normally.

**CAN Module**

---

**RECSUCC Bit**

The RECSUCC bit is set to "1" when the CAN module has received data normally. (It does not matter whether received message has been stored in the message slot or not.) If the received message is transmitted in loopback mode, the TRMSUCC bit is set to "1" and the RECSUCC bit is set to "0". The RECSUCC bit is set to "0" when the CAN module has transmitted data normally.

**TRMSTATE Bit**

The TRMSTATE bit is set to "1" when the CAN module is operating as a transmit node.  
The TRMSTATE bit is set to "0" when the CAN module is in a bus-idle state or starts operating as a receive node.

**RECSTATE Bit**

The RECSTATE bit is set to "1" when the CAN module is operating as a receive node.  
The RECSTATE bit is set to "0" when the CAN module is in a bus-idle state or starts operating as a transmit node.

**STATE\_RESET Bit**

After setting both RESET0 and RESET1 bits to "1" (CAN module is reset), the STATE\_RESET bit is set to "1" by completion of CAN module reset.  
The STATE\_RESET bit is set to "0" when setting the RESET0 or RESET1 bits to "0".

**STATE\_LOOPBACK Bit**

The STATE\_LOOPBACK bit indicates that the CAN module is operating in loopback mode when setting the STATE\_LOOPBACK bit to "1".  
The STATE\_LOOPBACK bit is set to "1" when the LOOPBACK bit in the C0CTRL0 register is set to "1" (loop back function enabled).  
The STATE\_LOOPBACK bit is set to "0" when the LOOPBACK bit is set to "0" (loop back function disabled).

**STATE\_BASICCAN Bit**

The CAN module operates in BasicCAN mode when setting the STATE\_BASICCAN bit to "1".  
Refer to "BASICCAN bit" in "1. CAN0 control register (C0CTRL0 register)" about BasicCAN mode.  
The STATE\_BASICCAN bit is set to "0" when setting the BASICCAN bit to "0" (BasicCAN mode function disabled).

**STATE\_BUSERROR Bit**

The STATE\_BUSERROR bit is set to "1" when an error is detected.  
The STATE\_BUSERROR bit is set to "0" when the CAN module has transmitted or received normally. It does not matter whether a received message has been stored into the message slot or not.  
Note :

1. When the STATE\_BUSERROR bit is set to "1", the STATE\_BUSERROR bit remains unchanged even if both RESET 0 and RESET1 bits are set to "1" (CAN module is reset).

## CAN Module

---

### **STATE\_ERRPAS Bit**

The STATE\_ERRPAS bit is set to "1" when values of the C0TEC or C0REC register exceeds 127 with the CAN module placed in an error-passive state.

The STATE\_ERRPAS bit is set to "0" when the CAN module is in another error state from in an error passive state.

The STATE\_ERRPAS bit is set to "0" when both RESET0 and RESET1 bits are set to "1" (CAN module is reset).

### **STATE\_BUSOFF Bit**

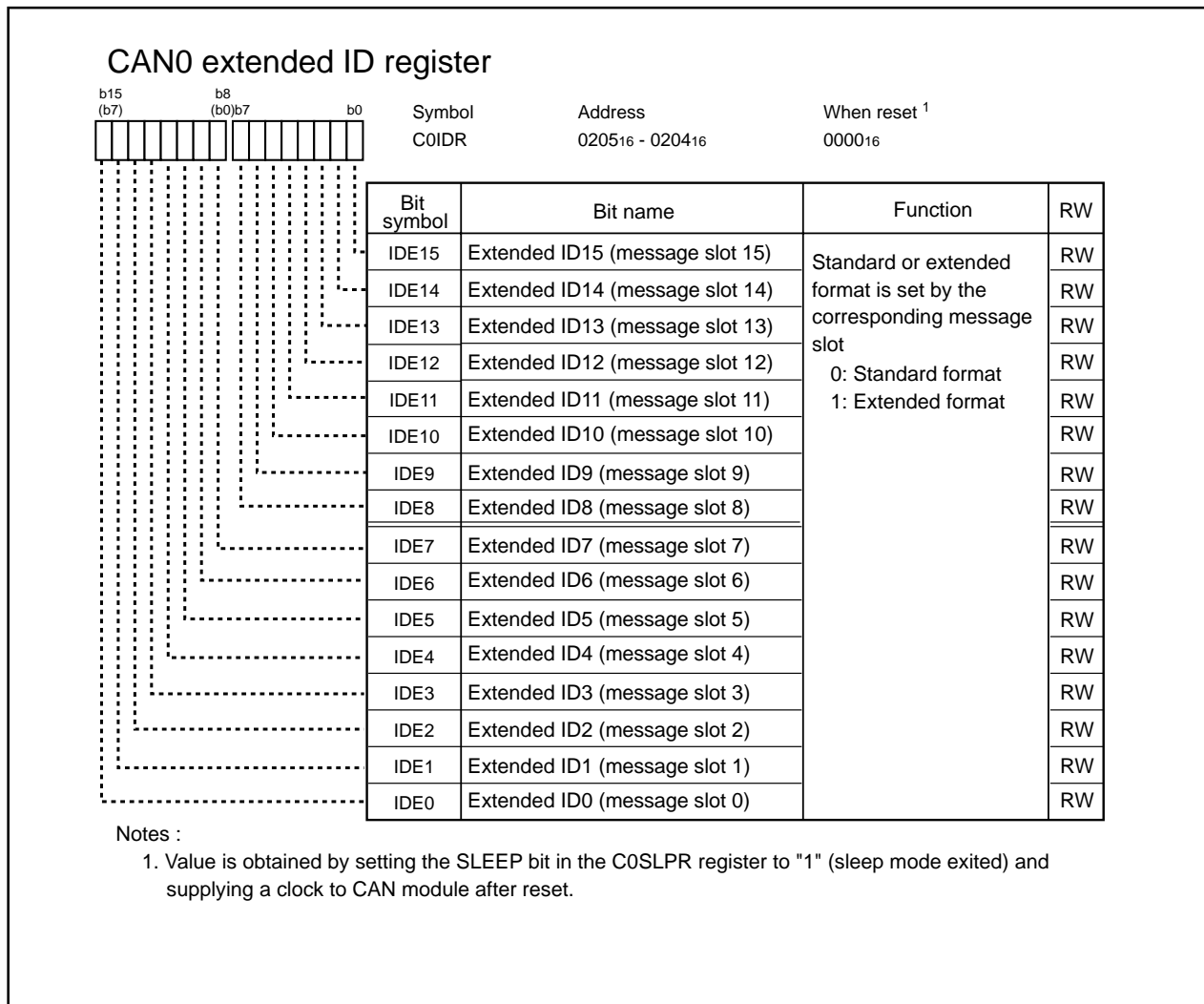
The STATE\_BUSOFF bit is set to "1" when value of the C0TEC register exceeds 255 with the CAN module placed in a bus-off state.

The STATE\_BUSOFF bit is set to "0" when the CAN module is in an error-active state again from in a bus-off state.

The STATE\_BUSOFF bit is set to "0" when both RESET0 and RESET1 bits are set to "1" (CAN module is reset).

CAN Module

**5. CAN0 Extended ID Register (C0IDR Register)**



**Figure 1.21.7. C0IDR Register**

Bits in the C0IDR register determines a frame format in the message slot for each bit.

Standard format is selected when setting the above bit to "0".

Extended format is selected when setting the above bit to "1".

Notes :

- Each bit in the C0IDR register should be set when neither transmit request nor receive request from the message slot is generated.

## 6. CAN0 Configuration Register (C0CONR Register)

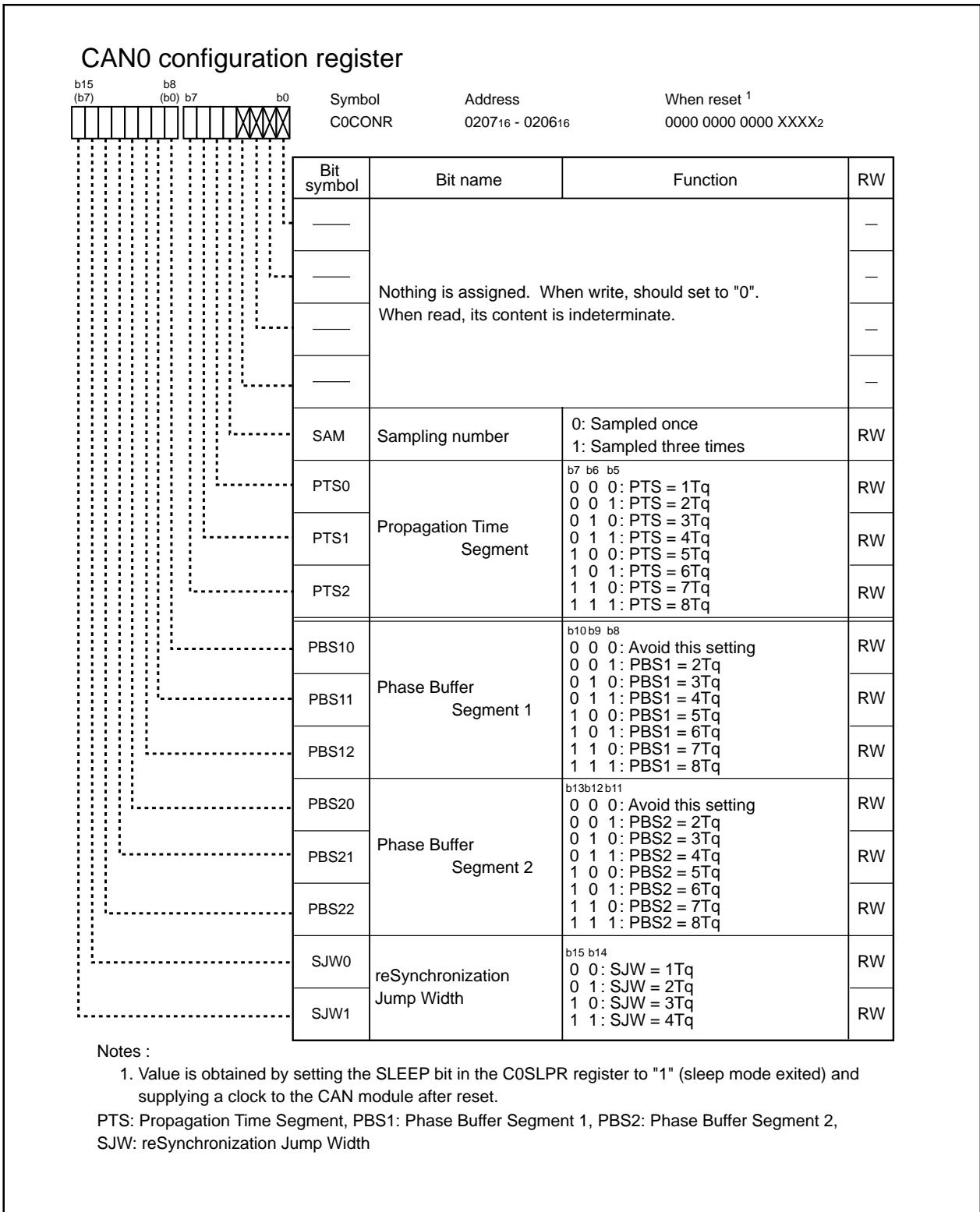


Figure 1.21.8. C0CONR Register

## CAN Module

**SAM Bit**

The SAM bit determines the number of samples of the CAN<sub>IN</sub> pin output to be taken per bit.

When setting the SAM bit to "0", only one sample is taken per bit at the end of Phase Buffer Segment 1 (PBS1) to determine the value of the bit.

When setting the SAM bit to "1", three samples per bit are taken at the end of PBS1, in one time quantum in two time quanta before the end of PBS1.

**PTS2 to PTS0 Bits**

The PTS2 to PTS0 bits determine Propagation Time Segment (PTS) wide.

**PBS12 to PBS10 Bits**

The PBS12 to PBS10 bits determine PBS1 wide. The PBS12 to 10 bits should be set to "0012" or more.

**PBS22 to PBS20 Bits**

The PBS22 to PBS20 bits determine PBS2 wide. The PBS22 to PBS20 bits should be set to "0012" or more.

**SJW1 to SJW0 Bits**

The SJW1 to SJW0 bits fix Resynchronization Jump Width (SJW) width. The SJW1 to SJW0 bits should be set to a value less than the PBS22 to PBS20 bits.

**Table 1.21.3 Bit Timing when CPU Clock = 30 MHz**

Baud rate	BRP	T <sub>q</sub> clock cycles (ns)	T <sub>q</sub> per bit	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%

## 7. CAN0 Time Stamp Register (C0TSR Register)

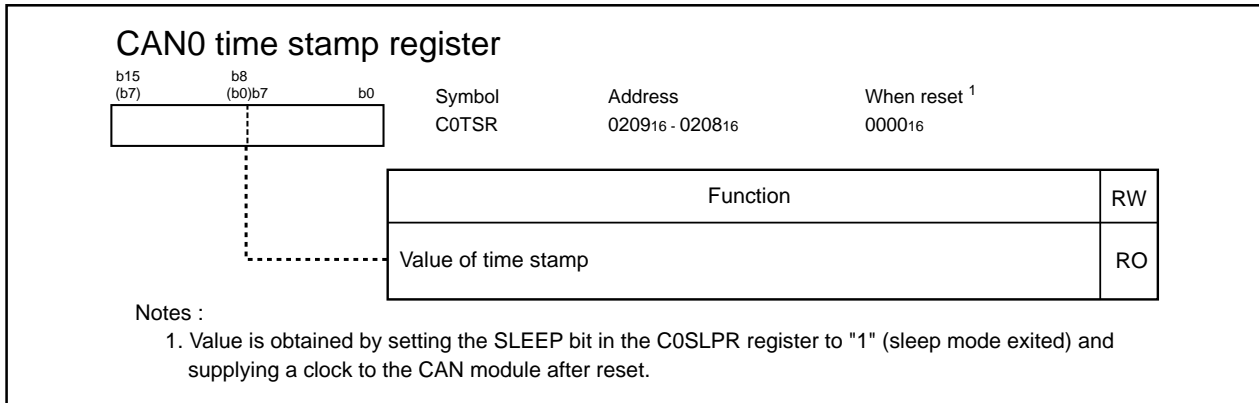


Figure 1.21.9. C0TSR Register

The C0TSR register is a 16-bit counter. The C0TSR register selects either CAN bus bit clock divided by 1, 2, 3 or 4 as a count source in the TSPRE0 and TSPRE1 bits in the C0CTRL0 register.

When transmission or reception is completed, a value of the C0TSR register is automatically stored into the message slot.

The C0TSR register starts counting up when the RESET0 and RESET1 bits in the C0CTRL0 register are set to "0".

The C0TSR register is set to "0000<sub>16</sub>" upon the following conditions:

- At the next count timing after the C0TSR register is set to "FFFF<sub>16</sub>"
- When the RESET0 and RESET1 bits are set to "1" (CAN module is reset) by program
- When the TSRESET bit is set to "1" (C0TSR register reset) by program

In loopback mode, when either data frame or remote frame receive message slot that stores message is available, a value of the C0TSR register is stored into the corresponding message slot when a frame reception is completed. (A value of the C0TSR register is not stored when a frame transmission is completed.)

## 8. CAN0 Transmit Error Count Register (C0TEC Register)

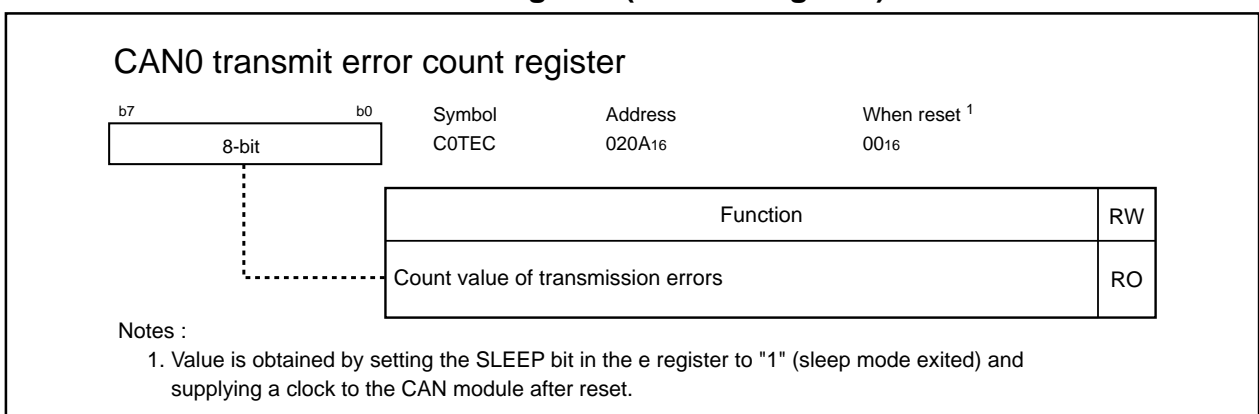


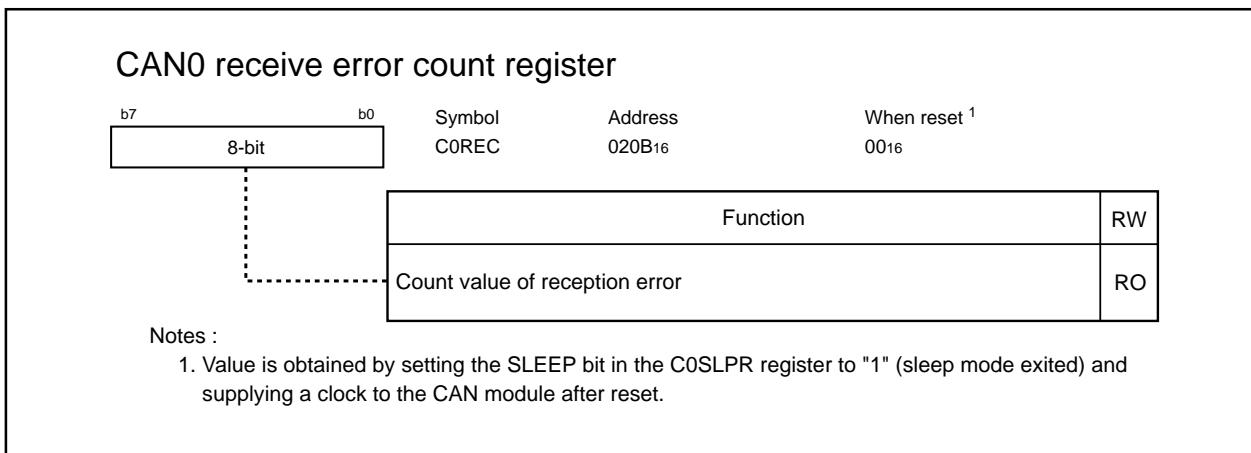
Figure 1.21.10. C0TEC Register

In an error active or an error passive state, a count value of the transmission error is stored into the C0TEC register. The counter is decremented when the CAN module has transmitted normally or it is incremented when an error occurs while transmitting.

In a bus-off state, an indeterminate value is stored into the C0TEC register. The C0TEC register is set to "00<sub>16</sub>" when the CAN module is placed in an error active state again.

CAN Module

**9. CAN0 Receive Error Count Register (C0REC Register)**



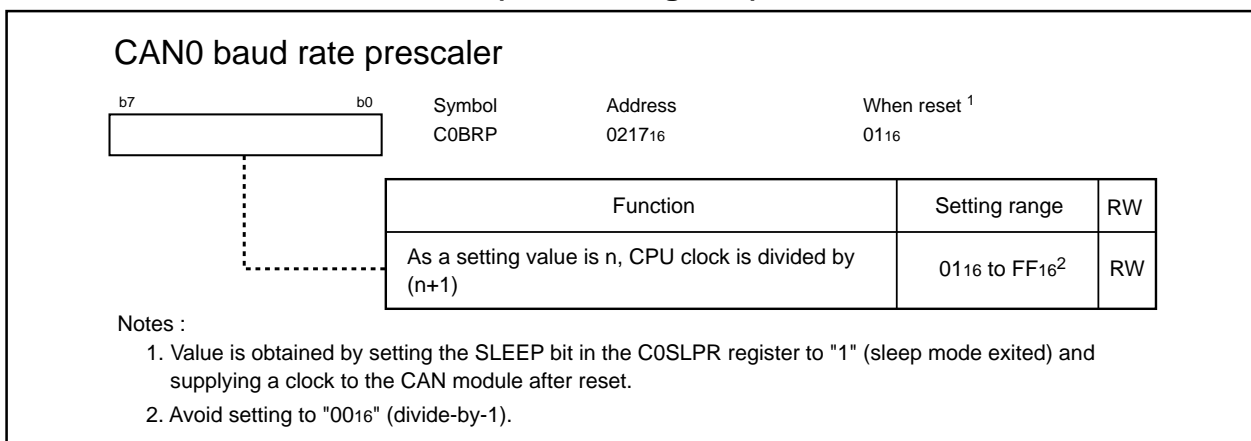
**Figure 1.21.11. C0REC Register**

In an error active or an error passive state, a count value of the reception error is stored into the C0REC register. The counter is decremented when the CAN module has received normally or it is incremented when an error occurs while receiving.

When CAN module has received normally with the C0REC register being equal to or more than 128 (error passive state), the C0REC register is set to 127.

In a bus-off state, an indeterminate value is stored into the C0REC register. The C0REC register is set to "00<sub>16</sub>" the CAN module is placed in entering an error active state again.

**10. CAN0 Baud Rate Prescaler (C0BPR Register)**



**Figure 1.21.12. C0BPR Register**

The C0BPR register determines the Tq clock cycle, which is used to build up an respective bit timing. The baud rate is obtained from Tq clock cycle x Tq per bit.

$$Tq \text{ clock cycle} = (BRP+1) / f_1$$

$$\text{Baud rate} = \frac{1}{Tq \text{ clock cycle} \times Tq \text{ per bit}}$$

$$Tq \text{ per bit} = SS + PTS + PBS1 + PBS2$$

Tq: Time quantum/quanta

BRP: Setting value of the C0BPR register, 1-255

SS: Synchronization Segment, 1 Tq

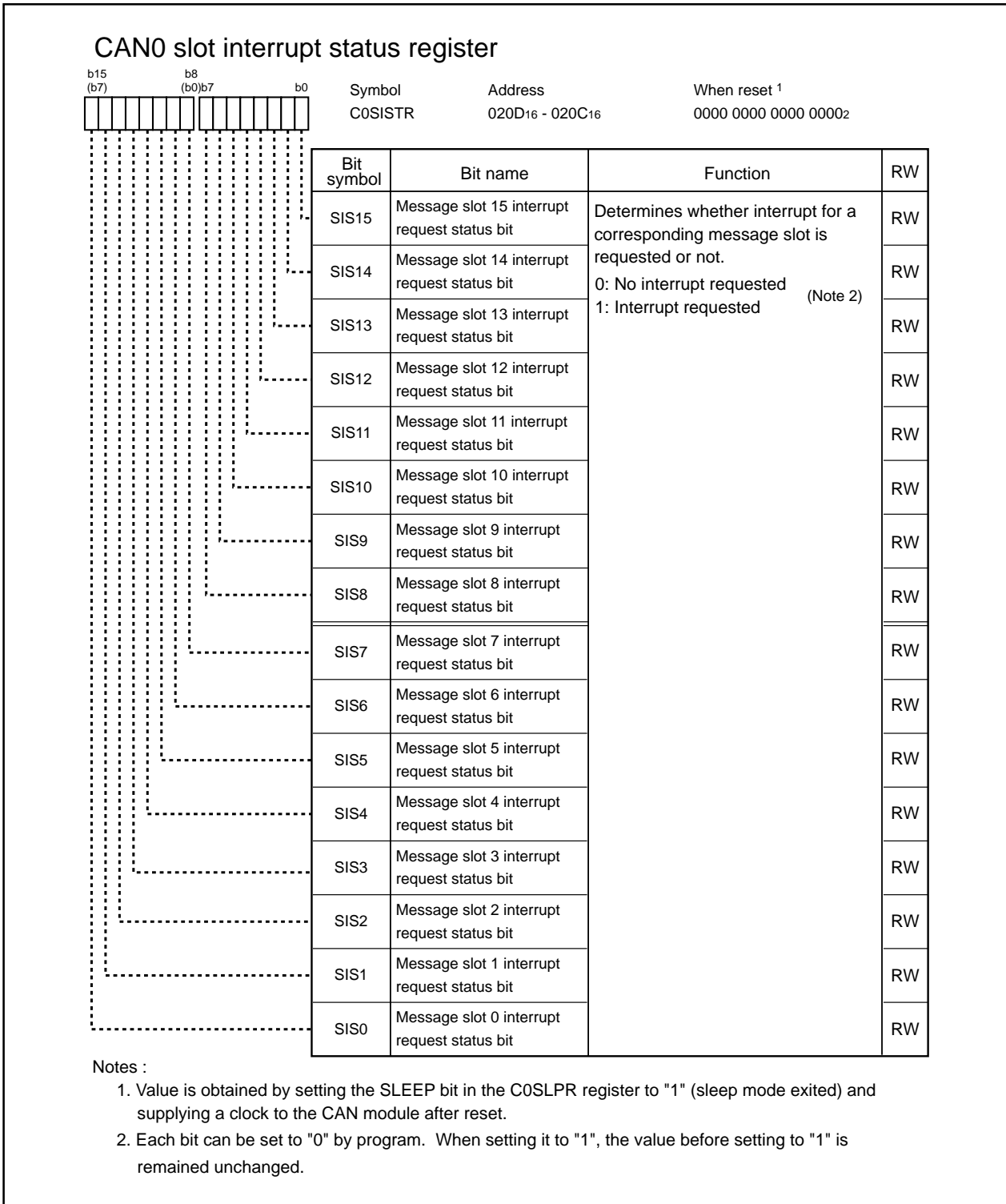
PTS: Propagation Time Segment, 1 to 8 Tq

PBS1: Phase Buffer Segment 1, 2 to 8 Tq

PBS2: Phase Buffer Segment 2, 2 to 8 Tq

CAN Module

**11. CAN0 Slot Interrupt Status Register (C0SISTR Register)**



**Figure 1.21.13. C0SISTR Register**

## CAN Module

---

With the CAN interrupt, the C0SISTR register determines which message slot requests an interrupt. The SISI bits (i=0 to 15) are not automatically set to "0" (no interrupt request) although an interrupt is acknowledged. The SISI bits should be set to "0" by program<sup>1</sup>.

Refer to the paragraph "CAN interrupt" for details.

### • Message Slot Set for Transmission

The SISI bit is set to "1" (interrupt request) when the CAN module stores a value of the C0TSR register into the message slot i after transmission completes.

### • Message Slot Set for Reception

The SISI bit is set to "1" when the CAN module stores a received message in the message slot i after reception completes.

Notes :

1. The MOV instruction, instead of the bit clear instruction, should be used to set the SISI bit to "0". Bits to remain unchanged should be set to "1".

e.g. To set the SIS0 bit to "0"

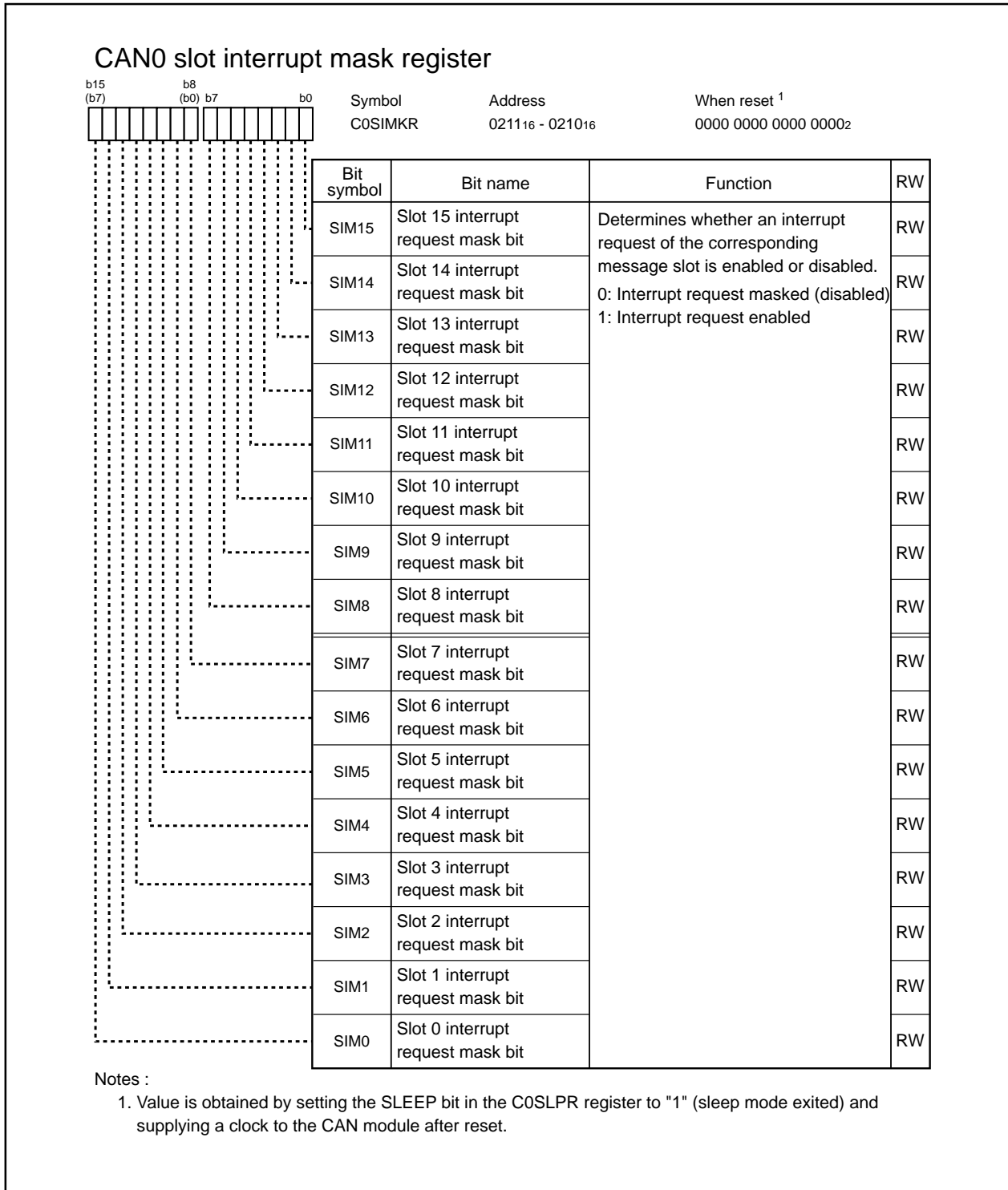
Assembler language: `mov.w #07FFFh, C0SISTR`

C language: `c0sistr = 0x7FFF;`

2. If the automatic answering function is enabled with the message slot to receive remote frames, the SISI bit is set to "1" after receiving a remote frame and transmitting a data frame.
3. With the message slot to transmit remote frames, the SISI bit is set to "1" after transmitting a remote frame and receiving a data frame.
4. The SISI bit is set to "1" when setting the SISI bit is set to "1" by an interrupt request and to "0" by program simultaneously.

CAN Module

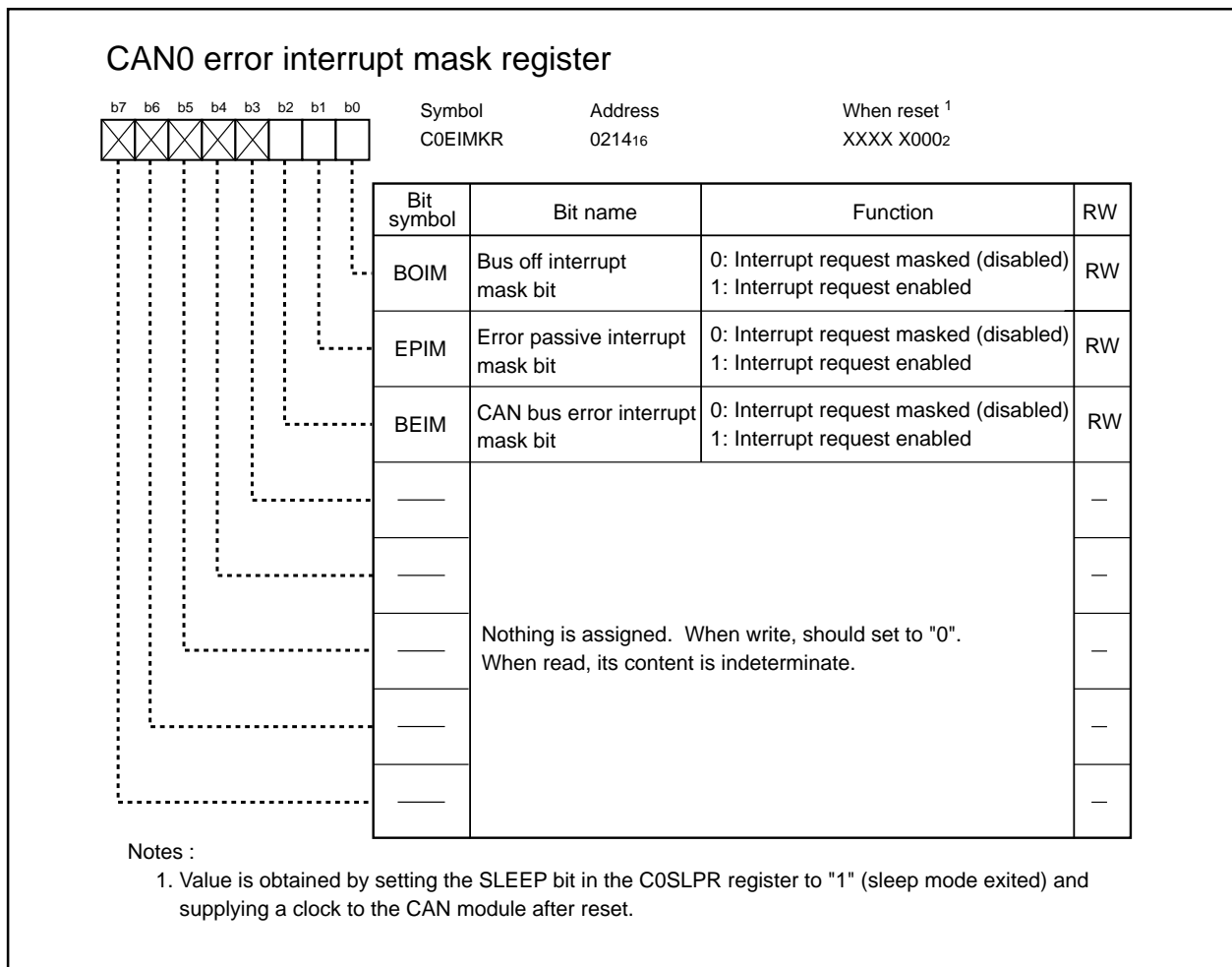
**12. CAN0 Slot Interrupt Mask Register (C0SIMKR Register)**



**Figure 1.21.14. C0SIMKR Register**

The C0SIMKR register determines whether an interrupt request generated by transmitting or receiving the corresponding message slot is enabled or disabled. When setting the SIM<sub>i</sub> bit to "1", an interrupt request generated by transmitting or receiving the corresponding message slot is enabled. Refer to the paragraph "CAN interrupt" for details.

### 13. CAN0 Error Interrupt Mask Register (C0EIMKR Register)



**Figure 1.21.15. C0EIMKR Register**

**BOIM Bit**

The BOIM bit determines whether an interrupt request is enabled or disabled when the CAN module is placed in a bus-off state. When setting the BOIM bit to "1", a bus-off interrupt request is enabled.

**EPIM Bit**

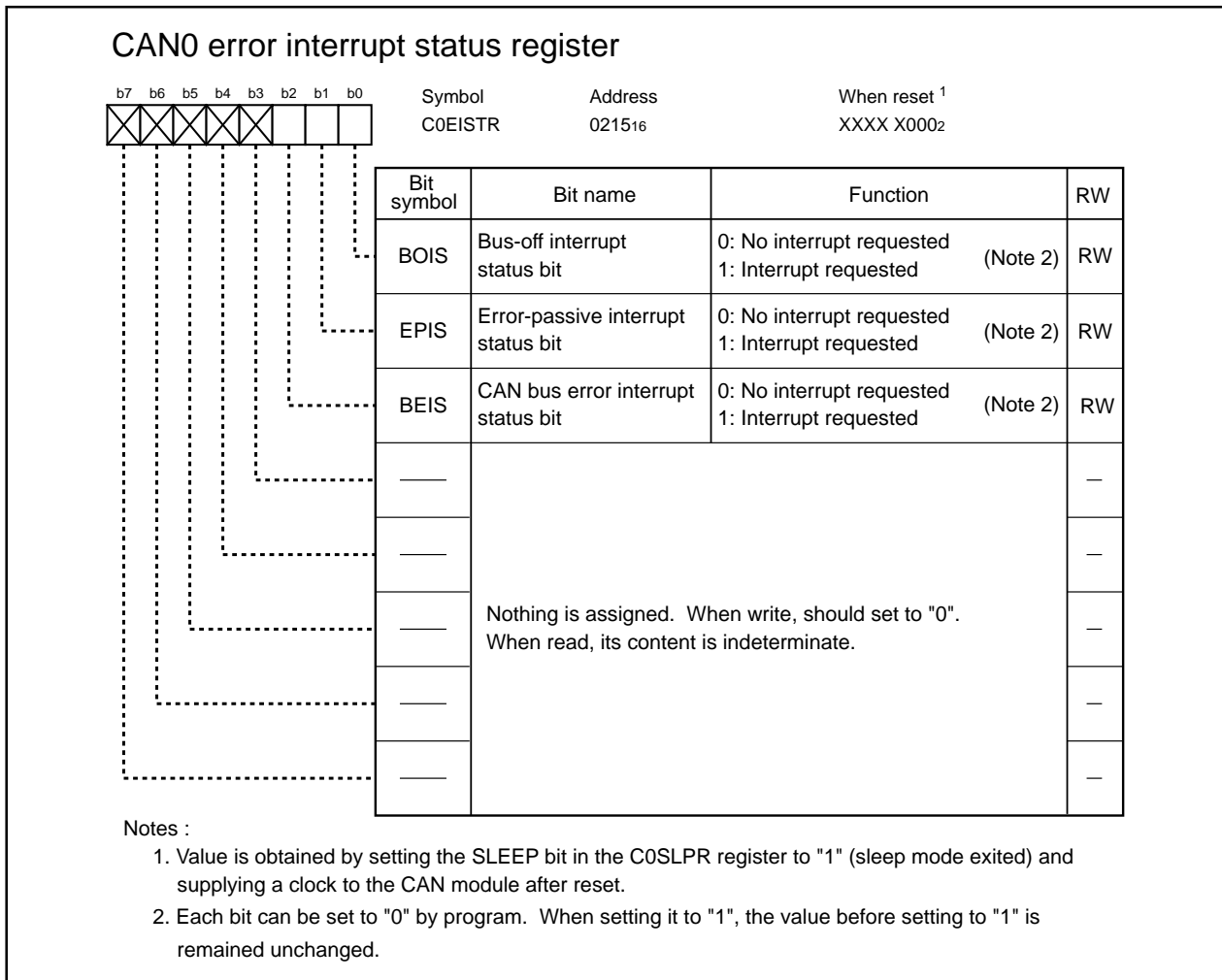
The EPIM bit determines whether an interrupt request is enabled or disabled when the CAN module is placed in an error passive state. When setting the EPIM bit to "1", an error passive interrupt request is enabled.

**BEIM Bit**

The BEIM bit determines whether an interrupt request is enabled or disabled when a CAN bus error occurs. When setting the BEIM bit to "1", a CAN bus error interrupt request is enabled. Refer to the paragraph "CAN interrupt" for details.

**CAN Module**

**14. CAN0 Error Interrupt Status Register (C0EISTR Register)**



**Figure 1.21.16. C0EISTR Register**

With the CAN interrupt, the C0EISTR register determines how an interrupt caused by occurring an error is generated. The BOIS, EPIS and BEIS bits are not automatically set to "0" (no interrupt request) even if an interrupt is acknowledged. These bits should be set to "0" by program.<sup>1</sup> Refer to the paragraph "CAN interrupt" for details.

**BOIS Bit**

The BOIS bit is set to "1" when the CAN module is in a bus-off state.

**EPIS Bit**

The EPIS bit is set to "1" when the CAN module is in an error passive state.

**BEIS Bit**

The BEIS bit is set to "1" when a CAN bus error is detected.

Notes :

- When setting any bits in the C0EISTR register to "0", the MOV instruction, instead of the bit clear instruction, should be used . Bits to remain unchanged should be set to "1".

e.g. To set the BOIS bit to "0"

Assembler language: MOV.B #006h, C0EISTR

C language: c0eistr = 0x06;

## 15. CAN0 Global Mask Register, CAN0 Local Mask Register A and CAN0 Local Mask Register B (C0GMRj (j=0 to 4), C0LMARj and C0LMBRj Registers)

The C0GMRj, C0LMARj and C0LMBRj registers are used with the acceptance filter.

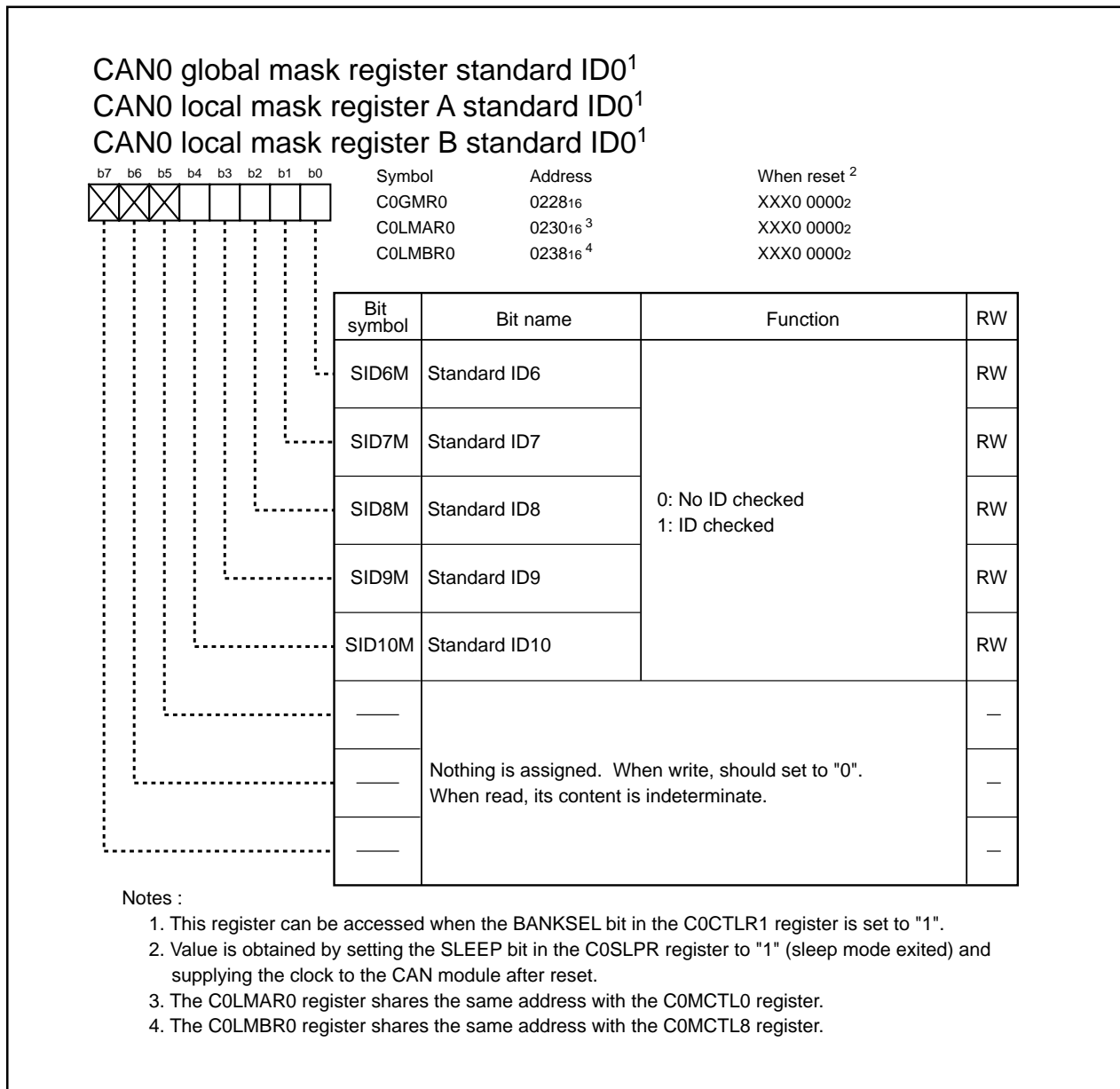
The C0GMRj register determines whether an identifier (ID) check for the message slots 0 to 13 is executed. The C0LMARj register determines whether an ID check for the message slot 14 is executed. The C0LMBRj register determines whether an ID check for the message slot 15 is executed.

- When bits in these registers are set to "0", each bit (ID bit) in the corresponding message slots i standard ID0 to 1 and the message slots i extended ID0 to 2 is masked while acceptance filtering. (The corresponding bit is assumed to have a matched ID.)
- When bits in these registers are set to "1", its corresponding ID bits are compared with a received ID. If the received ID matches an ID in the message slot i, the received data having the matched ID is stored into that slot.

Notes :

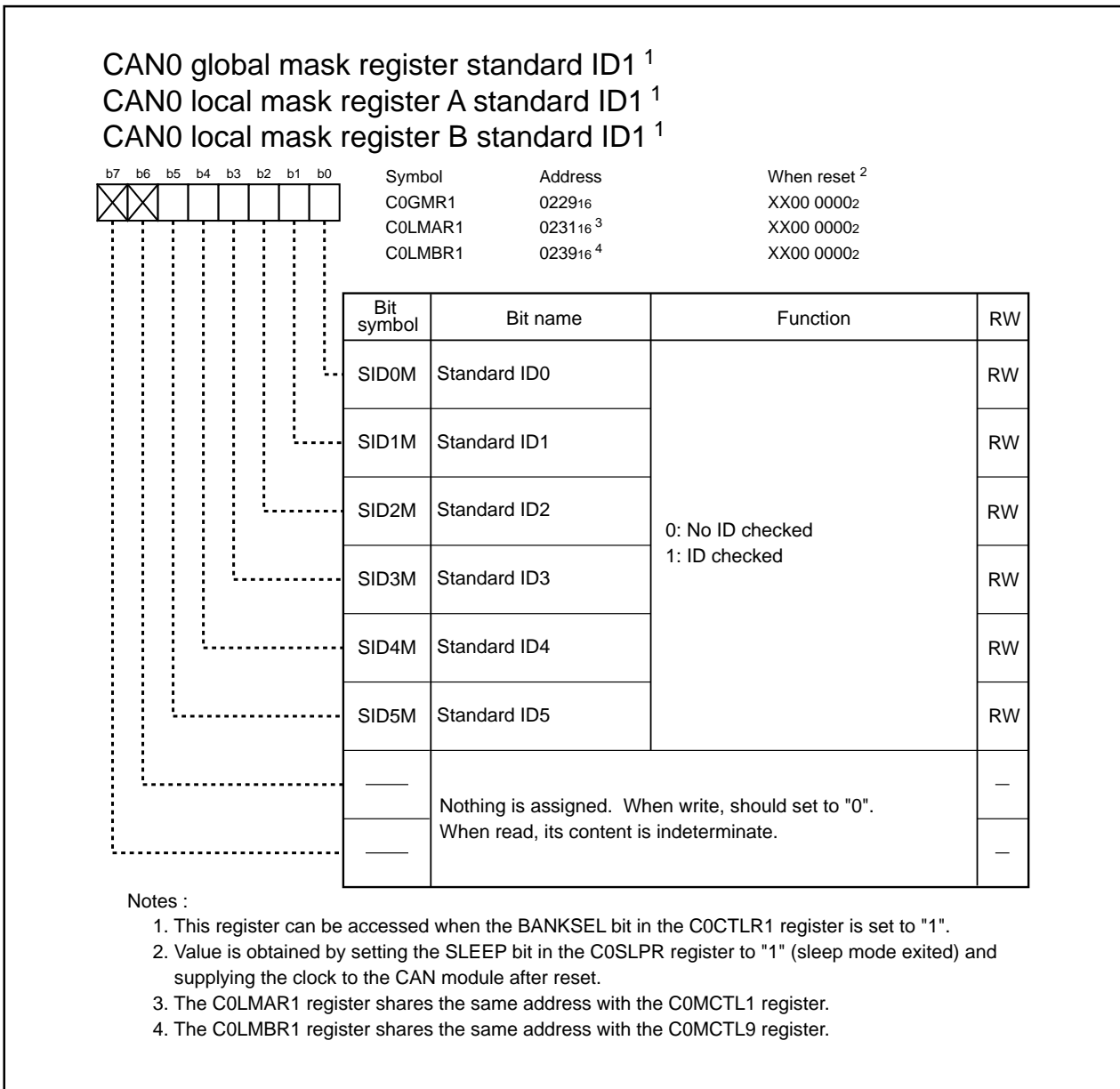
1. The C0MARj register should be changed when the message slots 0 to 13 have no receive request.
2. The C0MARj register should be changed when the message slot 14 has no receive request.
3. The C0MBRj register should be changed when the message slot 15 has no receive request.

**CAN Module**



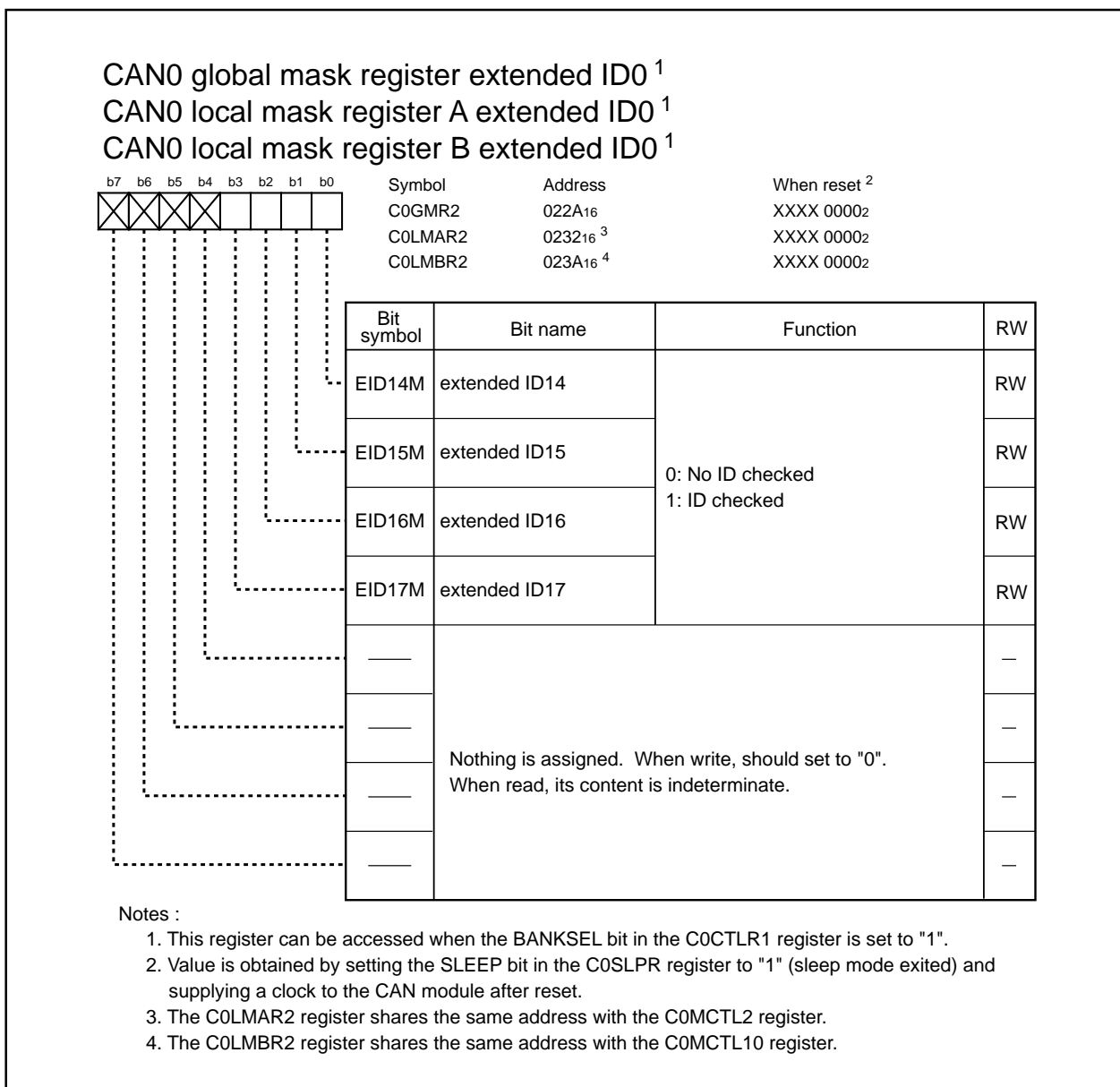
**Figure 1.21.17. C0GMR0, C0LMAR0 and C0LMBR0 Registers**

**CAN Module**



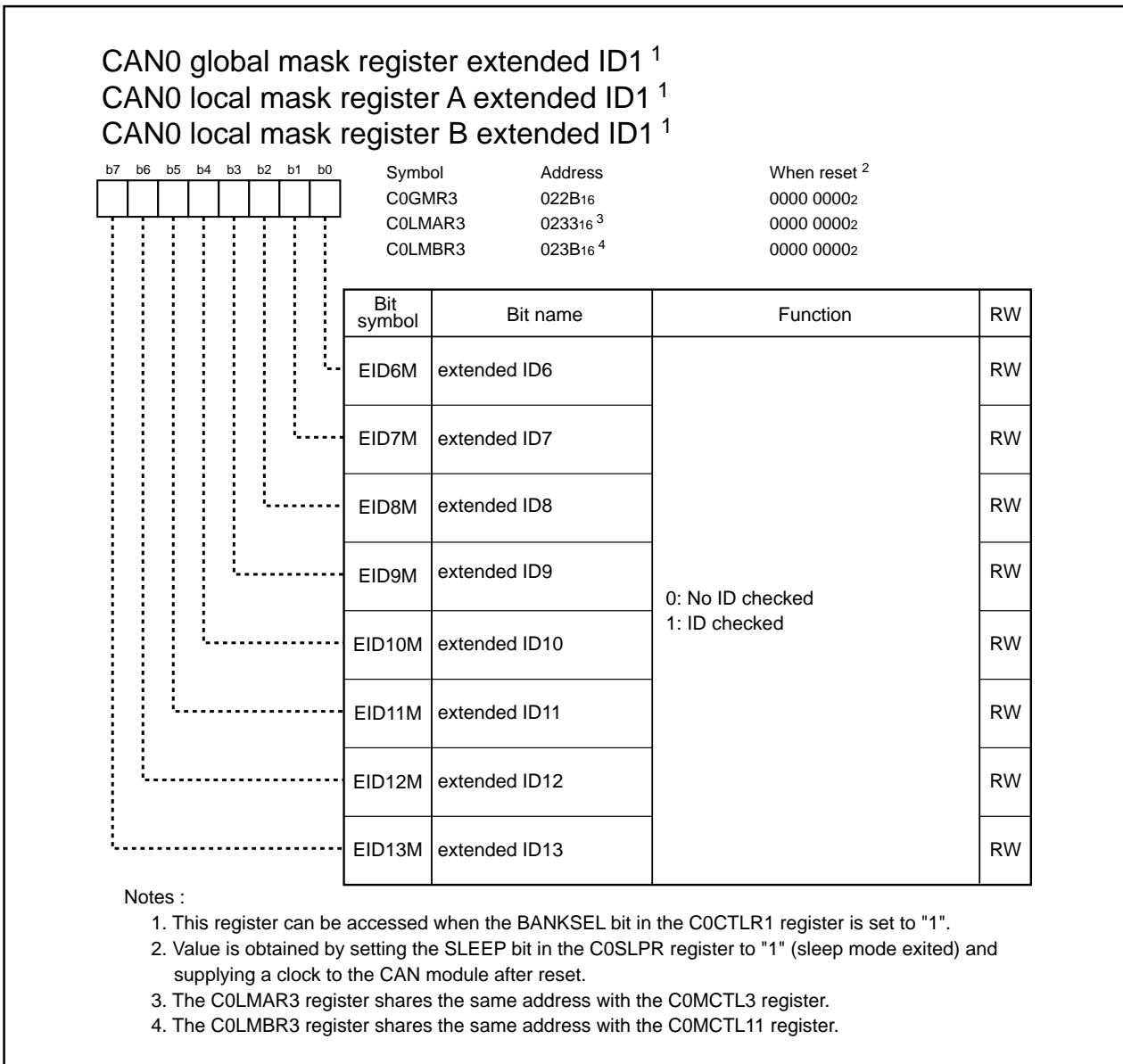
**Figure 1.21.18. C0GMR1, C0LMAR1 and C0LMBR1 Registers**

**CAN Module**



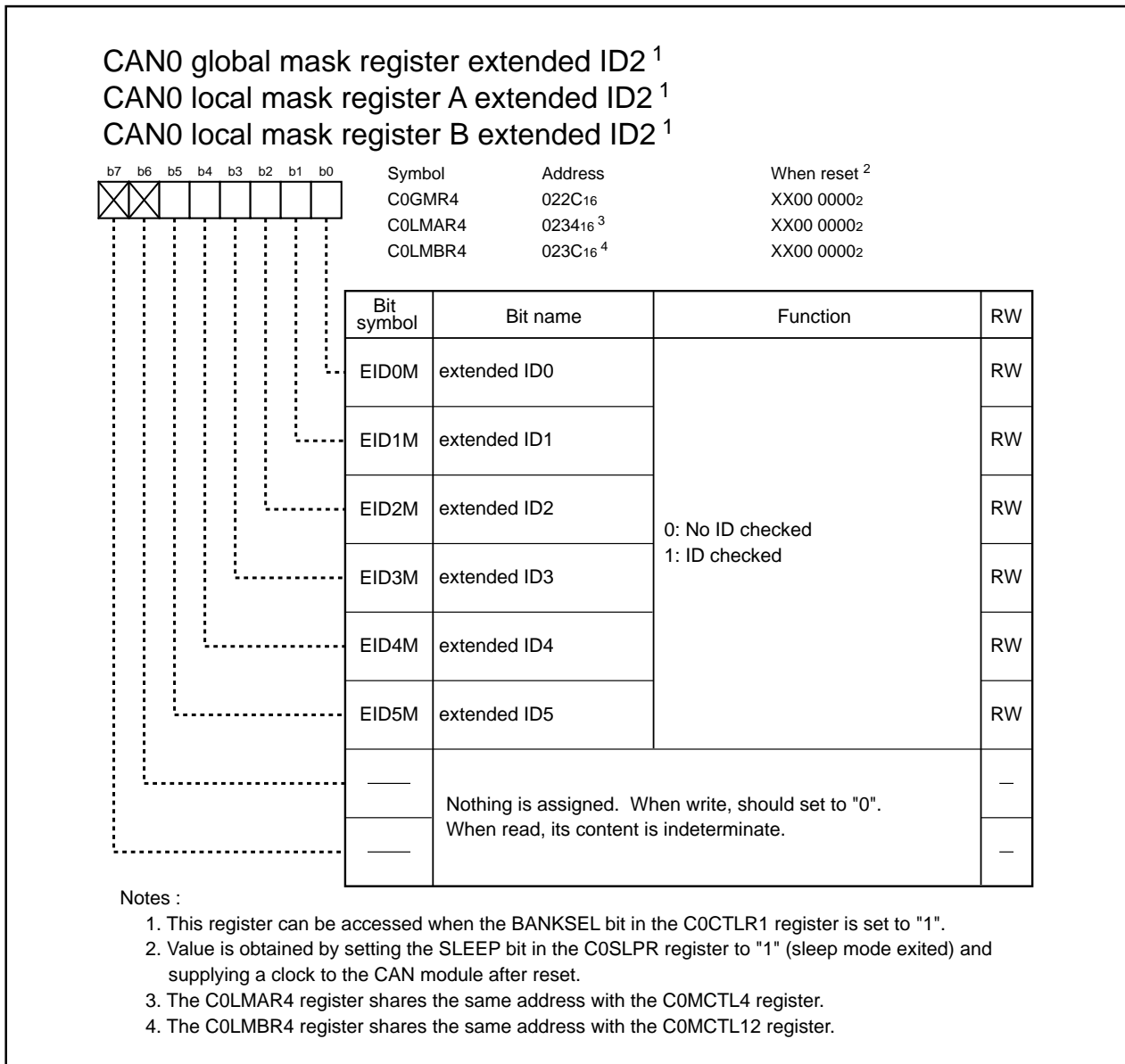
**Figure 1.21.19. C0GMR2, C0LMAR2 and C0LMBR2 Registers**

**CAN Module**



**Figure 1.21.20. C0GMR3, C0LMAR3 and C0LMBR3 Registers**

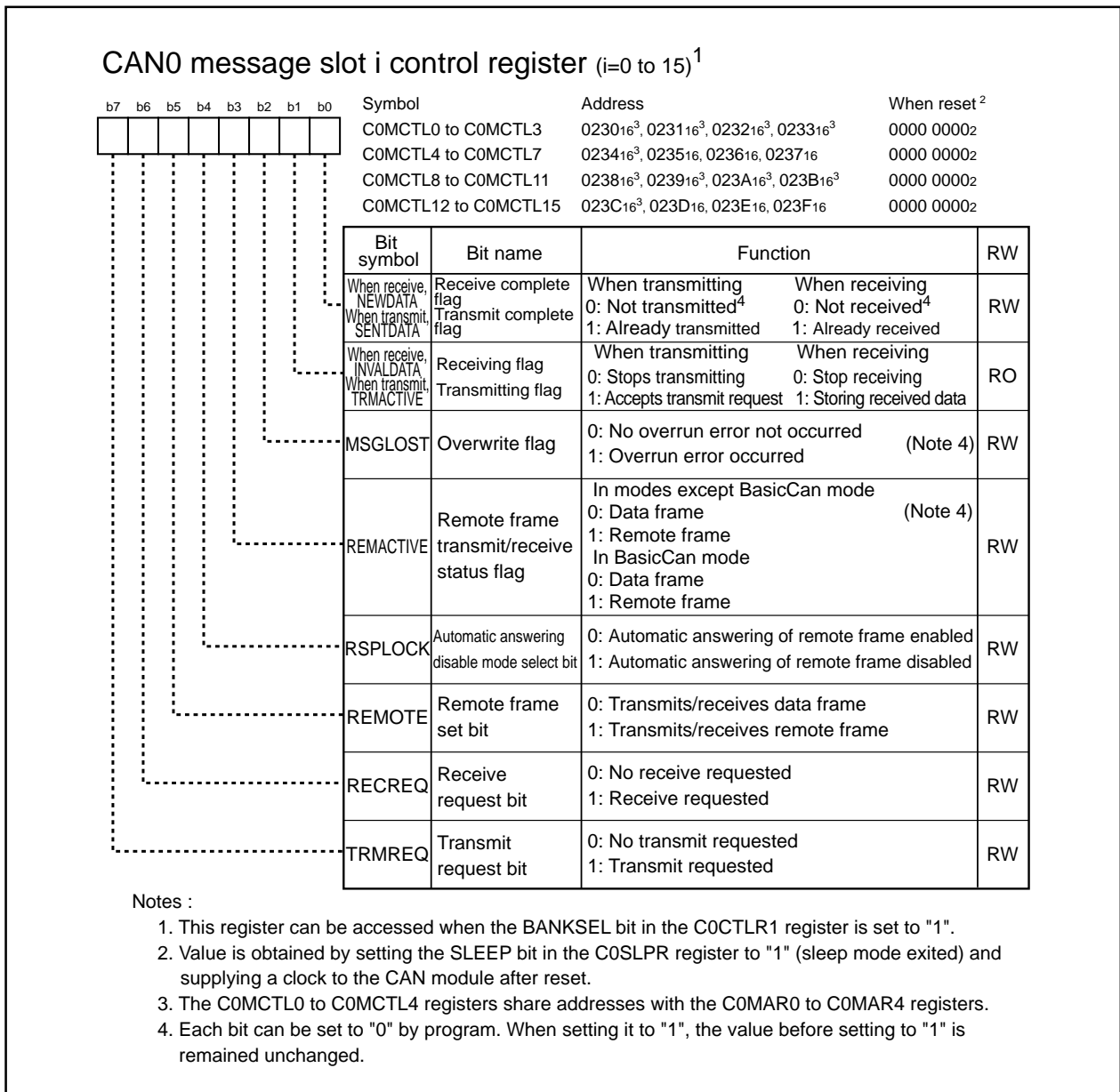
**CAN Module**



**Figure 1.21.21. C0GMR4, C0LMAR4 and C0LMBR4 Registers**

**CAN Module**

**16. CAN0 Message Slot i Control Register (COMCTLi Register)(i=0 to 15)**



**Figure 1.21.22. COMCTL0 to COMCTL15 Registers**

**Table 1.21.4 COMCTLi register(i = 0 to 15) Settings and Transmit/Receive Mode**

Settings for the COMCTLi register								Transmit/receive mode
TRMREC	RECREQ	REMOTE	RSPLOCK	REMACTIVE	MSGLOST	TRMACTIVE INVALIDDATA	SENTDATA NEWDATA	
0	0	-	-	0	0	-	0	No transmission/reception
0	1	0	0	0	0	-	0	Data frame received
0	1	1	1 or 0	0	0	-	0	Remote frame received (After remote frame is received, data frame is transmitted.)
1	0	0	0	0	0	-	0	Data frame transmitted
1	0	1	0	0	0	-	0	Remote frame transmitted (After remote frame is transmitted, data frame is received.)

## CAN Module

---

### SENTDATA/NEWDATA Bit

The SENTDATA/NEWDATA bit indicates that the CAN module has transmitted or received a CAN message. The SENTDATA/NEWDATA bit should be set to "0" (not transmitted or not received) by program before starting transmission and reception. The SENTDATA/NEWDATA bit is not automatically set to "0". When the TRMACTIVE/INVALIDDATA bit is set to "1" (accepts transmit request or storing received data), the SENTDATA/NEWDATA bit cannot be set to "0".

**SENTDATA** : The SENTDATA bit is set to "1" (already transmitted) when a transmission is completed in the message slot.

**NEWDATA** : The NEWDATA bit is set to "1" (already received) when a message to be stored into the message slot i is normally received in the message slot.

#### Notes :

1. The NEWDATA bit should be set to "0" before reading a received data from the message slot i. If the NEWDATA bit is set to "1" after reading, it indicates that new received data has been stored into the message slot while reading and that the data contains an indeterminate value. In this case, the data with indeterminate value should be discarded and data be read after setting the NEWDATA bit to "0".
2. When transmitting and receiving a remote frame, the SENTDATA/NEWDATA bit remains unchanged while a complete remote frame transmission or reception is completed. The SENTDATA/NEWDATA bit is set to "1" when a subsequent data frame transmission or reception is completed.

### TRMACTIVE/INVALIDDATA Bit

The TRMACTIVE/INVALIDDATA bit indicates that the CAN module is transmitting or receiving a message and accessing the message slot i. The TRMACTIVE/INVALIDDATA bit is set to "1" when the CAN module is accessing the message slot and to "0" when not accessing the message slot.

**TRMACTIVE** : The TRMACTIVE bit is set to "1" (accepts transmission request) when accepting a transmission request in the message slot. If the CAN module loses arbitration, CAN bus error occurs or the TRMACTIVE bit is set to "0" (stops transmitting) when CAN message transmission is completed.

**INVALIDDATA** : The INVALIDDATA bit is set to "1" (storing received data) when receiving a message and storing a received data into the message slot. Data, which is read from the message slot while setting this bit to "1", is indeterminate.

### MSGLOST Bit

The MSGLOST bit is valid only when setting the receive message slots. The MSGLOST bit is set to "1" (overrun error occurred) when the message slot i is rewritten by a new received message with the NEWDATA bit set to "1" (already received).

The MSGLOST bit is not automatically set to "0". This bit should be set to "0" (no overrun error occurred) by program.

### REMACTIVE Bit

The REMACTIVE bit in the C0MCTL0 to C0MCTL13 registers and in the C0MCTL14 and C0MCTL15 registers have different functions.

- The C0MCTL0 to C0MCTL13 registers, the C0MCTL14 and C0MCTL15 registers when setting the STATE\_BASICCAN bit to "0" (except BasicCAN mode):

When setting the message slot i to transmit or receive a remote frame, the REMACTIVE bit is set to "1" (remote frame). The REMACTIVE bit is set to "0" (data frame) after a remote frame has been transmitted and received.

- The COMCTL14 and COMCTL15 registers when setting the STATE\_BASICCAN bit to "1" (BasicCAN mode):  
When the REMACTIVE bit is set to "0", it indicates that a message stored into the message slot is a data frame. When the REMACTIVE bit is set to "1", it also indicates a message stored into the message slot is a remote frame.

#### RSPLOCK Bit

The RSPLOCK bit is valid only when selecting a remote frame reception shown in Table 1.21.4. The RSPLOCK determines a remote frame processing after receiving.

When the RSPLOCK bit is set to "0" (automatic answering of remote frame enabled), the slot automatically changes to a transmit slot after receiving a remote frame. A message stored into the message slot is transmitted as a data frame.

When the RSPLOCK bit is set to "1" (automatic answering of remote frame disabled), the slot halts operating after receiving a remote frame.

The RSPLOCK bit should be set to "0" whenever selecting any transmit/receive modes other than the remote frame reception.

#### REMOTE Bit

The REMOTE bit selects a transmit/receive mode shown in Table 1.21.4. The REMOTE bit should be set to "0" to transmit and receive a data frame. The REMOTE bit should be set to "1" to transmit and receive a remote frame.

If transmitting and receiving a remote frame, the following occurs:

- Transmitting a remote frame

A message stored into the message slot *i* is transmitted as a remote frame. The slot automatically changes to the message slot to receive a data frame after transmitting.

If a data frame is received before transmitting a remote frame, the data frame is stored into the message slot *i*. The remote frame is not transmitted.

- Receiving a remote frame

The message slot receives a remote frame. The RSPLOCK bit determines how to process a received remote frame.

#### RECREQ Bit

The RECREQ bit selects a transmit/receive mode shown in Table 1.21.4. The RECREQ bit should be set to "1" (receive requested) when receiving a data frame or a remote frame. The RECREQ bit should be set to "0" (no receive requested) when transmitting a data frame or a remote frame.

When automatically transmitting a data frame after receiving a remote frame, the RECREQ bit remains unchanged in "1". When transmitting a remote frame, the RECREQ bit should be set to "0". After transmitting a remote frame, a data frame is automatically received when the RECREQ bit remains unchanged in "0".

When setting the TRMREQ bit to "1" (transmit requested), avoid setting the RECREQ bit to "1" (receive requested).

#### TRMREQ Bit

The TRMREQ bit selects a transmit/receive mode shown in Table 1.21.4. The TRMREQ bit should be set to "1" (transmit requested) when transmitting a data frame or a remote frame.

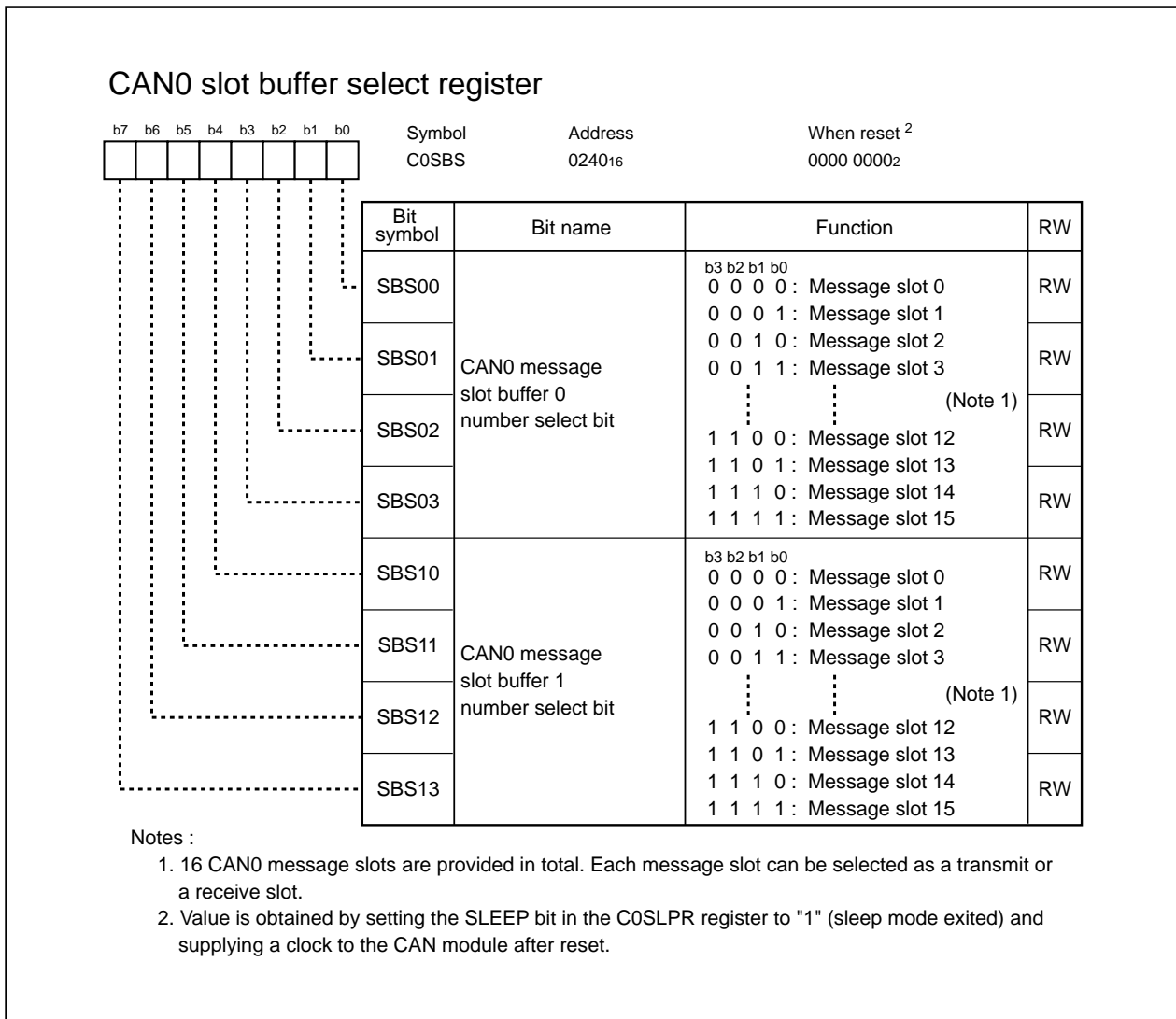
The TRMREQ bit should be set to "0" (no transmit requested) when receiving a data frame or a remote frame.

When automatically receiving a data frame after transmitting a remote frame, the TRMREQ bit remains unchanged in "1". When receiving a remote frame, the TRMREQ bit should be set to "0". After receiving a remote frame, a data frame is automatically transmitted when the TRMREQ bit remains unchanged in "0".

If the RECREQ bit is set to "1" (receive requested), avoid setting the TRMREQ bit to "1" (transmit requested).

**CAN Module**

**17. CAN0 Slot Buffer Select Register (C0SBS Register)**



**Figure 1.21.23. C0SBS Register**

**SBS03 to SBS00 Bits**

Assume a number selected in the SBS03 to SBS00 bits is i, the message slot i is allocated to the message slot buffer 0. The message slot i can be accessed via the allocated addresses (01E0<sub>16</sub> to 01EF<sub>16</sub>).

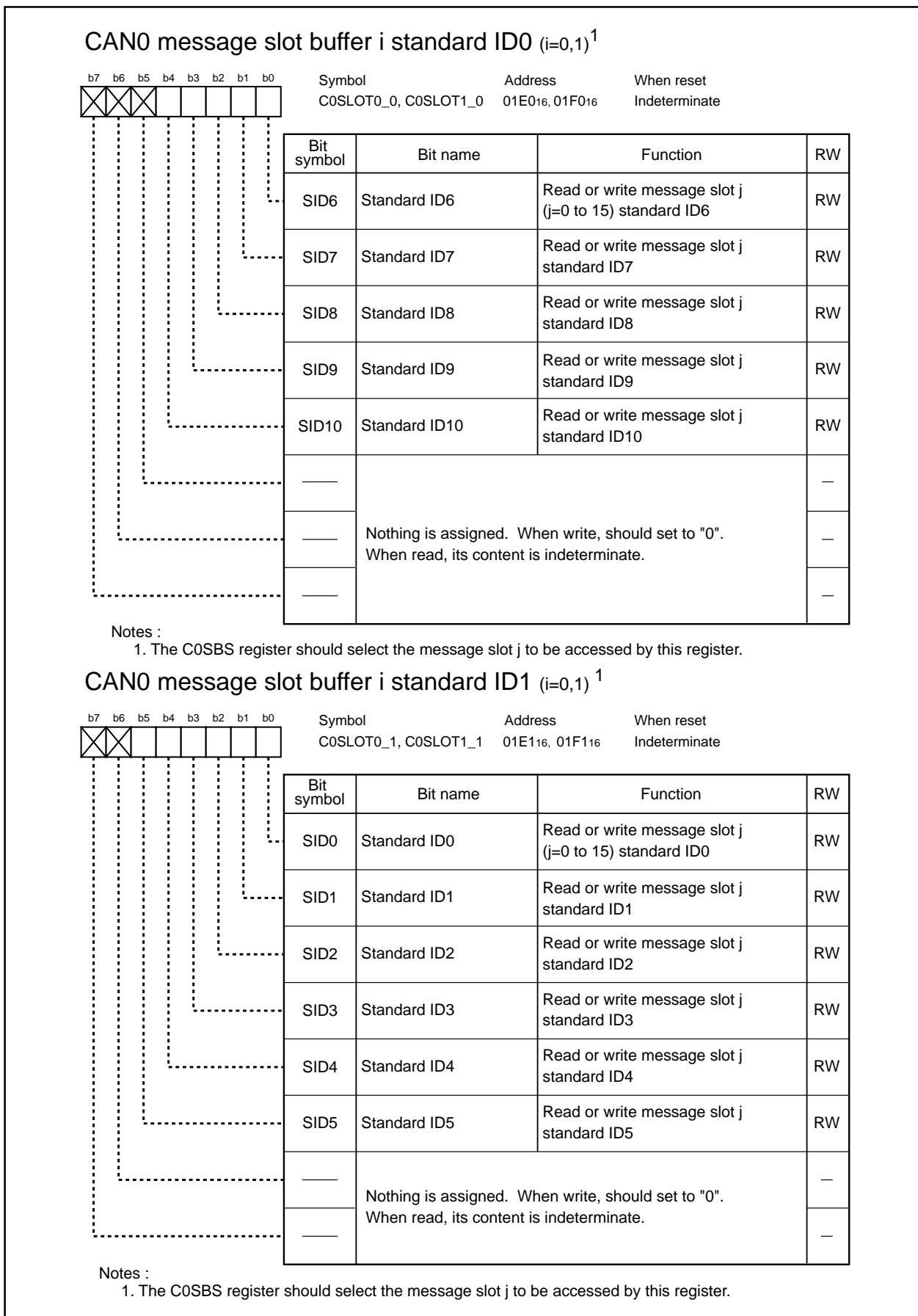
**SBS13 to SBS10 Bits**

Assume a number selected in the SBS13 to SBS10 bits is i, the message slot i is allocated to the message slot buffer 1. The message slot i can be accessed via the allocated addresses (01F0<sub>16</sub> to 01FF<sub>16</sub>).

**18. Message Slot Buffer**

The message slot selected by the C0SBS register can be read when reading the message slot buffer. A message can be written in the message slot selected by the C0SBS register when writing a message in the message slot buffer.

**CAN Module**



**Figure 1.21.24. C0SLOT0\_0, C0SLOT1\_0, C0SLOT0\_1 and C0SLOT1\_1 Registers**

CAN Module

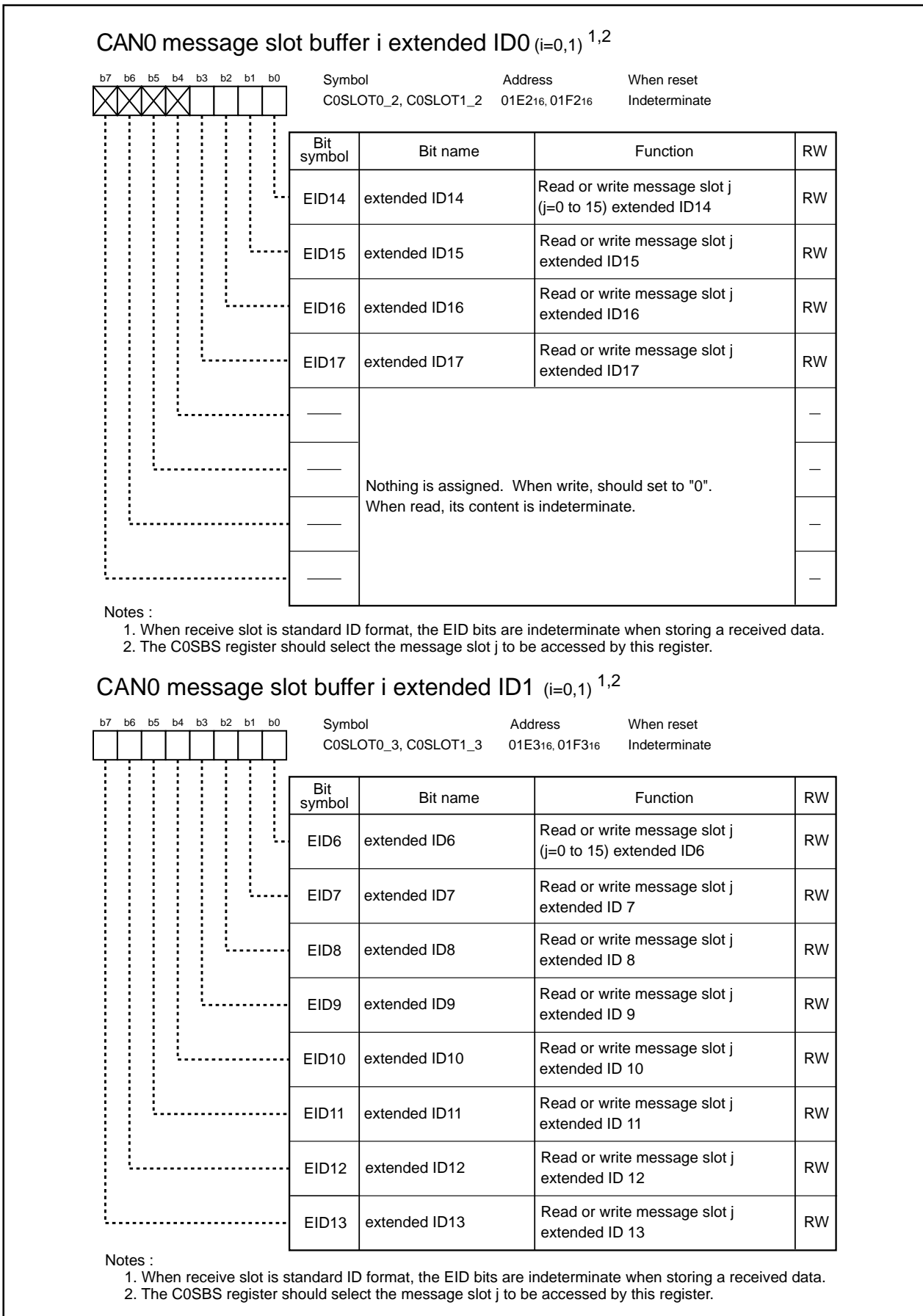
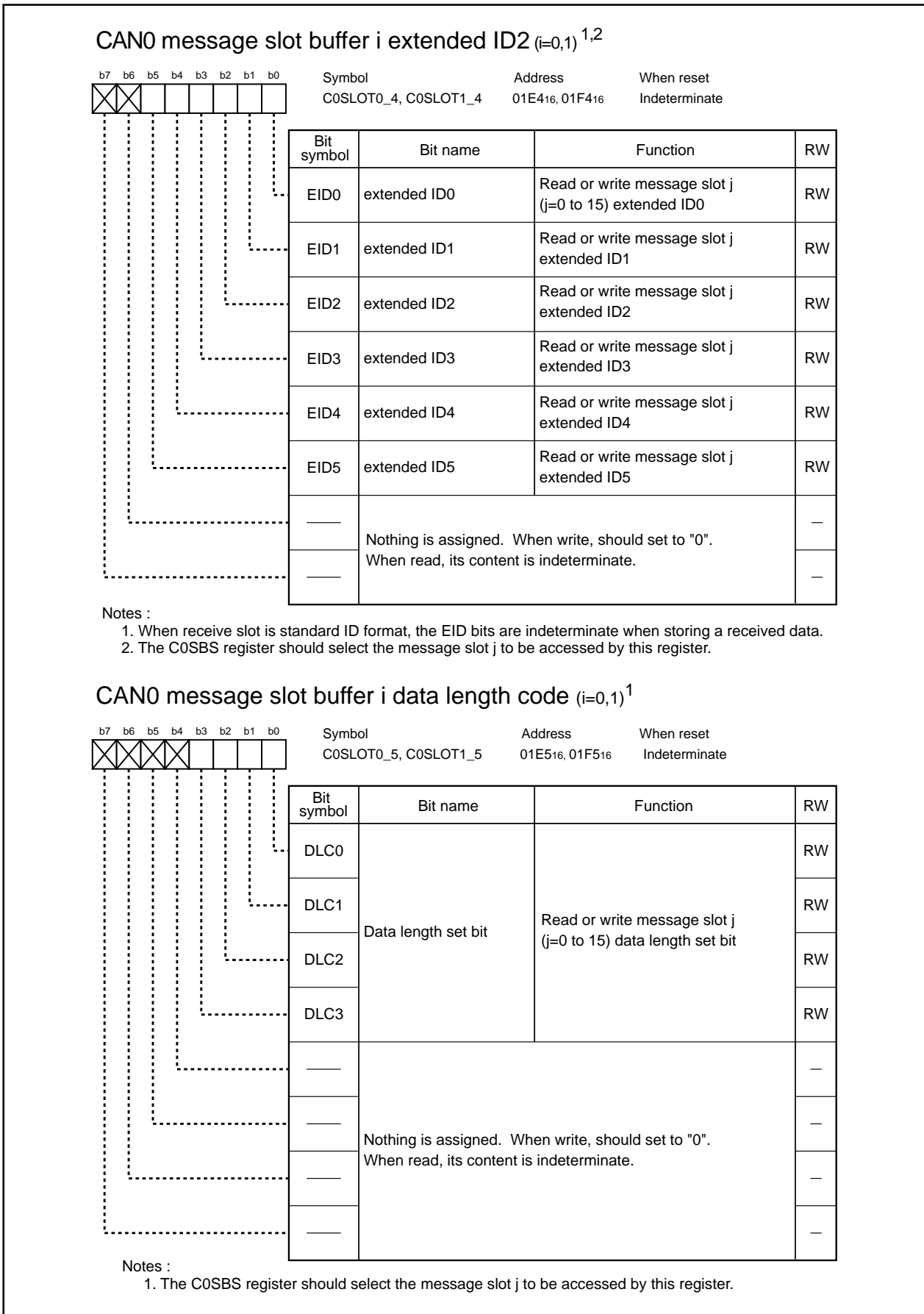


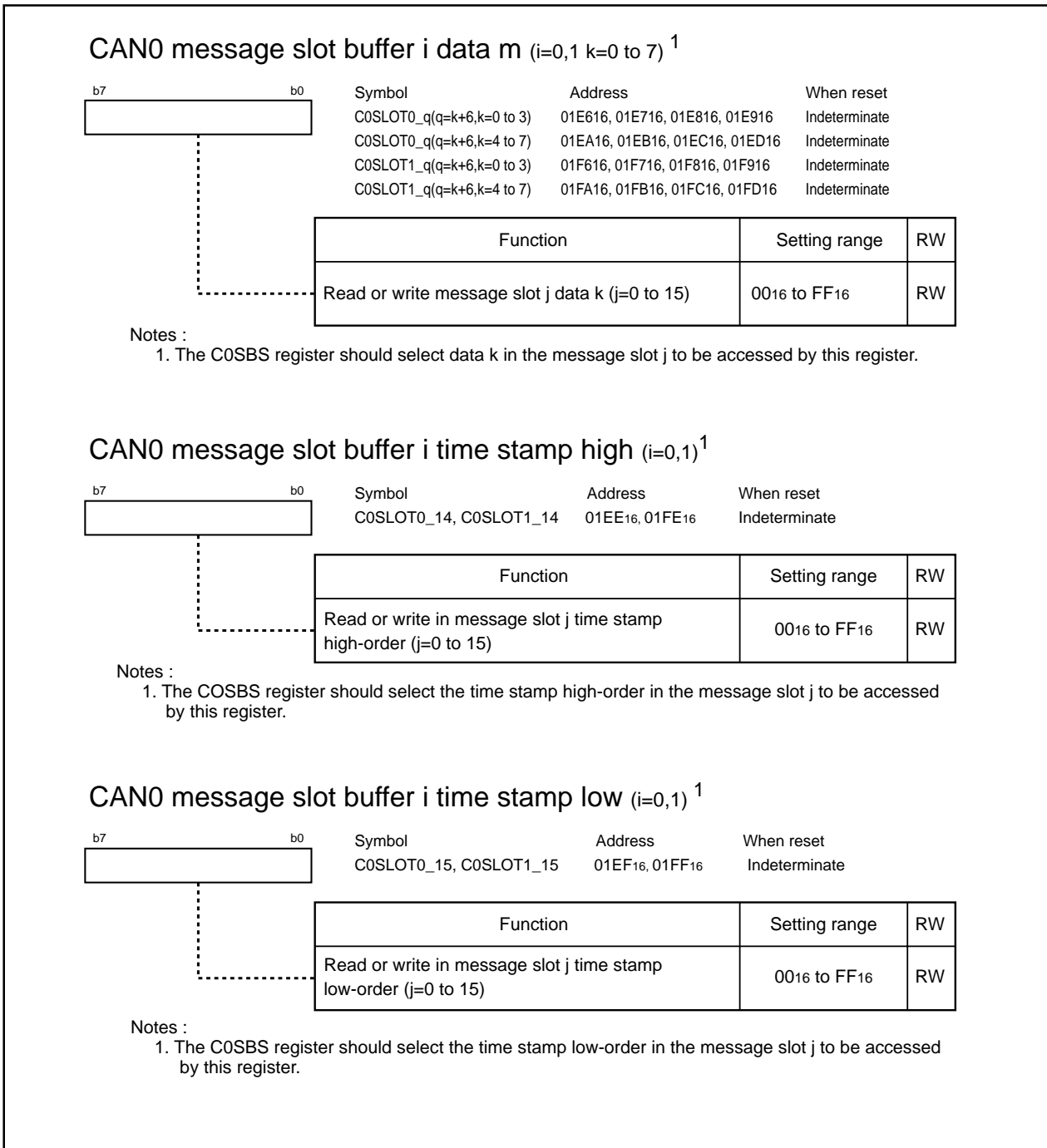
Figure 1.21.25. C0SLOT0\_2, C0SLOT1\_2, C0SLOT0\_3 and C0SLOT1\_3 Registers

**CAN Module**



**Figure 1.21.26. C0SLOT0\_4, C0SLOT1\_4, C0SLOT0\_5 and C0SLOT1\_5 Registers**

**CAN Module**



**Figure 1.21.27. C0SLOT0\_6 to C0SLOT0\_13, C0SLOT1\_6 to C0SLOT1\_13, C0SLOT0\_14, C0SLOT1\_14, C0SLOT0\_15 and C0SLOT1\_15 Registers**

CAN Module

### 19. CAN0 Acceptance Filter Support Register

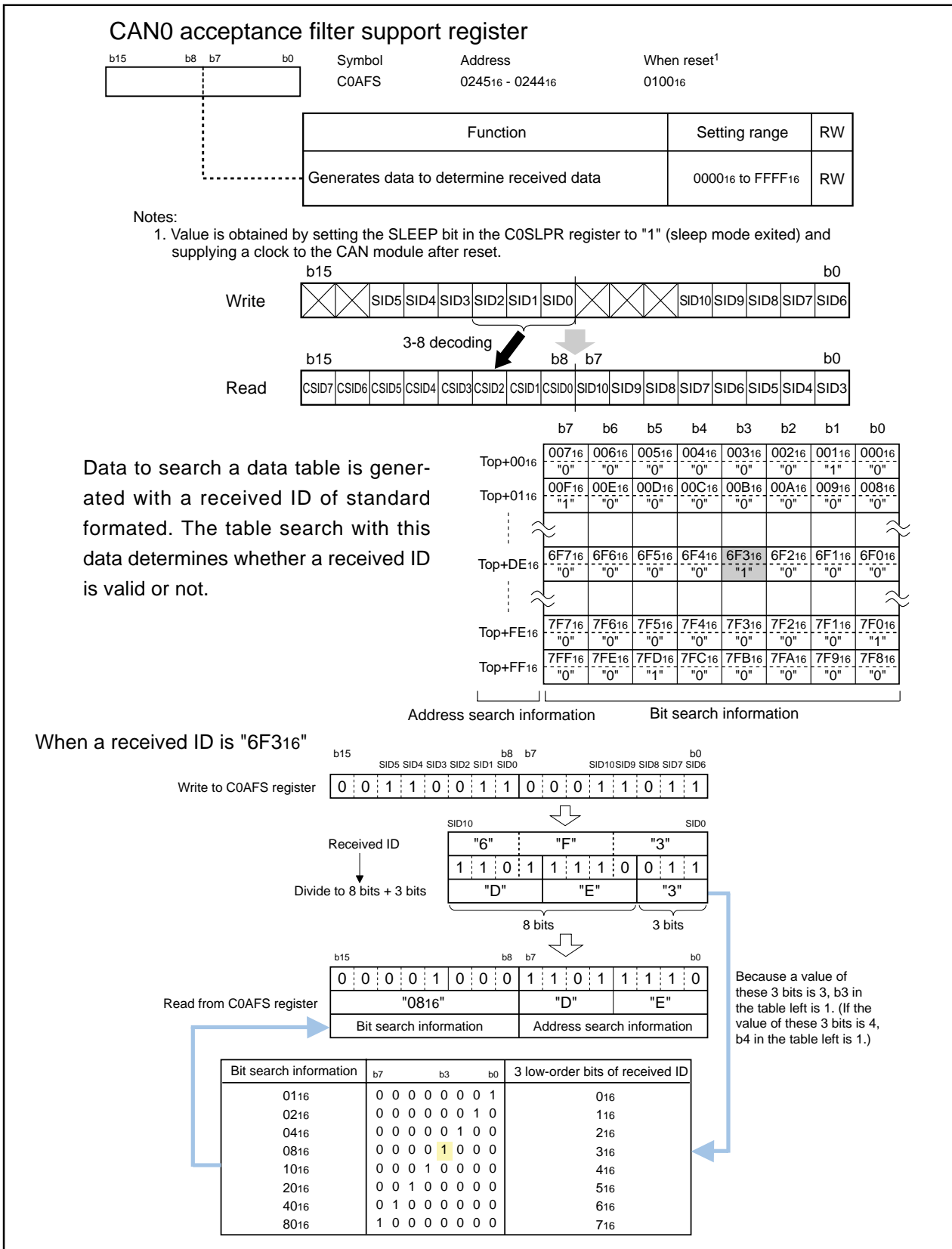


Figure 1.21.28. C0AFS Register

The table search determines whether a received ID is valid or not as soon as it can. This function is for a standard-formatted ID only.

## CAN Module

## Timing of CAN-Associated Registers

## 1. CAN Module Reset

Figure 1.21.29 shows an example of an operation timing when the CAN module is reset.

- (1) The CAN module can be reset when the RESET0 and RESET1 bits in the C0CTRL0 register are set to "1" (CAN module is reset) and the STATE\_RESET bit in the C0STR register is set to "1" (CAN module reset is completed).
- (2) Set necessary CAN-associated registers.
- (3) CAN communication can be established when setting the RESET0 and RESET1 bits to "0" (CAN module is not reset) and STATE\_RESET bit to "0" (resetting).

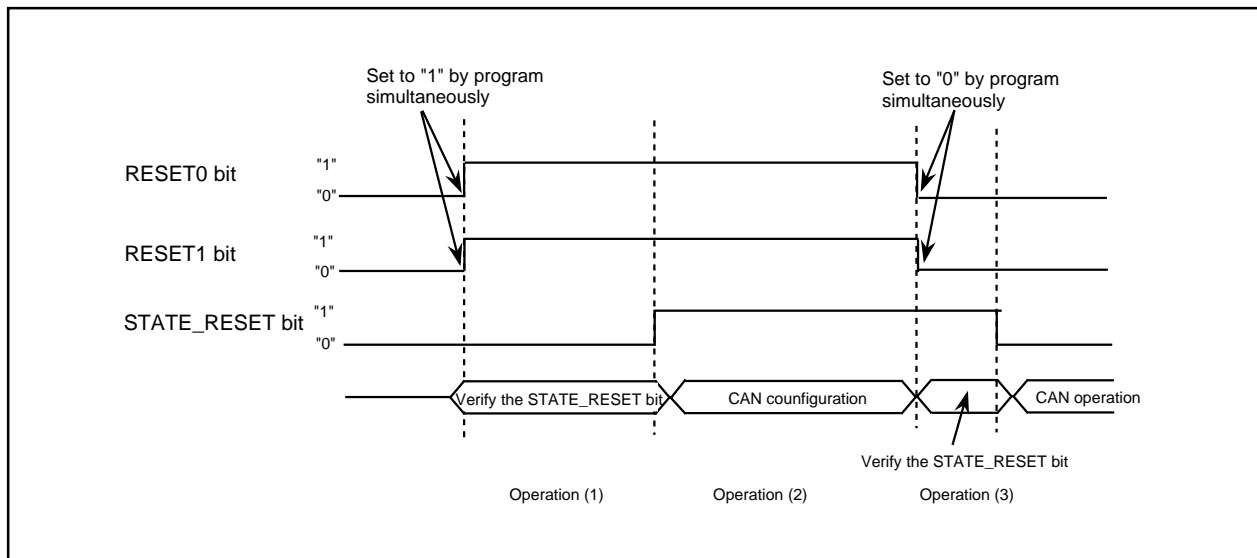


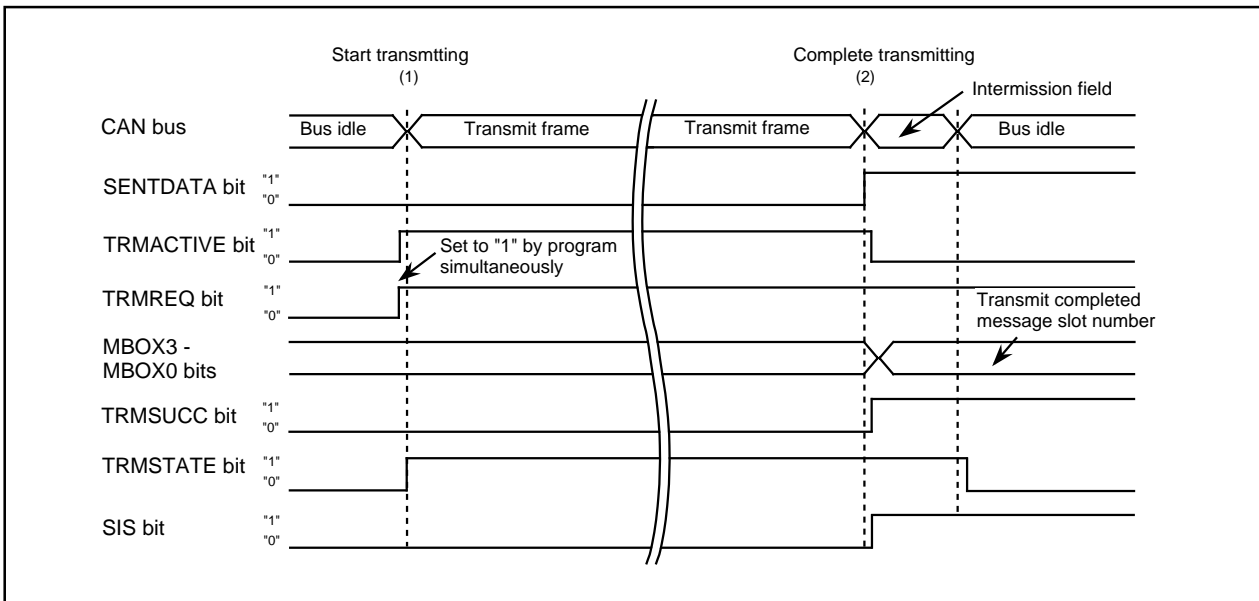
Figure 1.21.29 Operation Timing at CAN Module Reset

## 2. CAN Transmit Timing

Figure 1.21.30 shows an example of an operation timing when the CAN data frame is transmitted.

- (1) When setting the TRMREQ bit to "1" (transmit requested) in a bus idle state, the TRMACTIVE bit in the COMCTLi register is set to "1" (accepts transmit request) and the TRMSTATE bit in the C0STR register is set to "1" (transmitting). The CAN data frame is started transmitting.
- (2) After transmitting the CAN data frame, the SENTDATA bit in the COMCTLi register is set to "1" (transmit completed), the TRMSUCC bit in the C0STR register is set to "1" (transmit completed) and the SISi bit in the C0SISTR register to "1" (interrupt request). A message slot number transmitted to the MBOX3 to MBOX0 bits in the C0STR register is stored.

**CAN Module**



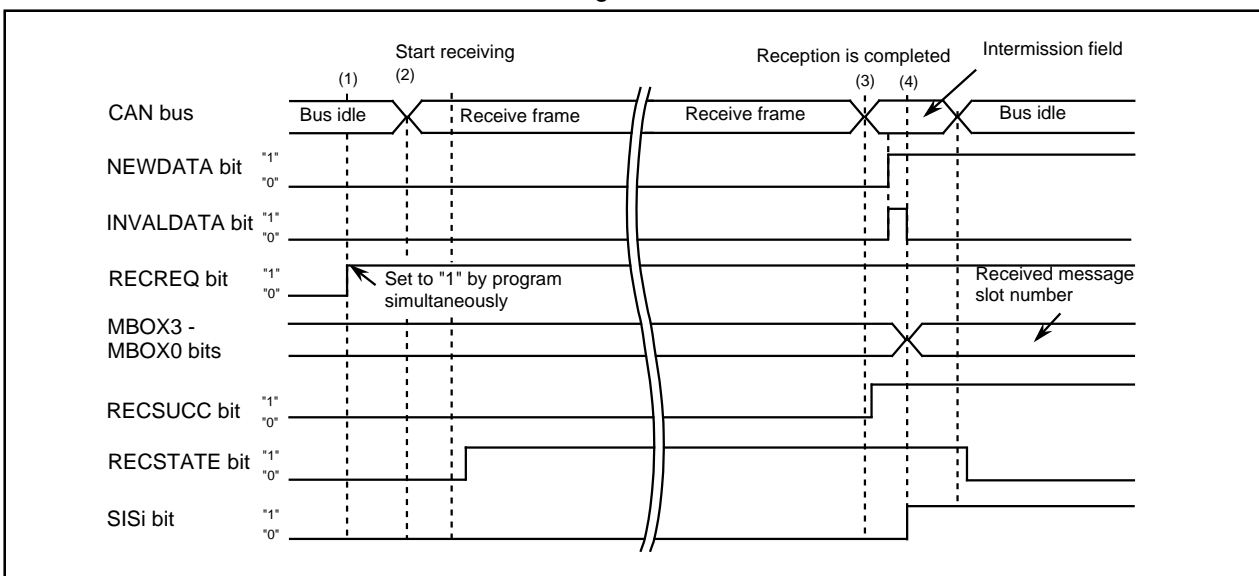
**Figure 1.21.30 Operation Timing at CAN data frame transmission**

**3. CAN Receive Timing**

Figure 1.21.31 shows an example of an operation timing when the CAN data frame is received.

- (1) When the RECREQ bit in the C0MCTLi register (i = 0 to 15) is set to "1" (transmit requested), the CAN data frame can be received anytime.
- (2) When starting a reception of the CAN data frame, the RECSTATE bit in the C0STR register is set to "1" (transmitting).
- (3) After receiving the CAN data frame, the INVALIDDTA bit in the C0MCTL0 register is set to "1" (store a received data), the NEWDATA bit in the C0MCTL0 register is set to "1" (receive completed) and the RECSUCC bit in the C0STR register is set to "1" (receive completed).
- (4) After writing to the message slot, The INVALIDDATA bit is set to "0" (stops receiving) and the SISi bit is set to "1" (interrupt request). Received slot numbers are stored into the MBOX3 to MBOX0 bits.

The MBOX3 to MBOX0 bits stores message slot numbers received.



**Figure 1.21.31 Operation Timing at CAN data frame reception**

## CAN Module

## 4. CAN bus Error Timing

Figure 1.21.32 shows an example of an operation timing when a CAN bus error occurs.

- (1) When detecting a CAN bus error, the STATE\_BUSERROR bit in the C0STR register is set to "1", (error occurred) and the BEIS bit in the C0EISTR register is set to "1" (interrupt request). The error frame is started transmitting.

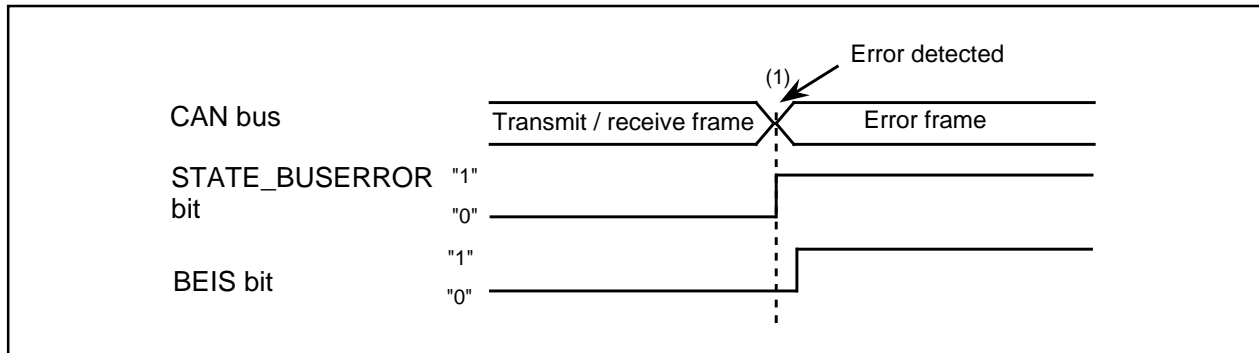


Figure 1.21.32. Operation timing when a CAN bus error occurred

## CAN Interrupts

CAN<sub>j</sub> interrupts (j=0 to 2) are provided as the CAN interrupt. Figure 1.21.33 shows a block diagram of the CAN interrupts.

The followings cause the CAN-associated interrupt request to be generated.

- CAN0 slot i transmission completed (i=0 to 15)
- CAN0 slot i reception completed
- CAN0 slot i bus error detection
- CAN0 slot i error-passive transaction
- CAN0 slot i bus-off transaction

If the CAN-associated interrupt is generated by the above interrupt request(s), a corresponding bit in the C0SISTR register is set to "1" (interrupt request) when CAN0 transmission or reception is completed. A corresponding bit in the C0EISTR register is set to "1" (interrupt request) with CAN0 bus error detection, CAN0 error-passive transaction or CAN0 bus-off transaction. When a corresponding bit in the C0SISTR or C0EISTR is set to "1" and a corresponding bit in the C0SIMKR or C0EIMKR is set to "1", the CAN0 interrupt request signal is set to "1".

When the CAN0 interrupt request signal change "0" to "1" all CAN<sub>j</sub>R bits in the IIO9IR to IIO11IR registers are set to "1" (interrupt request).

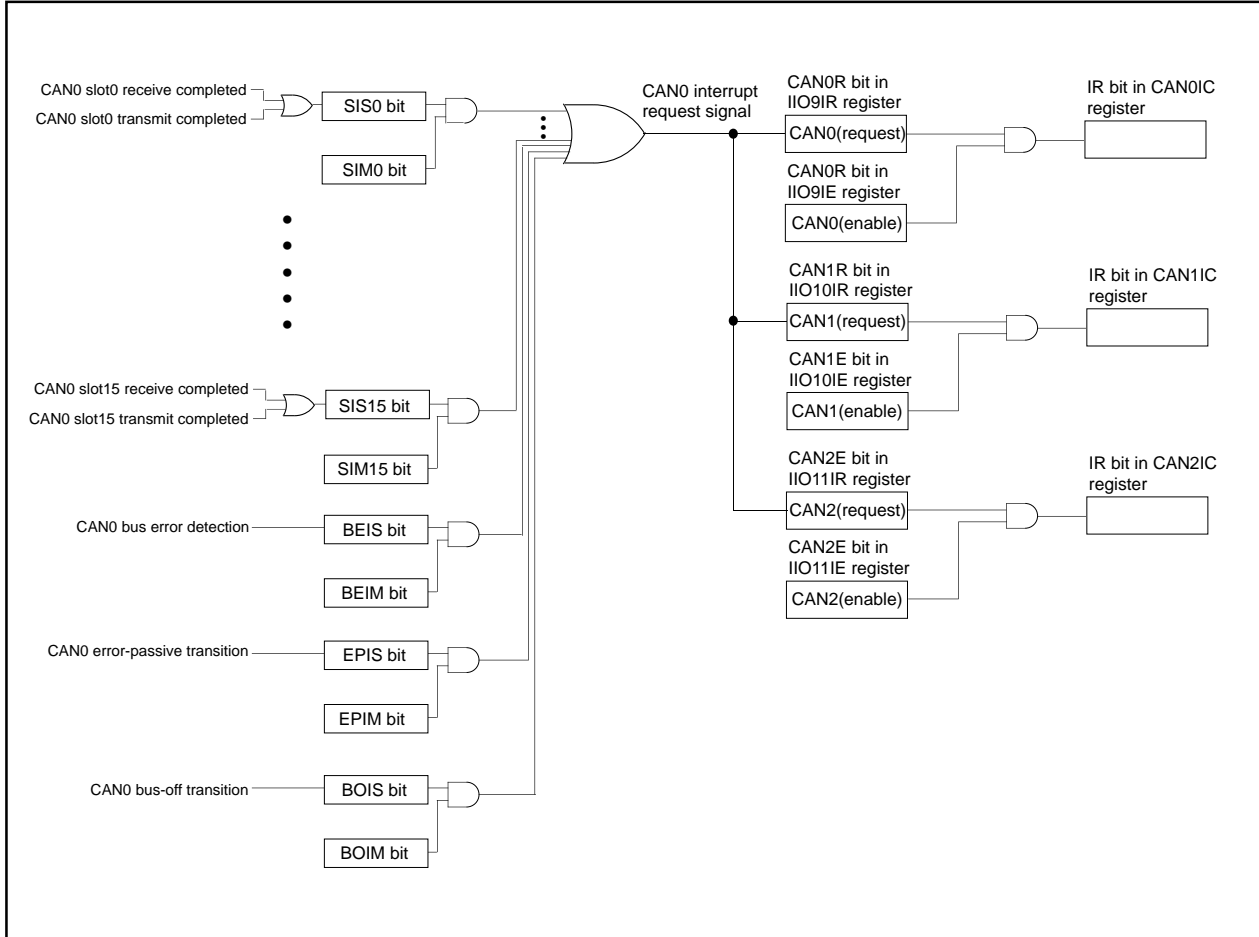
If at least one of the CAN<sub>j</sub>E bits in the IIO9IR to IIO11IR registers is set to "1", the IR bit in the corresponding CAN<sub>j</sub>IC register is set to "1". If another interrupt request causes a corresponding bit in the C0SISTR or C0EISTR to be set to "1" and a corresponding bit in the C0SIMKR or C0EIMKR to be set to "1", the CAN0 interrupt request signal remains unchanged in "1". The CAN<sub>j</sub>R and IR bits also remain unchanged.

Bits in the C0SISTR or C0EISTR register and CAN<sub>j</sub>R bits in the IIO9IR to IIO11IR registers are not set to "0" automatically regardless of an interrupt acknowledgment. These bits should be set to "0" by program.

The IIO9IR to IIO11IR registers enable CAN<sub>j</sub>R bits to acknowledge an interrupt. The C0SIMKR or C0EIMKR register enables the C0SISTR or C0EISTR register to acknowledge an interrupt.

**CAN Module**

The CANj interrupts are acknowledged when both the CANjR bits and corresponding bits in the C0SISTR or C0EISTR register are set to "0". If these bits are left unchanged in "1", all interrupt requests generated are disabled.



**Figure 1.21.33. CAN Interrupts**

## Intelligent I/O

The intelligent I/O is a multi-functional I/O port for time measurement, waveform generation, clock synchronous serial I/O, clock asynchronous serial I/O (UART), IEBus<sup>1</sup> communications, HDLC data processing and more.

The intelligent I/O consists of 4 groups and each group has one 16-bit base timer for free-running operation, eight 16-bit registers for time measurement and waveform generation and two 8-bit shift registers (or one 16-bit register) for communications.

Table 1.22.1 lists functions and channels of the intelligent I/O.

Notes :

1. IEBus is a trademark of NEC Corporation.

**Table 1.22.1. Intelligent I/O Functions and Channels**

Function	Group 0	Group 1	Group 2	Group 3	Group 0,1 cascaded
Time measurement <sup>1</sup>	8 channels (3 channels) <sup>2</sup>	4 channels (2 channels)	Not Available	Not Available	8 channels (3 channels)
Digital filter	8 channels (3 channels)	4 channels (2 channels)			8 channels (3 channels)
Trigger input prescaler	2 channels	2 channels			2 channels
Trigger input gate	2 channels	2 channels			2 channels
Waveform generation <sup>1</sup>	4 channels (2 channels)	8 channels (3 channels)	8 channels (3 channels)	8 channels (2 channels)	8 channels (3 channels)
Single-phase waveform output	Available	Available	Available	Available	Available
Phase-delayed waveform output					
SR waveform output					
Bit modulation PWM output	Not Available	Not Available			Not Available
RTP output					
Parallel RTP output					
Communication	8 bits fixed		Variable	8 bits or 16 bits	Not Available
Clock synchronous serial I/O mode	Available		Available	Available	Not Available
UART mode			Not Available	Not Available	
HDLC data processing mode					
IE mode	Not Available		Available		

Notes :

1. The time measurement function share pins with the waveform generation function.
2. Channels for the 100-pin package is in parentheses.

The time measurement function and waveform generation function can be selected in each channel.

The communication function is available by a combination of multiple channels.

Figures 1.22.1 to 1.22.4 show block diagrams of the intelligent I/O group 0 to 3.

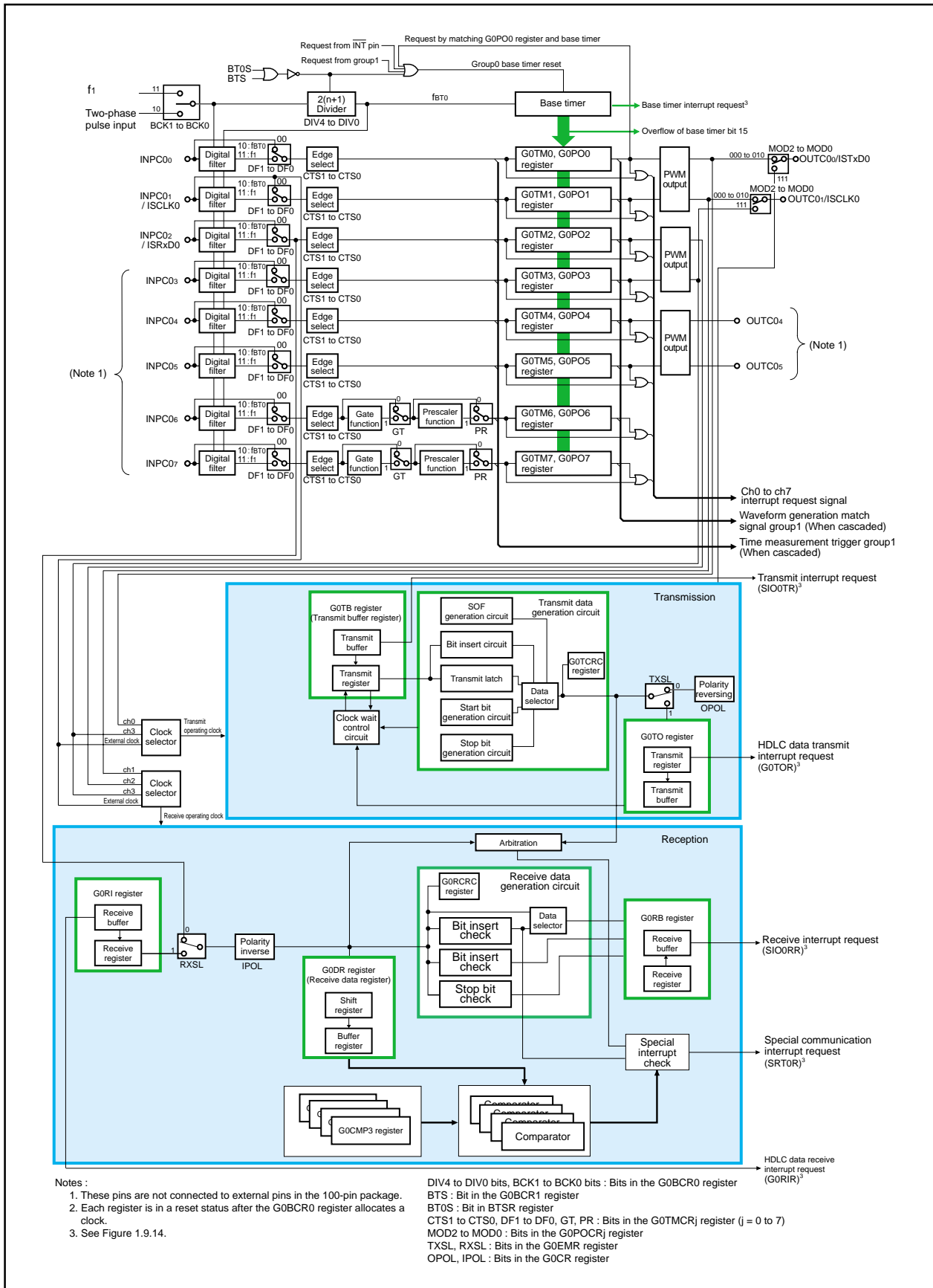


Figure 1.22.1. Intelligent I/O Group 0 Block Diagram

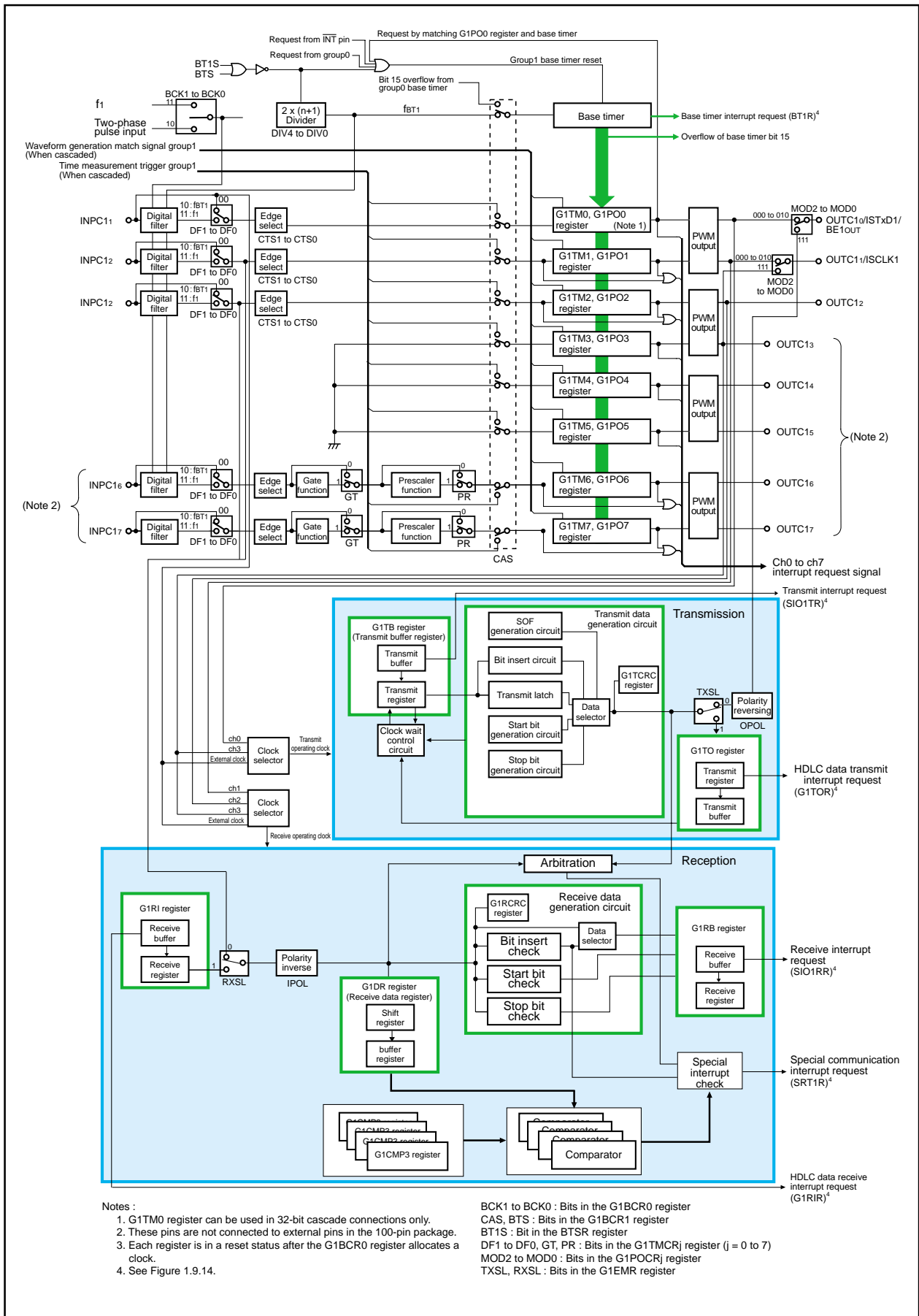


Figure 1.22.2. Intelligent I/O Group 1 Block Diagram

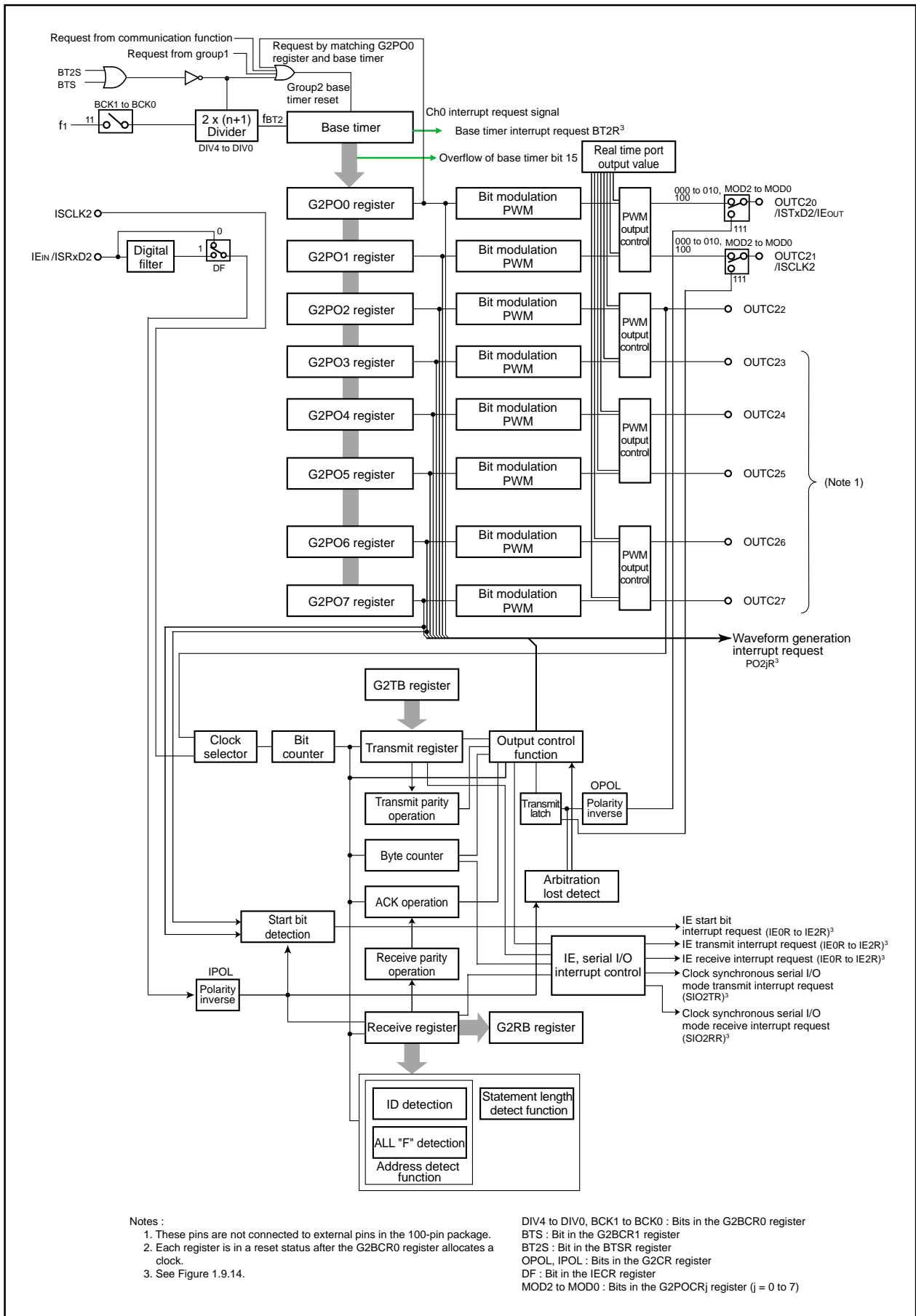


Figure 1.22.3. Intelligent I/O Group 2 Block Diagram

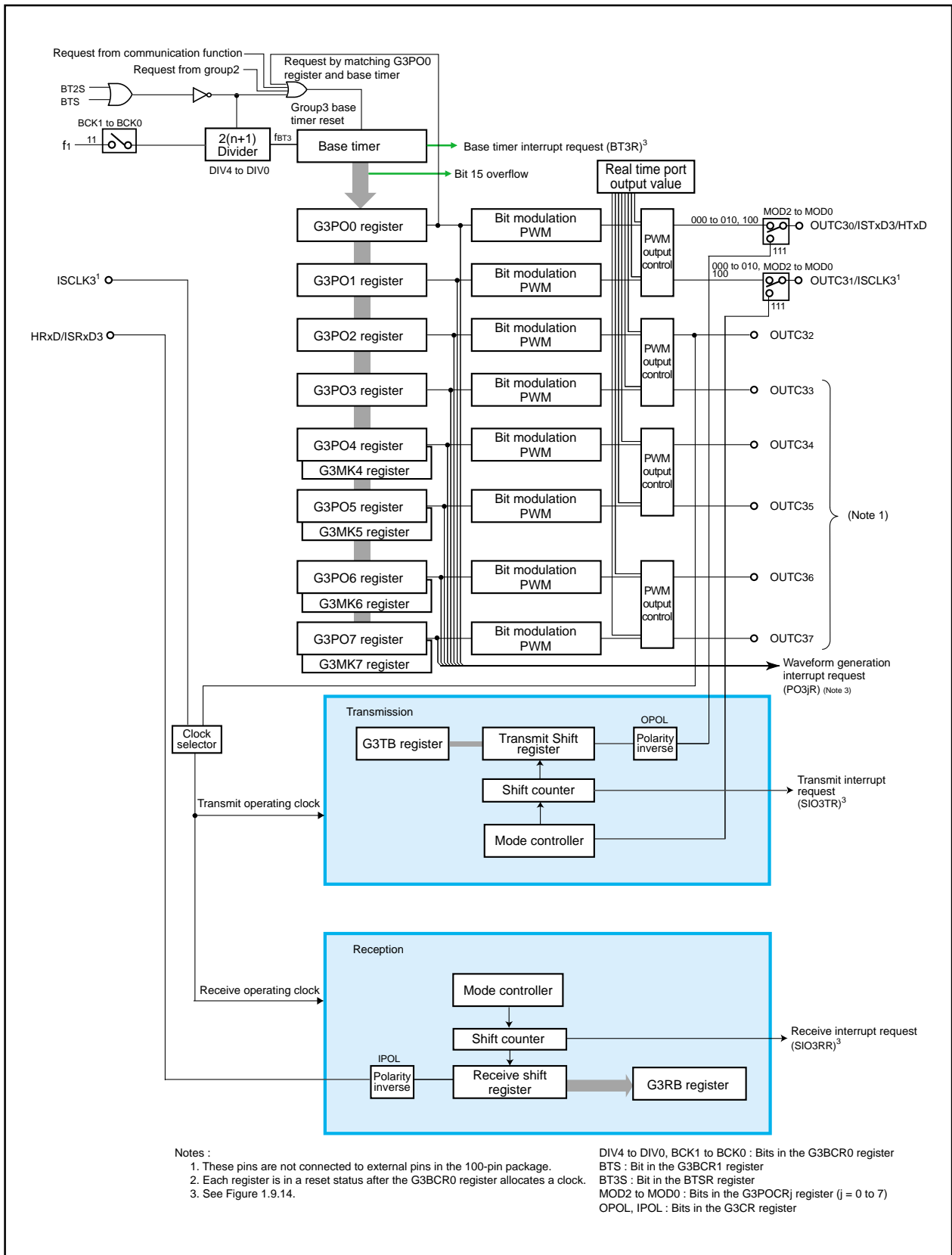


Figure 1.22.4. Intelligent I/O Group 3 Block Diagram

Figures 1.22.5 to 1.22.15 show registers associated with the intelligent I/O base timer, the time measurement function and waveform generation function. (For registers associated with the communication function, see Figures 1.22.32 to 1.22.38, 1.22.42 to 1.22.45 and 1.22.47 to 1.22.49.)

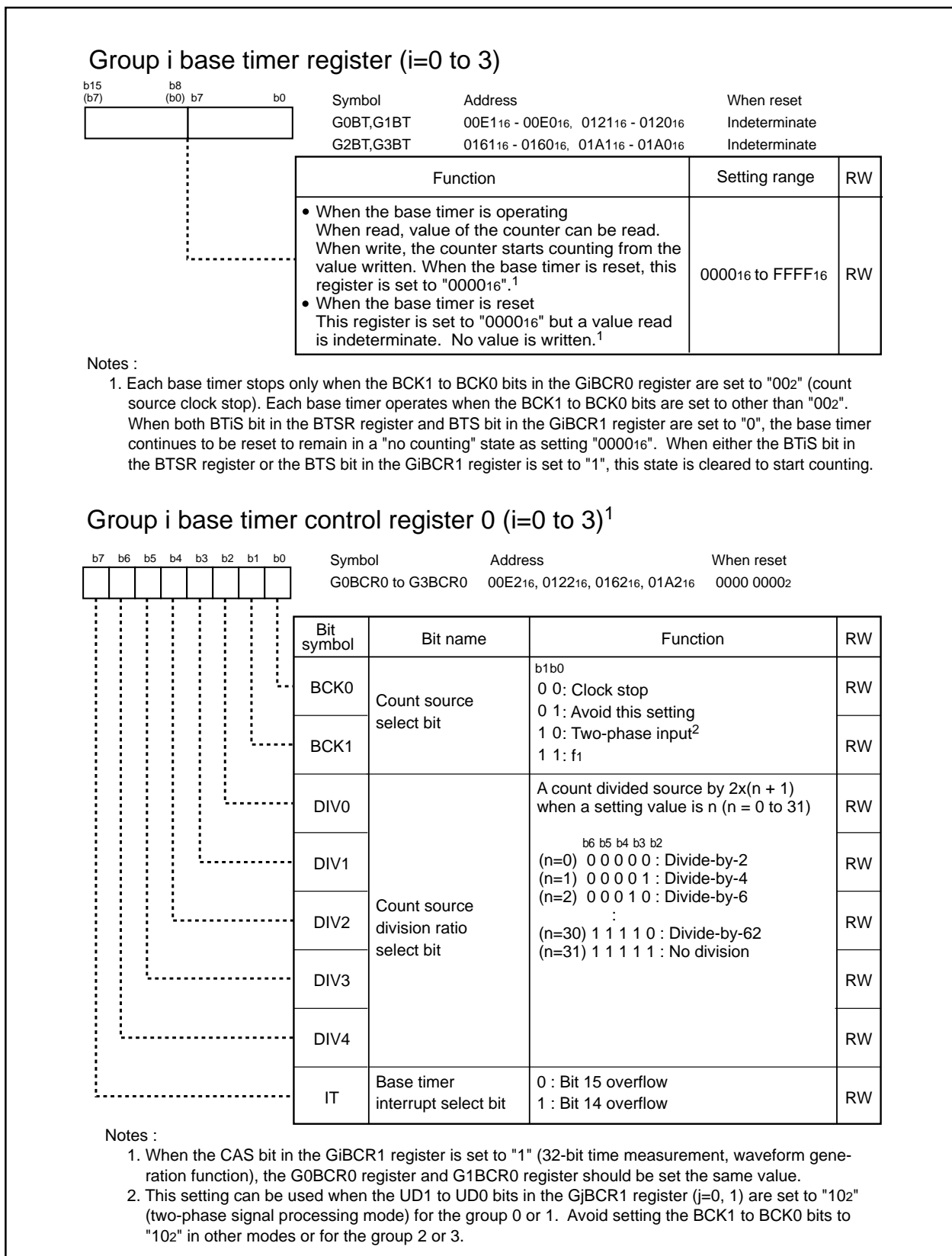


Figure 1.22.5. G0BT to G3BT Registers and G0BCR0 to G3BCR0 Registers

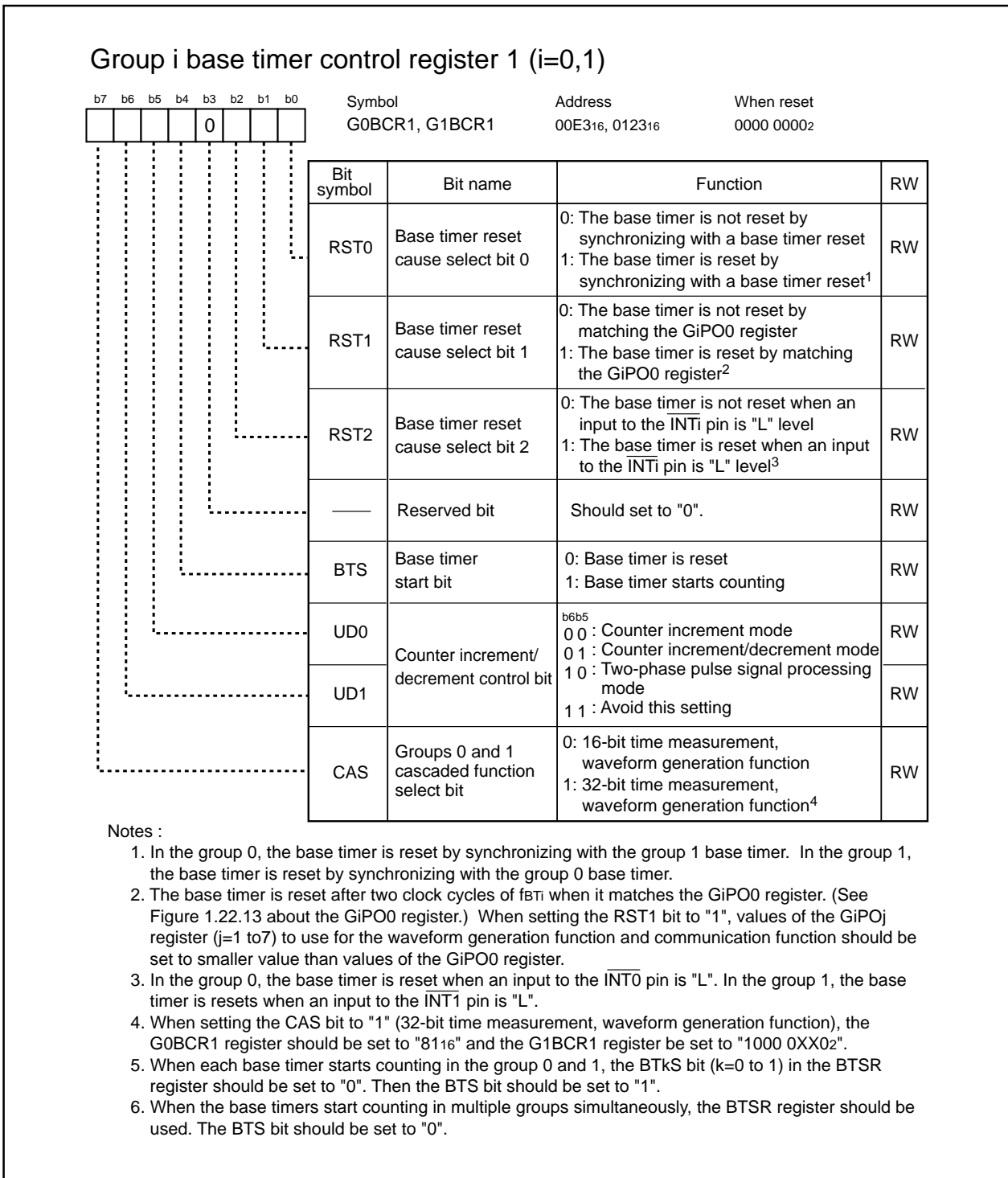


Figure 1.22.6. G0BCR1 and G1BCR1 Registers

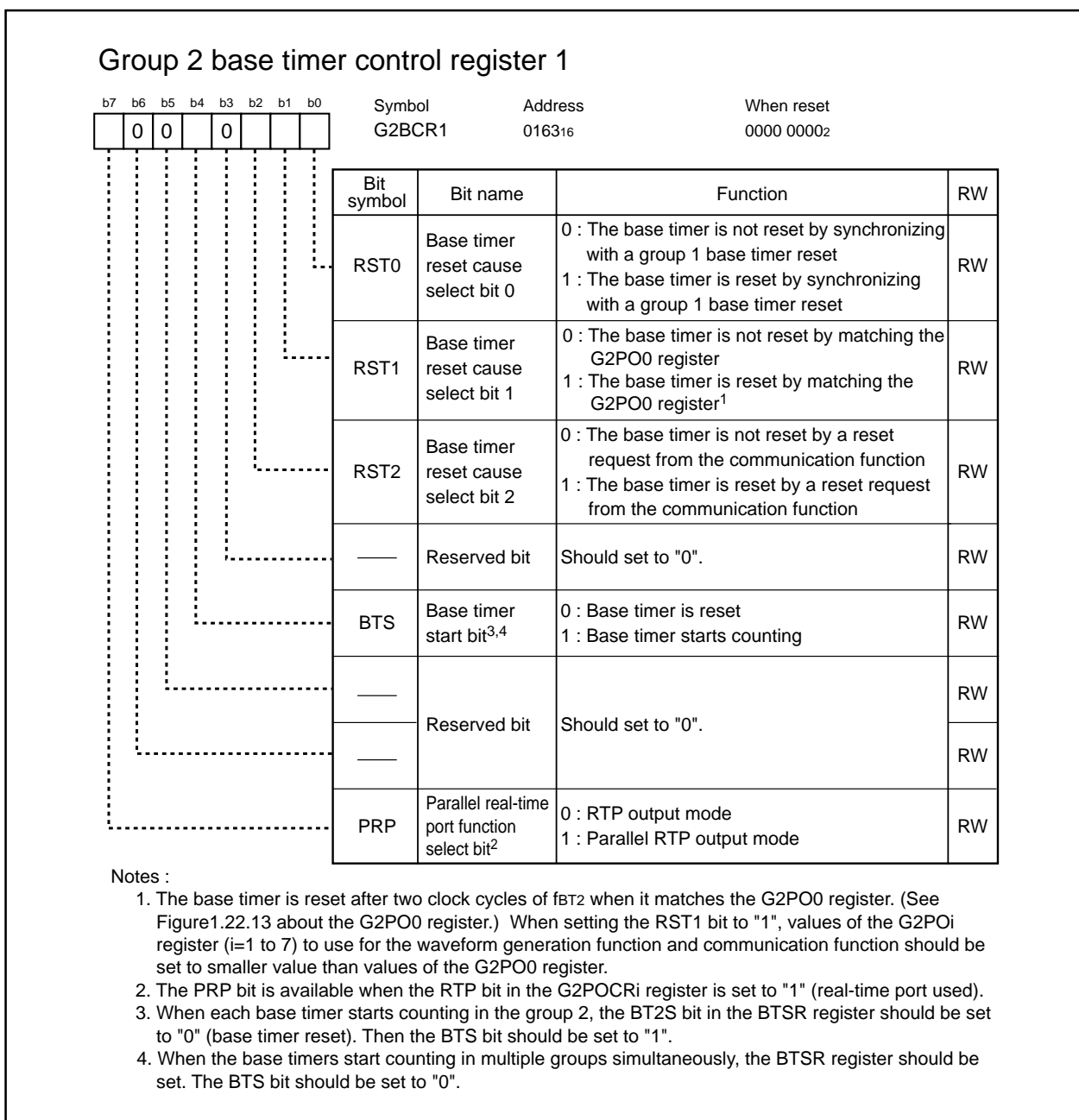


Figure 1.22.7. G2BCR1 Register

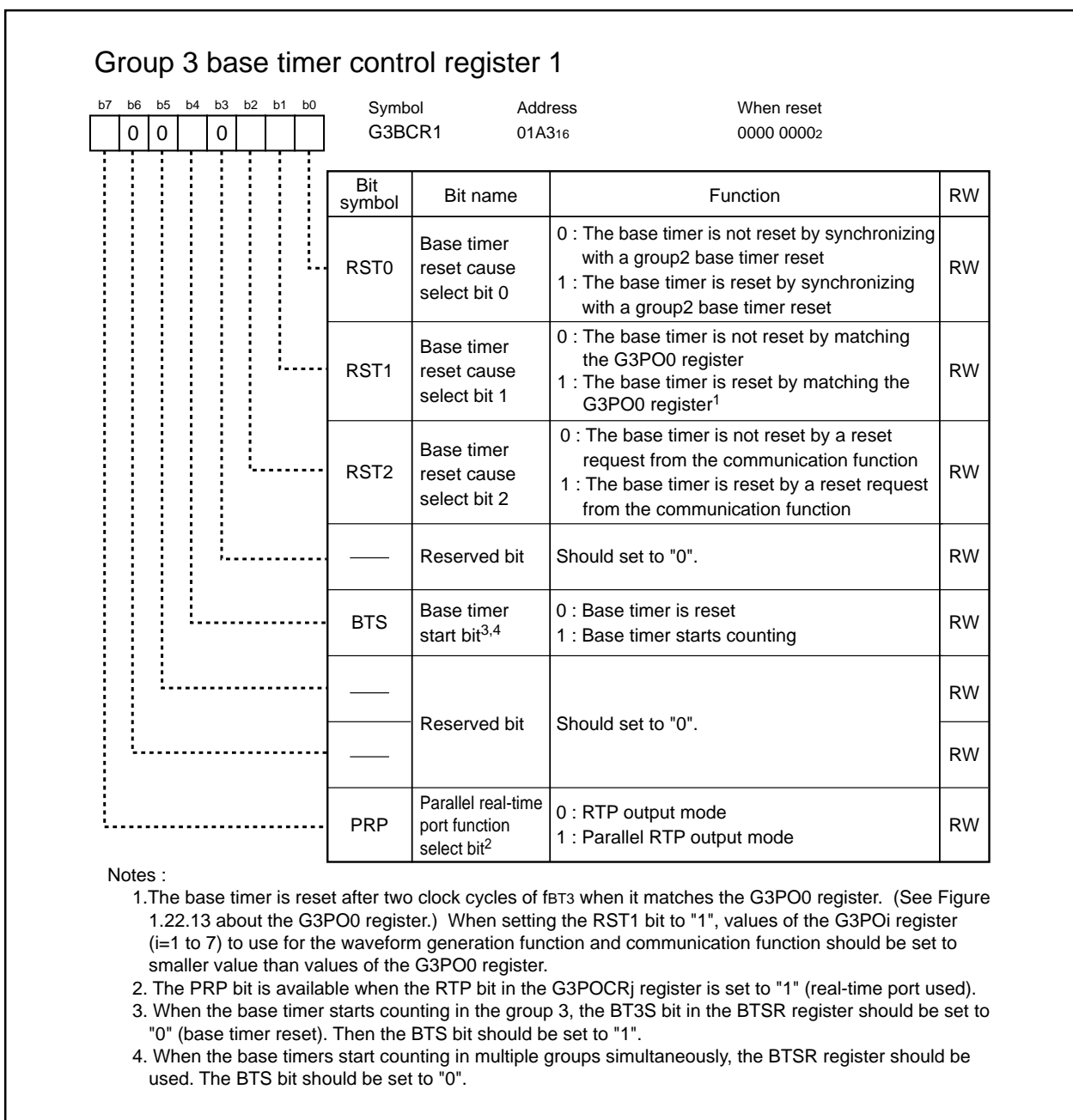


Figure 1.22.8. G3BCR1 Register

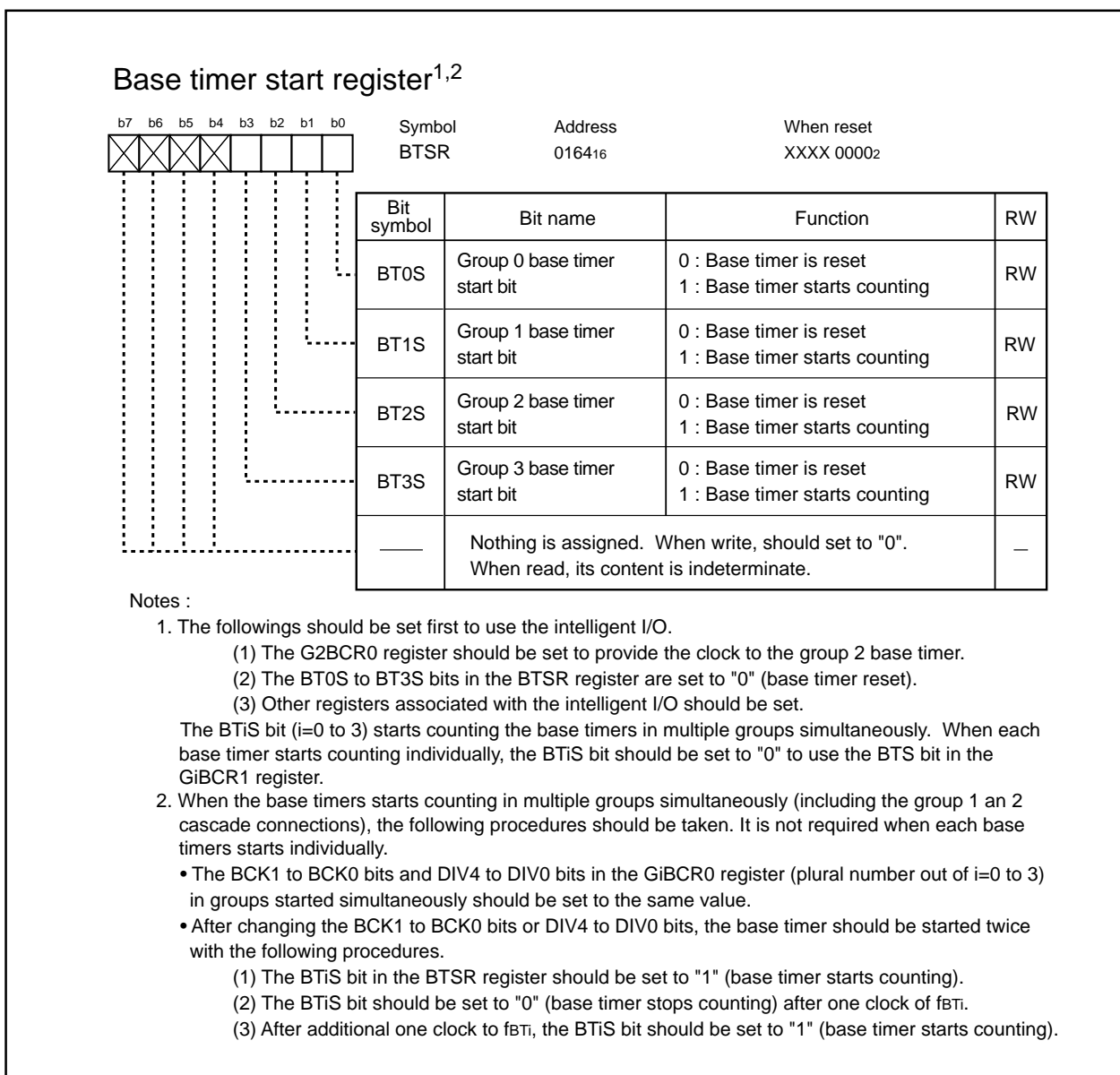
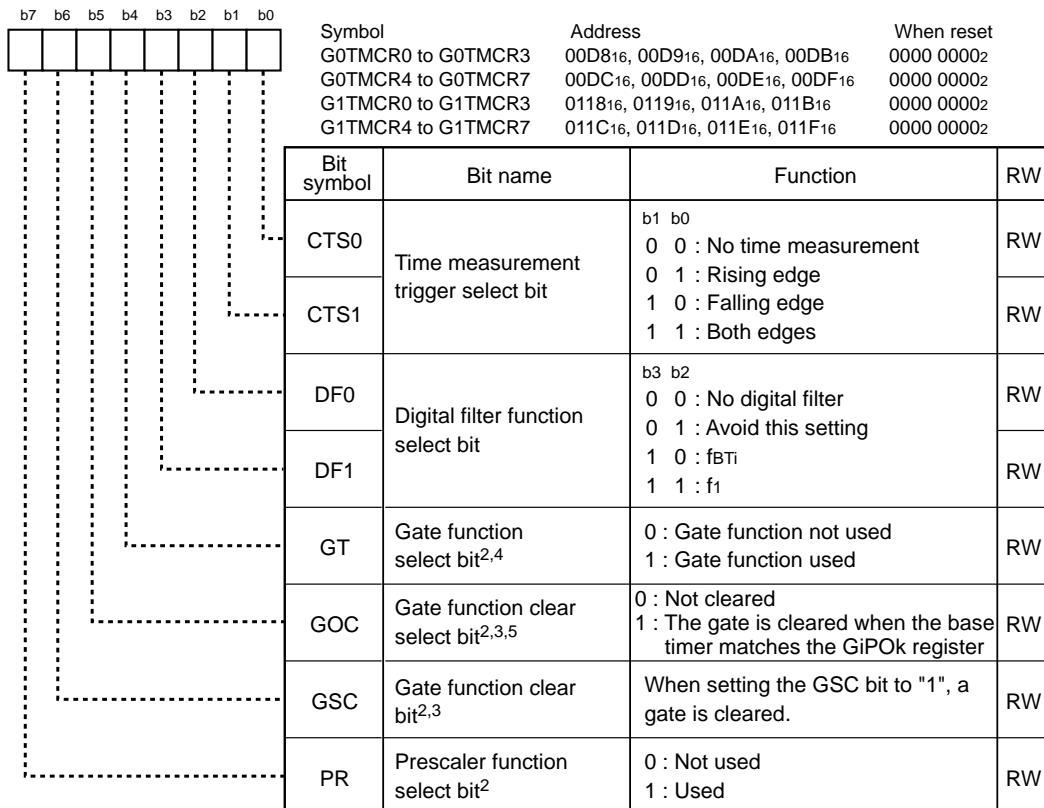


Figure 1.22.9. BTSR Register

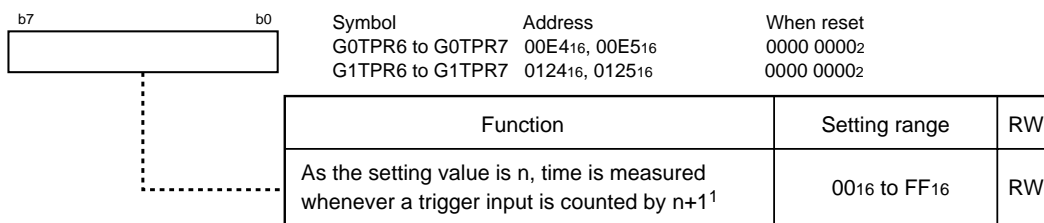
### Group i time measurement control register j<sup>1</sup>(i=0,1 / j=0 to 7)



Notes :

- When the CAS bit in the GiBCR1 register is set to "0" (16-bit time measurement function), the G1TMCR0 and G1TMCR3 to G1TMCR5 registers cannot be used. When write, these register should be set to "00<sub>16</sub>".  
When setting the CAS bit to "1" (32-bit time measurement function), the G0TMCRj and G1TMCRj registers should be set to same value.
- This bit is in the GiTMCR6 and GiTMCR7 registers.  
All bits 4 to 7 in the GiTMCR0 to GiTMCR5 registers should set to "0".
- These bits are available when setting the GT bit to "1".
- When the CAS bit in the GiBCR1 register is set to "1" (32-bit time measurement function), the GT bit should be set to "0". The gate function cannot be used.
- The GOC bit is set to "0" after the gate function is cleared. See Figure 1.22.13 about the GiPOk register (k=4 when j=6 and k=5 when j=7).

### Group i time measurement prescale register j (i=0,1 / j=6,7)

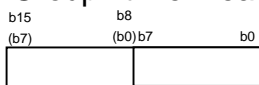


Notes :

- The first prescaler, after the PR bit in the GiTMCRj register changes "0" (prescaler function used) to "1" (prescaler function unused), may be divided by n regarding the setting value n, not by n+1. Subsequent prescaler is divided by n+1.

Figure 1.22.10. G0TMCR0 to G0TMCR7 and G1TMCR0 to G1TMCR7 Registers, G0TPR6, G0TPR7 and G1TPR6, G1TPR7 Registers

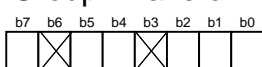
### Group i time measurement register j (i=0,1 / j=0 to 7)



Symbol	Address	When reset
G0TM0 to G0TM2	00C1 <sub>16</sub> - 00C0 <sub>16</sub> , 00C3 <sub>16</sub> - 00C2 <sub>16</sub> , 00C5 <sub>16</sub> - 00C4 <sub>16</sub>	Indeterminate
G0TM3 to G0TM5	00C7 <sub>16</sub> - 00C6 <sub>16</sub> , 00C9 <sub>16</sub> - 00C8 <sub>16</sub> , 00CB <sub>16</sub> - 00CA <sub>16</sub>	Indeterminate
G0TM6 to G0TM7	00CD <sub>16</sub> - 00CC <sub>16</sub> , 00CF <sub>16</sub> - 00CE <sub>16</sub>	Indeterminate
G1TM0 to G1TM2	0101 <sub>16</sub> - 0100 <sub>16</sub> , 0103 <sub>16</sub> - 0102 <sub>16</sub> , 0105 <sub>16</sub> - 0104 <sub>16</sub>	Indeterminate
G1TM3 to G1TM5	0107 <sub>16</sub> - 0106 <sub>16</sub> , 0109 <sub>16</sub> - 0108 <sub>16</sub> , 010B <sub>16</sub> - 010A <sub>16</sub>	Indeterminate
G1TM6 to G1TM7	010D <sub>16</sub> - 010C <sub>16</sub> , 010F <sub>16</sub> - 010E <sub>16</sub>	Indeterminate

Function	Setting range	RW
Value of the base timer is stored every time measurement. When the CAS bit in the GiBCR1 register is set to "1" (32-bit time measurement), low-order 16 bits are stored into the G0TMj register and high-order 16 bits are into stored the G1TMj register.	—	RO

### Group i waveform generation control register j<sup>1</sup> (i=0 to 1/ j=0 to 7)



Symbol	Address	When reset
G0POCR0 to G0POCR3	00D0 <sub>16</sub> , 00D1 <sub>16</sub> , 00D2 <sub>16</sub> , 00D3 <sub>16</sub>	0X00 X000 <sub>2</sub>
G0POCR4 to G0POCR7	00D4 <sub>16</sub> , 00D5 <sub>16</sub> , 00D6 <sub>16</sub> , 00D7 <sub>16</sub>	0X00 X000 <sub>2</sub>
G1POCR0 to G1POCR3	0110 <sub>16</sub> , 0111 <sub>16</sub> , 0112 <sub>16</sub> , 0113 <sub>16</sub>	0X00 X000 <sub>2</sub>
G1POCR4 to G1POCR7	0114 <sub>16</sub> , 0115 <sub>16</sub> , 0116 <sub>16</sub> , 0117 <sub>16</sub>	0X00 X000 <sub>2</sub>

Bit symbol	Bit name	Function	RW
MOD0	Operation mode select bit	b2b1b0 0 0 0: Single waveform output mode 0 0 1: SR waveform output mode <sup>2</sup> 0 1 0: Phase-delayed waveform output mode	RW
MOD1		0 1 1: Avoid this setting 1 0 0: Avoid this setting 1 0 1: Avoid this setting <sup>3</sup> 1 1 0: Avoid this setting <sup>3</sup>	RW
MOD2		1 1 1: Use communication function output <sup>4</sup>	RW
—	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.	—	—
IVL	Output initial value select bit	0: Outputs "L" as an initial value 1: Outputs "H" as an initial value	RW
RLD	GiPOj register value reload timing select bit	0: Reloads the GiPOj register when the CPU writes a counter 1: Reloads the GiPOj register when the base timer is reset	RW
—	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.	—	—
INV	Inverse output function select bit <sup>5</sup>	0: Output is not inverted 1: Output is inverted	RW

Notes :

- The Group 0 and 1 have the 16-bit and 32-bit waveform generation functions. When the CAS bit in the GiBCR1 register is set to "0" (16-bit waveform generation function), the G0POCR2, 3, 6 and 7 registers cannot be available. When write, should set to "0016". When setting the CAS bit to "1" (32-bit waveform generation function), the G0POCRj and G1POCRj registers should set to the same value.
- This setting is enabled only on even channels. In SR waveform output mode, value written to the corresponding odd channel (next channel after an even channel) are ignored. An even channel outputs waveform. An odd channel outputs no waveform.
- When receiving in UART mode of the group 0 and 1, the GiPOCR2 register should be set to "0000 0110<sub>2</sub>".
- This setting is enabled only for channel 0 and 1. When using ISTxDi, the GiPOCR0 register should set to "1112". When using ISCLKi for output, the MOD2 to MOD0 bits in the GiPOCR1 register should set to "1112". Avoid setting the MOD2 to MOD0 bits to "1112" other than channels 0 and 1.
- The inverse output function is performed as a final step on a process of waveform generation. When setting the INV bit to "1" (output inverse), "H" is output with setting the IVL bit to "0" (output "L" as an initial value) and "L" with setting the IVL bit to "1" (output "H" as an initial value).

Figure 1.22.11. G0TM0 to G0TM7 and G1TM0 to G1TM7 Registers, G0POCR0 to G0POCR7 and G1POCR0 to G1POCR7 Registers

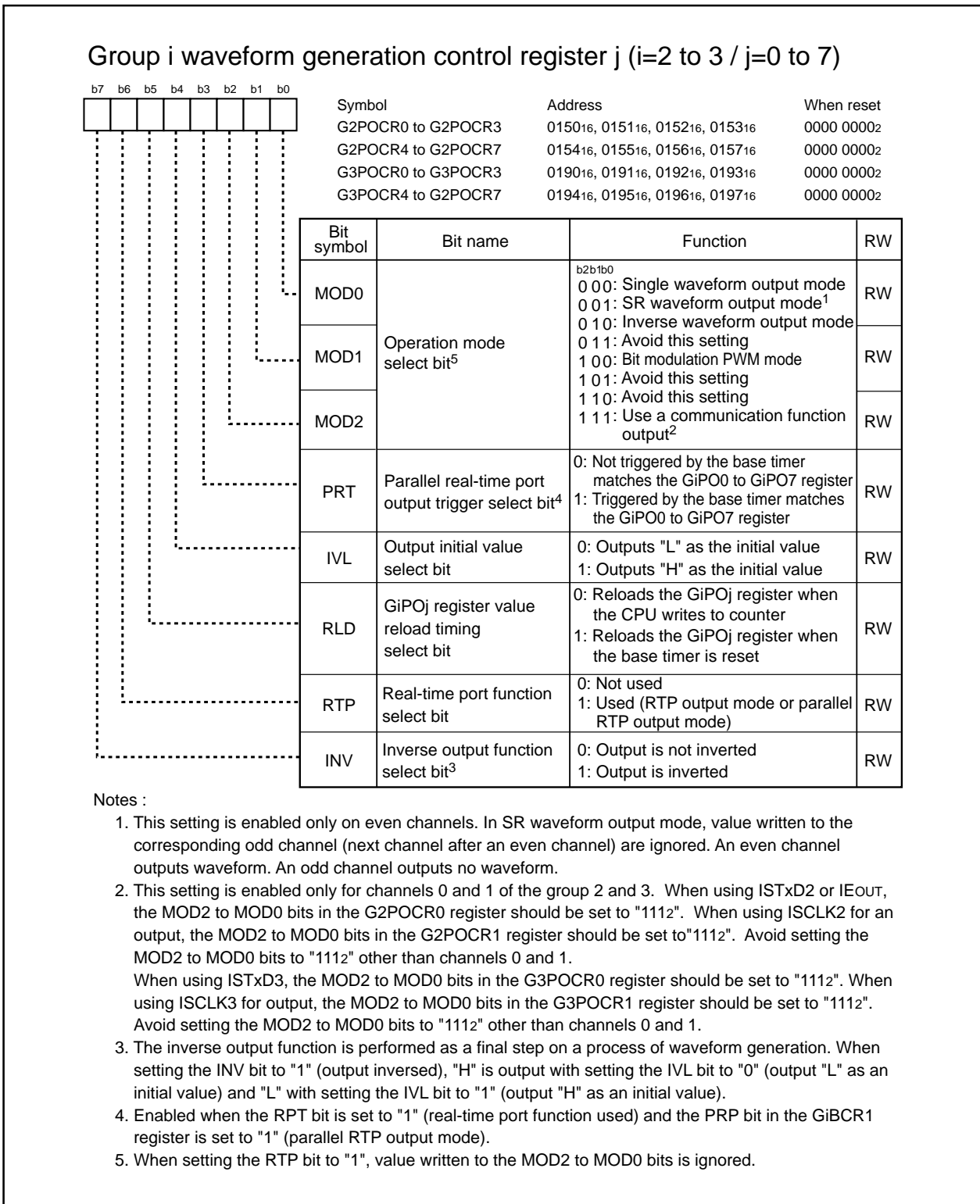


Figure 1.22.12. G2POCR0 to G2POCR7 and G3POCR0 to G3POCR7 Registers

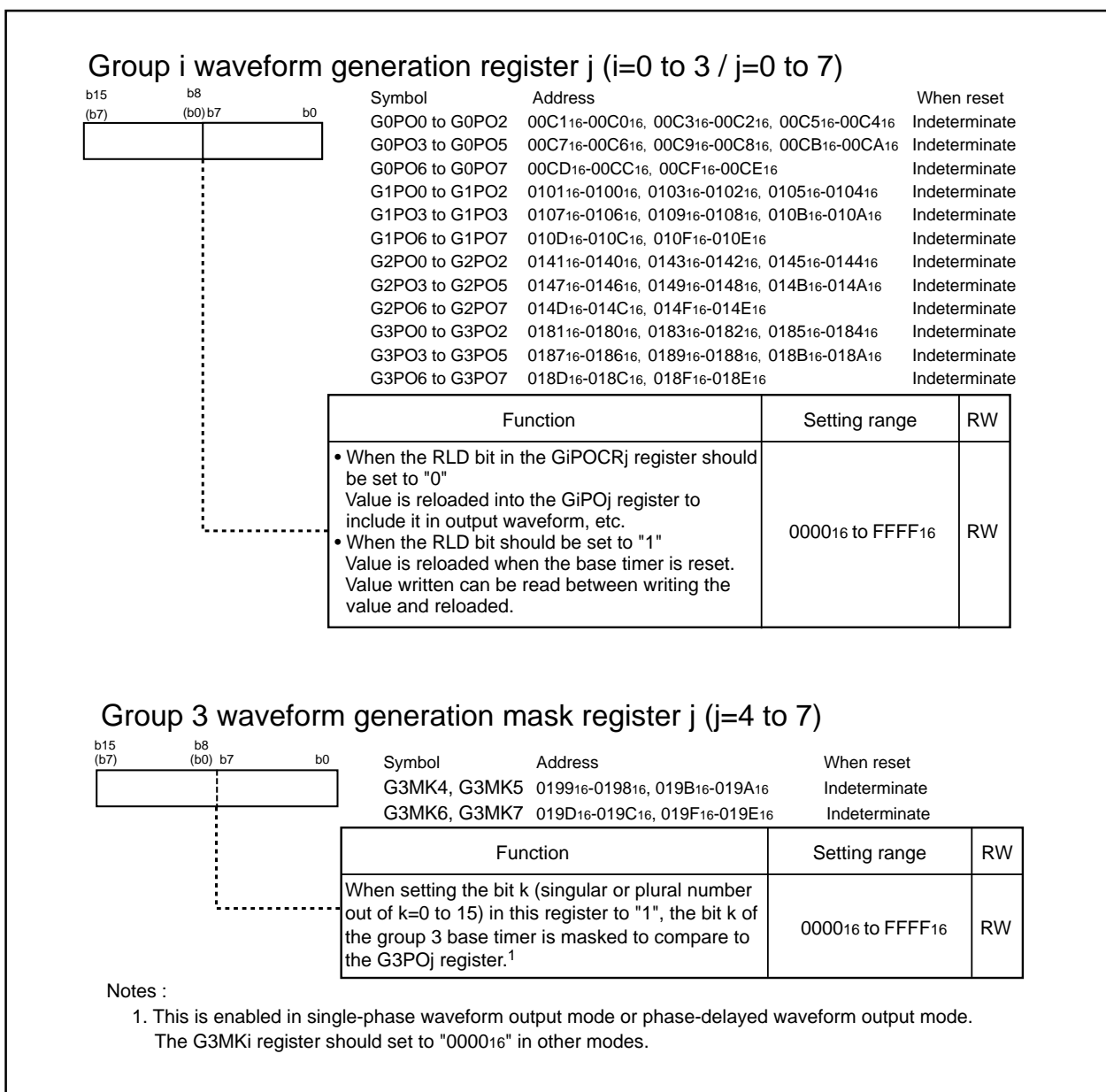


Figure 1.22.13. G0PO0 to G0PO7, G1PO0 to G1PO7, G2PO0 to G2PO7 and G3PO0 to G3PO7 Registers, G3MK4 to G3MK7 Registers

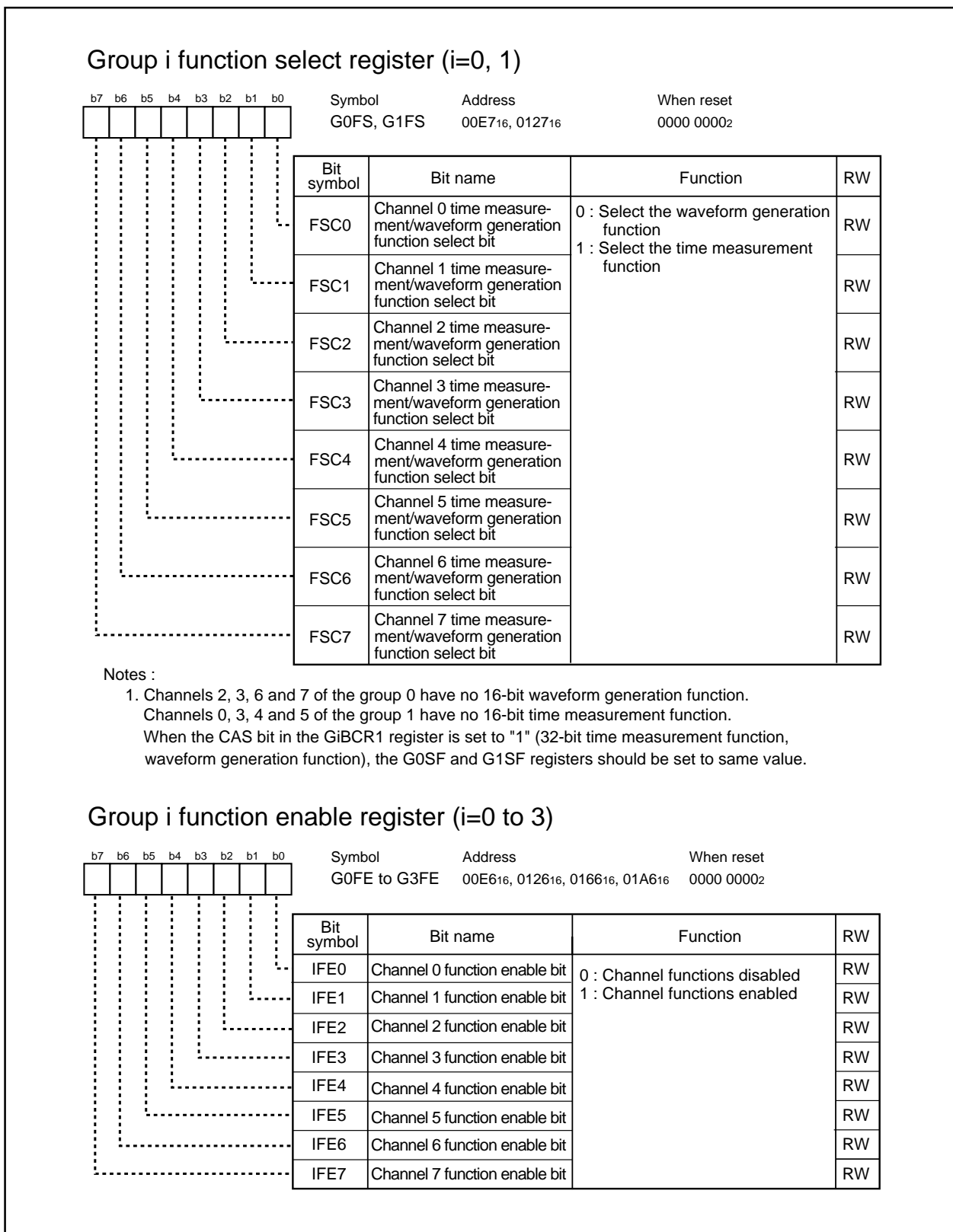
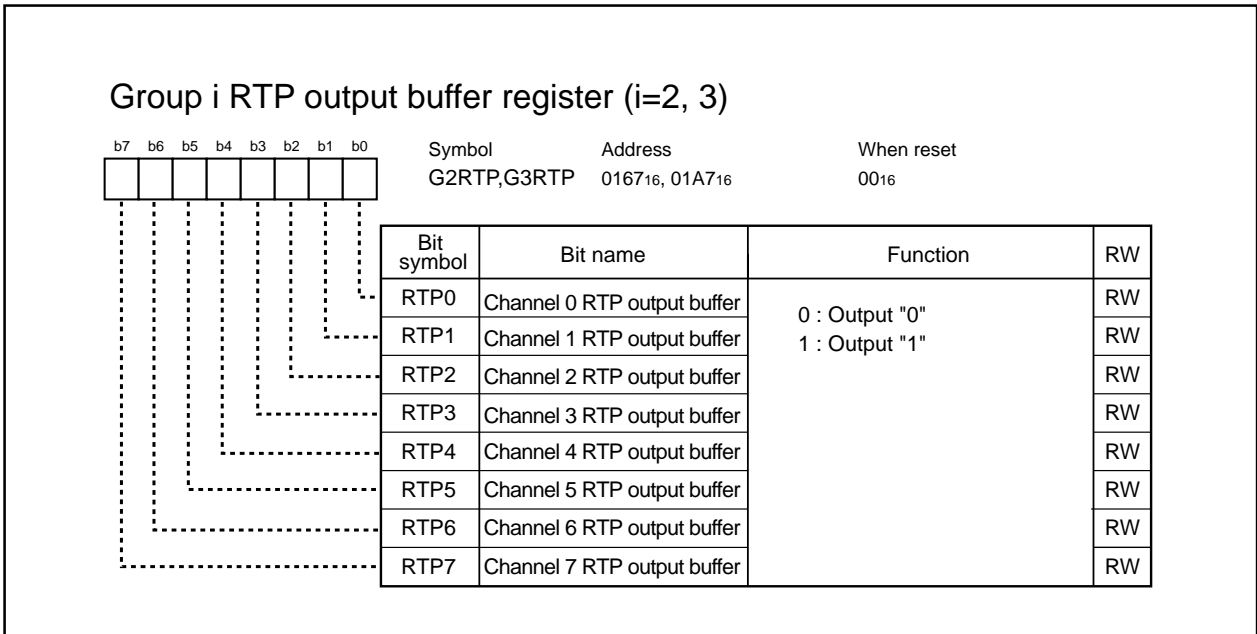


Figure 1.22.14. G0FS and G1FS Registers and G0FE to G3FE Registers



**Figure 1.22.15. G2RTP and G3RTP Registers**

## Base Timer (Group 0 to 3)

The base timer counts an internally generated count source with free-running.

Table 1.22.2 lists specifications of the base timer. Figures 1.22.5 to 1.22.9 show registers associated with the base timer. Figure 1.22.16 shows a block diagram of the base timer. Figure 1.22.17 shows an example of the base timer (group 0, 1) in counter increment mode. Figure 1.22.18 shows an example of the base timer (group 0, 1) in counter increment/decrement mode. Figure 1.22.19 shows a cascaded connection. Figure 1.22.20 shows an example of two-phase pulse signal processing mode.

**Table 1. 22.2. Base Timer Specifications**

Item	Specification
Count source( $f_{BTi}$ )( $i=0$ to 3)	$f_1$ divided by $2^{(n+1)}$ (group 0 to 3), two pulse input divided by $2^{(n+1)}$ (group 0, 1) n: The DIV4 to DIV0 bits in the GiBCR0 register determines. n=0 to 31 In $f_1$ and two pulses input when $n=31$ , a count source is not divided
Counting operation	The base timer increments the counter The base timer decrements the counter (see the group 0, 1 select function) Two-phase pulse processing (see the group 0, 1 select function)
Count start condition	<ul style="list-style-type: none"> <li>When the base timers individually start counting The BTS bit in the GiBCR1 register should be set to "1" (base timer starts counting)</li> <li>When the base timers in multiple groups simultaneously started counting The BTiS bit in the B TSR register should be set to "1" (base timer starts counting)</li> </ul>
Count stop condition	The BTiS bit in the B TSR register is set to "0" (base timer reset) and the BTS bit in the GiBCR1 register is set to "0" (base timer reset)
Base timer reset condition	(1) Synchronized with the base timer reset in different groups Group 0: synchronized with the base timer reset in the group 1 Group 1: synchronized with the base timer reset in the group 0 Group 2: synchronized with the base timer reset in the group 1 Group 3: synchronized with the base timer reset in the group 2 (2) Value of the base timer matches value of the GiPO0 register (3) Input "L" to external interrupt pins Group 1: the INT0 pin      Group 2: the INT1 pin (4) Requested a reset from the communication function (group 2, 3)
Value for base timer reset	"0000 <sub>16</sub> "
Interrupt request	In bit 14 or bit 15 overflow of the base timer, the BTiR bit in the interrupt request register is set to "1" (See Figure 1.9.14.)
Read from timer	<ul style="list-style-type: none"> <li>While the base timer is running, the GiBT register indicates a counter value</li> <li>When the base timer is reset, a counter value is indeterminate</li> </ul>
Write to timer	When a value is written while the base timer is running, the value written is counted first. No value can be written while the base timer is reset.
Selectable function	<ul style="list-style-type: none"> <li>Cascaded connection (group0, 1) The group 1 base timer increments its counter whenever the group 0 base timer overflows. (See Figure 1.22.19.)</li> <li>Counter increment/decrement mode (group0, 1) After starting counting, the base timer increments the counter from "0000<sub>16</sub>" until reaching "FFFF<sub>16</sub>" to decrement the counter. The base timer increments the counter again when reaching the next "0000<sub>16</sub>". (See Figure 1.22.18.)</li> </ul>

Table 1.22.2. Base Timer Specifications (Continued)

Item	Specification
Selectable function	<ul style="list-style-type: none"> <li>Two-phase pulse processing mode (group 0, 1)</li> </ul> <p>In the group 0, two-phase pulses from P76 and P77 pins are counted In the group 1, two-phase pulses from P80 and P81 pins are counted as well (See Figure 1.22.20)</p>

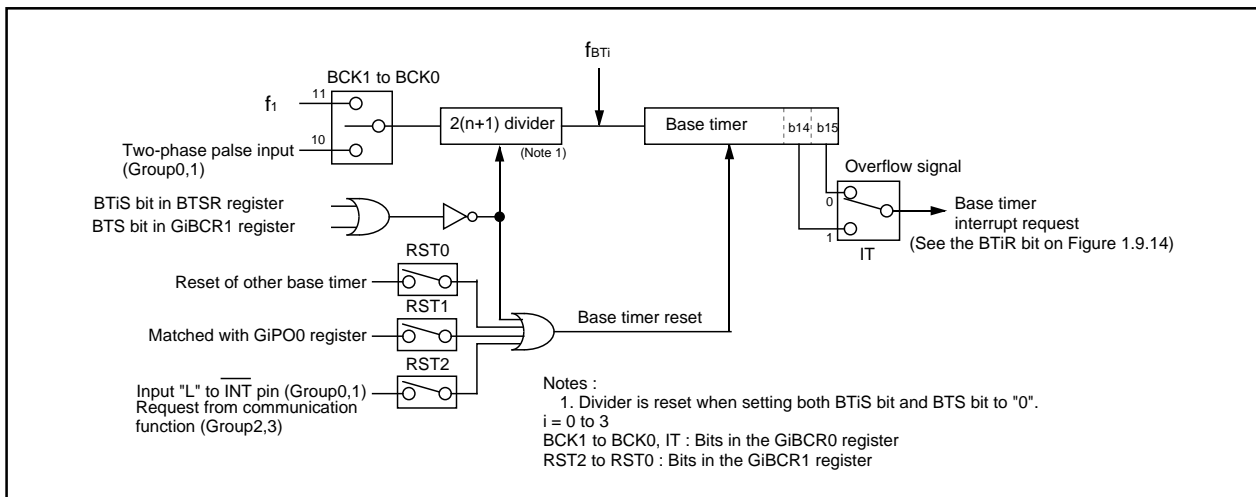


Figure 1.22.16. Base Timer Block Diagram

Table 1.22.3. Base Timer Associated Register Settings (Time Measurement Function, Waveform Generation Function, Communication Function)

Register	Bit	Function
G2BCR0	-	Provides the clock to the BTSR register. Set to "0111 11112"
BTSR	-	Set to "0000 00002"
GiBCR0	BCK1 to BCK0	Select a count source
	DIV4 to DIV0	Select a divide ratio of a count source
	IT	Select the base timer interrupt
GiBCR1	RST2 to RST0	Select base timer reset timing
	BTS	Used when starting the base timer independently
	UD1 to UD0	Select how to count (Group0,1)
	CAS	Select cascaded connection (Group0,1)
GiBT	-	Base timer value to read or to write

When setting the RST bit to "1" (base timer reset when base timer matches GiPO0), the following registers require to be setup.

GiPOCR0	MOD2 to MOD0	Set to "0002" (single-phase waveform output mode)
GiPO0	-	Set reset cycle
GiFS	FSC0	Set to "0" (waveform generation function)
GiFE	IFE0	Set to "1" (channel operation start)

i : 0 to 3

Bit configuration and function vary depending on which group is used.

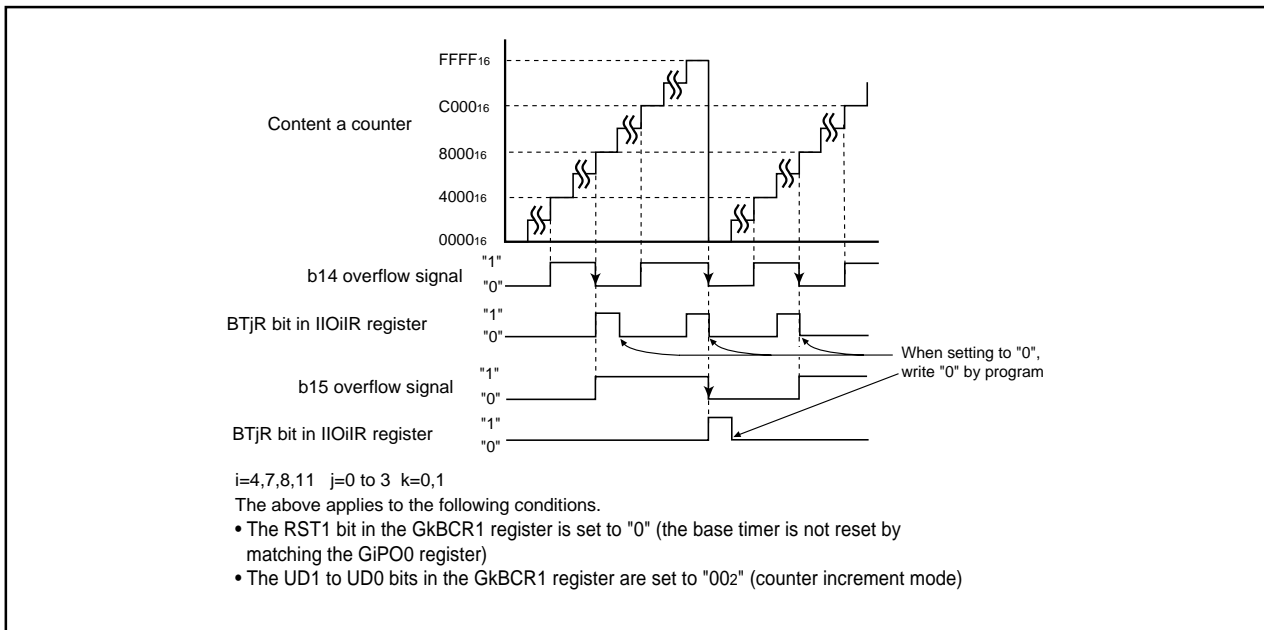


Figure 1.22.17. Counter Increment Mode (Group 0, 1)

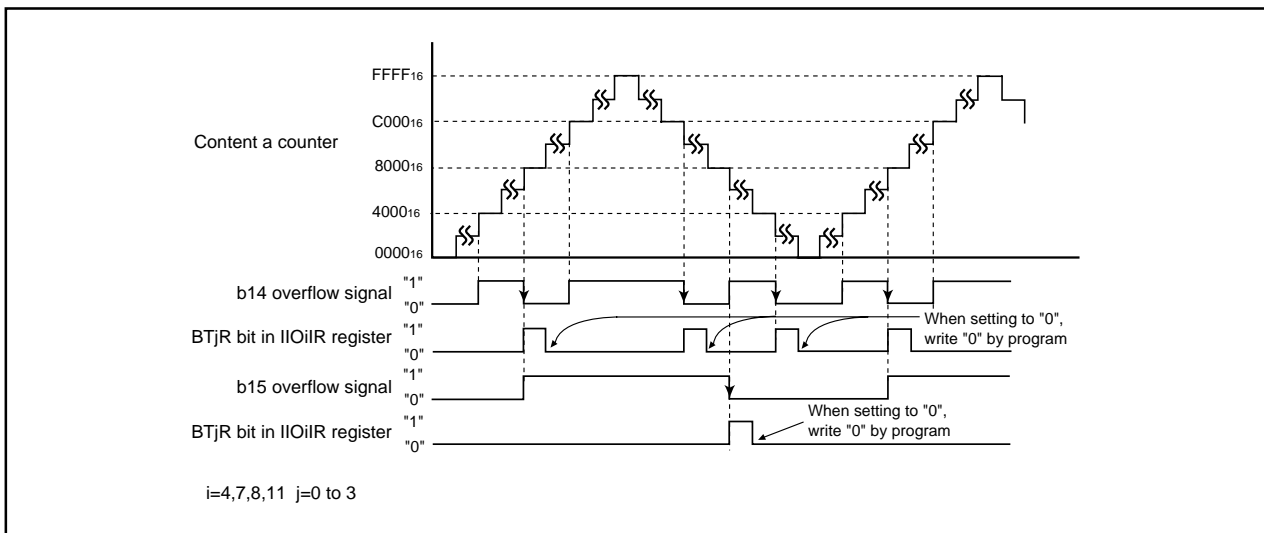


Figure 1.22.18. Counter Increment/Decrement Mode (Group 0,1)

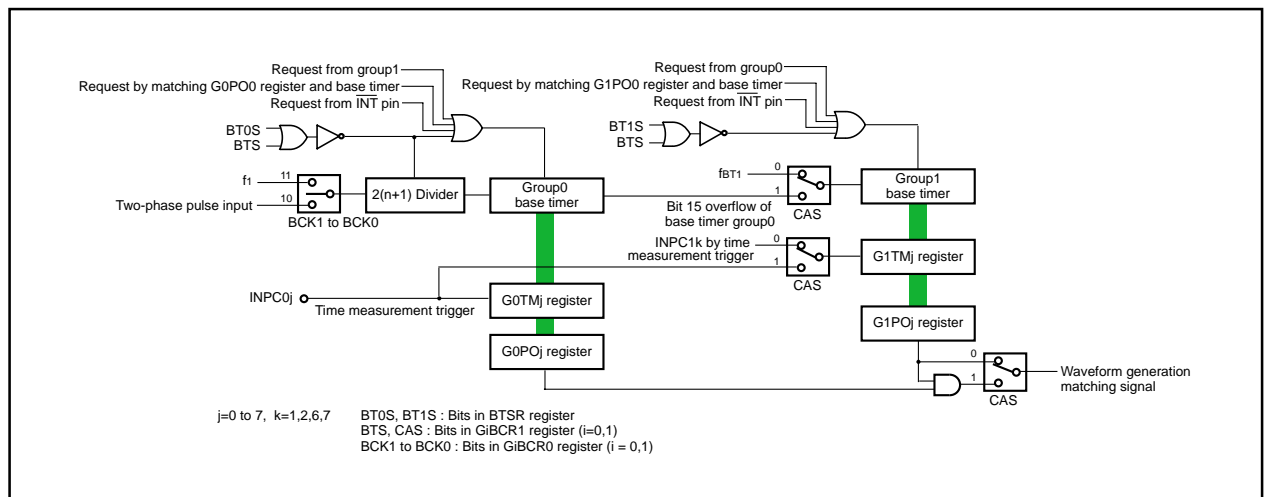


Figure 1.22.19. Cascaded Connection

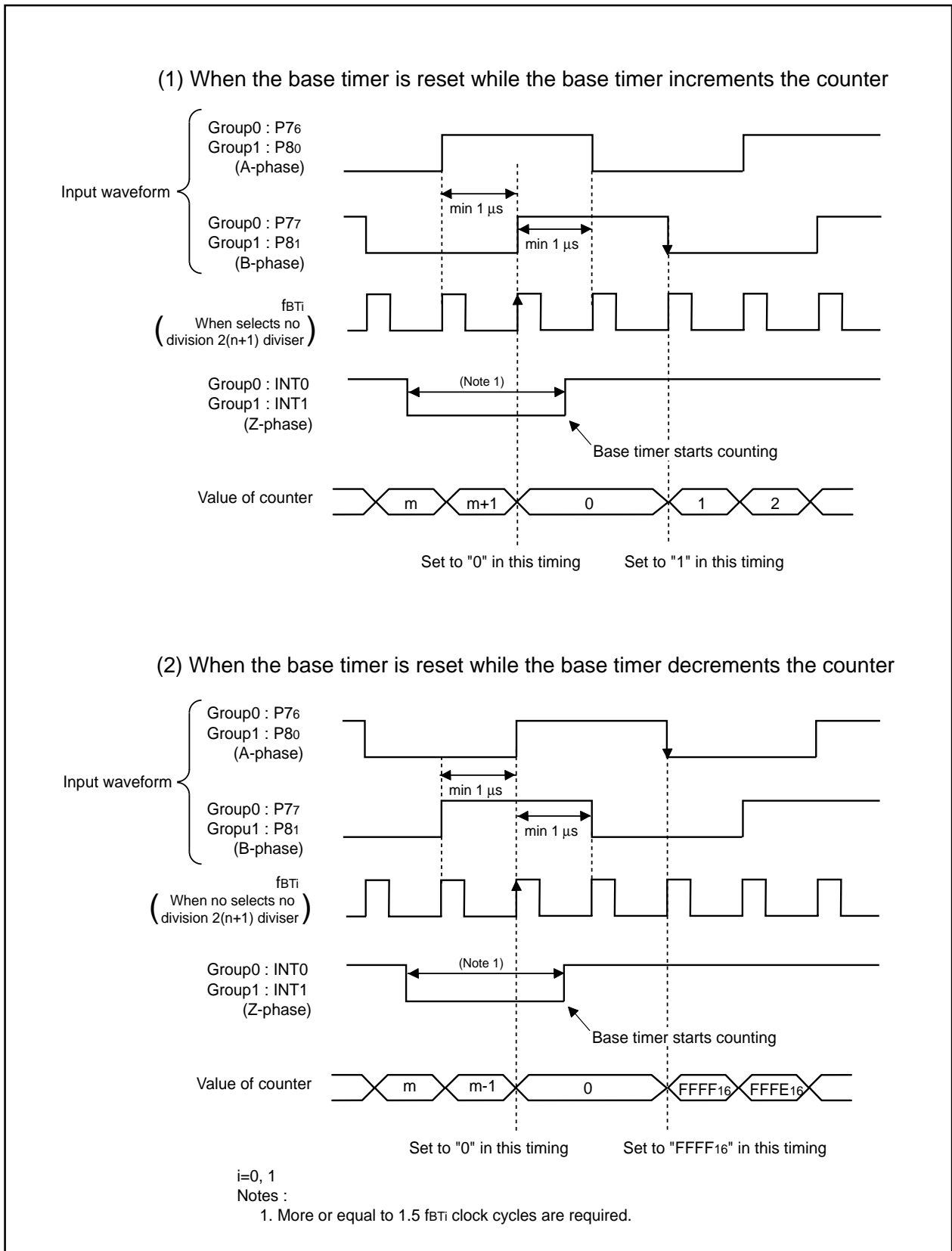


Figure 1.22.20. Base Timer Operation in Two-phase Pulse Signal Processing Mode (Group 0,1)

## Time Measurement Function (group 0 and 1)

With synchronizing an external trigger input, value of the base timer are stored into the GiTMj register (i=0 to 1, j=0 to 7). Table 1.22.4 shows specifications of the time measurement function. Figures 1.22.21 and 1.22.22 show an operating timing of the time measurement function. Figure 1.22.23 shows an operating timing of the prescaler function and gate function.

**Table 1.22.4. Time Measurement Function Specifications**

Item	Specification
Measurement channel	Group 0: channel 0 to 7 Group 1: channel 1, 2, 6, 7
Selecting trigger input polarity	Rising edge, falling edge, both edges of the INPCij pin <sup>1</sup>
Measurement start condition	The IFEj bit in the GiFE register should be set to "1" (channels j function enabled) when the FSCj bit (i=0, 1, j=0 to 7) in the GiFS register is set to "1" (time measurement function selected).
Measurement stop condition	The IFEj bit should be set to "0" (channel j function disabled)
Time measurement timing	<ul style="list-style-type: none"> <li>•No prescaler : every a trigger is input</li> <li>•Prescaler (for channel 6 and channel 7) : every value of the GiTPRk register (k=6,7) +1 trigger input</li> </ul>
Interrupt request generation timing	The TMijR bit in the interrupt request register (See Figure 1.9.14) is set to "1" at time measurement timing
INPCij pin function <sup>1</sup>	Trigger input pin
Selectable function	<ul style="list-style-type: none"> <li>• Digital filter function The digital filter samples a trigger input level every f1 or fBTi to pass pulses matching a trigger input level three times</li> <li>• Cascaded connection function Group 0 and 1 are connected to operate as a 32-bit timer</li> <li>• Prescaler function (for channel 6 and channel 7) Trigger inputs are counted to perform time measurement whenever value of the GiTPRk register + 1 trigger is input</li> <li>• Gate function (for channel 6 and channel 7) When a trigger input is inhibited with setting the GOC bit in the GiTMCRk register to "1" (gate cleared by matching the GiPOp register (p=4 when k=6, p=5 when k=7)) after time measurement by first trigger input, a trigger input is enabled to receive again by matching the base timer with the GiPOp register</li> </ul>

Notes :

1. The INPC00 to INPC07, INPC11 to INPC12 and INPC16 to INPC17 pins (INPC00 to INPC07 pins in cascaded connection)

**Table 1.22.5. Pin settings for Time Measurement Function**

Pin <sup>2</sup>	Bit and Setting		
	PS1, PS2, PS5, PS8, PS9 registers	PD7, PD8, PD11, PD14, PD15 registers	IPS register
P74/INPC11	PS1_4 = 0	PD7_4 = 0	IPS1 = 0
P75/INPC12	PS1_5 = 0	PD7_5 = 0	
P76/INPC00	PS1_6 = 0	PD7_6 = 0	IPS0 = 0
P77/INPC01	PS1_7 = 0	PD7_7 = 0	IPS0 = 0
P80/INPC02	PS2_0 = 0	PD8_0 = 0	IPS0 = 0
P111/INPC11 <sup>1</sup>	PS5_1 = 0	PD11_1 = 0	IPS1 = 1
P112/INPC11 <sup>1</sup>	PS5_2 = 0	PD11_2 = 0	
P142/INPC16 <sup>1</sup>	PS8_2 = 0	PD14_2 = 0	-
P143/INPC17 <sup>1</sup>	PS8_3 = 0	PD14_3 = 0	
P150/INPC00 <sup>1</sup>	PS9_0 = 0	PD15_0 = 0	
P151/INPC01 <sup>1</sup>	PS9_1 = 0	PD15_1 = 0	IPS0 = 1, IPS2 = 0
P152/INPC02 <sup>1</sup>	-	PD15_2 = 0	
P153/INPC03 <sup>1</sup>	-	PD15_3 = 0	
P154/INPC04 <sup>1</sup>	PS9_4 = 0	PD15_4 = 0	IPS2 = 0
P155/INPC05 <sup>1</sup>	PS9_5 = 0	PD15_5 = 0	
P156/INPC06 <sup>1</sup>	-	PD15_6 = 0	
P157/INPC07 <sup>1</sup>	-	PD15_7 = 0	

Notes :

1. This port can be provided in the 144-pin package only.
2. When the CAS bit in the GiBCR register is set to "1" (32-bit time measurement function), a trigger should be input to the INPC0j pin (j=0 to 7). Trigger input to the INPC1k1 pin (k=1, 2, 6, 7) is disabled.

**Table 1.22.6. Registers Setting Associated with the Time Measurement Function (group0, 1)**

Register	Bit	Function
GiTMCRj	CTS1 to CTS0	Select time measurement trigger
	DF1 to DF0	Select the digital filter function
	GT, GOC, GSC	Select the gate function
	PR	Select the prescaler function
GiTPRk	-	Setting value of prescaler
GiFS	FSCj	Set to "1" (time measurement function)
GiFE	IFEj	Set to "1" (channel j function enabled)

i = 0 j = 0 to 7 k = 6, 7

Bit configuration and function vary depending on which group or channel is used.

Registers associated with to the time measurement function should be set after setting registers associated with the base time.

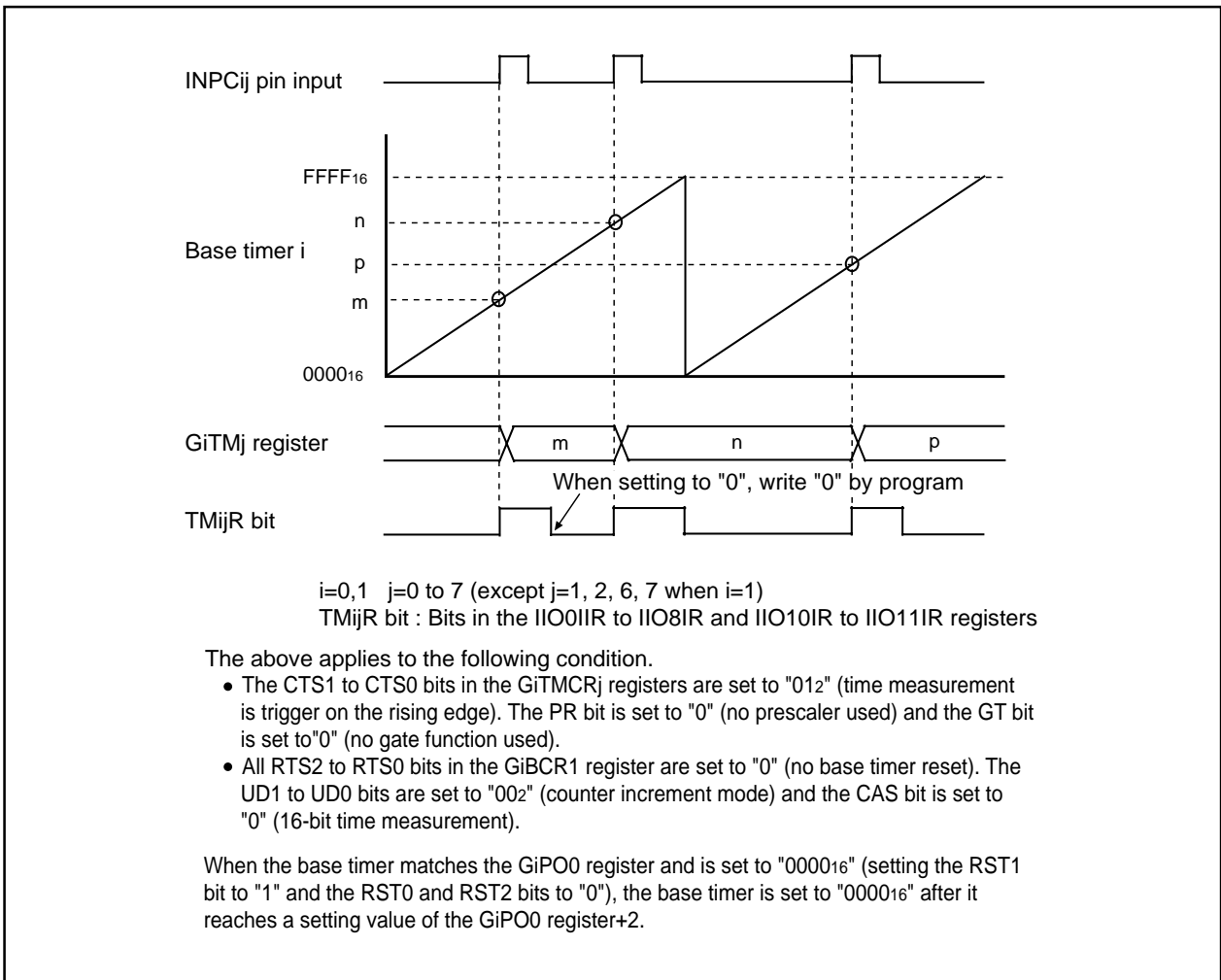


Figure 1.22.21. Time Measurement Function (1)

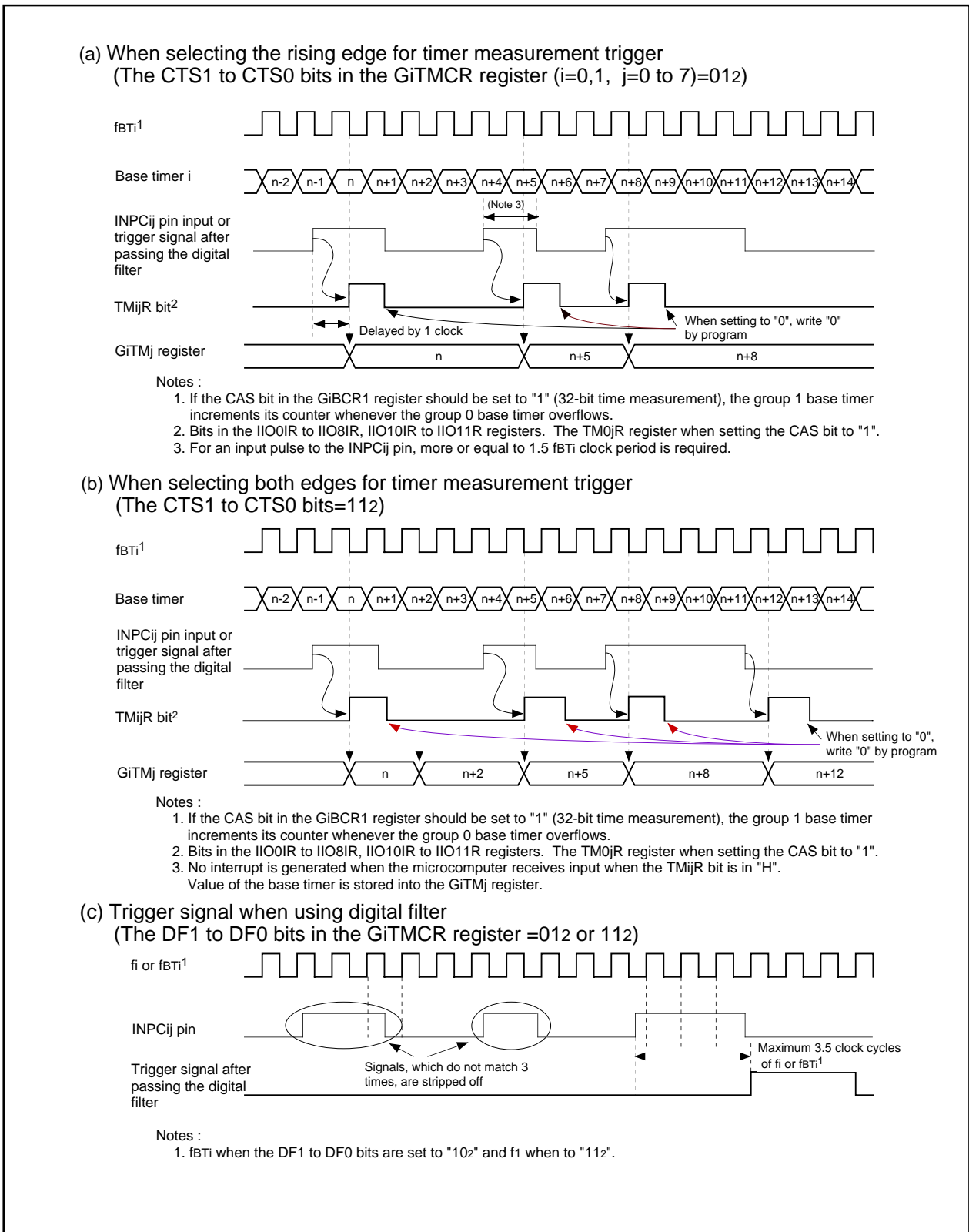
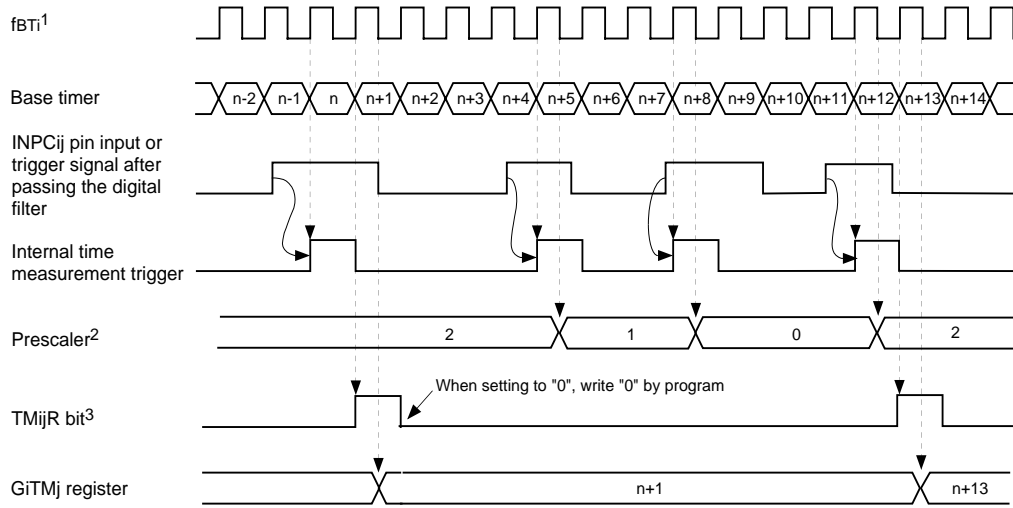


Figure 1.22.22. Time Measurement Function (2)

(a) When using the prescaler function

(When the GiTPRj register (i=0, 1, j=6, 7) =0216, PR bit in the GiTMCR register=1)

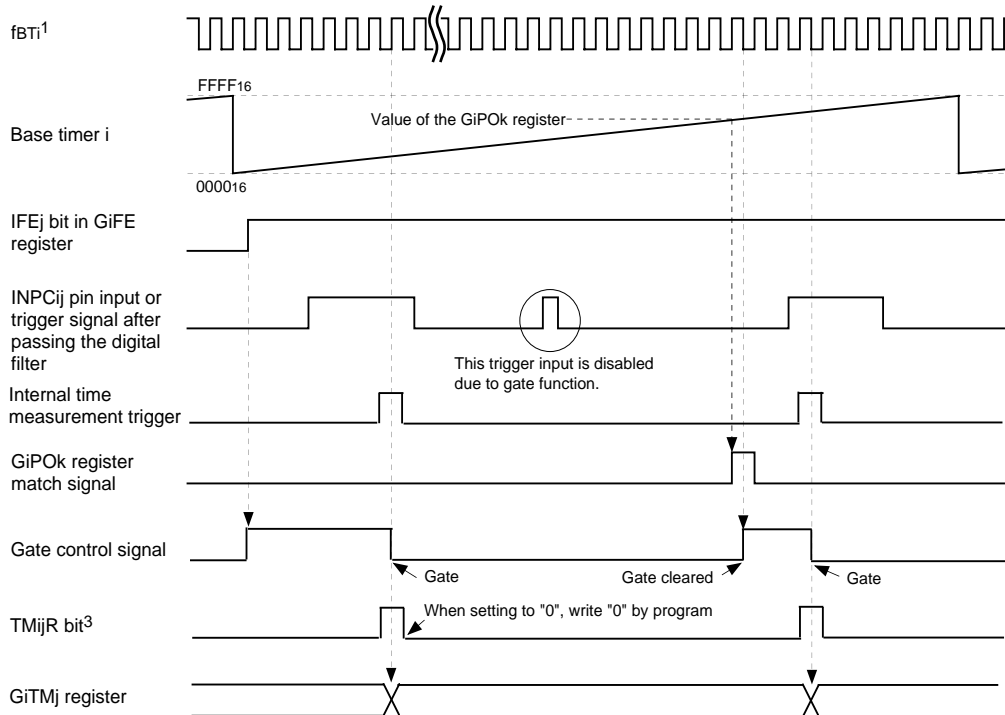


Notes :

1. If the CAS bit in the GiBCR1 register should be set to "1" (32-bit time measurement), the group 1 base timer increments its counter whenever the group 0 base timer overflows.
2. This applies to the second period that the GiTPRj register decrements after setting the PR bit in the GiTMCRj register to "1" (prescaler used).
3. Bits in the IIO0iR to IIO8iR, IIO10iR to IIO11R registers. The TM0jR register when setting the CAS bit to "1".

(b) When using the gate function

(Gate function is cleared by matching the GiPOk register and base timer, the GT bit in the GiTMCRj register=1, the GOC bit=1)



Notes :

1. If the CAS bit in the GiBCR1 register should be set to "1" (32-bit time measurement), the group 1 base timer increments its counter whenever the group 0 base timer overflows.
2. Bits in the IIO0iR to IIO8iR, IIO10iR to IIO11R registers. The TM0jR register when setting the CAS bit to "1".

Figure 1.22.23. Prescaler Function and Gate Function

## Waveform Generation Function (Group 0 to 3)

Waveforms are generated when value of the base timer matches GiPOj register (i=0 to 3, j=0 to 7).

The waveform generation function has the following six modes :

- Single-phase waveform output mode (group 0 to 3)
- Phase-delayed waveform output mode (group 0 to 3)
- Set/Reset waveform output (SR waveform output) mode (group 0 to 3)
- Bit modulation PWM output mode (group 2 and 3)
- Real-time port output (RTP output) mode (group 2 and 3)
- Parallel real-time port output (parallel RTP output) mode (group 2 and 3)

Table 1.22.7 lists pin settings of the waveform generation function. Table 1.22.8 lists registers associated with the waveform generation function.

**Table 1.22.7. Pin Settings for Waveform Generation Function**

Pin	Bit and setting		
	PS0 to PS2, PS5 to PS9 registers	PSL0, PSL1, PSL2 registers	PSC register
P64/OUTC21	PS0_4 = 1	PSL0_4 = 1	-
P70/OUTC20	PS1_0 = 1	PSL1_0 = 0	PSC_0 = 1
P71/OUTC22	PS1_1 = 1	PSL1_1 = 0	PSC_1 = 1
P73/OUTC10 <sup>2</sup>	PS1_3 = 1	PSL1_3 = 0	PSC_3 = 1
P74/OUTC11 <sup>2</sup>	PS1_4 = 1	PSL1_4 = 0	PSC_4 = 1
P75/OUTC12 <sup>2</sup>	PS1_5 = 1	PSL1_5 = 1	-
P76/OUTC00 <sup>2</sup>	PS1_6 = 1	PSL1_6 = 0	PSC_6 = 1
P77/OUTC01 <sup>2</sup>	PS1_7 = 1	-	-
P81/OUTC30	PS2_1 = 1	PSL2_1 = 1	-
P82/OUTC32	PS2_2 = 1	PSL2_2 = 0	-
P92/OUTC20	PS3_2 = 1	PSL3_2 = 1	-
P110/OUTC10 <sup>1,2</sup>	PS5_0 = 1	-	-
P111/OUTC11 <sup>1,2</sup>	PS5_1 = 1		
P112/OUTC12 <sup>1,2</sup>	PS5_2 = 1		
P113/OUTC13 <sup>1,2</sup>	PS5_3 = 1		
P120/OUTC30 <sup>1</sup>	PS6_0 = 1	-	-
P121/OUTC31 <sup>1</sup>	PS6_1 = 1		
P122/OUTC32 <sup>1</sup>	PS6_2 = 1		
P123/OUTC33 <sup>1</sup>	PS6_3 = 1		
P124/OUTC34 <sup>1</sup>	PS6_4 = 1		
P125/OUTC35 <sup>1</sup>	PS6_5 = 1		
P126/OUTC36 <sup>1</sup>	PS6_6 = 1		
P127/OUTC37 <sup>1</sup>	PS6_7 = 1		

**Table 1.22.7. Pin Settings (Continued)**

Pin	Bit and setting		
	PS0 to PS2, PS5 to PS9 registers	PSL0, PSL1, PSL2 registers	PSC register
P130/OUTC24 <sup>1</sup>	PS7_0 = 1	-	-
P131/OUTC25 <sup>1</sup>	PS7_1 = 1		
P132/OUTC26 <sup>1</sup>	PS7_2 = 1		
P133/OUTC23 <sup>1</sup>	PS7_3 = 1		
P134/OUTC20 <sup>1</sup>	PS7_4 = 1		
P135/OUTC21 <sup>1</sup>	PS7_5 = 1		
P136/OUTC22 <sup>1</sup>	PS7_6 = 1		
P137/OUTC27 <sup>1</sup>	PS7_7 = 1		
P140/OUTC14 <sup>1,2</sup>	PS8_0 = 1	-	-
P141/OUTC15 <sup>1,2</sup>	PS8_1 = 1		
P142/OUTC16 <sup>1,2</sup>	PS8_2 = 1		
P143/OUTC17 <sup>1,2</sup>	PS8_3 = 1		
P150/OUTC00 <sup>1,2</sup>	PS9_0 = 1	-	-
P151/OUTC04 <sup>1,2</sup>	PS9_1 = 1		
P154/OUTC04 <sup>1,2</sup>	PS9_4 = 1		
P155/OUTC05 <sup>1,2</sup>	PS9_5 = 1		

Notes :

1. This port can be provided in the 144-pin package only.
2. When the CAS bit in the GiBCR1 register is set to "1" (32-bit time measurement function), the OUTC1j pin (j=0 to 7) outputs waveform and the OUTC0k pin (k=0, 1, 4, 5) set above outputs low-order 16-bit waveform.

**Table 1.22.8. Registers Related to the Waveform Generation Function Settings**

Register	Bit	Function
GiPOCRj	MOD2 to MOD0	Select output waveform mode
	PRT	In parallel RTP mode with this channel, set to "1"
	IVL	Select default value
	RLD	Select GiPOj register value reload timing
	RTP <sup>1</sup>	In parallel RTP mode with this channel, set to "1". When this bit is set to "1", the MOD2 to 0 bits are disabled.
	INV	Select inverse output
G2BCR1 G3BCR1	PRP	In parallel RTP mode with this channel, set to "1"
GiPOj	-	Select timing to output waveform inversed
G3MK4 to G3MK7	-	Set masked values of the base timer and G3PO4 to G3PO7 registers
GiFS	FSCj	Set to "0" (waveform generation function) (Group0, 1)
GiFE	IFEj	Set to "1" (enables function on channel j)
G2RTP, G3RTP	RTP0 to RTP7	Set a RTP output value in RTP output or parallel RTP output mode

i = 0 to 3, j = 0 to 7

Bit configuration and function vary depending on which group or channel is used.

Registers associated with the waveform generation function should be set after setting registers associated with the base time.

Note:

1. This is the bit in the G2POCRj and G3POCRj registers only.

### (1) Single-Phase Waveform Output Mode (Group 0 to 3)

Output level of the OUTCij pin is inversed when value of the base timer matches the one of the GiPOj register (i=0 to 3, j=0 to 7). The inversed output level is returned to a default output level when the base timer i reaches "000016". Table 1.22.9 lists specifications of single-phase waveform mode. Figure 1.22.24 lists an example of single-phase waveform mode operation.

**Table 1.22.9. Single-phase Waveform Output Mode Specifications**

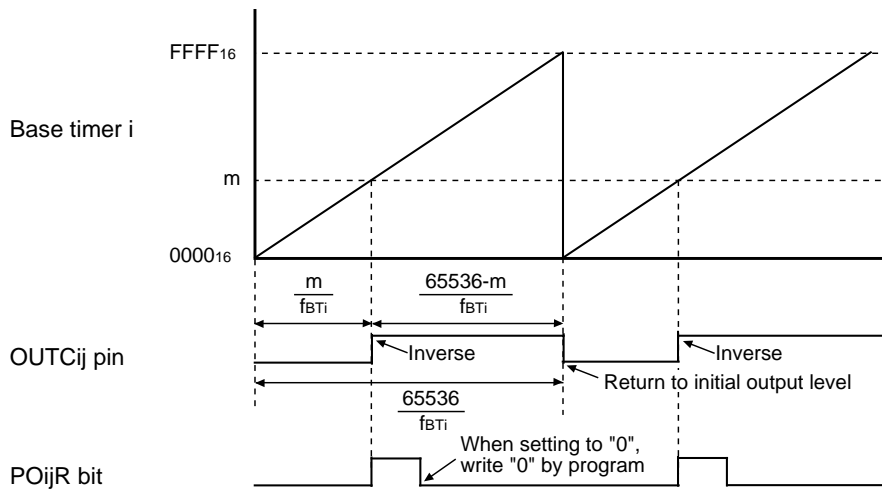
Item	Specification
Output waveform	<ul style="list-style-type: none"> <li>Free-running operation (the RST0 to RST2 bits in the GiBCR1 register (i=0 to 3) is set to "0" (no reset))</li> </ul> <p>Cycle : <math>\frac{65536}{f_{BTi}}</math></p> <p>Default output level : <math>\frac{m}{f_{BTi}}</math></p> <p>Inverse level : <math>\frac{65536-m}{f_{BTi}}</math></p> <ul style="list-style-type: none"> <li>The base timer is reset when the base timer matches the GiPO0 register (the RST1 bit is set to "1" and both RST0 and RST2 bits is to "0")</li> </ul> <p>Cycle : <math>\frac{n+2}{f_{BTi}}</math></p> <p>Default output level : <math>\frac{m}{f_{BTi}}</math></p> <p>Inverse level : <math>\frac{n+2-m}{f_{BTi}}</math></p> <p>m : setting value of the GiPOj register (j=0 to 7), 000116 to FFFD16 n : setting value of the GiPO0 register, 000116 to FFFD16  </p>
Waveform output start condition	The IFEj bit in the GiFE register should be set to "1" (channel j function enabled)
Waveform output stop condition	The IFEj bit should be set to "0" (channel j function disabled)
Interrupt request	The POijR bit in the interrupt request register is set to "1" when value of the base timer matches the one of the GiPOj register. (See Figure 1.9.14.)
OUTCij <sup>1</sup> pin	Pulse output
Selectable function	<ul style="list-style-type: none"> <li>Default value set function : Output level is set when waveform output starts</li> <li>Inverse output function : Waveform level is inversed to output waveform from the OUTCij pin</li> <li>Cascaded connection function : The groups 0 and 1 are connected to operate as a 32-bit timer</li> </ul>

Notes :

1. The OUTC00, OUTC01, OUTC04, OUTC05, OUTC10 to OUTC17, OUTC20 to OUTC27 and OUTC30 to OUTC37 pins (OUTC10 to OUTC17 pins in the groups 0 and 1 cascaded connection)

(1) Free-running operation

(All RST2 to RST0 bits in the GiBCR1 register are set to "0")

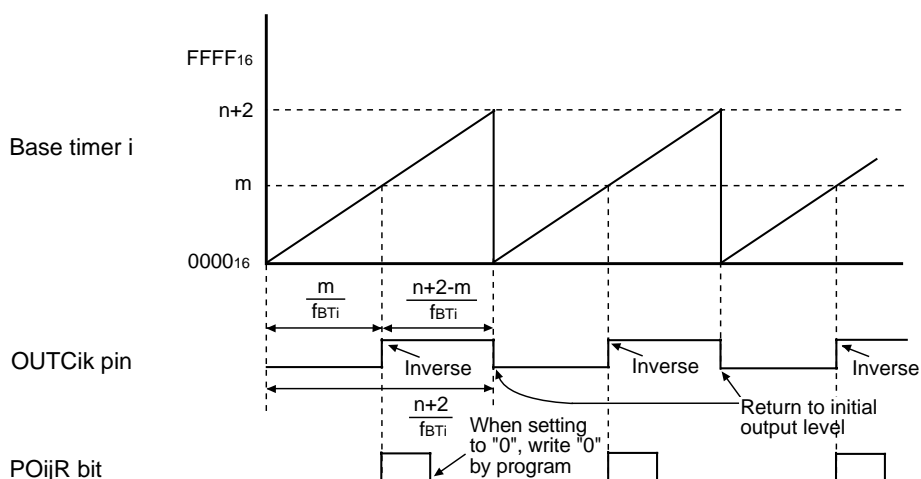


$i=0$  to  $3$ ,  $j=0$  to  $7$  ( $j=0, 1, 4, 5$  when  $i=0$ )  
 $m$  : Setting value of the GiPOj register  
 POijR bit : Bits in the IIO0iR to IIO11iR registers

The above applies to the following conditions.

- The IVL bit in the GiPOCRj register is set to "0" (output "L" as an initial value) and the INV bit is set to "0" (no output inverted).
- All RST2 to RST0 bits in the GiBCR1 register are set to "0" (no base timer reset), the UD1 to UD0 bits be set to "002" (counter increment mode) and the CAS bit be set to "0" (16-bit time measurement).

(2) Base timer is reset when the base timer matches the GiPO0 register  
 (The RST1 bit is set to "1" and both RST0 and RST2 bits are set to "0")



$i=0$  to  $3$ ,  $k=1$  to  $7$  ( $k=1, 4, 5$  when  $i=0$ )  
 $m$  : Setting value of the GiPOj register     $n$  : Setting value of the GiPO0 register  
 POijR bit : Bits in the IIO0iR to IIO11iR registers

The above applies to the following conditions.

- The IVL bit in the GiPOCRk register is set to "0" (output "L" as an initial value) and the INV bit be set to "0" (no output inverse).
- The UD1 to UD0 bits are set to "002" (counter increment mode). The CAS bit is set to "0" (16-bit time measurement).

Figure 1.22.24. Single-phase Waveform Output Mode

## (2) Phase-Delayed Waveform Output Mode (Group 0 to 3)

Output level of the OUTCij pin is inverted whenever value of the base timer matches the one of the GiPOj register value (i=0 to 3, j=0 to 7). Table 1.22.10 lists specifications of phase-delayed waveform mode. Figure 1.22.25 lists an example of phase-delayed waveform mode operation.

**Table 1.22.10. Phase-delayed Waveform Output Mode Specifications**

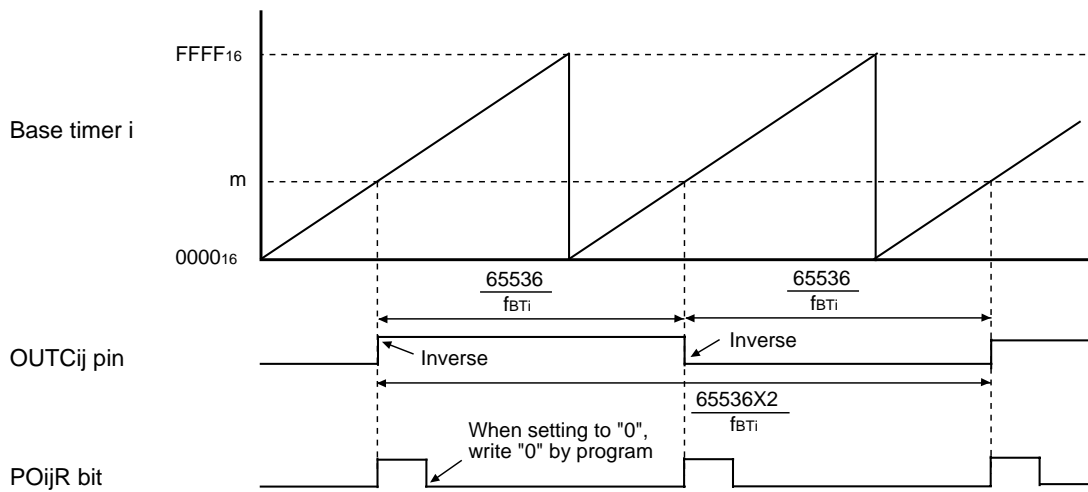
Item	Specification
Output waveform	<ul style="list-style-type: none"> <li>Free-running operation (all RST0 to RST2 bits in the GiBCR1 register (i=0 to 3) is set to "0" (no reset)) Cycle : <math>\frac{65536 \times 2}{f_{BTi}}</math> "H" and "L" width : <math>\frac{65536}{f_{BTi}}</math></li> <li>The base timer is reset when the base timer matches the GiPO0 register (the RST1 bit is set to "1" and both RST0 and RST2 bits is to "0") Cycle : <math>\frac{2(n+2)}{f_{BTi}}</math> "H" and "L" width : <math>\frac{n+2}{f_{BTi}}</math> n : setting value of the GiPO0 register, 0001<sub>16</sub> to FFFD<sub>16</sub></li> </ul>
Waveform output start condition <sup>1</sup>	The IFEj bit in the GiFE register should be set to "1" (channel j function enabled)
Waveform output stop condition	The IFEj bit should be set to "0" (channel j function disabled)
Interrupt request	The POijR bit in the interrupt request register is set to "1" when value of the base timer matches the one of the GiPOj register. (See Figure 1.9.14.)
OUTCij <sup>2</sup> pin	Pulse output
Selectable function	<ul style="list-style-type: none"> <li>Default value set function : Output level is set when waveform output starts</li> <li>Inverse output function : Waveform level is inverted to output waveform from the OUTCij pin</li> <li>Cascaded connection function : The groups 0 and 1 are connected to operate as a 32-bit timer</li> </ul>

Notes :

- The FSCj bit in the GiFS register should be set to "0" (waveform generation function selected) in the channels shared by the time measurement function and waveform generation function.
- The OUTC00, OUTC01, OUTC04, OUTC05, OUTC10 to OUTC17, OUTC20 to OUTC27 and OUTC30 to OUTC37 pins (OUTC10 to OUTC17 pins in group 0 and 1 cascaded connection).

(1) Free-running operation

(All RST2 to RST0 bits in the GiBCR1 register are set to "0")



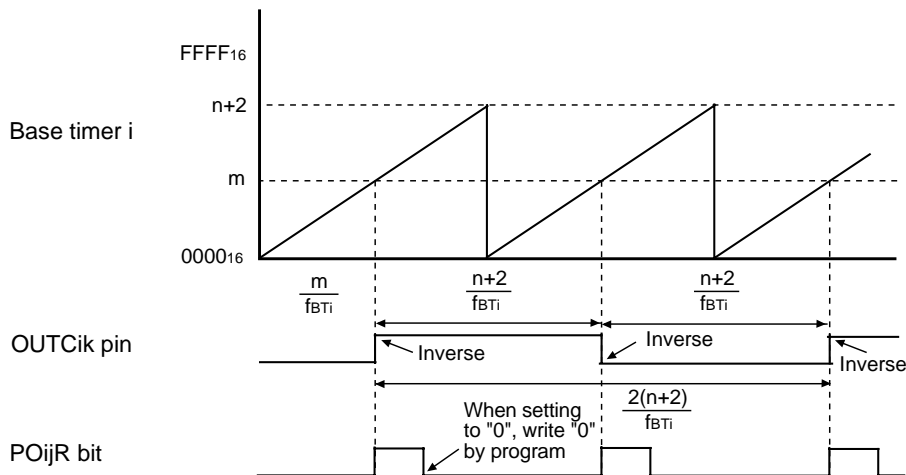
$i=0$  to  $3$ ,  $j=0$  to  $7$  ( $j=0, 1, 4, 5$  when  $i=0$ )  
 $m$  : Setting value of the GiPOj register  
 POijR bit : Bits in the IIO0IR to IIO11IR registers

The above applies to the following conditions.

- The IVL bit in the GiPOCRj register is set to "0" (output "L" as an initial value). The INV bit is set to "0" (no output inverted).
- All RST2 to RST0 bits in the GiBCR1 register are set to "0" (no base timer reset). The UD1 to UD0 bits are set to "002" (counter increment mode). The CAS bit is set to "0" (16-bit time measurement).

(2) Base timer is reset when the base timer matches GiPO0 register

(The RST1 bit should be set to "1" and both RST0 and RST2 bits be set to "0")



$i=0$  to  $3$ ,  $k=1$  to  $7$  ( $k=1, 4, 5$  when  $i=0$ )  
 $m$  : Setting value of the GiPOj register       $n$  : Setting value of the GiPO0 register  
 POijR bit : Bits in the IIO0IR to IIO11IR registers

The above applies to the following conditions.

- The IVL bit in the GiPOCRk register is set to "0" (output "L" as an initial value). The INV bit is set to "0" (no output inverted).
- The UD1 to UD0 bits are set to "002" (counter increment mode). The CAS bit is set to "0" (16-bit time measurement).

Figure 1.22.25. Phase-delayed Waveform Output Mode

### (3) Set/Reset Waveform Output (SR Waveform Output) Mode (Group 0 to 3)

Output level of the OUTCij pin is inversed when value of the base timer matches the one of the GiPOj register value (i=0 to 3, j=0, 2, 4, 6). It is returned to default output level when value of the base timer matches the one of the GiPOk register (k=j+1) and is set to "0". Table 1.22.11 lists specifications of SR waveform mode. Figure 1.22.26 lists an example of the SR waveform mode operation.

**Table 1.22.11. SR Waveform Output Mode Specifications**

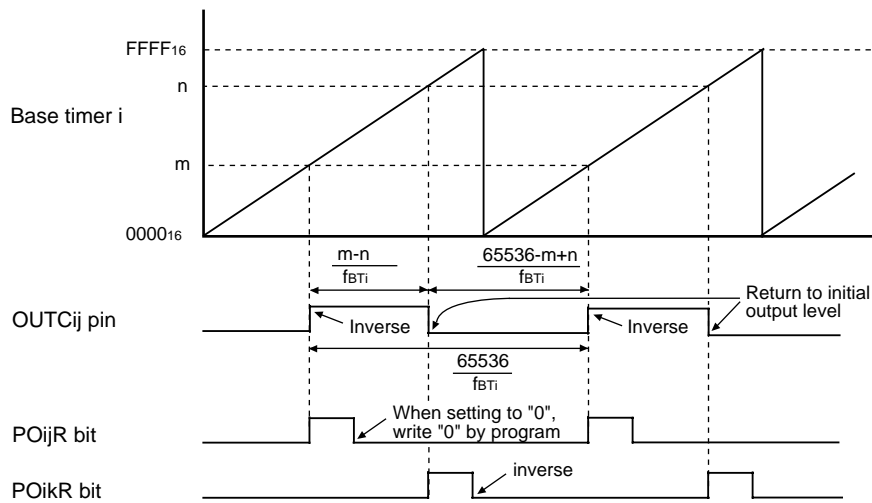
Item	Specification
Output waveform	<ul style="list-style-type: none"> <li>Free-running operation (the RST0 to RST2 bits in the GiBCR1 register (i=0 to 3) is set to "0" (no reset)) Cycle : <math>\frac{65536}{f_{BTi}}</math> Inverse level : <math>\frac{m-n}{f_{BTi}}</math></li> <li>The base timer is reset when the base timer matches the GiPO0 register<sup>1</sup> (the RST1bit is set to "1" and both RST0 and RST2 bits is to "0") Cycle : <math>\frac{p+2}{f_{BTi}}</math> Inverse level : <math>\frac{m-n}{f_{BTi}}</math> m : setting value of the GiPOj register (j=0, 2, 4, 6 ) n : setting value of the GiPOk register (k=j+1) p : setting value of the GiPO0 register all m, n, p: 0001<sub>16</sub> to FFFD<sub>16</sub></li> </ul>
Waveform output start condition <sup>3</sup>	The IFEj bit in the GiFE register should be set to "1" (channel j function enabled)
Waveform output stop condition	The IFEj bit should be set to "0" (channel j function disabled)
Interrupt request	The POijR bit in the interrupt request register is set to "1" when value of the base timer matches the one of the GiPOj register. The POikR bit in the interrupt request register is set to "1" when value of the base timer matches the one of the GiPOk register (See Figure 1.9.14.)
OUTCij <sup>4</sup> pin	Pulse output
Selectable function	<ul style="list-style-type: none"> <li>Default value set function : Output level is set when waveform output starts</li> <li>Inverse output function : Waveform level is inversed to output waveform from the OUTCij pin</li> <li>Cascaded connection function : The groups 0 and 1 are connected to operate as a 32-bit timer</li> </ul>

Notes :

- When the GiPO0 register resets the base timer, the SR waveform generation function with channels 0 and 1 cannot be used.
- The waveform generation register of odd channel should have greater value than the one of even channel has.
- The FSCj bit in the GiFS register should be set to "0" (waveform generation function selected) in the channels shared by the time measurement function and waveform generation function.
- The OUTC0<sub>0</sub>, OUTC0<sub>4</sub>, OUTC1<sub>0</sub>, OUTC1<sub>2</sub>, OUTC1<sub>4</sub>, OUTC1<sub>6</sub>, OUTC2<sub>0</sub>, OUTC2<sub>2</sub>, OUTC2<sub>4</sub>, OUTC2<sub>6</sub>, OUTC3<sub>0</sub>, OUTC3<sub>2</sub>, OUTC3<sub>4</sub> and OUTC3<sub>6</sub> pins (OUTC1<sub>0</sub>, OUTC1<sub>2</sub>, OUTC1<sub>4</sub>, OUTC1<sub>6</sub> pins in the groups 0 and 1 cascaded connection).

(1) Free-running operation

(All RST2 to RST0 bits in the GiBCR1 register are set to "0")



$i=0$  to 3,  $j=0, 2, 4, 6$  ( $j=0, 4$  when  $i=0$ )  $k=j+1$

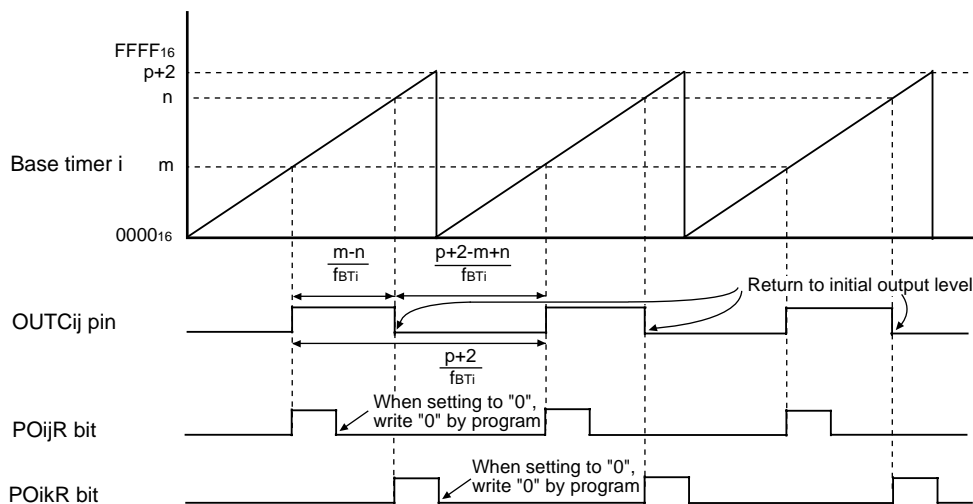
$m$  : Setting value of the GiPOj register       $n$  : Setting value of the GiPOk register

POijR, POikR bits : Bits in the IIO0iR to IIO11iR registers

The above applies to the following conditions.

- The IVL bit in the GiPOCRj register is set to "0" (output "L" as an initial value). The INV bit is set to "0" (no output inverted).
- All RST0 to RST2 bits in the GiBCR1 register are set to "0" (no base timer reset). The UD1 to UD0 bits are set to "002" (counter increment mode). The CAS bit is set to "0" (16-bit time measurement).

(2) Base timer is reset when the base timer matches the GiPO0 register  
(the RST bit is set to "1" and both RST0 and RST bits are set to "0")



$i=0$  to 3,  $j=2, 4, 6$  ( $j=4$  when  $i=0$ )  $k=j+1$

$m$  : Setting value of the GiPOj register       $n$  : Setting value of the GiPOk register

$p$  : Setting value of the GiPO0 register

POijR bit : Bits in the IIO0iR to IIO11iR registers

The above applies to the following conditions.

- The IVL bit in the GiPOCRk register is set to "0" (output "L" as an initial value). The INV bit is set to "0" (no output inverted).
- The UD1 to UD0 bits are set to "002" (counter increment mode). The CAS bit is set to "0" (16-bit time measurement).

Figure 1.22.26. SR Waveform Output Mode

**(4) Bit Modulation PWM Output Mode (Group 2 and 3)**

Output waveform "L" and "H" width can be changed in bit PWM output mode, "L" width become longer and "H" width is shorter by one cycle of  $f_{BTi}$  every  $1024/m$  ( $m$  : low-order 10 bits in the GiPOj register).

**Table 1.22.12. Bit Modulation PWM Mode Specifications**

Item	Specification
Output waveform	When $\frac{64}{f_{BTi}}$ width is defined as one leg, "L" width of $\frac{n+1}{f_{BTi}}$ is out put in $m$ out of 1024 legs. "L" width of $\frac{n}{f_{BTi}}$ is output in $(1024-m)$ legs. Average "L" width : $\frac{64}{f_{BTi}} \times (n + \frac{n}{f_{BTi}})$ $n$ : setting value (6 high-order bits) of the GiPOj register ( $i=2$ to $3$ , $j=0$ to $7$ ) $m$ : setting value (10 low-order bits) of the GiPOj register $m, n$ : 0001 <sub>16</sub> to FFFD <sub>16</sub>
Waveform output start condition	The IFEj bit in the GiFE register should be set to "1" (channel j function enabled)
Waveform output stop condition	The IFEj bit should be set to "0" (channel j function disabled)
Interrupt request	The POijR bit in the interrupt request register is set to "1" when value of the base timer matches the one of the GiPOj register (see Figure 1.9.14).
OUTCij pin	Pulse output
Selectable function	<ul style="list-style-type: none"> <li>• Default value set function : Output level is set when waveform output starts</li> <li>• Inverse output function : Waveform level is inversed to output waveform from the OUTCij pin</li> </ul>

**Table 1.22.13. Modulation-added Leg and Minimum Bit Width Leg t (with Free-Running Operation)**

Number of modulation-added leg	Minimum bit-added leg
00 0000 0000 <sub>2</sub>	none
00 0000 0001 <sub>2</sub>	t512
00 0000 0010 <sub>2</sub>	t256, t768
00 0000 0100 <sub>2</sub>	t128, t384, t640, t896
00 0000 1000 <sub>2</sub>	t64, t192, t320, t448, t576, t704, t832, t960
⋮	⋮
10 0000 0000 <sub>2</sub>	t1, t3, t5, t7, ... t1019, t1021, t1023

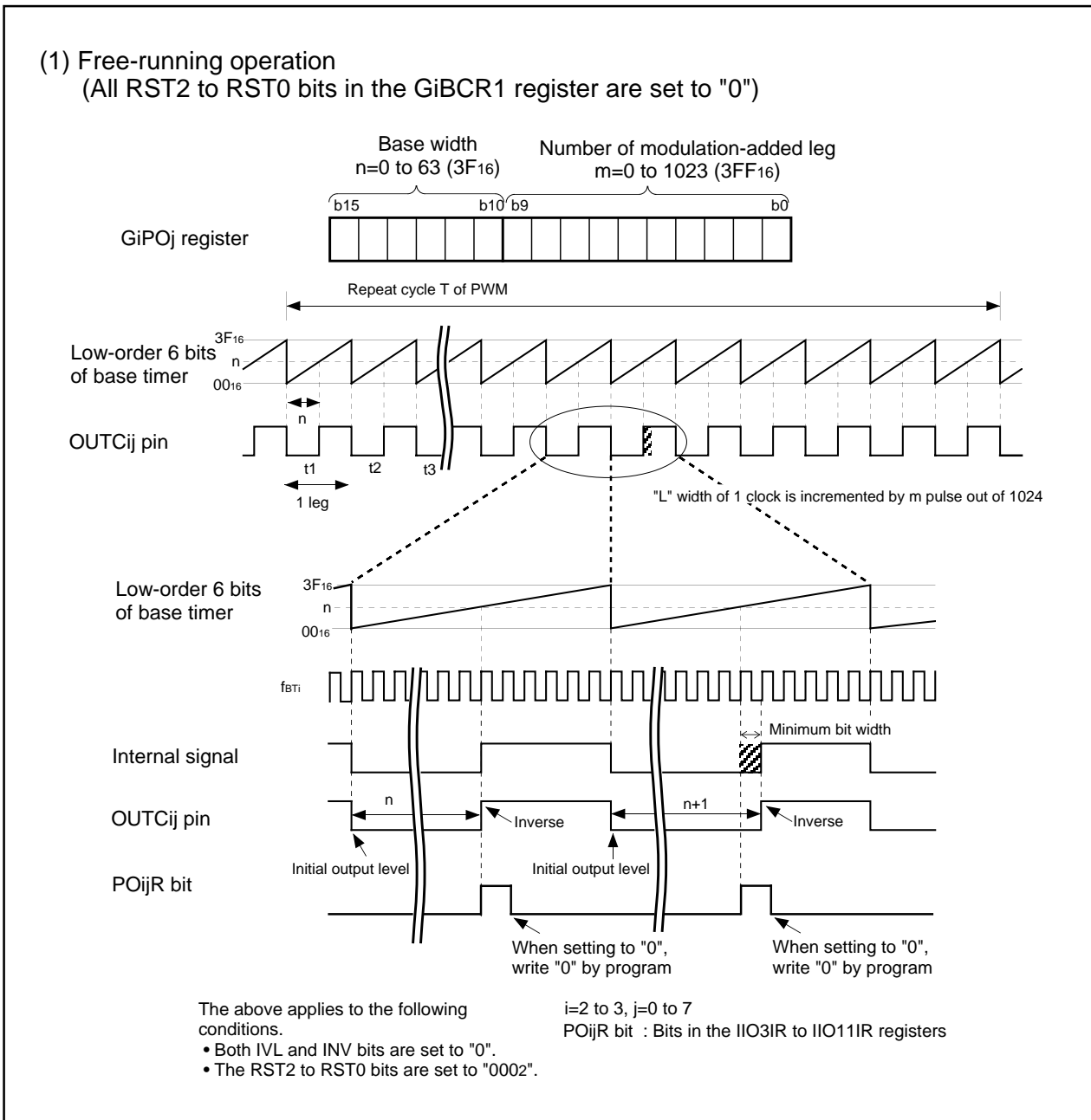


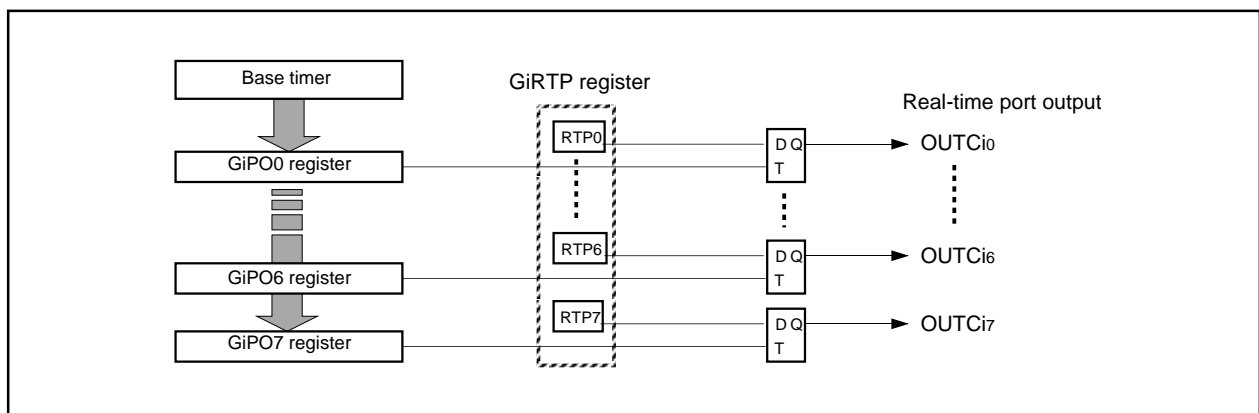
Figure 1.22.27. Bit Modulation PWM mode

**(5) Real-Time Port (RTP) Output Mode (Group 2 and 3)**

The OUTCij pin outputs value set by the GiRTP register when value of the base timer matches the one of the GiPOj register (i=2 to 3, j=0 to 7). Table 1.22.14 lists specifications of RTP output mode. Figure 1.22.28 shows a block diagram of the RTP output function. Figure 1.22.29 shows an example of RTP output mode operation.

**Table 1.22.14. RTP Output Mode Specifications**

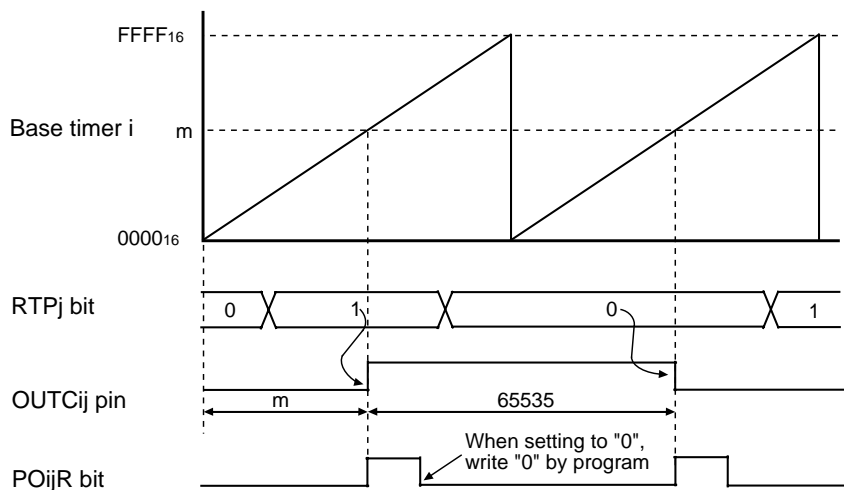
Item	Specification
Waveform output start condition	The IFEj bit in the GiFE register (i=2 to 3, j=0 to 7) should be set to "1" (channel j function enabled)
Waveform output stop condition	The IFEj bit should be set to "0" (channel j function disabled)
Interrupt request	The POijR bit in the interrupt request register is set to "1" when value of the base timer matches the one of the GiPOj register (0001 <sub>16</sub> to FFFD <sub>16</sub> ). (See Figure 1.9.14.)
OUTCij pin	RTP output
Selectable function	<ul style="list-style-type: none"> <li>• Default value set function : Output level is set when waveform output starts.</li> <li>• Inverse output function : Waveform level is inverted to output waveform from the OUTCij pin</li> </ul>



**Figure 1.22.28. Real-time Port Output Function Block Diagram**

(1) Free-running operation

(All RST2 to RST0 bits in the GiBCR1 register are set to "0")



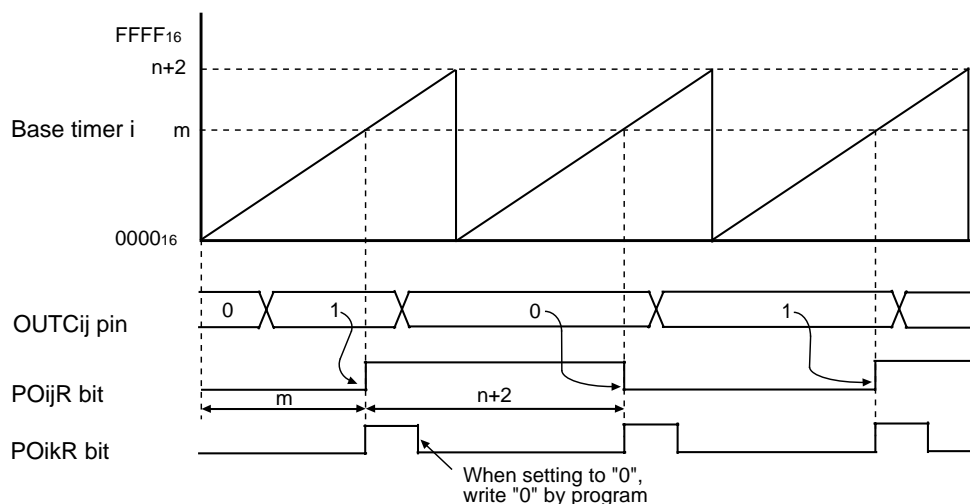
$i=2$  to  $3$ ,  $j=0$  to  $7$   
 $m$  : Setting value of the GiPOj register  
 POijR bit : Bits in the IIO3IR to IIO11IR registers

The above applies to the following condition

- The IVL bit in the GiPOCRj register is set to "0" (output "L" as an initial value). The INV bit is set to "0" (no output inverse).
- All RST0 to RST2 bits in the GiBCR1 register are set to "0" (no base timer reset).

(2) The base timer is reset when the base timer matches the GiPO0 register

(The RST1 bit is set to "1" and both RST0 and RST2 bits are set to "0")



$i=2$  to  $3$ ,  $j=1$  to  $7$   
 $m$  : Setting value of the GiPOj register       $n$  : Setting value of the GiPO0 register  
 POijR bit : Bits in the IIO0IR to IIO11IR registers

The above applies to the following condition

- The IVL bit in the GiPOCRj register is set to "0" (output "L" as an initial value). The INV bit is set to "0" (no output inverse).

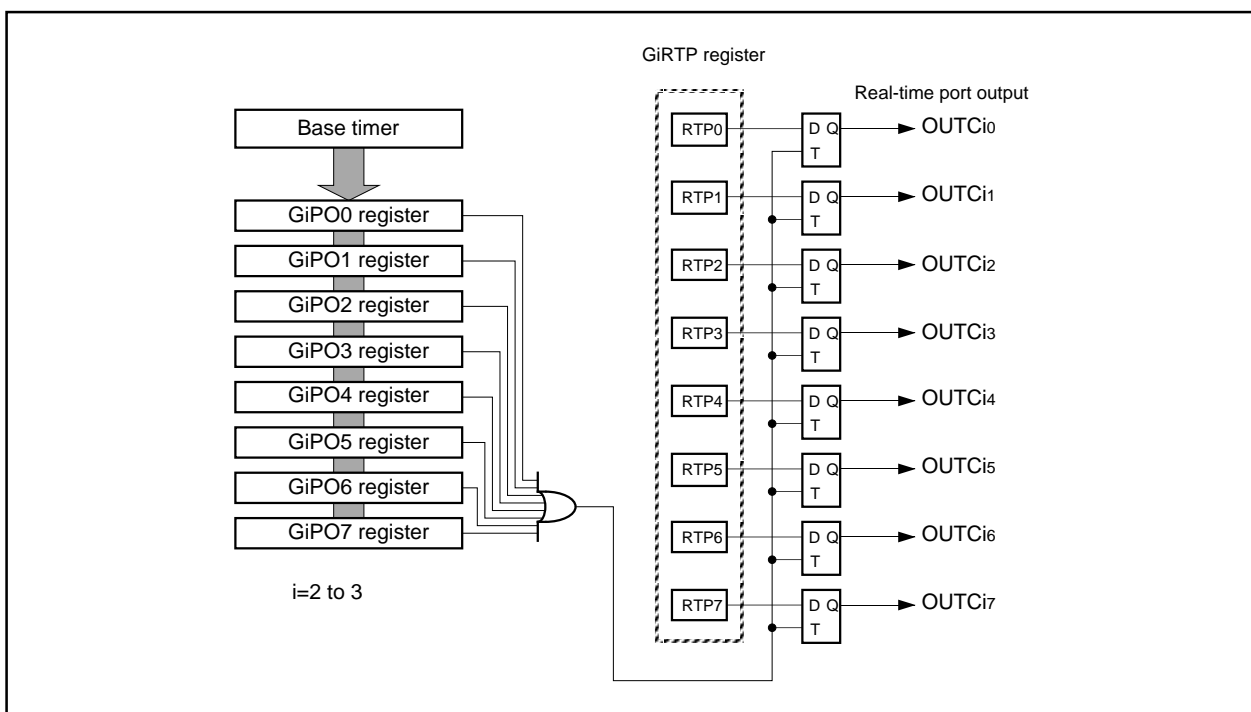
Figure 1.22.29. Real-time Port Output Mode

**(6) Parallel Real-Time Port Output Mode (Group 2 and 3)**

The OUTCij pin outputs value set by the GiRTP register when value of the base timer matches the one of the GiPOj register (i=2 to 3, j=0 to 7). Table 1.22.15 lists specifications of the parallel RTP output mode. Figure 1.22.30 shows a block diagram of the parallel RTP output function. Figure 1.22.31 shows an example of the parallel RTP output mode operation. (See Figure 1.22.7 about the G2BCR1 register and Figure 1.22.8 about the G3BCR1 register.)

**Table 1.22.15. Parallel RTP Output Mode Specifications**

Item	Specification
Waveform output start condition	The IFEj bit in the GiFE register (i=2 to 3, j=0 to 7) should be set to "1" (channel j function enabled)
Waveform output stop condition	The IFEj bit should be set to "0" (channel j function disabled)
Interrupt request	The POijR bit in the interrupt request register is set to "1" when value of the base timer matches the one of the GiPOj register value (0001 <sub>16</sub> to FFFD <sub>16</sub> ). (See Figure 1.9.14.)
OUTCij pin	RTP output
Selectable function	<ul style="list-style-type: none"> <li>• Default value set function: Output level is set when waveform output starts.</li> <li>• Inverse output function: Waveform level is inversed to output waveform from the OUTCij pin</li> </ul>



**Figure 1.22.30. Parallel RTP Output Function Block Diagram**

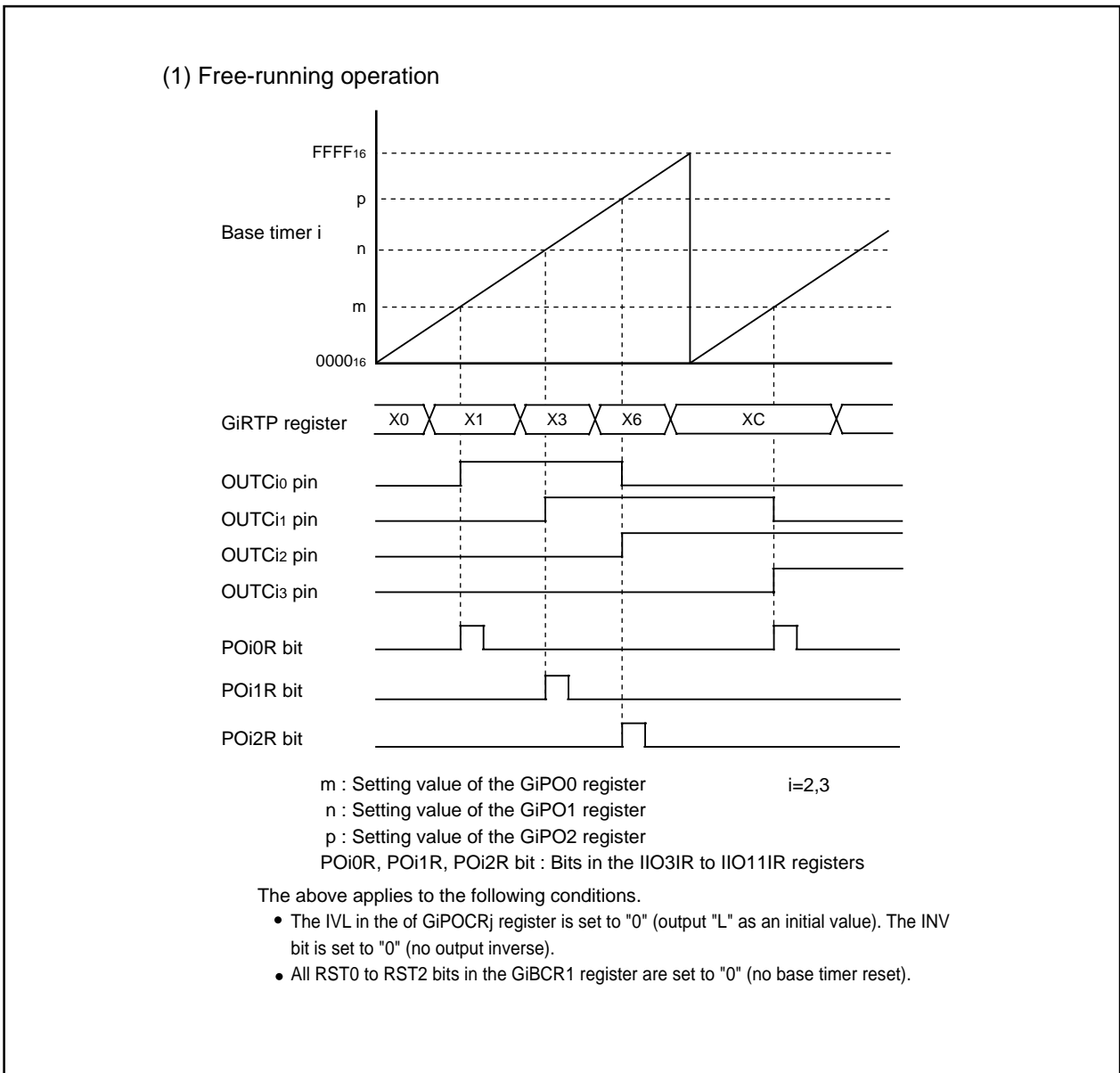


Figure 1.22.31. Parallel RTP Output Mode

### Communication Function (Group 0 to 3)

The communication function is available when two 8-bit shift registers use with either timer measurement function or waveform generation function.

#### (1) Group 0 and 1 Communication Function

In the intelligent I/O groups 0 and 1, 8-bit clock synchronous serial I/O, 8-bit clock asynchronous serial I/O (UART) or HDLC data processing is available.

Figures 1.22.32 to 1.22.38 show registers associated with the communication function.

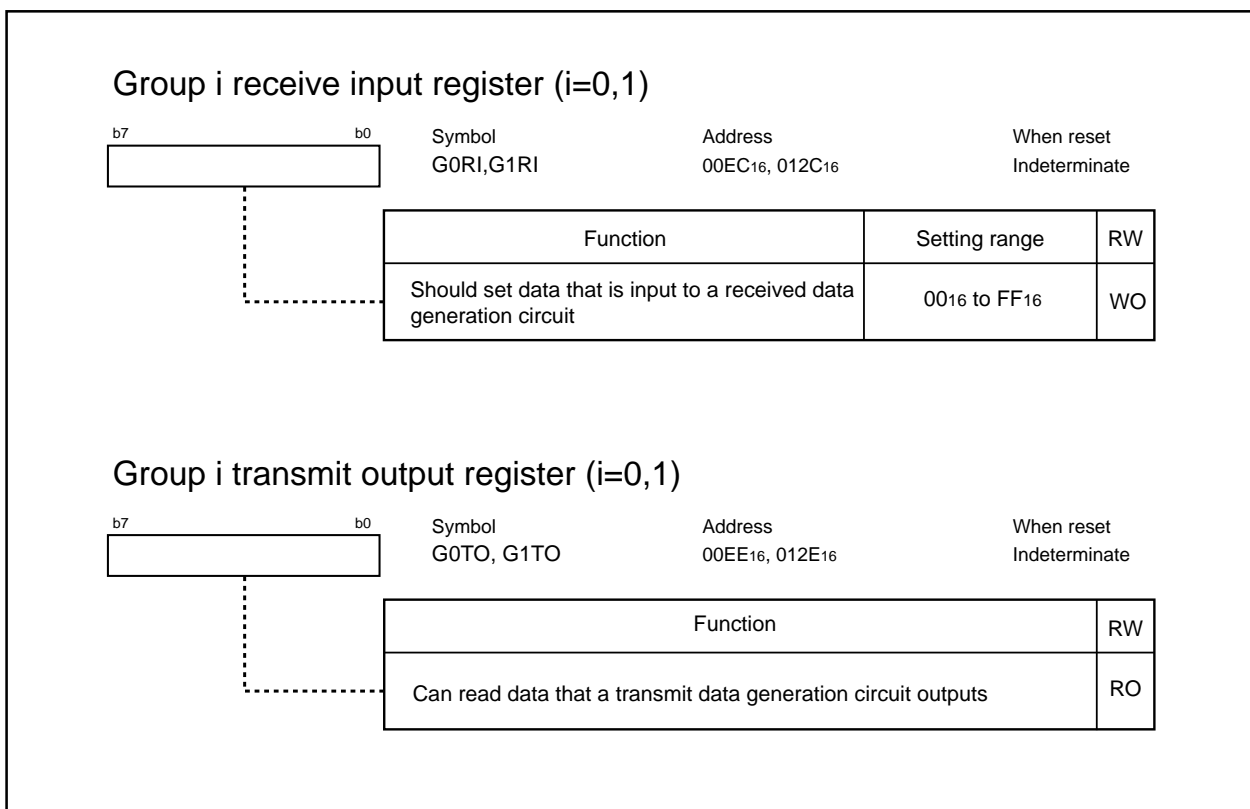
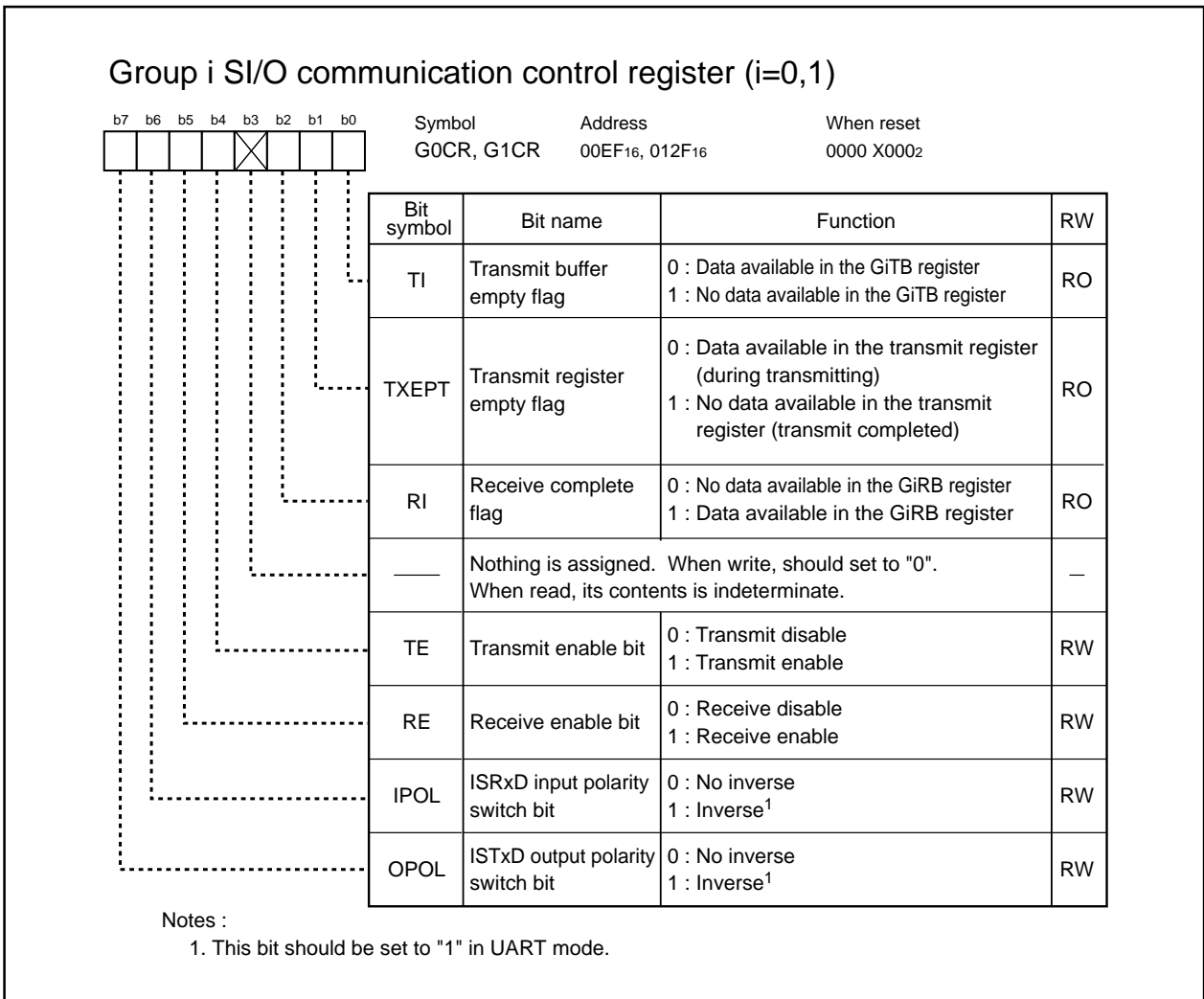


Figure 1.22.32. G0RI to G1RI Registers and G0TO to G1TO Registers



**Figure 1.22.33. G0CR to G1CR Registers**

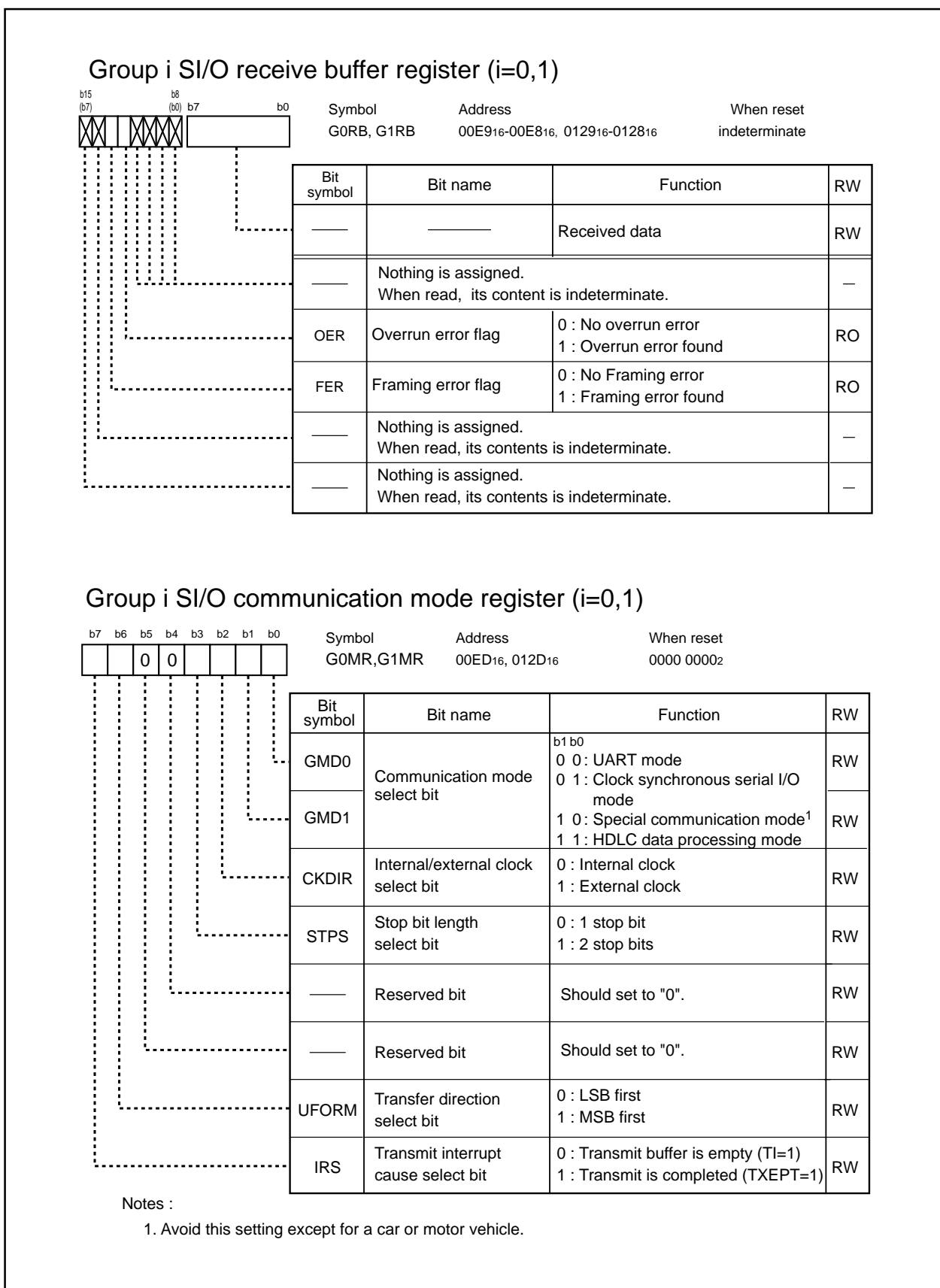


Figure 1.22.34. G0RB to G1RB Registers and G0MR to G1MR Registers

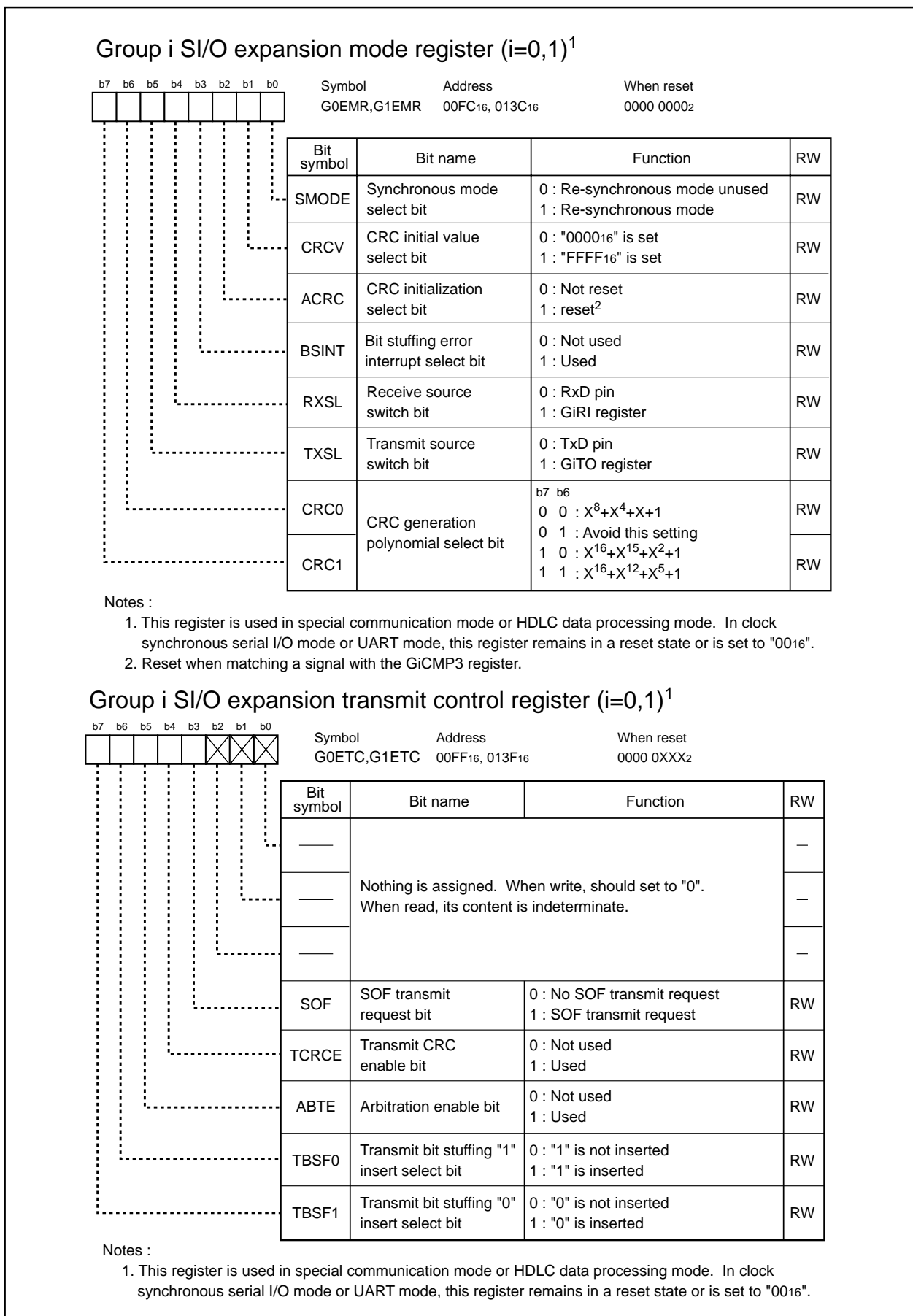


Figure 1.22.35. G0EMR to G1EMR Registers and G0ETC to G1ETC Registers

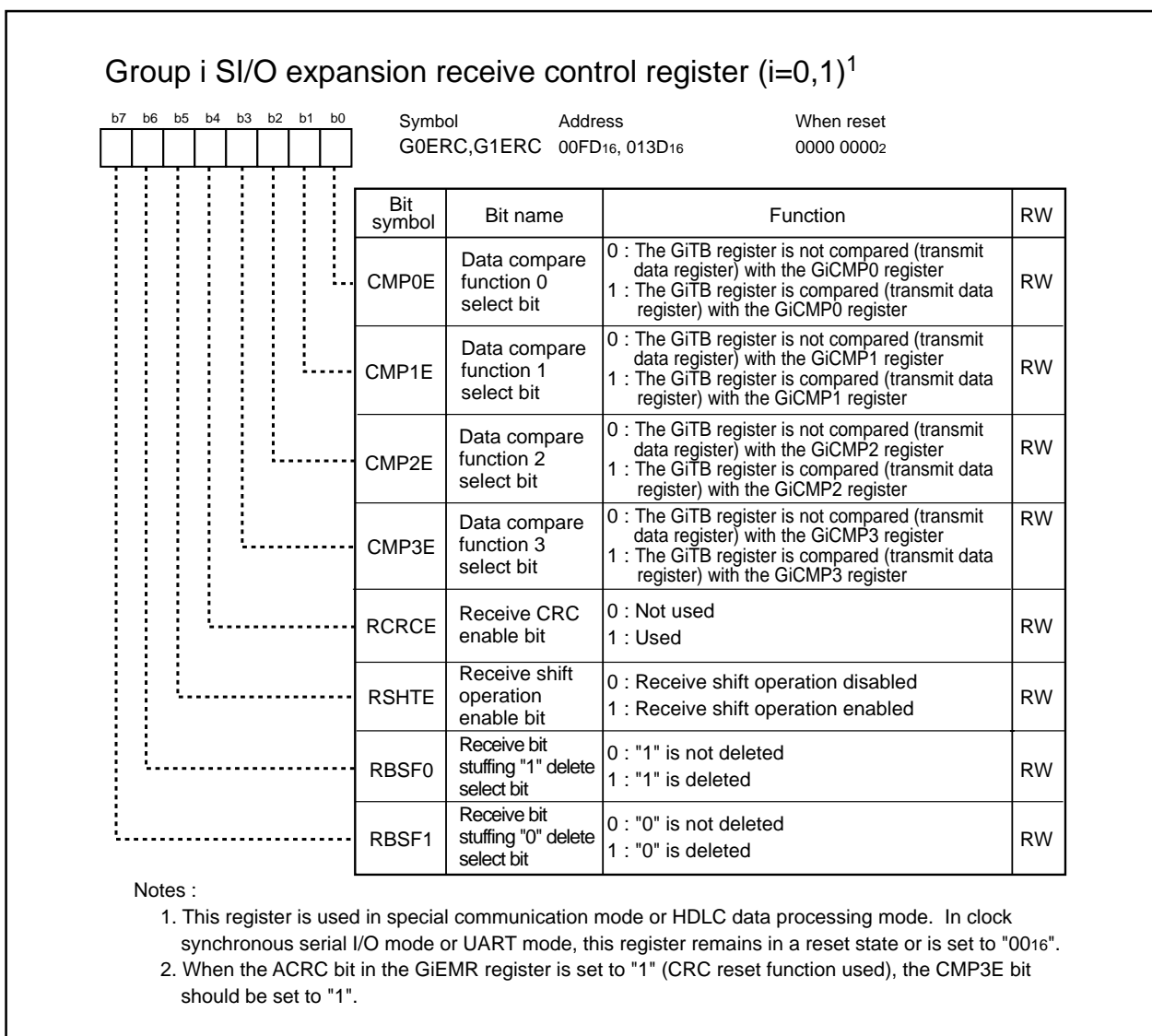


Figure 1.22.36. G0ERC to G1ERC Registers

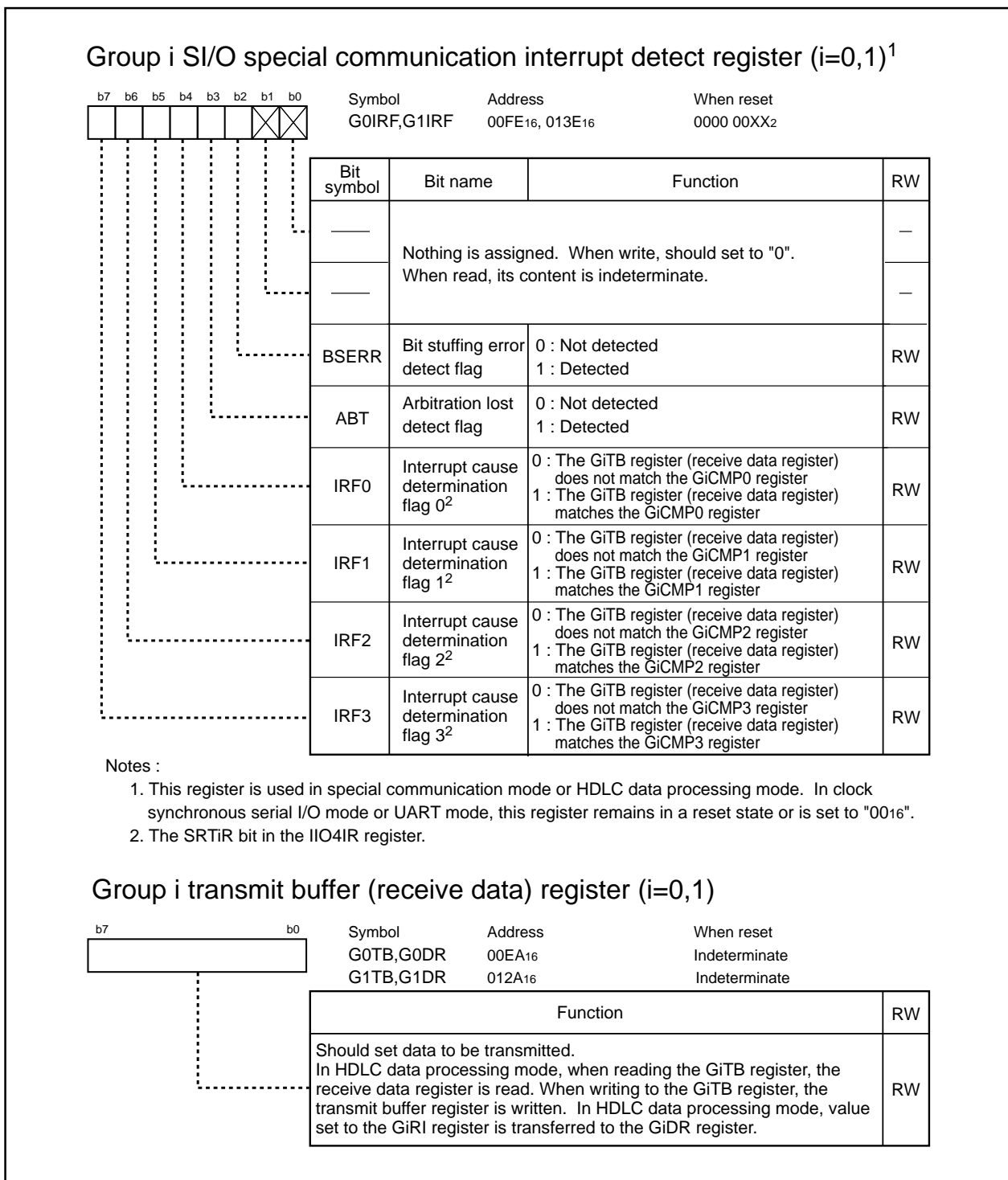


Figure 1.22.37. G0IRF to G1IRF registers and G0TB to G1TB Registers

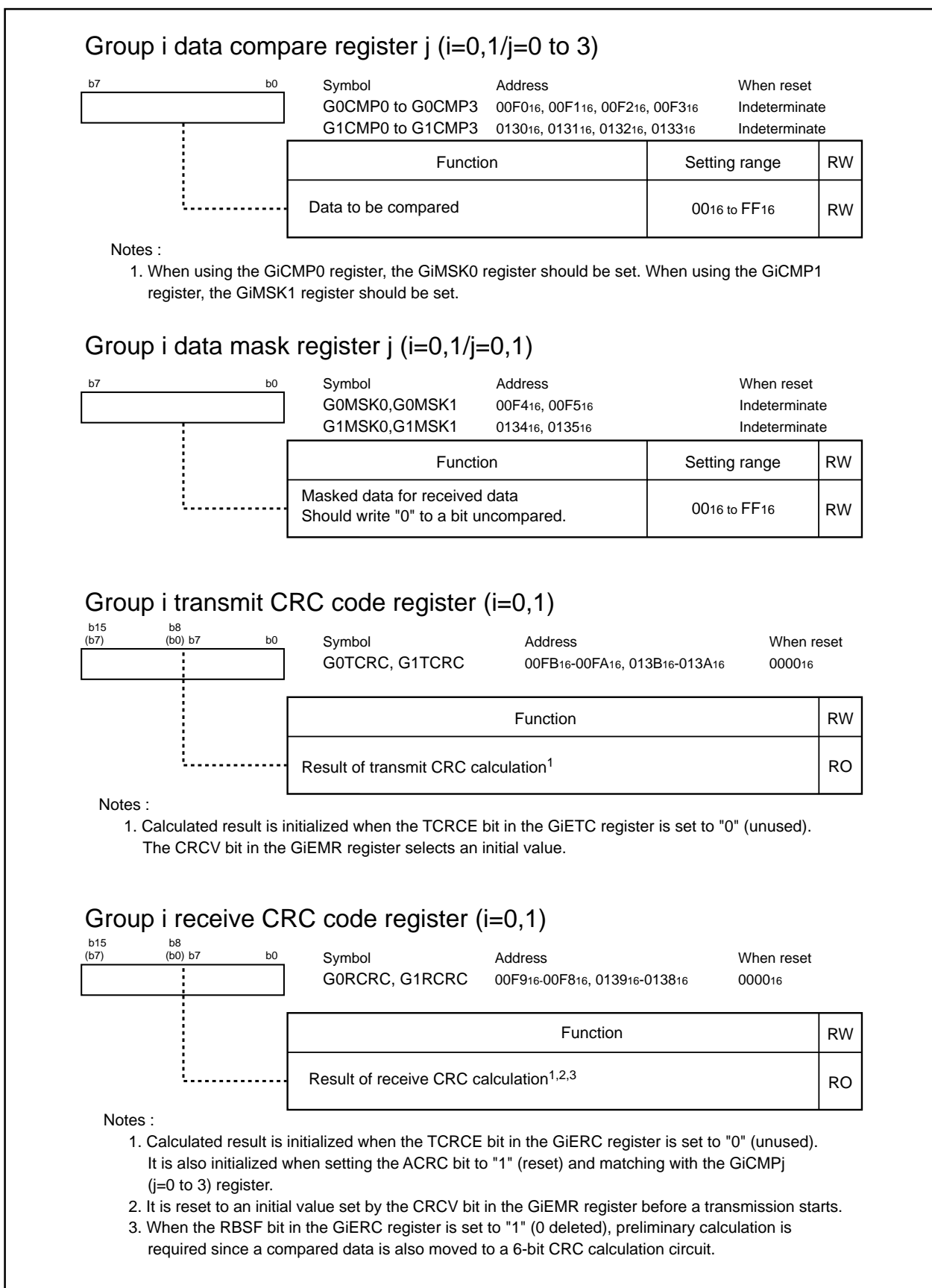


Figure 1.22.38. G0CMP0 to G0CMP3 and G1CMP0 to G1CMP3 Registers, G0MSK0 to G0MSK1 and G1MSK0 to G1MSK1 Registers, G0TCRC to G1TCRC Registers and G0RCRC to G1RCRC Registers

### • Clock Synchronous Serial I/O Mode (Group 0 and 1)

In clock synchronous serial I/O mode, data is transmitted and received with the synchronous clock. When the internal clock is selected as the synchronous clock, channels 0 and 3 generate the internal clock and share pins with ISTxDi, ISCLKi, ISRxDi, INPCi0 to INPCi2 or OUTCi0 to OUTCi2.

Table 1.22.16 lists specifications of clock synchronous serial I/O mode group 0 and 1. Table 1.22.17 lists registers and to be used settings. Tables 1.22.18 to 1.22.21 list pin settings. Figure 1.22.39 shows an example of transmit and receive operation.

**Table 1.22.16. Clock Synchronous Serial I/O Mode Specifications (Group 0 and 1)**

Item	Specification
Transfer data format	• Transfer data : 8 bits long
Transfer clock <sup>1,2</sup>	<ul style="list-style-type: none"> <li>• When the CKDIR bit in the GiMR register (i=0, 1) is set to "0" (internal clock) : <math>\frac{f_{BTi}}{2(n+2)}</math>  n : setting value of the GiPO0 register, 0000<sub>16</sub> to FFFF<sub>16</sub> <ul style="list-style-type: none"> <li>– The GiPO0 register determines the baud rate and the transfer clock is generated with the channel 3 waveform generation function and in phase-delayed waveform output mode</li> </ul> </li> <li>• When the CKDIR bit is set to "1" (external clock) : input from the ISCLK1 pin</li> </ul>
Transmit start condition	Registers associated with the waveform generation function, the GiMR register and GiERC register should be set and the following values be set after one cycle. <ul style="list-style-type: none"> <li>• The TE bit in the GiCR register is set to "1" (transmit enable)</li> <li>• The TI bit in the GiCR register is set to "0" (data in the GiTB register)</li> </ul>
Receive start condition	Registers associated with the waveform generation function, the GiMR register and GiERC register should be set and the following values be set after one cycle. <ul style="list-style-type: none"> <li>• The RE bit in the GiCR register is set to "1" (receive enable)</li> <li>• The TE bit is set to "1" (transmit enable)</li> <li>• The TI bit is set to "0" (data in the GiTB register)</li> </ul>
Interrupt request	<ul style="list-style-type: none"> <li>• While transmitting, the following condition can be selected to set the SIOiTR bit to "1" (see Figure 1.9.14) : <ul style="list-style-type: none"> <li>– When the IRS bit in the GiMR register is set to "0" (interrupt with the GiTB register empty) and data is transferred to the transmit register from the GiTB register</li> <li>– When the IRS bit is set to "1" (interrupt at reception completed) and data transfer from the transmit register is completed</li> </ul> </li> <li>• While receiving  When data is transferred to the GiRB register from the receive register (reception completed), the SIOiRR bit is set to "1" (see Figure 1.9.14)</li> </ul>
Error detection	Overrun error <sup>3</sup> This error occurs when receiving the 8th bit of the next data before reading the GiRB register
Selectable function	<ul style="list-style-type: none"> <li>• LSB first/MSB first  Whether data is transmitted/received in bit 0 or in bit 7 can be selected</li> <li>• ISTxDi and ISRxDi I/O polarity inverse  ISTxDi pin output and ISRxDi pin input levels are inverted</li> </ul>

Notes :

1. The transfer clock should be  $f_{BTi}$  divided by six or more.
2. In clock synchronous serial I/O mode, the RSHTe bit in the GiERC register (i=0, 1) should be set to "1".
3. When an overrun error occurs, the GiRB register is indeterminate.

When the OPOL bit in the GiCR register is set to "0" (no TxD output polarity inverted), the ISTxD pin outputs "H" between a selection of operation mode and a start of a transfer. When the OPOL bit is set to "1", the ISTxD pin outputs "L".

**Table 1.22.17. Registers to be Used and Settings**

Register	Bit	Function
GiBCR0	BCK1 to BCK0	Set to "112"
	DIV4 to DIV0	Select a divide ratio of a count source
	IT	Set to "0"
GiBCR1	7 to 0	Set to "0001 0010 <sub>2</sub> "
GiPOCR0	7 to 0	Set to "0000 0111 <sub>2</sub> "
GiPOCR1	7 to 0	Set to "0000 0111 <sub>2</sub> "
GiPOCR3	7 to 0	Set to "0000 0010 <sub>2</sub> " <sup>1</sup>
GiPO0	15 to 0	Set a baud rate $\frac{f_{BTi}}{2 \times (\text{setting value} + 2)} = \text{transfer clock frequency}^1$
GiPO3	15 to 0	Set a smaller value than the GiPO0 register <sup>1</sup>
GiFS	FSC3,1,0	Set to "0"
GiFE	IFE3,1,0	Set to "1"
GiERC	7 to 0	Set to "0010 0000 <sub>2</sub> "
GiMR	GMD1 to GMD0	Set to "01 <sub>2</sub> "
	CKDIR	Select the internal clock or external clock
	STPS	Set to "0"
	UFORM	Select either LSB first or MSB first
	IRS	Select how the transmit interrupt is generated
GiCR	TI	Transmit buffer empty flag
	TXEPT	Transmit register empty flag
	RI	Receive complete flag
	TE	Set to "1" for transmit and receive enable
	RE	Set to "1" for receive enable
	IPOL	Select an ISRxD input polarity (usually set to "0")
	OPOL	Select an ISTxD output polarity (usually set to "0")
GiTB	7 to 0	Write data to be transmitted
GiRB	15 to 0	Received data and error flag are stored

i = 0 to 1

Notes :

1. The CKDIR bit in the GiMR register is set to "0" (internal clock).

**Table 1.22.18. Pin Settings**

Port name	Function	Bit and setting value					register <sup>1</sup>
		PS1 register	PSL1 register	PSC register	PD7 register	IPS register	
P7 <sub>3</sub>	ISTxD1 output	PS1_3 = 1	PSL1_3 = 0	PSC_3 = 1	-	-	G1POCR0
P7 <sub>4</sub>	ISCLK1 input	PS1_4 = 0	-	-	PD7_4 = 0	IPS1 = 0	-
	ISCLK1 output	PS1_4 = 1	PSL1_4 = 0	PSC_4 = 1	-	-	G1POCR1
P7 <sub>5</sub>	ISRxD1 input	PS1_5 = 0	-	-	PD7_5 = 0	IPS1 = 0	-
P7 <sub>6</sub>	ISTxD0 output	PS1_6 = 1	PSL1_6 = 0	PSC_6 = 0	-	-	G0POCR0
P7 <sub>7</sub>	ISCLK0 input	PS1_7 = 0	-	-	PD7_7	IPS0 = 0	-
	ISCLK0 output	PS1_7 = 1	-	-	-	-	G0POCR1

Notes :

1. The MOD2 to MOD0 bits in the corresponding register should be set to "111<sub>2</sub>" (output of the communication function is used).

**Table 1.22.19. Pin Settings (Continued)**

Port name	Function	Bit and setting			register
		PS2 register	PD8 register	IPS register	
P8 <sub>0</sub>	ISRxD0 input	PS2_0 = 0	PD8_2 = 0	IPS0 = 0	-

**Table 1.22.20. Pin Settings (Continued)**

Port name	Function	Bit and setting			register <sup>1</sup>
		PS5 register	PD11 register	IPS register	
P110	ISTxD1 output	PS5_0 = 1	-	-	G1POCR0
P111	ISCLK1 input	PS5_1 = 0	PD11_1 = 0	IPS1 = 1	-
	ISCLK1 output	PS5_1 = 1	-	-	G1POCR1
P112	ISRxD1 input	PS5_2 = 0	PD11_2 = 0	IPS1 = 1	-

Notes :

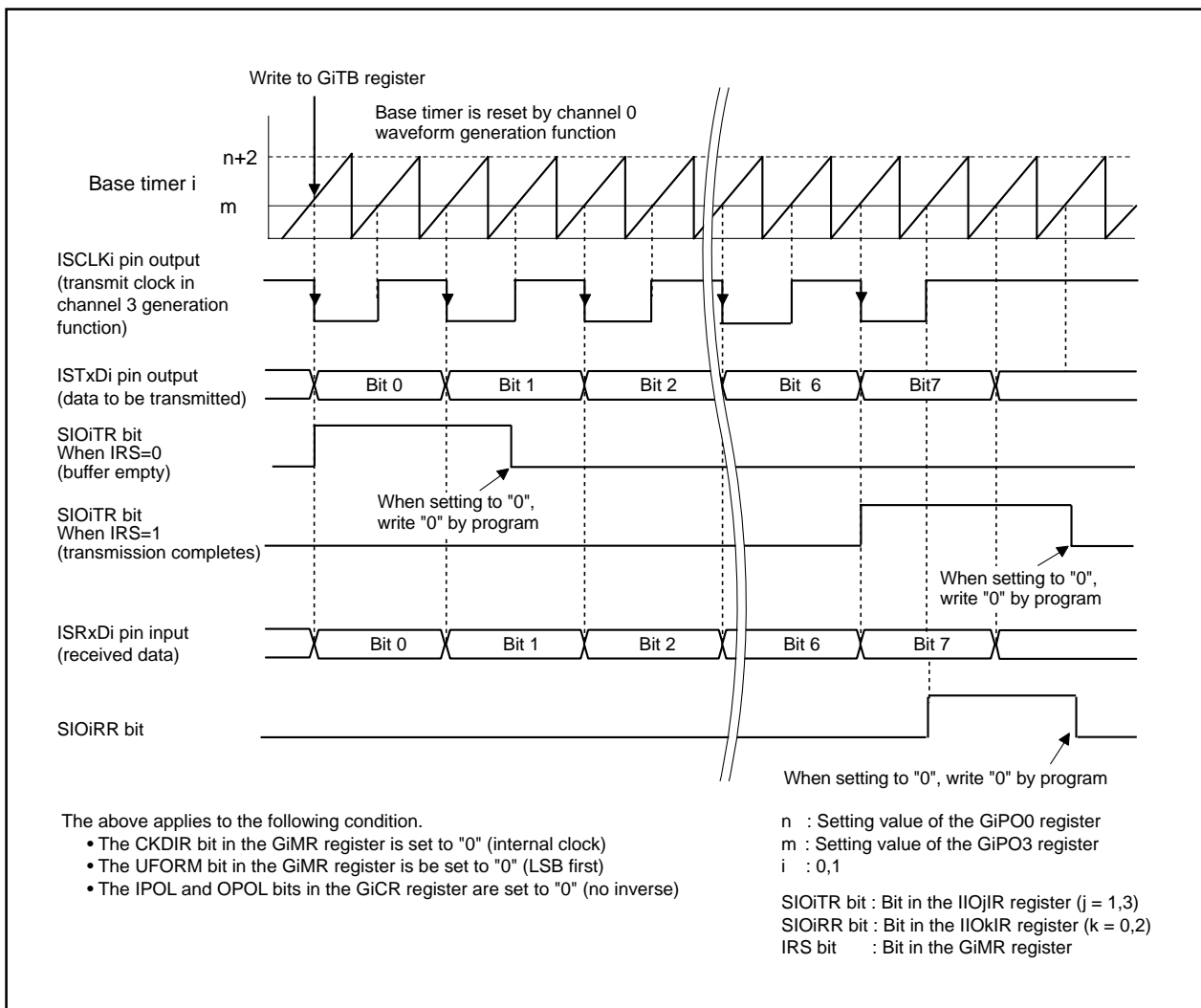
1. The MOD2 to MOD0 bits in the corresponding register should be set to "1112" (output of the communication function is used).

**Table 1.22.21. Pin Settings (Continued)**

Port name	Function	Bit and setting			register <sup>1</sup>
		PS9 register	PD15 register	IPS register	
P150	ISTxD1 output	PS9_0 = 1	-	-	G0POCR0
P151	ISCLK1 input	PS9_1 = 0	PD15_2 = 0	IPS0 = 1	-
	ISCLK1 output	PS9_1 = 1	-	-	G0POCR1
P152	ISRxD1 input	PS9_2 = 0	PD15_2 = 0	IPS0 = 1	-

Notes :

1. The MOD2 to MOD0 bits in the corresponding register should be set to "1112" (output of the communication function is used).



**Figure 1.22.39. Transmit and Receive Operation**

### • Clock Asynchronous Serial I/O Mode (UART) (Group 0 and 1)

Table 1.22.22 lists specifications of UART mode group 0 and 1. Table 1.22.23 lists registers to be used and settings. Tables 1.22.24 to 1.22.27 list pin settings. Figure 1.22.40 shows an example of transmit operation. Figure 1.22.41 shows an example of receive operation.

**Table 1.22.22. UART Mode Specifications**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>• Character Bit (transfer data) : 8 bits long</li> <li>• Start bit : 1 bit long</li> <li>• Stop bit : selectable from 1 bit or 2 bits long</li> </ul>
Transfer clock <sup>1,2</sup>	<ul style="list-style-type: none"> <li>• When the CKDIR bit in the GiMR register (i=0, 1) is set to "0" (internal clock) : <math>\frac{f_{BTi}}{2(n+2)}</math>  n : setting value of the GiPO0 register, 0000<sub>16</sub> to FFFF<sub>16</sub>. <ul style="list-style-type: none"> <li>– The he GiPO0 register determines the baud rate and the transfer clock is generated in phase-delayed waveform output mode</li> <li>Transmit clock is generated with the channel 3 waveform generation function.</li> <li>Receive clock is generated with the channel 2 time measurement function.</li> </ul> </li> </ul>
Transmit start condition	<p>Registers associated with the waveform generation function, the GiMR register and GiERC register should be set and the following values be set after one cycle</p> <ul style="list-style-type: none"> <li>• The TE bit in the GiCR register is set to "1" (transmit enable)</li> <li>• The TI bit in the GiCR register is set to "0" (data written to the GiTB register)</li> </ul>
Receive start condition	<p>Registers associated with the waveform generation function, the GiMR register and GiERC register should be set and the following values be set after one cycle</p> <ul style="list-style-type: none"> <li>• The RE bit in the GiCR register is set to "1" (receive enable)</li> <li>• The start bit is detected</li> </ul>
Interrupt request	<ul style="list-style-type: none"> <li>• While transmitting, the following condition can be selected to set the SIOiTR bit to "1" (See Figure 1.9.14.) : <ul style="list-style-type: none"> <li>– When the IRS bit in the GiMR register is set to "0" (interrupt with the GiTB register empty) and data is transferred to the transmit register from the GiTB register.</li> <li>– When the IRS bit is set to "1" (interrupt at reception completed) and data transfer from the transmit register is completed</li> </ul> </li> <li>• While receiving  When data is transferred to the GiRB register from the receive register (receive completed), the SIOiRR bit is set to "1" (see Figure 1.9.14)</li> </ul>
Error detection	<ul style="list-style-type: none"> <li>• Overrun error<sup>3</sup>  This error occurs when receiving the last bit of the next data before reading the GiRB register</li> <li>• Flaming Error  This error occurs when the number of the stop bits set is not detected</li> </ul>
Selectable function	<ul style="list-style-type: none"> <li>• Stop bit length  The stop bit is selectable from 1 bit or 2 bits long</li> <li>• LSB first/MSB first  Whether data is transmitted/received in bit 0 or in bit 7 can be selected</li> </ul>

Notes :

1. The transfer clock should be  $f_{BTi}$  divided by six or more.
2. The GiPOCRj register and the GiTMCRj register should be set.
3. When an overrun error occurs, the GiRB register is indeterminate.

**Table 1.22.23. Registers to be Used and Settings**

Register	Bit	Function
GiBCR0	BCK1 to BCK0	Set to "112"
	DIV4 to DIV0	Select a divide ratio of a count source
	IT	Set to "0"
GiBCR1	7 to 0	Set to "0001 00102"
GiPOCR0	7 to 0	Set to "0000 01112"
GiPOCR2	7 to 0	Set to "0000 01102"
GiPOCR3	7 to 0	Set to "0000 00102"
GiTMCR2	7 to 0	Set to "0000 00102"
GiPO0	15 to 0	Set a baud rate $\frac{f_{BTi}}{2 \times (\text{setting value} + 2)} = \text{transfer clock frequency}$
GiPO3	15 to 0	Set a smaller value than the GiPO0 register
GiFS	FSC3 to FSC0	Set to "01002"
GiFE	IFE3 to IFE0	Set to "11012"
GiMR	GMD1 to GMD0	Set to "0016"
	CKDIR	Set to "0"
	STPS	Select a stop bit length
	UFORM	Set to "0"
	IRS	Select how the receive interrupt is generated
GiCR	TI	Transmit buffer empty flag
	TXEPT	Transmit register empty flag
	RI	Receive complete flag
	TE	Set to "1" for transmit and receive enable
	RE	Set to "1" for receive enable
	IPOL	Set to "1"
	OPOL	Set to "1"
GiTB	7 to 0	Write data to be transmitted
GiRB	15 to 0	Received data and error flag are stored

i = 0 to 1

**Table 1.22.24. Pin Settings in UART Mode**

Port name	Function	Bit and setting					Register <sup>1</sup>
		PS1 register	PSL1 register	PSC register	PD7 register	IPS register	
P73	ISTxD1 output	PS1_3 = 1	PSL1_3 = 0	PSC_3 = 1	-	-	G1POCR0
P75	ISRxD1 input	PS1_5 = 0	-	-	PD7_5 = 0	IPS1 = 0	-
P76	ISTxD0 output	PS1_6 = 1	PSL1_6 = 0	PSC_6 = 0	-	-	G0POCR0

Notes :

1. The MOD2 to MOD0 bits in the corresponding register should be set to "1112" (output of the communication function is used).

**Table 1.22.25. Pin Settings (Continued)**

Port name	Function	Bit and setting				Register
		PS2 register	PSL2 register	PD8 register	IPS register	
P80	ISRxD0 input	PS2_0 = 0	-	PD8_0 = 0	IPS0 = 0	-

**Table 1.22.26. Pin Settings (Continued)**

Port name	Function	Bit and setting			Register <sup>1</sup>
		PS5 register	PD11 register	IPS register	
P110	ISTxD1 output	PS5_0 = 1	-	-	G1POCR0
P112	ISRxD1 input	PS5_2 = 0	PD11_2 = 0	IPS1 = 1	-

Notes :

1. The MOD2 to MOD0 bits in the corresponding register should be set to "1112" (output of the communication function is used).

**Table 1.22.27. Pin Settings (Continued)**

Port name	Function	Bit and setting			Register <sup>1</sup>
		PS9 register	PD15 register	IPS register	
P150	ISTxD0 output	PS9_0 = 1	-	-	G0POCR0
P152	ISRxD0 input	-	PD15_2 = 0	IPS0 = 1	-

Notes :

1. The MOD2 to MOD0 bits in the corresponding register should be set to "1112" (output of the communication function is used).

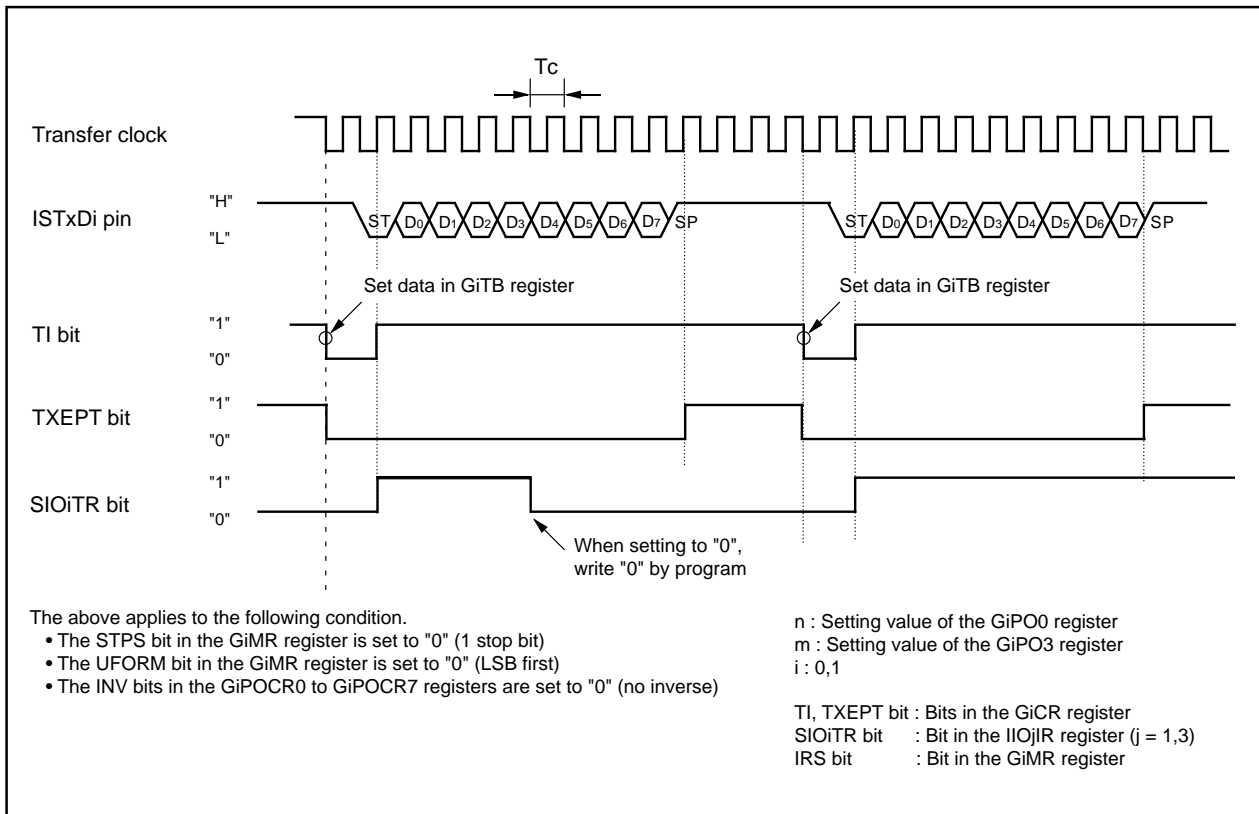


Figure 1.22.40. Transmit Operation

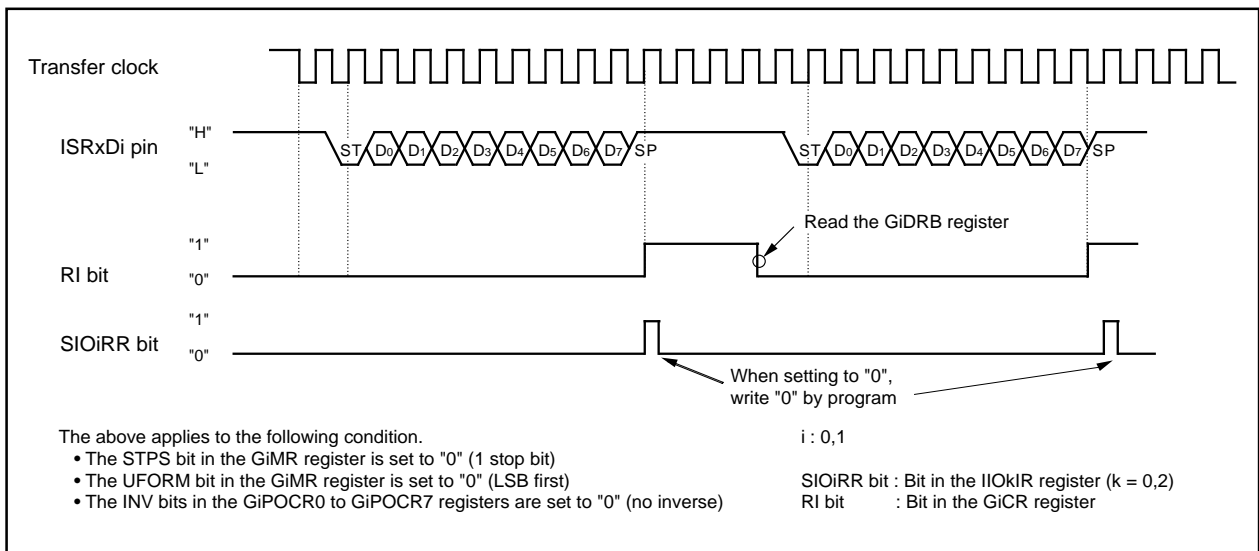


Figure 1.22.41. Receive Operation

### • HDLC Data Processing Mode (Group 0 and 1)

In HDLC data processing mode, bit stuffing, flag detection, abort detection and CRC processing are available to be required for HDLC control. Channel 0 and 1 generate the transfer clock. No pin is used.

Data to be transmitted is written to the GiTB register (i=0,1). After data is converted, data conversion result is restored. If data is restored into the GiTO register after the data conversion, a conversion is terminated. If no data is restored, bit stuffing is executed regardless of no data in the transmit output buffer. CRC value is calculated whenever one bit is converted. If no data is in the GiRI register, received data conversion is terminated.

Table 1.22.28 list specifications of the HDLC data processing mode. Table 1.22.29 lists registers to be used and their settings.

**Table 1.22.28. HDLC Processing Mode Specifications**

Item	Specification
Input data format	8 bits long fixed, a bit alignment as optional
output data format	8 bits long fixed
Transfer clock	<ul style="list-style-type: none"> <li>• When the CKDIR bit in the GiMR register (i=0, 1) is set to "0" (internal clock) : <math>\frac{f_{BTi}}{n+2}</math>  n : setting value of the GiPO0 register 0000<sub>16</sub> to FFFF<sub>16</sub>  - The GiPO0 register determines the baud rate and the transfer clock is generated with the channel 2 waveform generation function and in phase-delayed waveform output mode</li> <li>• When the RSHTTE bit in the GiERC register is set to "1" (reception shifting operation enable), the transfer clock is generated in the receiver</li> </ul>
I/O method	<ul style="list-style-type: none"> <li>• While transmitting  Value set in the GiTB register is converted in HDLC data processing mode to transfer it to the GiTO register</li> <li>• While receiving  Value set in the GiRI register is converted in HDLC data processing mode to transfer it to the GiRB register.  Value set in the GiRI register is also transferred to the GiTB register (received data register).</li> </ul>
Bit stuffing	While transmitting, "0" following five continuous "1" is inserted. While receiving, "0" following five continuous "1" is deleted.
Flag detection	The flag data "7E <sub>16</sub> " should be written in the GiCMPj register for the special communication interrupt (the SRTiR bit in the IIO4IR register)
Abort detection	The masked data "01 <sub>16</sub> " should be written in the GiMSKj register
CRC	The CRC1 to CRC0 bits are set to "112" ( $X^{16}+X^{12}+X^5+1$ ) The CRV bit is set to "1" (set to FFFF <sub>16</sub> ) <ul style="list-style-type: none"> <li>• While transmitting, CRC calculation result is stored into the GiTCRC register  The TCRCE bit in the GiETC register is set to "1" (CRC for data to be transmitted is used).  CRC calculation result is reset when the TCRCE bit is set to "0" (CRC for data to be transmitted is not used).<sup>1</sup></li> <li>• While receiving, CRC calculation result is stored into the GiRCRC register  The RCRCE bit in the GiERC register is set to "1" (CRC for data to be received is used).  CRC calculation result is reset when the result matches a value of the GiCMP3 register by comparing the flag data "7E<sub>16</sub>". The ACRC bit in the GiEMR register is set to "1" (CRC reset)<sup>2</sup></li> </ul>

**Table 1.22.28. HDLC Processing Mode Specifications (Continued)**

Item	Specification
Transmit start condition	The following conditions are required to start transmitting <ul style="list-style-type: none"> <li>• The TE bit in the GiCR register is set to "1" (transmit enable)</li> <li>• Data written to the GiTB register are required</li> </ul>
Receive start condition	The following conditions are required to start transmitting <ul style="list-style-type: none"> <li>• The RE bit in the GiCR register is set to "1" (transmit enable)</li> <li>• Data written to the GiTB register</li> </ul>
Interrupt request <sup>3</sup>	<ul style="list-style-type: none"> <li>• While transmitting, <ol style="list-style-type: none"> <li>(1) The following condition can be selected to set the GiTOR register bit to "1" <ul style="list-style-type: none"> <li>– When the IRS bit in the GiMR register is set to "0" (interrupt with the GiTB register empty) and data is transferred to the transmit register from the GiTB register.</li> <li>– When the IRS bit is set to "1" (interrupt at reception completed) and data transfer from the transmit register is completed.</li> </ul> </li> <li>(2) When data, which is already converted as HDLC data, is transmitted to the GiTO register from the transmit buffer, the GiTOR bit is set to "1".</li> </ol> </li> <li>• While receiving, <ol style="list-style-type: none"> <li>(1) When data is transferred to the GiRB register from the GiRI register (reception completes), the GiRIR bit is set to "1". (See Figure 1.9.14)</li> <li>(2) When received data is transferred to the receive register from the GiRI register, the GiRIR bit is set to "1".</li> <li>(3) When the GiTB register is compared to the GiCMPj register (j=0 to 3), the SRTiR bit is set to "1".</li> </ol> </li> </ul>

## Notes :

1. The CRCV bit and ACRC bit in the GiEMR register should be set to "1".
2. The CRC calculation circuit is reset after the GiRCRC register stores CRC data.
3. See Figure 1.9.14 about the GiTOR bit, GiRIR bit and SRTiR bit.

Table 1.22.29. Registers to be Used and Settings

Register	Bit	Function
GiBCR0	BCK1 to BCK0	Select a count source
	DIV4 to DIV0	Select a divide ratio of a count source
	IT	Select the base timer interrupt
GiBCR1	7 to 0	Set to "0001 0010 <sub>2</sub> "
GiPOCR0	7 to 0	Set to "0000 0000 <sub>2</sub> "
GiPOCR1	7 to 0	Set to "0000 0000 <sub>2</sub> "
GiPO0	15 to 0	Set a baud rate
GiPO1	15 to 0	Set a timing of the rising edge of the transfer clock. Timing of the rising edge ("H" width of the transfer clock) is fixed. Setting value of GiPO1 ≤ setting value of GiPO0 .
GiFS	FSC0 to FSC1	Set to "00 <sub>2</sub> "
GiFE	IFE0 to IFE1	Set to "11 <sub>2</sub> "
GiMR	GMD1 to GMD0	Set to "11 <sub>2</sub> "
	CKDIR	Set to "0"
	UFORM	Set to "0"
	IRS	Select how the transmit interrupt is generated
GiEMR	7 to 0	Set to "1111 0110 <sub>2</sub> "
GiCR	TI	Transmit buffer empty flag
	TXEPT	Transmit register empty flag
	RI	Receive complete flag
	TE	Transmit enable bit
	RE	Receive enable bit
GiETC	SOF	Set to "0"
	TCRCE	Select whether a transmit CRC is used or not
	ABTE	Set to "0"
	TBSF0, TBSF1	Transmit bit stuffing
GiERC	CMP0E to CMP2E	Select whether a received data is compared or not
	CMP3E	Set to "1"
	RCRCE	Select receive CRC used or not
	RSHTE	When using for reception, set to "1"
	RBSF0, RBSF1	Receive bit stuffing
GiIRF	BSERR, ABT	Set to "0"
	IRF0 to IRF3	Select how an interrupt is generated
GiCMP0, GiCMP1	7 to 0	Write "FE <sub>16</sub> " to abort processing
GiCMP2	7 to 0	Data to be compared
GiCMP3	7 to 0	Write "7E <sub>16</sub> "
GiMSK0, GiMSK1	7 to 0	Write "01 <sub>16</sub> " to abort processing
GiTCRC	15 to 0	Transmit CRC calculation results can be read
GiRCRC	15 to 0	Receive CRC calculation results can be read
GiTO	7 to 0	Data, which is output from a transmit data generation circuit, can be read
GiRI	7 to 0	Set data input to a receive data generation circuit
GiRB	7 to 0	Received data is stored
GiTB	7 to 0	For transmitting : write data to be transmitted For receiving : Received data to compare is stored

i = 0,1

## (2) Group 2 Communication Function

In the intelligent I/O group 2, variable clock synchronous serial I/O or IE Bus<sup>1</sup> communication function is available. Figures 1.22.42 to 1.22.44 show registers associated with the communication function.

Noes :

1. IEBus is a trademark of NEC Corporation.

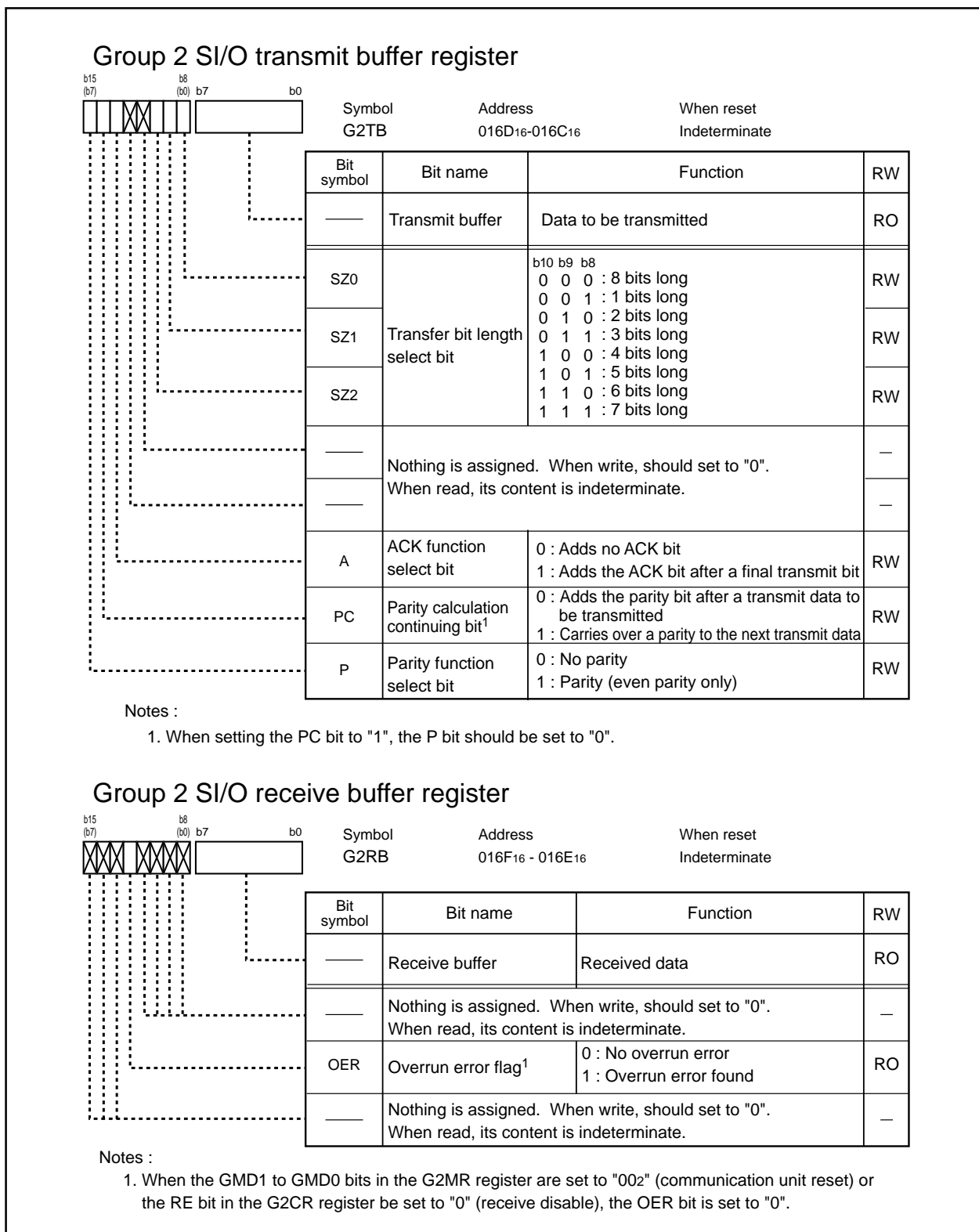


Figure 1.22.42. G2TB and G2RB Register

### Group 2 SI/O communication mode register

		Symbol G2MR	Address 016A <sub>16</sub>	When reset 00XX X000 <sub>2</sub>
Bit symbol	Bit name	Function	RW	
GMD0	Communication mode select bit	b <sub>1</sub> b <sub>0</sub> 0 0 : Communication unit is reset (The OER bit is set to "0") <sup>1</sup> 0 1 : Clock synchronous serial I/O mode <sup>2</sup>	RW	
GMD1		1 0 : IE mode <sup>2</sup> 1 1 : Avoid this setting	RW	
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock	RW	
—	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—	
—			—	
—			—	
UFORM	Transfer direction select bit	0 : LSB first 1 : MSB first	RW	
IRS	Transmit interrupt cause select bit	0 : Transmit buffer is empty 1 : Transmission is completed	RW	

Notes :

1. After setting the GMD1 to GMD0 bits to "00<sub>2</sub>" (communication part reset), the base timer clock should be run in one cycle or more.
2. The GMD1 to GMD0 bits should be rewritten to "01<sub>2</sub>" (clock synchronous serial I/O mode) or "10<sub>2</sub>" (IE mode), while the base timer clock stops.

### Group 2 SI/O communication control register

		Symbol G2CR	Address 016B <sub>16</sub>	When reset 0000 X000 <sub>2</sub>
Bit symbol	Bit name	Function	RW	
TE	Transmit enable bit	0 : Transmit disable 1 : Transmit enable	RW	
TXEPT	Transmit register empty flag	0 : Data available in transmit register (during transmission) 1 : No data available in transmit register (transmission is completed)	RO	
TI	Transmit buffer empty flag	0 : Data available in the G2TB register 1 : No data available in the G2TB register	RO	
—	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—	
RE	Receive enable bit <sup>1</sup>	0 : Receive disable 1 : Receive enable	RW	
RI	Receive complete flag	0 : No data available in the G2RB register 1 : Data available in the G2RB register	RO	
OPOL	TxD output polarity switch bit	0 : No inverse 1 : Inverse	RW	
IPOL	RxD input polarity switch bit <sup>1</sup>	0 : No inverse 1 : Inverse	RW	

Notes :

1. When this bit is rewritten, the group2 base timer may be reset. To avoid resetting, the RST2 bit in the G2BCR1 register should be set to "0" (no base timer reset by a reset request with the communication function).

Figure 1.22.43. G2MR and G2CR Register

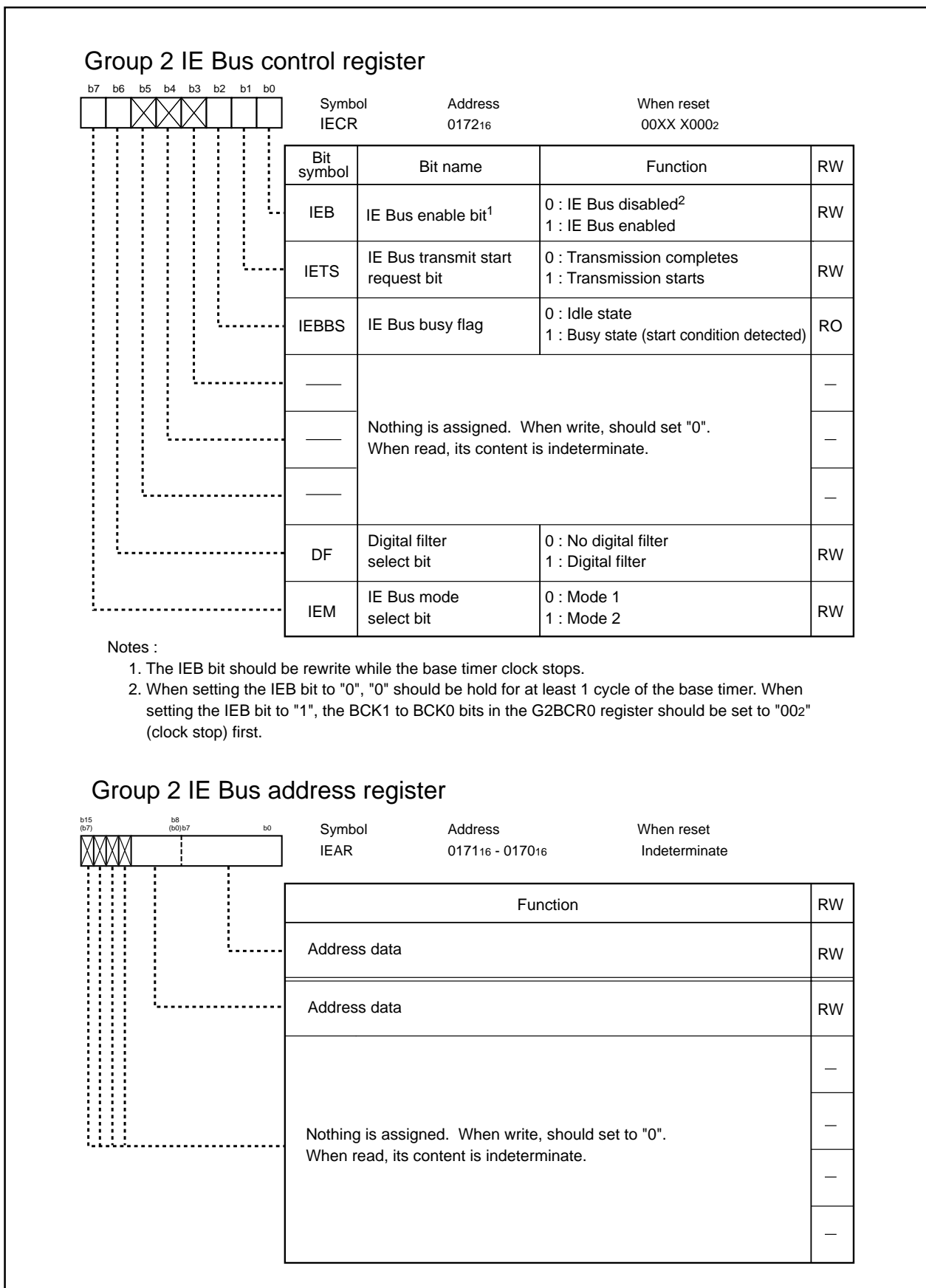


Figure 1.22.44. IECR and IEAR Registers

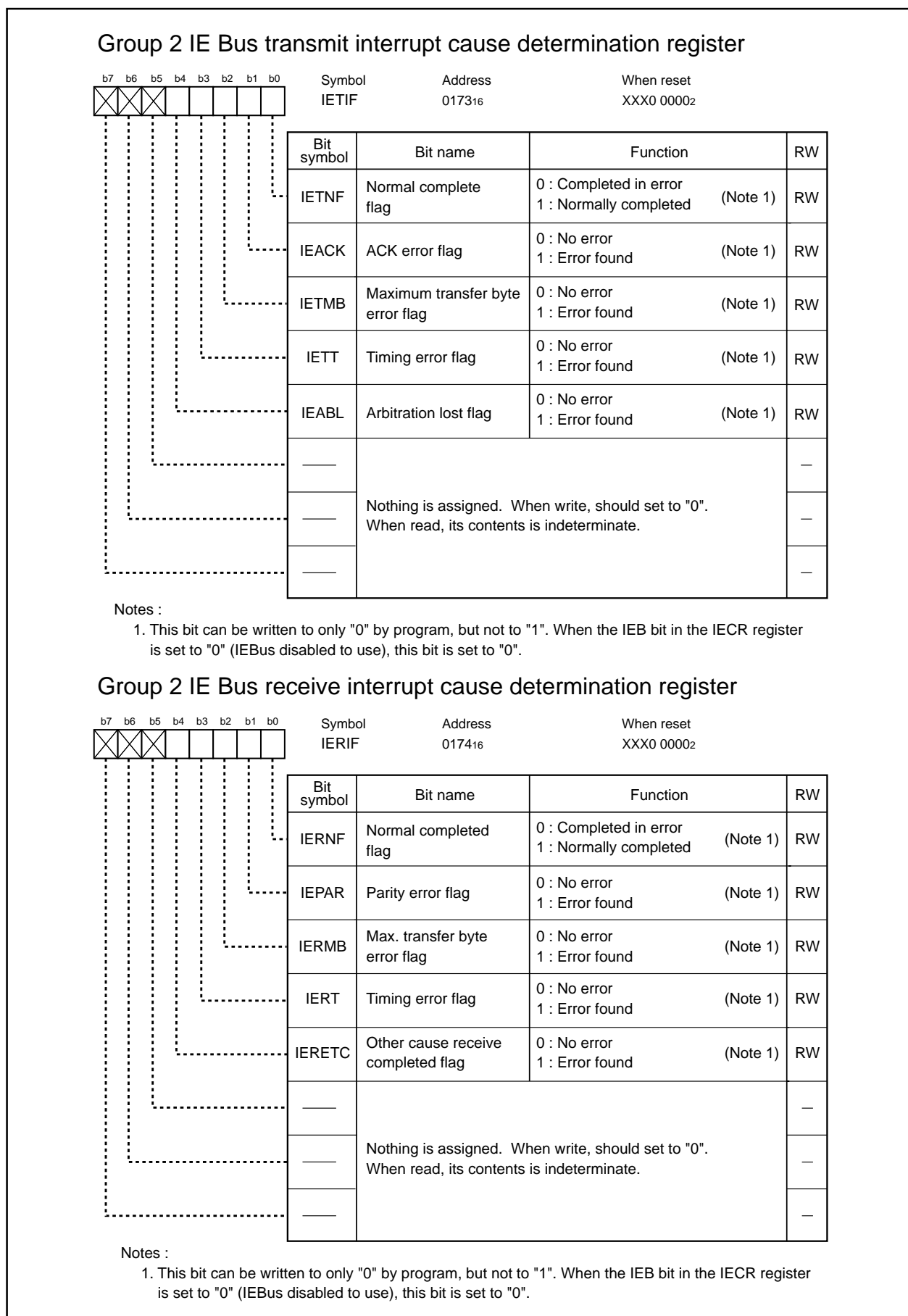


Figure 1.22.45. IETIF and IERIF Registers

### • Clock Synchronous Serial I/O Mode (Group 2)

Table 1.22.30 lists specifications of clock synchronous serial I/O mode group 2. Table 1.22.31 lists register to be used and settings. Tables 1.22.32 to 1.22.34 list pin settings. Figure 1.22.46 shows an example of transmit and receive operation.

**Table 1.22.30. Clock Synchronous Serial I/O Mode Specifications (Group 2)**

Item	Specification
Transfer data format	• Transfer data : Variable
Transfer clock <sup>1</sup>	<ul style="list-style-type: none"> <li>• When the CKDIR bit in the G2MR register is set to "0" (internal clock) : <math>\frac{f_{BT2}}{2(n+2)}</math>  n : setting value of the G2PO0 register 0000<sub>16</sub> to FFFF<sub>16</sub></li> <li>- The G2PO0 register determines the baud rate and the transfer clock is generated with the channel 2 waveform generation function and in phase-delayed waveform output mode</li> <li>• When the CKDIR bit is set to "1" (external clock) : input from the ISCLK2 pin<sup>2</sup></li> </ul>
Transmit start condition	<ul style="list-style-type: none"> <li>• To start transmitting, the following conditions are required : <ul style="list-style-type: none"> <li>- The TE bit in the G2CR register is set to "1" (transmit enable)</li> <li>- Data written to the G2TB register</li> </ul> </li> </ul>
Receive start condition	<ul style="list-style-type: none"> <li>• To start receiving, the following conditions are required : <ul style="list-style-type: none"> <li>- The RE bit in the G2CR register should be set to "1" (receive enable)</li> <li>- The TE bit in the G2CR register should be set to "1" (transmit enable)</li> <li>- Data written to the G2TB register</li> </ul> </li> </ul>
Interrupt request	<ul style="list-style-type: none"> <li>• While transmitting, the following condition can be selected to set the SIO2TR bit in the IIO6IR register to "1" (see Figure 1.9.14) : <ul style="list-style-type: none"> <li>- When the IRS bit in the G2MR register is set to "0" (interrupt with the G2TB register empty) and data is transferred to the transmit register from the G2TB register</li> <li>- When setting the IRS bit to "1" (interrupt at reception completed) and data transfer from the transmit register is completed</li> </ul> </li> <li>• While receiving  When data is transferred to the G2RB register from the receive register (reception completed), the SIO2RR bit in the IIO5IR register is set to "1" (see Figure 1.9.14)</li> </ul>
Error detection	<p>Overrun error<sup>3</sup></p> <p>This error occurs when receiving the 8th bit of the next data before reading the G2RB register</p>
Selectable function	<ul style="list-style-type: none"> <li>• LSB first/MSB first  Whether data is transmitted/received in bit 0 or in bit 7 can be selected</li> <li>• ISTxD2 and ISRxD2 I/O polarity inverse  ISTxD2 pin output and ISRxD2 pin input levels are inverted</li> <li>• Data transfer bit length  Transfer bit is selectable from 1 bits to 8 bits</li> </ul>

Notes :

1. The transfer clock should be  $f_{BT2}$  divided by six or more when both transfer clock and transfer data are transmitted. Other than the above, the transfer clock should be  $f_{BT2}$  divided by 20 or more.
2. Additional transfer clock should be input  $f_{BT2}$  divided by 20 or more.
3. When an overrun error occurs, the G2RB register is indeterminate.

**Table 1.22.31. Register to be Used and Settings**

Register	Bit	Function
G2BCR0	BCK1 to BCK0	Set to "112"
	DIV4 to DIV0	Select a divide ratio of a count source
	IT	Set to "0"
G2BCR1	7 to 0	Set to "0001 00102"
G2POCR0	7 to 0	Set to "0000 01112"
G2POCR1	7 to 0	Set to "0000 01112"
G2BCR2	7 to 0	Set to "0000 00102"
G2PO0	15 to 0	Set a baud rate $\frac{f_{BT2}}{2 \times (\text{setting value} + 2)} = \text{transfer clock frequency}$
G2PO2	15 to 0	Set a smaller value than G2PO0 register
G2FE	IFE2 to IFE0	Set to "1112"
G2MR	GMD1 to GMD0	Set to "012"
	CKDIR	Select an internal or external clock
	UFORM	Select either LSB first or MSB first
	IRS	Select how the transmit interrupt is generated
G2CR	TE	When transmission is enabled, set to "1"
	TXEPT	Transmit register empty flag
	TI	Transmit buffer empty flag
	RE	When reception is enabled, set to "1"
	RI	Receive complete flag
	OPOL	TxD output polarity inverse (usually set to "0")
	IPOL	RxD input polarity inverse (usually set to "0")
G2TB	15 to 0	Write a transfer bit length and transmit data
G2RB	15 to 0	Received data and error flag are stored

**Table 1.22.32. Pin Settings**

Port name	Function	Bit and setting					Register <sup>2</sup>
		PS1 register	PSL1 register	PSC register	PD7 register	IPS register	
P70 <sup>1</sup>	ISTxD2 output	PS1_0 = 1	PSL1_0 = 0	PSC_0 = 1	-	-	G2POCR0
P71	ISRxD2 input	PS1_1 = 0	-	-	PD7_1 = 0	IPS5 to 4 = 002	-

Notes :

1. Output is N-channel open drain output.
2. The MOD2 to MOD0 bits in the corresponding register should be set to "1112" (output of the communication function is used).

**Table 1.22.33. Pin Settings (Continued)**

Port name	Function	Bit and setting				Register <sup>2</sup>
		PS3 register	PSL3 register	PD9 register <sup>1</sup>	IPS register	
P91	ISRxD2 input	PS3_1=0	-	PD9_1=0	IPS5 to 4=012	-
P92	ISTxD2 output	PS3_2=1	PSL3_2=1	-	-	G2POCR0

Notes :

1. The PD9 and PS3 registers should be set with subsequent to the instruction that the PRC2 bit in the PRCR register is set to "1" (write enable). Avoid an interrupt and DMA transfer between the instruction to set the PRC2 bit to "1" and the instruction to write to the PD9 and PS3 registers.
2. The MOD2 to MOD0 bits in the corresponding register should be set to "1112" (output of the communication function is used).

**Table 1.22.34. Pin Settings (Continued)**

Port name	Function	Bit and setting				Register <sup>1</sup>
		PS0 register	PSL0 Register	PD6 register	IPS register	
P64	ISCLK2 input	PS0_4 = 0	-	PD6_4 = 0	IPS6 = 0	-
	ISCLK2 output	PS0_4 = 1	PSL0_4 = 1	-	-	G2POCR1

Notes :

1. The MOD2 to MOD0 bits in the corresponding register should be set to "1112" (output of the communication function is used).

**Table 1.22.35. Pin Settings (Continued)**

Port name	Function	Bit and setting			Register <sup>1</sup>
		PS7 register	PD13 register	IPS register	
P134	ISTxD2 output	PS7_4 = 1	-	-	G2POCR0
P135	ISRxD2 input	PS7_5 = 0	PD13_5 = 0	IPS5 to 4 = 102	-
P136	ISCLK2 input	PS7_6 = 0	PD13_6 = 0	IPS6_1 = 1	-
	ISCLK2 output	PS7_6 = 1	-	-	G2POCR1

Notes :

1. The MOD2 to MOD0 bits in the corresponding register should be set to "1112" (output of the communication function is used).

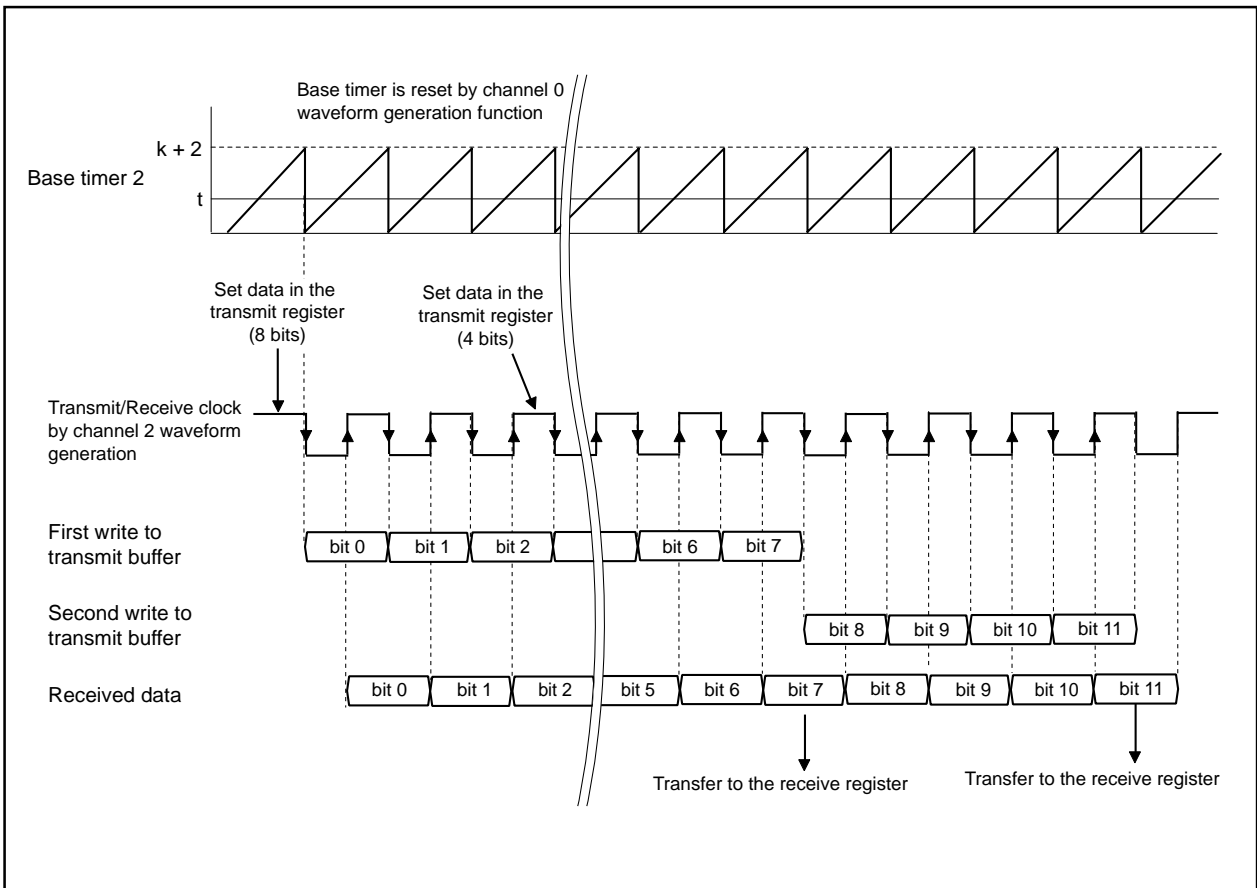


Figure 1.22.46. Transmit and Receive Operation

• **IE Bus Mode (Group 2)**

Table 1.22.36 lists specifications of IE Bus mode. Table 1.22.37 lists registers to be used and settings.  
Table 1.22.38 lists pin settings.

**Table 1.22.36. IE Bus Mode Specification**

Item	Specification
Transfer data format	<ul style="list-style-type: none"> <li>• Transfer data : Variable</li> </ul>
Transfer clock	<ul style="list-style-type: none"> <li>• When the CKDIR bit in the G2MR register is set to "0" (internal clock) :               <ul style="list-style-type: none"> <li>n : setting value of the G2PO0 register, 0000<sub>16</sub> to FFFF<sub>16</sub>.</li> <li>- The G2PO0 register determines the baud rate and the transfer clock is generated with the channel 2 waveform generation function and in phase-delayed waveform output mode.</li> <li>- The G2PO2 register = (n+2)/2</li> </ul> </li> <li>• When the CKDIR bit is set to "1" (external clock) : input from the ISCLK2 pin<sup>2</sup></li> </ul>
Transmit start condition	To start transmitting, the following conditions are required : <ul style="list-style-type: none"> <li>• The TE bit in the G2CR register is set to "1" (transmit enable)</li> <li>• Write data to G2TB register</li> </ul>
Receive start condition	To start receiving, the following requirements must be met: <ul style="list-style-type: none"> <li>• The RE bit in the G2CR register should be set to "1" (receive enable)</li> <li>• The TE bit in the G2CR register should be set to "1" (transmit enable)</li> <li>• Data written to the G2TB register</li> </ul>
Interrupt request	<ul style="list-style-type: none"> <li>• While transmitting, the following condition can be selected to set the SIO2TR bit in the IIO6R register to "1" (see Figure 1.9.14) :               <ul style="list-style-type: none"> <li>- When the IRS bit in the G2MR register is set to "0" (interrupt with the G2TB register empty) and data is transferred to the transmit register from the G2TB register (transmission starts)</li> <li>- When the IRS bit is set to "1" (interrupt at reception completed) and data transfer from the transmit register is completed</li> </ul> </li> <li>• While receiving               <ul style="list-style-type: none"> <li>When data is transferred to G2RB register from receive register (reception completes), the SIO2RR bit in the IIO5R register is set to "1" (see Figure 1.9.14).</li> </ul> </li> </ul>
Error detection	Overrun error <sup>3</sup> This error occurs when receiving the 8th bit of the next data before reading the G2RB register
Selectable function	<ul style="list-style-type: none"> <li>• LSB first/MSB first select               <ul style="list-style-type: none"> <li>Whether data is transmitted/received in bit 0 or in bit 7 can be selected</li> </ul> </li> <li>• ISTxD2 and ISRxD2 I/O polarity inverse               <ul style="list-style-type: none"> <li>ISTxD2 pin output and ISRxD2 pin input levels are inverted</li> </ul> </li> <li>• Data transfer bit length               <ul style="list-style-type: none"> <li>Transfer bit is selectable from 1 bit to 8 bits</li> </ul> </li> </ul>

Notes :

1. The transfer clock should be  $f_{BT2}$  divided by six or more when both transfer clock and transfer data are transmitted. Other than the above, the transfer clock should be  $f_{BT2}$  divided by 20 or more.
2. Additional transfer clock should be input  $f_{BT2}$  divided by 20 or more.
3. When an overrun error occurs, the G2RB register is indeterminate.

**Table 1.22.37. Registers to be Used and Settings**

Register	Bit Function	
G2BCR0	BCK1 to BCK0	Set to "112"
	DIV4 to DIV0	Select a divide ratio of a count source
	IT	Set to "0"
G2BCR1	7 to 0	Set to "000100102"
G2POCR0 to G2POCR7	MOD2 to MOD0	Set to "1112"
	PRT	Set to "0"
	IVL	Set to "0"
	RLD	Set to "0"
	RTP	Set to "0"
G2PO0 to G2PO7	15 to 0	Set compared data for waveform generation
	G2FE	7 to 0
G2MR	GMD1 to GMD0	Select serial I/O mode
	CKDIR	Select the internal clock or external clock
	UFORM	Select either LSB first or MSB first
	IRS	Select how the transmit interrupt is generated
G2CR	TI	Transmit buffer empty flag
	TXEPT	Transmit register empty flag
	RI	Receive complete flag
	TE	When transmission is enabled, set to "1"
	RE	When reception is enabled, set to "1"
	IPOL	Set to "0"
	OPOL	Set to "0"
IECR	IEB	Set to "1"
	IETS	When transmission starts, set to "1"
	IEBBS	Select IE Bus busy flag
	DF	Select whether the digital filter is available or not
	IEM	Select mode
IEAR	11 to 0	Set address data
IETIF	IETNF	Normal complete flag when transmitting
	IEACK	ACK error flag when transmitting
	IETMB	Maximum transfer byte error flag when transmitting
	IETT	Timing error flag when transmitting
	IEABL	Arbitration lost flag when transmitting
IERIF	IERNF	Normal complete flag when receiving
	IEPAR	Parity error flag when receiving
	IERMB	Maximum transfer byte error flag when receiving
	IERT	Timing error flag when receiving
	IERETC	Other cause receive completed flag when receiving
G2RB	7 to 0	Received data and error flag are stored
	OER	Overrun error flag
G2TB	7 to 0	Write transfer bit length and data to be transmitted

**Table 1.22.38. Pin Settings**

Port name	Function	Bit and setting					Register <sup>1</sup>
		PS1 register	PSL1 register	PSC register	PD7 register	IPS register	
P70 <sup>2</sup>	IEOUT output	PS1_0 = 1	PSL1_0 = 0	PSC_0 = 1	-	-	G2POCR0
P71 <sup>2</sup>	IEIN input	PS1_1 = 0	-	-	PD7_1 = 0	IPS5 to 4 = 002	-

Notes :

1. The MOD2 to MOD0 bits in the G2POCR0 register should be set to "1112".
2. Output is N-channel open drain.

**Table 1.22.39. Pin Settings (Continued)**

Port name	Function	Bit and setting				Register <sup>1</sup>
		PS3 register <sup>2</sup>	PSL3 register	PD9 register <sup>2</sup>	IPS register	
P91	IEIN input	PS3_1 = 0	-	-	IPS5 to 4 = 012	-
P92	IEOUT output	PS3_2 = 1	PSL3_2 = 1	PD9_2 = 0	-	G2POCR0

Notes :

1. The MOD2 to MOD0 bits in the G2POCR0 register should be set to "1112".
2. The PD9 and PS3 registers should be set with subsequent to the instruction that the PRC2 bit in the PRCR register is set to "1" (write enable). Avoid an interrupt and DMA transfer between the instruction to set the PRC2 bit to "1" and the instruction to write to the PD9 and PS3 registers.

**Table 1.22.40. Pin Settings (Continued)**

Port name	Function	Bit and setting			Register <sup>1</sup>
		PS7 register	PSL7 register	IPS register	
P134	IEOUT output	PS7_4 = 1	-	-	G2POCR0
P135	IEIN input	PS7_5 = 0	PD13_5 = 0	IPS5 to 4 = 102	-

Notes :

1. The MOD2 to MOD0 bits in the G2POCR0 register should be set to "1112".

### (3) Group 3 Communication Function

In the intelligent I/O group 3, 8-bit or 16-bit synchronous communication function is available. Figures 1.22.47 to 1.22.49 show registers associated with the communication function.

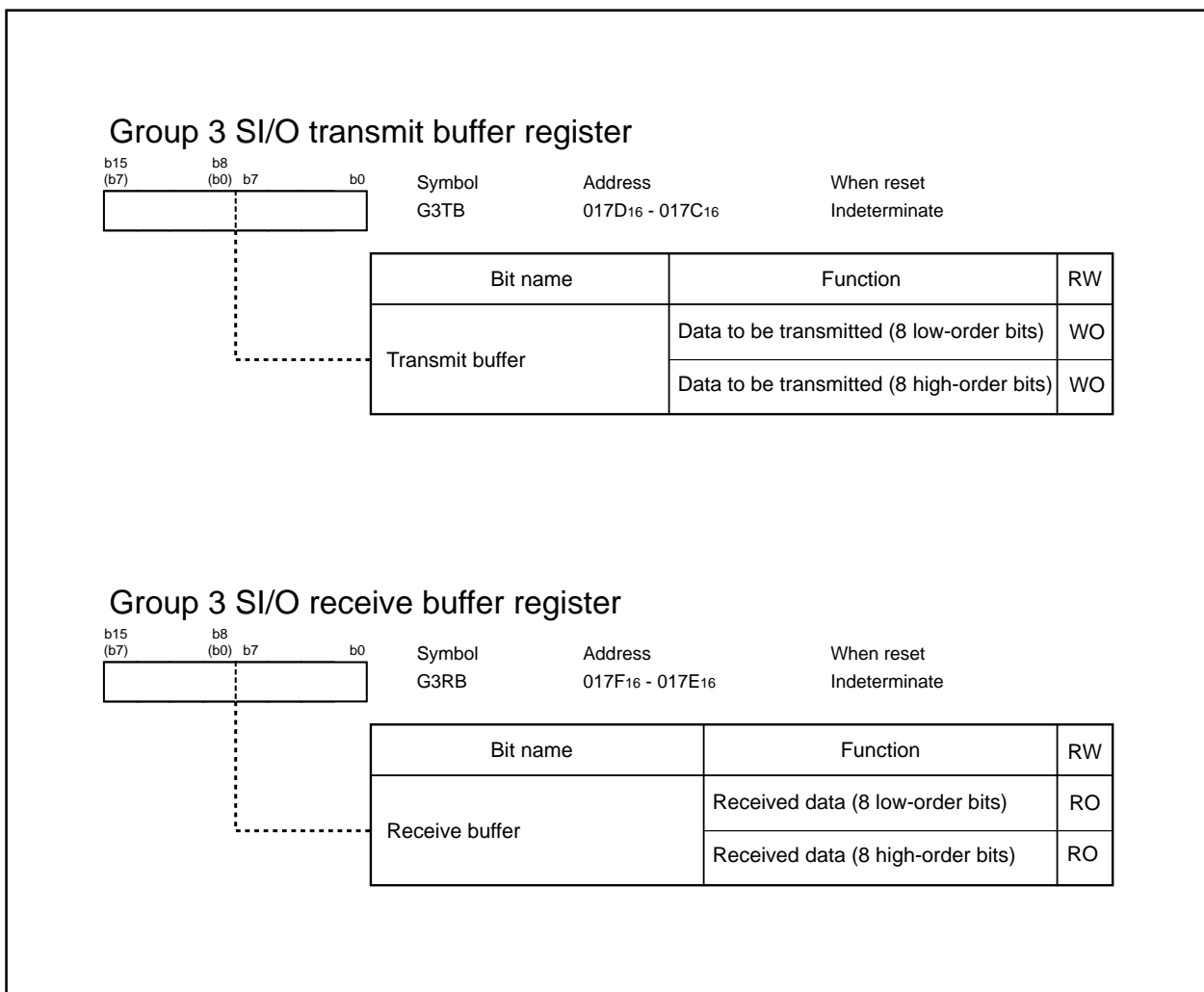


Figure 1.22.47. G3TB Register and G3RB Register

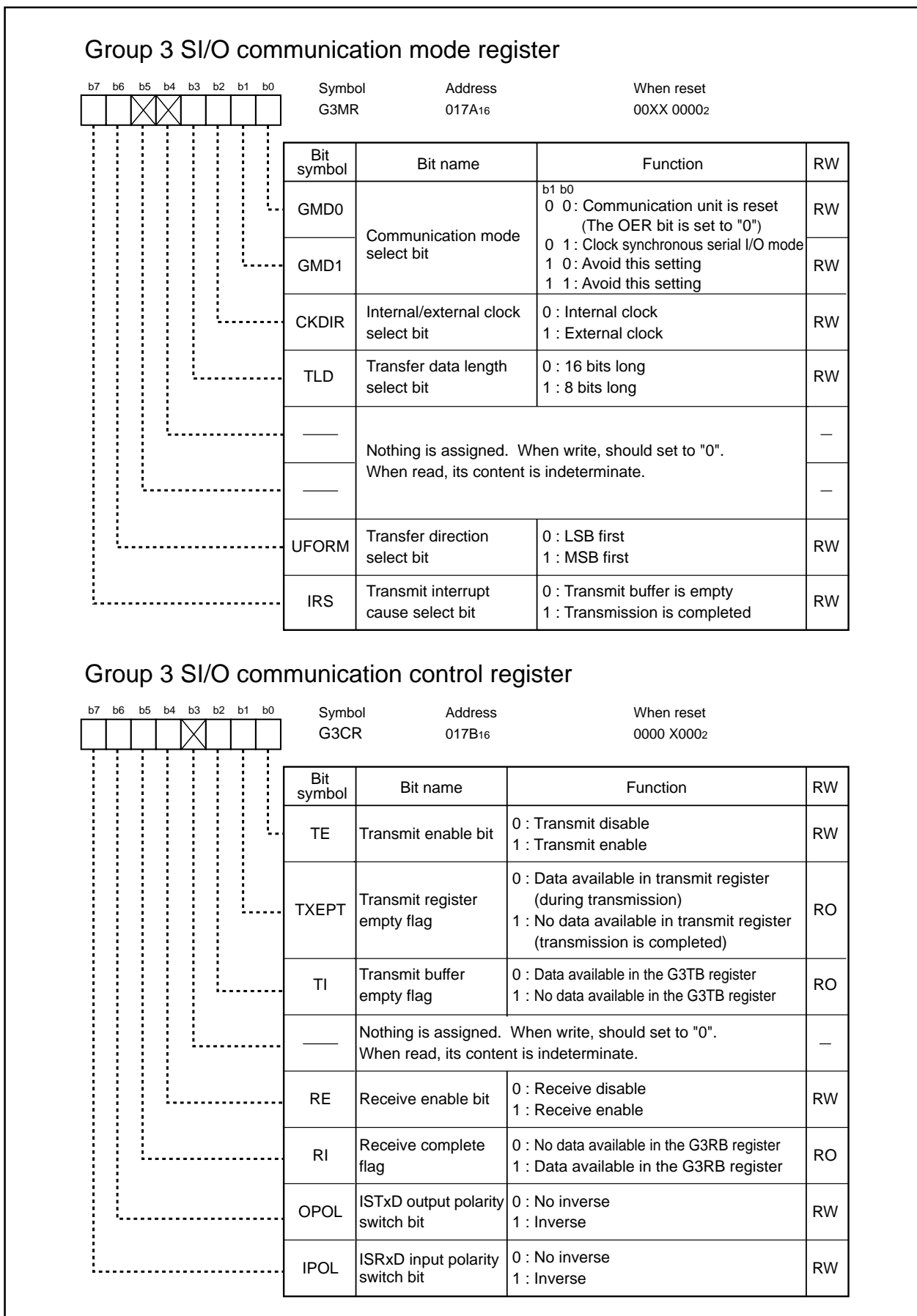


Figure 1.22.48. G3MR Register and G3CR Register

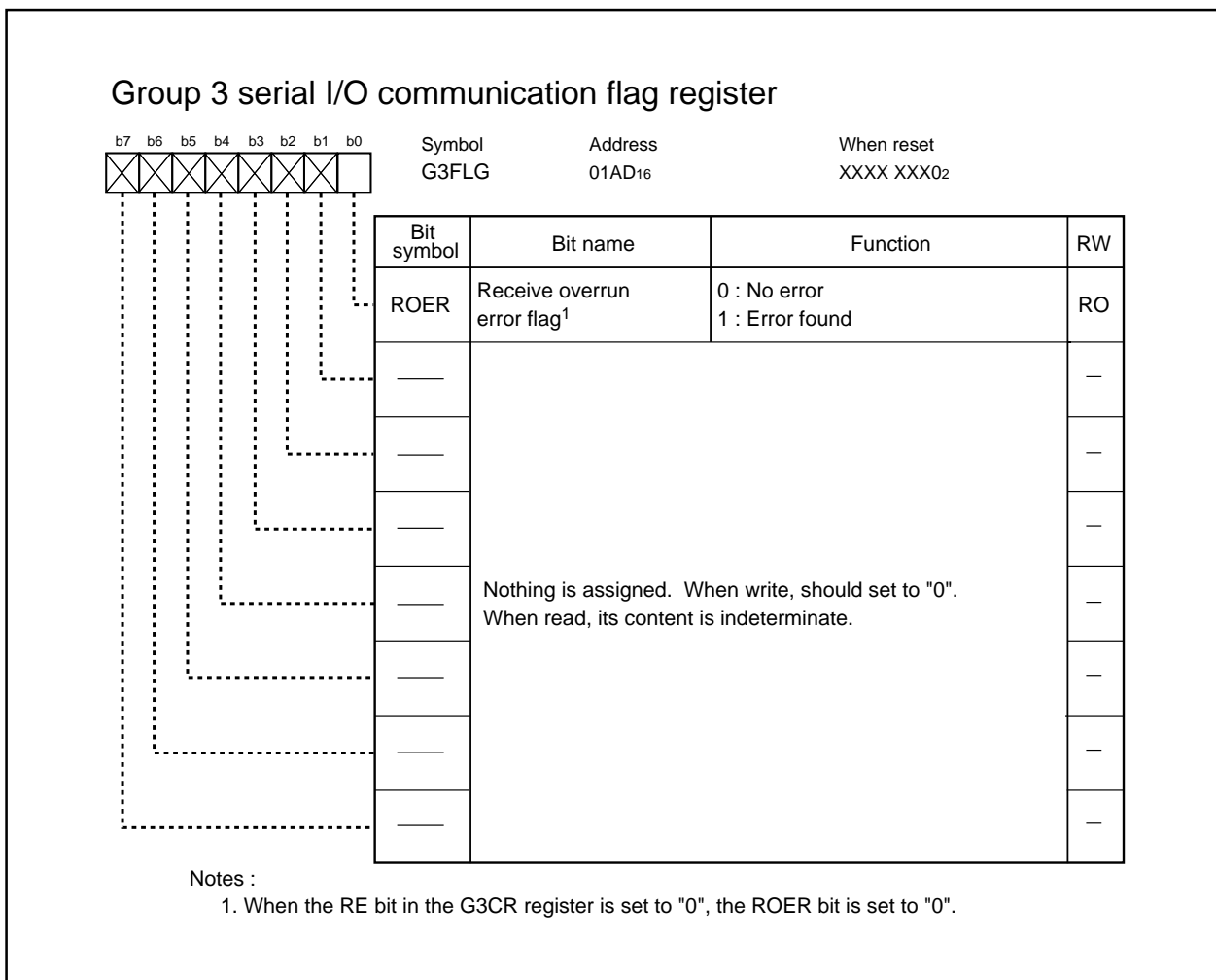


Figure 1.22.49. G3FLG Register

### • 8-bit or 16-bit Clock Synchronous Serial I/O Mode (Group 3)

In 8-bit or 16-bit clock synchronous serial I/O mode, data is transmitted and received with a synchronous clock. When the internal clock is selected as a synchronous clock, channel 0 and 2 generate the transfer clock. Each ISTxD3, ISCLK3 and ISRxD3 pin shares pins with OUTC30 to OUTC32 and can be used in the 144-pin package.

Table 1.22.41 lists specifications of synchronous serial I/O mode. Table 1.22.42 lists registers to be used and settings. Tables 1.22.43 to 1.22.44 list pin settings. Figures 1.22.50 to 1.22.51 show examples of transmit and receive operation.

**Table 1.22.41. Clock Synchronous Serial I/O Mode (Group 3)**

Item	Specification
Transfer data format	• Transfer data : 8 bits or 16 bits long
Transfer clock <sup>1</sup>	<ul style="list-style-type: none"> <li>• When the CKDIR bit in the G3MR register is set to "0" (internal clock) : <math>\frac{f_{BT3}}{2(n+2)}</math>  n : setting value of the G3PO0 register 0000<sub>16</sub> to FFFF<sub>16</sub>  Baud rate is determined by the G3PO0 register and generated with the channel 2 waveform generation function and in phase-delayed waveform output mode</li> <li>• When setting the CKDIR bit to "1" (external clock) : input from the ISCLK3 pin</li> </ul>
Transmit start condition	Registers associated with the waveform generation function and the G3MR register should be set to set the following values after one cycle. <ul style="list-style-type: none"> <li>• The TE bit in the G3CR register is set to "1" (transmit enable)</li> <li>• The TI bit in the G3CR register is set to "0" (data written to the G3TB register)</li> </ul>
Receive start condition	Registers associated with the waveform generation function and the G3MR register should be set to set the following values after one cycle. <ul style="list-style-type: none"> <li>• The TE bit should be set to "1" (transmit enable)</li> <li>• The RE bit in the G3CR register is set to "1" (receive enable)</li> <li>• The TI bit is set to "0" (data written to the G3TB register)</li> </ul>
Interrupt request	<ul style="list-style-type: none"> <li>• While transmitting, the following condition can be selected to set the SIO3TR bit in the IIO10R register to "1". (See Figure 1.9.14.) <ul style="list-style-type: none"> <li>– In one transfer clock cycle after data transmission starts, when the IRS bit in the G3MR register is set to "0" (transmit buffer is empty).</li> <li>– In 15 transfer clock cycles after data transmission starts, when the IRS bit is set to "1" (reception completes), in 16-bit clock synchronous serial I/O mode (setting the DLS bit in the G3MR register to "0")  In 7 transfer clock cycles after data transmission starts in 8-bit clock synchronous serial I/O mode (setting the DLS bit to "1").</li> </ul> </li> <li>• While receiving <ul style="list-style-type: none"> <li>– In 15.5 transfer clock cycles after data transmission starts, when the SIO3RR bit in the IIO9IR register is set to "1", in 16-bit clock synchronous serial I/O mode.</li> <li>– In 7.5 transfer clock cycles, after data transmission starts, when the SIO3RR bit in the IIO9IR register is set to "1", in 8-bit clock synchronous serial I/O mode. (See Figure 1.9.14.)</li> </ul> </li> </ul>
Error detection	<ul style="list-style-type: none"> <li>• Overrun error<sup>3</sup>  This error occurs in 16-bit clock synchronous serial I/O mode when receiving the 15th bit of the next data before reading the G3RB register. This error occurs in 8-bit clock synchronous serial I/O mode when receiving the 8th bit of the next data before reading the G3RB register.</li> </ul>
Selectable function	<ul style="list-style-type: none"> <li>• LSB first/MSB first :  Whether data is transmitted/received in bit 0 or in bit 7 can be selected</li> <li>• ISTxD3 and ISRxD3 I/O polarity inverse :  ISTxD3 pin output and ISRxD3 pin input levels are inverted.</li> </ul>

Notes :

1. The transfer clock should be  $f_{BT3}$  divided by six or more.
2. Transmit interrupt request is generated as soon as the TE bit is set to "1". The interrupt-associated registers should be set after setting the TE bit.
3. When an overrun error occurs, the G3RB register is indeterminate.

**Table 1.22.42. Registers to be Used and Settings**

Register	Bit	Function
G3BCR0	BCK1 to BCK0	Set to "112"
	DIV4 to DIV0	Select a divide ratio of a count source
	IT	Set to "0"
G3BCR1	7 to 0	Set to "0001 00102"
G3POCR0	7 to 0	Set to "0000 01112"
G3POCR1	7 to 0	Set to "0000 01112"
G3POCR2	7 to 0	Set to "0000 00102"
G3PO0	15 to 0	Set a baud rate $\frac{f_{BT3}}{2 \times (\text{setting value} + 2)} = \text{transfer clock frequency}$
G3PO2	15 to 0	Set a smaller value than the G3PO0 register
G3FE	7 to 0	Set to "0000 01112"
G3MR	GMD1 to GMD0	Set to "012"
	CKDIR	Select the internal clock or external clock
	TLD	Select a transfer data length
	UFORM	Select either LSB first or MSB first
	IRS	Select how the transmit interrupt is generated
G3CR	TE	When transmission is enabled, set to "1"
	TXEPT	Transmit register empty flag
	TI	Transmit buffer empty flag
	RE	When reception is enabled, set to "1"
	RI	Receive complete flag
	OPOL	TxD output polarity inverse (usually set to "0")
	IPOL	RxD input polarity inverse (usually set to "0")
G3TB	15 to 0	Write data to be transmitted
G3RB	15 to 0	Receive a data is stored

**Table 1.22.43. Pin Setting in Clock Synchronous Serial I/O Mode (Group 3)**

Port name	Function	Bit and setting				Register <sup>1</sup>
		PS2 register	PSL2 register	PD8 register	IPS register	
P81	ISTxD3 output	PS2_1 = 1	PSL2_1 = 1	-	-	G3POCR0
P82	ISRxD3 input	PS2_2 = 0	-	PD8_2 = 0	IPS7 = 0	-

Notes :

1. The MOD2 to MOD0 bits in the corresponding register should be set to "1112" (output of the communication function is used).

**Table 1.22.44. Pin Setting (Continued)**

Port name	Function	Bit and setting			Register <sup>1</sup>
		PS6 register	PD12 register	IPS register	
P120	ISTxD3 output	PS6_0 = 1	-	-	G3POCR0
P121	ISCLK3 input	PS6_1 = 0	PD12_1 = 0	-	-
	ISCLK3 output	PS6_1 = 1	-	-	G3POCR1
P122	ISRxD3 input	PS6_2 = 0	PD12_2 = 0	IPS7 = 1	-

Notes :

1. The MOD2 to MOD0 bits in the corresponding register should be set to "1112" (output of the communication function is used).

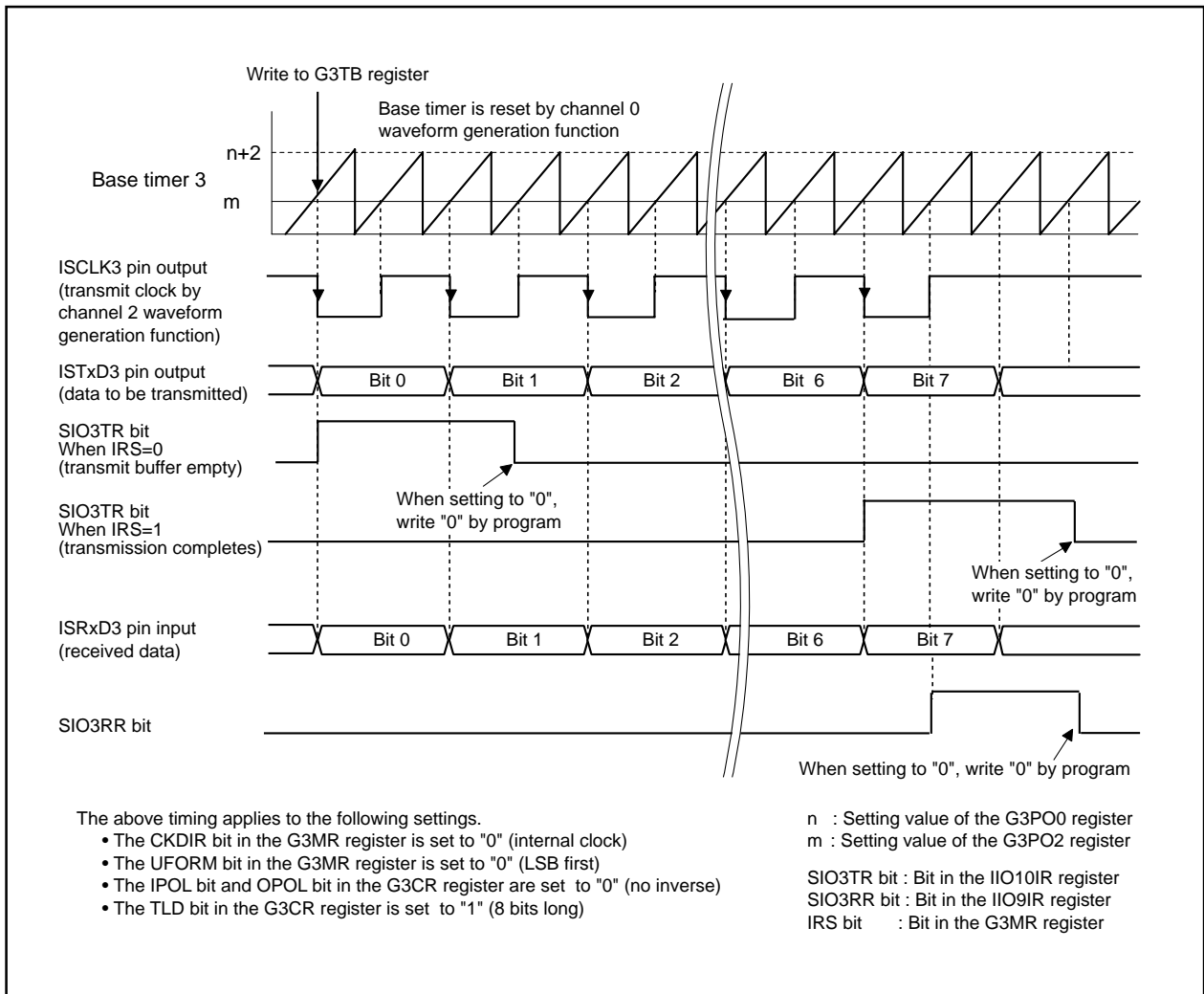


Figure 1. 22. 50. Transmit and Receive Operation (8-bit Length)

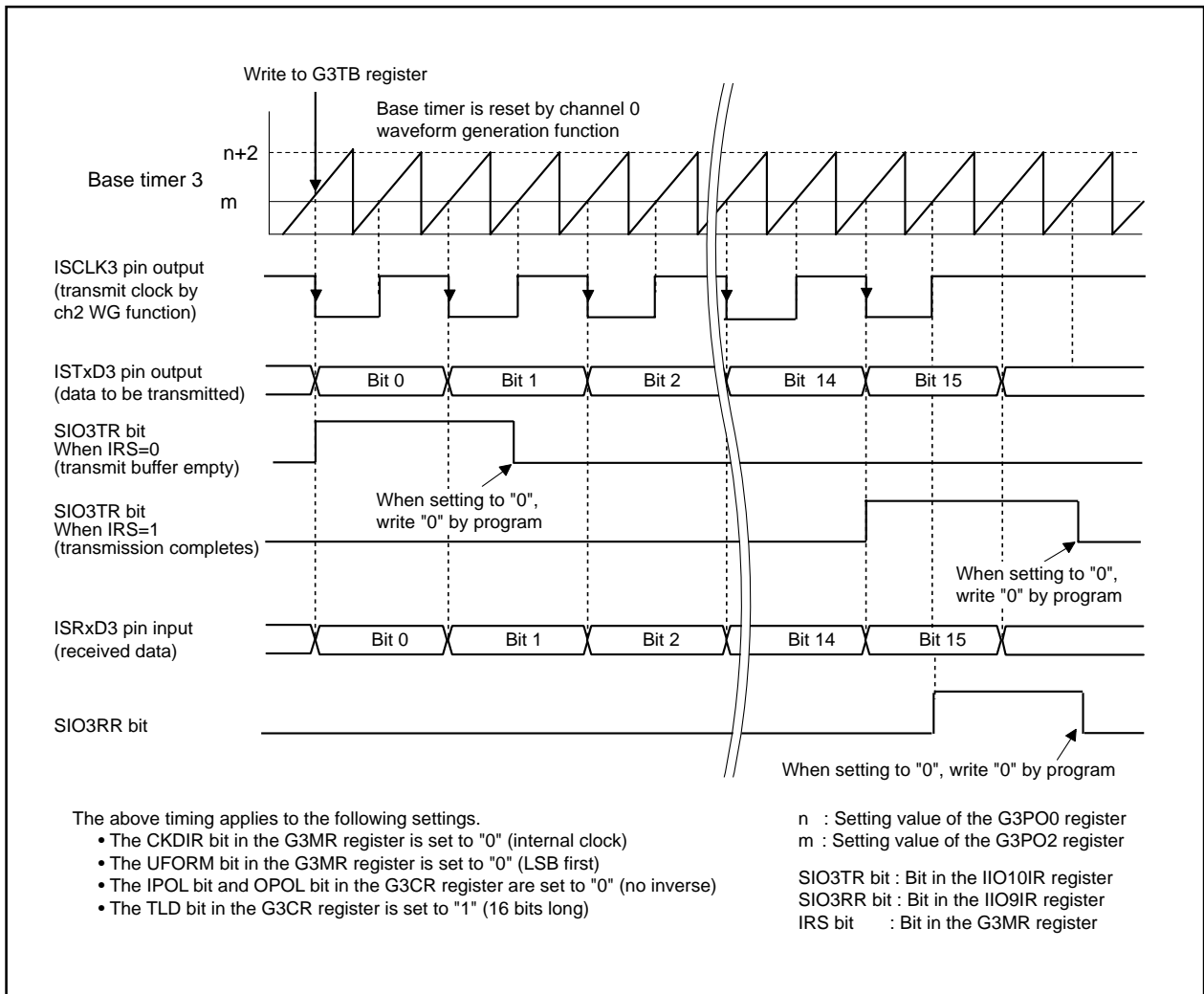


Figure 1. 22. 51. Transmit and Receive Operation (16-bit Length)

## A-D Converter

---

### A-D Converter

The A-D converter consists of two A-D converter circuits based on 10-bit successive approximation method configured with a capacitive-coupling amplifier.

Result of A-D conversion is stored into the A-D registers corresponding to selected pins.

Table 1.23.1 lists specifications of the A-D converter. Figure 1.23.1 shows a block diagram of the A-D converter. Table 1.23.2 lists differences between the A-D0 and A-D1 converters, which adopt the same conversion method. A-D0 and A-D1 can start a conversion simultaneously. Table 1.23.3 lists settings of the following pins; AN0 to AN7, AN00 to AN07, AN20 to AN27, AN150 to AN157, ANEX0, ANEX1,  $\overline{\text{ADTRG}}$ . Figures 1.24.2 to 1.24.7 show registers associated with the A-D converter.

#### Note

This section is described in the 144-pin package only as an example.  
In the 100-pin package, AN150 to AN157 pins are not provided.

## A-D Converter

Table 1.23.1. A-D Converter Specifications

Item	Specification
A-D conversion method	Successive approximation (capacitive coupling-amplifier)
Analog input voltage <sup>1</sup>	0V to AVCC (VCC)
Operating clock $\phi_{AD}$ <sup>2</sup>	fAD, fAD/2, fAD/3, fAD/4
Resolution	Selectable from 8 bits or 10 bits
Operating mode	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 : each 8 pins Extended input : 2 pins (ANEX0 and ANEX1)
A-D conversion start condition	<ul style="list-style-type: none"> <li>• Software trigger <ul style="list-style-type: none"> <li>- The ADST bit in the ADiCON0 register (i = 0,1) is set to "1" (A-D conversion starts) by program</li> <li>- The PST bit in the AD0CON2 register is set to "1" (A-D0 and A-D1 start a conversion simultaneously) by program</li> </ul> </li> <li>• External trigger (re-trigger enabled) When setting the ADST bit to "1" by program, <math>\overline{ADTRG}</math> pin input changes "H" to "L"</li> <li>• Hardware trigger (re-trigger enabled) When setting the ADST bit to "1" by program, one of the following interrupt is generated <ul style="list-style-type: none"> <li>- Timer B2 interrupt request of the three-phase motor control timer functions (after the ICTB2 register completes counting)</li> <li>- Intelligent I/O interrupt request Group2 channel 1 (A-D0), group3 channel 1 (A-D1)</li> </ul> </li> </ul>
Conversion rate per pin	<ul style="list-style-type: none"> <li>• Without the sample and hold function <ul style="list-style-type: none"> <li>8-bit resolution : 49 <math>\phi_{AD}</math> cycles</li> <li>10-bit resolution : 59 <math>\phi_{AD}</math> cycles</li> </ul> </li> <li>• With the sample and hold function <ul style="list-style-type: none"> <li>8-bit resolution : 28 <math>\phi_{AD}</math> cycles</li> <li>10-bit resolution : 33 <math>\phi_{AD}</math> cycles</li> </ul> </li> </ul>

## Notes:

1. Analog input voltage does not vary whether the sample and hold function is used or not.
2.  $\phi_{AD}$  frequency must be under 10 MHz.  
Without the sample and hold function, the  $\phi_{AD}$  frequency should be 250kHz or more.  
With the sample and hold function, set the  $\phi_{AD}$  frequency should be 1MHz or more.

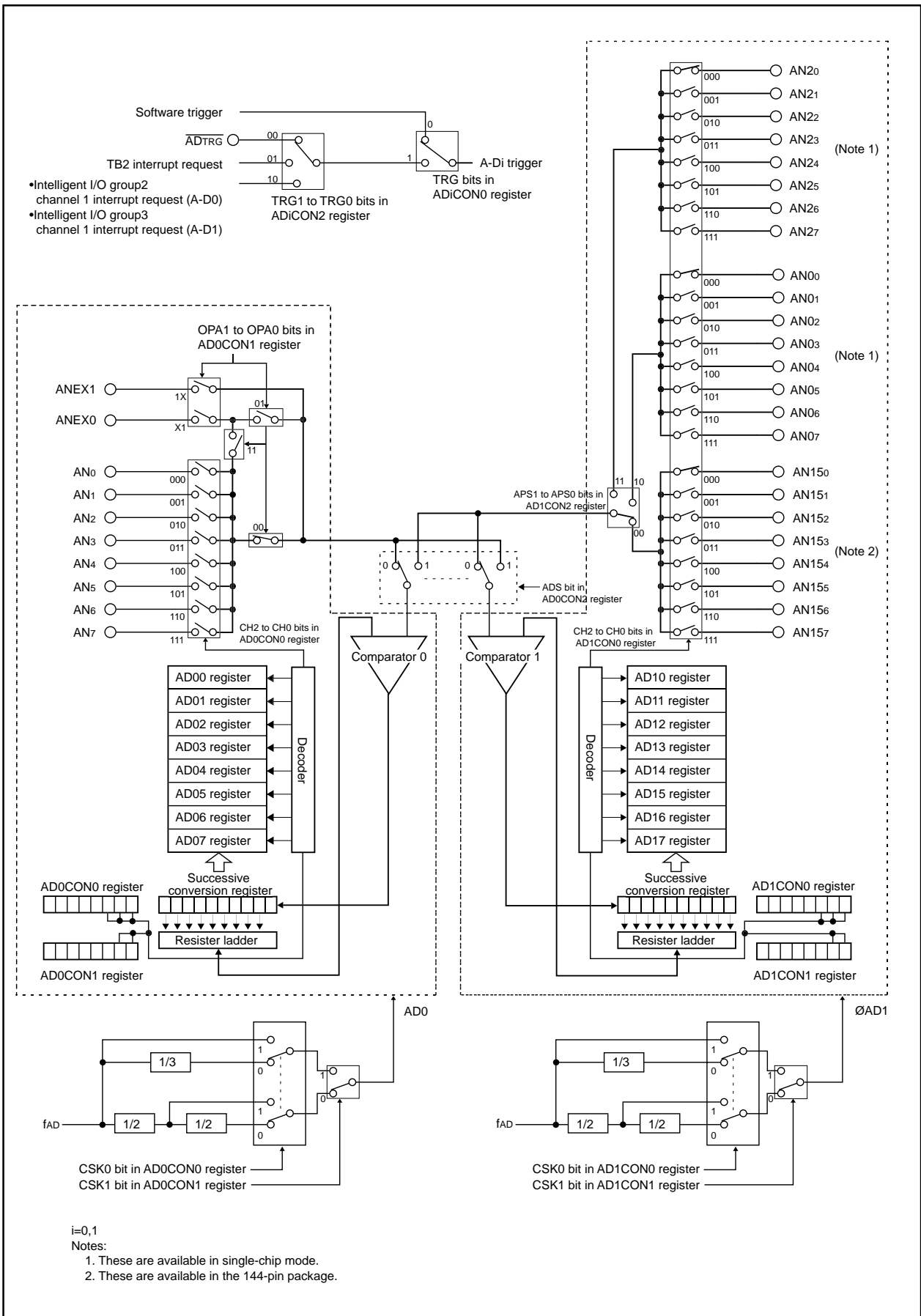
Table 1.23.2. Difference between A-D0 and A-D1

Item	A-D0	A-D1
Analog input pins <sup>1</sup>	AN (AN0 to AN7)	Selectable from AN0 (AN00 to AN07), AN2 (AN20 to AN27), AN15 (AN150 to AN157)
Extended Analog input pins	ANEX0, ANEX1	Not provided
External op-amp <sup>1</sup>	Enable	Disable
Intelligent I/O used as a trigger	Group2 channel 1	Group3 channel 1

## Notes:

1. When the ADS bit in the AD0CON2 register is set to "0" (channel replace disable).

**A-D Converter**



**Figure 1.23.1. A-D Converter Block Diagram**

## A-D Converter

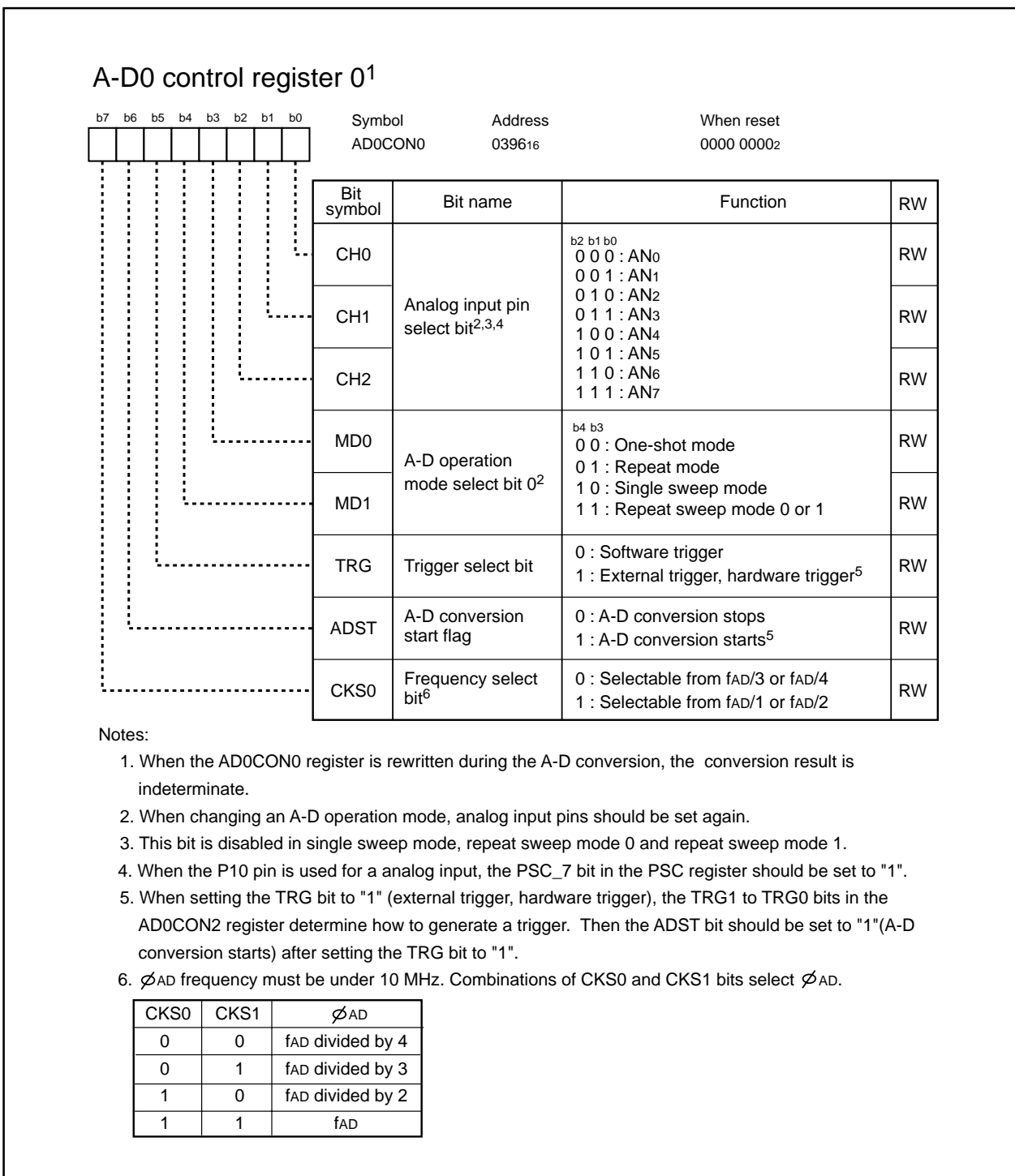
Table 1.23.3. Pin Settings

Port name	Function	Bit and setting					
		PD10, PD0, PD2, PD15, PD9 <sup>3</sup> registers	PS3 <sup>3</sup> , PS9 registers	PSL3, IPS registers	PUR0, PUR3, PUR4 registers		
P100	AN <sub>0</sub>	PD10_0 = 0	-	-	PU30 = 0		
P101	AN <sub>1</sub>	PD10_1 = 0					
P102	AN <sub>2</sub>	PD10_2 = 0					
P103	AN <sub>3</sub>	PD10_3 = 0					
P104	AN <sub>4</sub>	PD10_4 = 0			-	-	PU31 = 0
P105	AN <sub>5</sub>	PD10_5 = 0					
P106	AN <sub>6</sub>	PD10_6 = 0					
P107	AN <sub>7</sub>	PD10_7 = 0					
P00	AN0 <sub>0</sub> <sup>1</sup>	PD0_0 = 0	-	-	PU00 = 0		
P01	AN0 <sub>1</sub> <sup>1</sup>	PD0_1 = 0					
P02	AN0 <sub>2</sub> <sup>1</sup>	PD0_2 = 0					
P03	AN0 <sub>3</sub> <sup>1</sup>	PD0_3 = 0					
P04	AN0 <sub>4</sub> <sup>1</sup>	PD0_4 = 0			-	-	PU01 = 0
P05	AN0 <sub>5</sub> <sup>1</sup>	PD0_5 = 0					
P06	AN0 <sub>6</sub> <sup>1</sup>	PD0_6 = 0					
P07	AN0 <sub>7</sub> <sup>1</sup>	PD0_7 = 0					
P20	AN2 <sub>0</sub> <sup>1</sup>	PD2_0 = 0	-	-	PU04 = 0		
P21	AN2 <sub>1</sub> <sup>1</sup>	PD2_1 = 0					
P22	AN2 <sub>2</sub> <sup>1</sup>	PD2_2 = 0					
P23	AN2 <sub>3</sub> <sup>1</sup>	PD2_3 = 0					
P24	AN2 <sub>4</sub> <sup>1</sup>	PD2_4 = 0			-	-	PU05 = 0
P25	AN2 <sub>5</sub> <sup>1</sup>	PD2_5 = 0					
P26	AN2 <sub>6</sub> <sup>1</sup>	PD2_6 = 0					
P27	AN2 <sub>7</sub> <sup>1</sup>	PD2_7 = 0					
P150	AN15 <sub>0</sub> <sup>2</sup>	PD15_0 = 0	PS9_0 = 0	IPS2 = 1	PU42 = 0		
P151	AN15 <sub>1</sub> <sup>2</sup>	PD15_1 = 0	PS9_1 = 0				
P152	AN15 <sub>2</sub> <sup>2</sup>	PD15_2 = 0	-				
P153	AN15 <sub>3</sub> <sup>2</sup>	PD15_3 = 0	-				
P154	AN15 <sub>4</sub> <sup>2</sup>	PD15_4 = 0	PS9_4 = 0		-	PU43 = 0	
P155	AN15 <sub>5</sub> <sup>2</sup>	PD15_5 = 0	PS9_5 = 0				
P156	AN15 <sub>6</sub> <sup>2</sup>	PD15_6 = 0	-				
P157	AN15 <sub>7</sub> <sup>2</sup>	PD15_7 = 0	-				
P95	ANEX0	PD9_5 = 0	PS3_5 = 0	PSL3_5 = 1	PU27 = 0		
P96	ANEX1	PD9_6 = 0	PS3_6 = 0	PSL3_6 = 1			
P97	ANTRG	PD9_7 = 0	PS3_7 = 0	-	-		

## Notes:

1. This is available in single chip mode.
2. This is available in the 144-pin package.
3. The PD9 and PS3 registers should be set with subsequent to the instruction that the PRC2 bit in the PRCR register is set to "1" (write enable). Avoid generating an interrupt and operating a DMA transfer between the instruction to set the PRC2 bit to "1" and the instruction to write to the PD9 and PS3 registers.

**A-D Converter**



**Figure 1.23.2. AD0CON0 Register**

A-D Converter

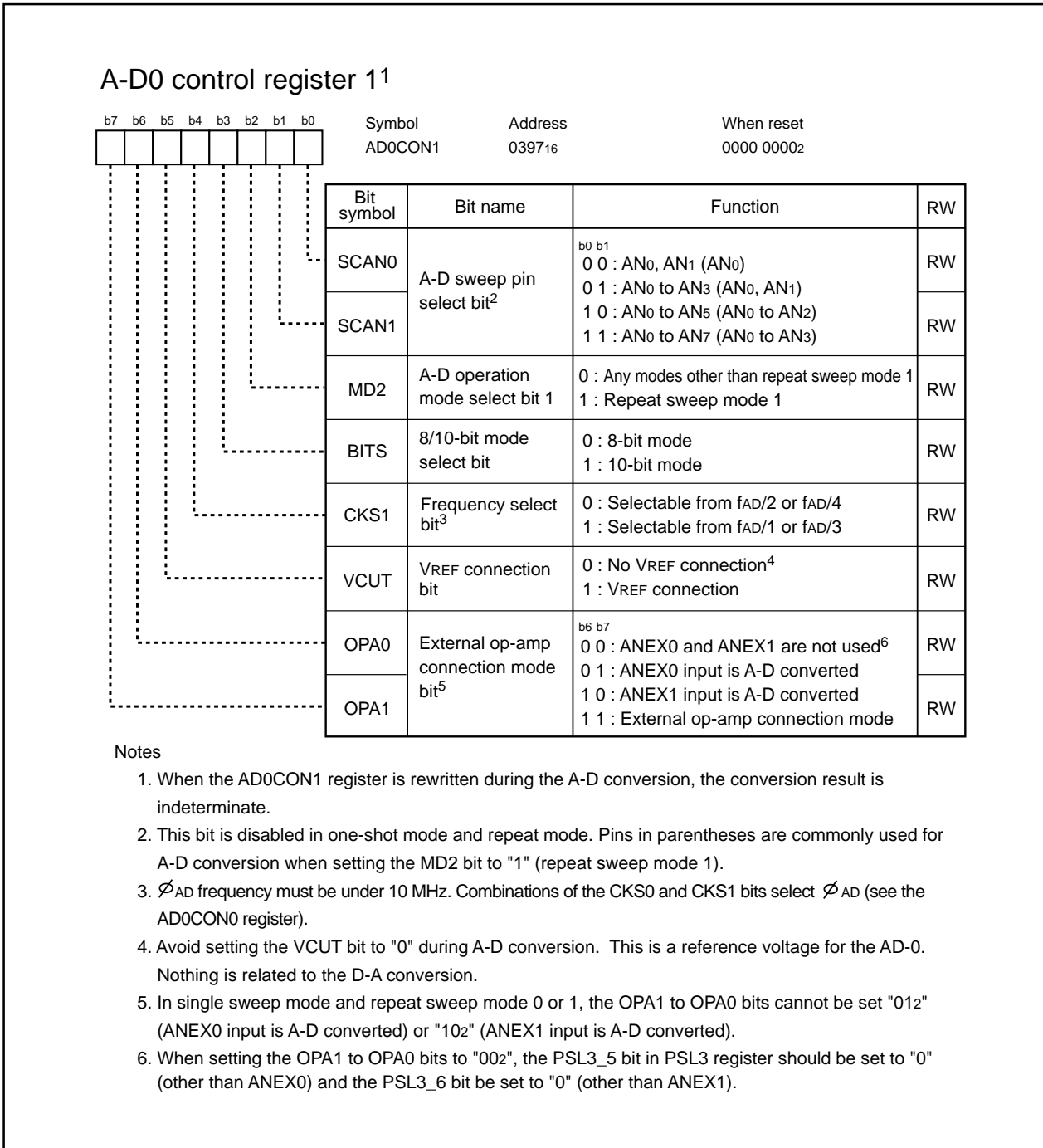
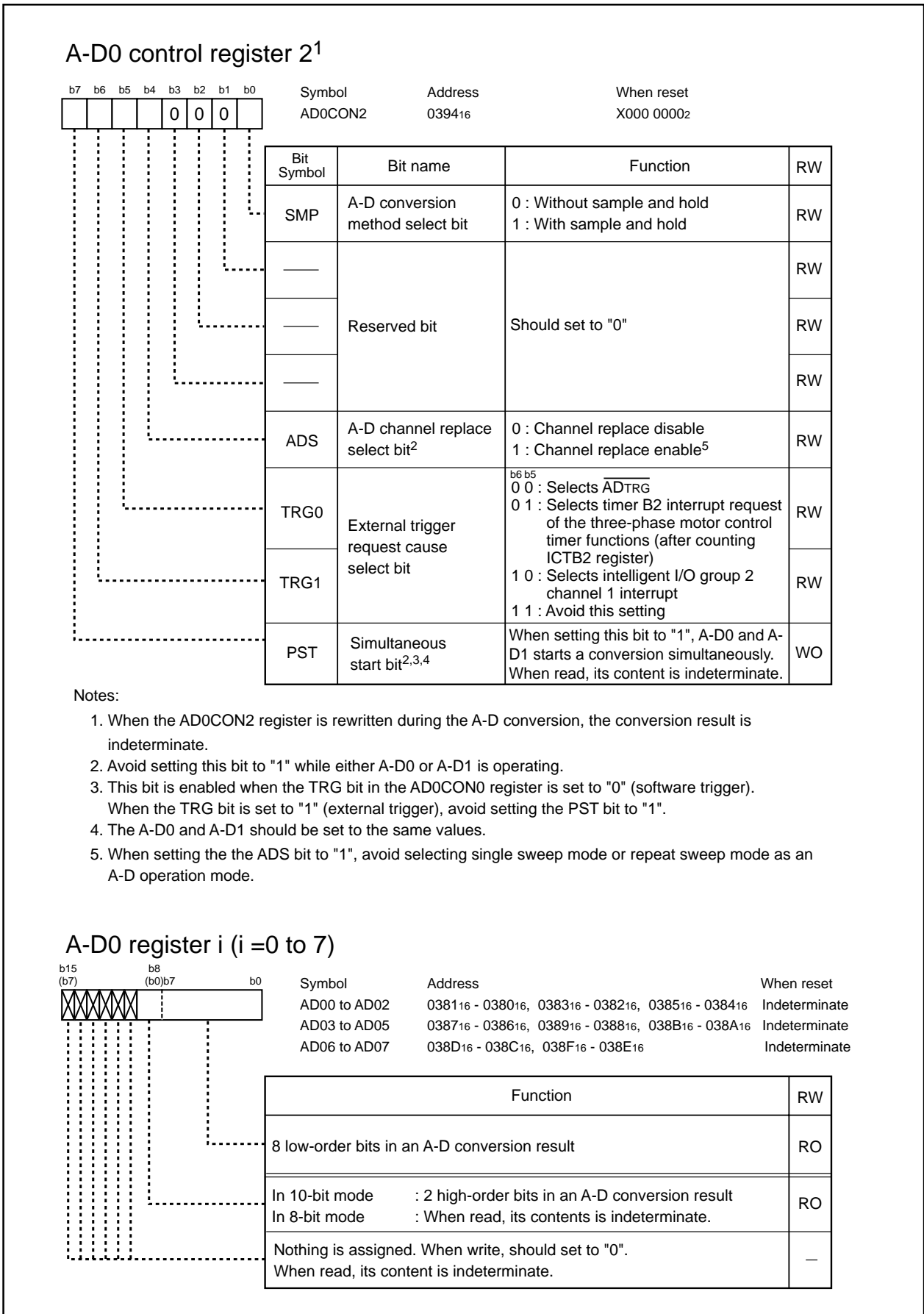


Figure 1.23.3. AD0CON1 Register

**A-D Converter**



**Figure 1.23.4. AD0CON2 Register and AD00 to AD07 Registers**

A-D Converter

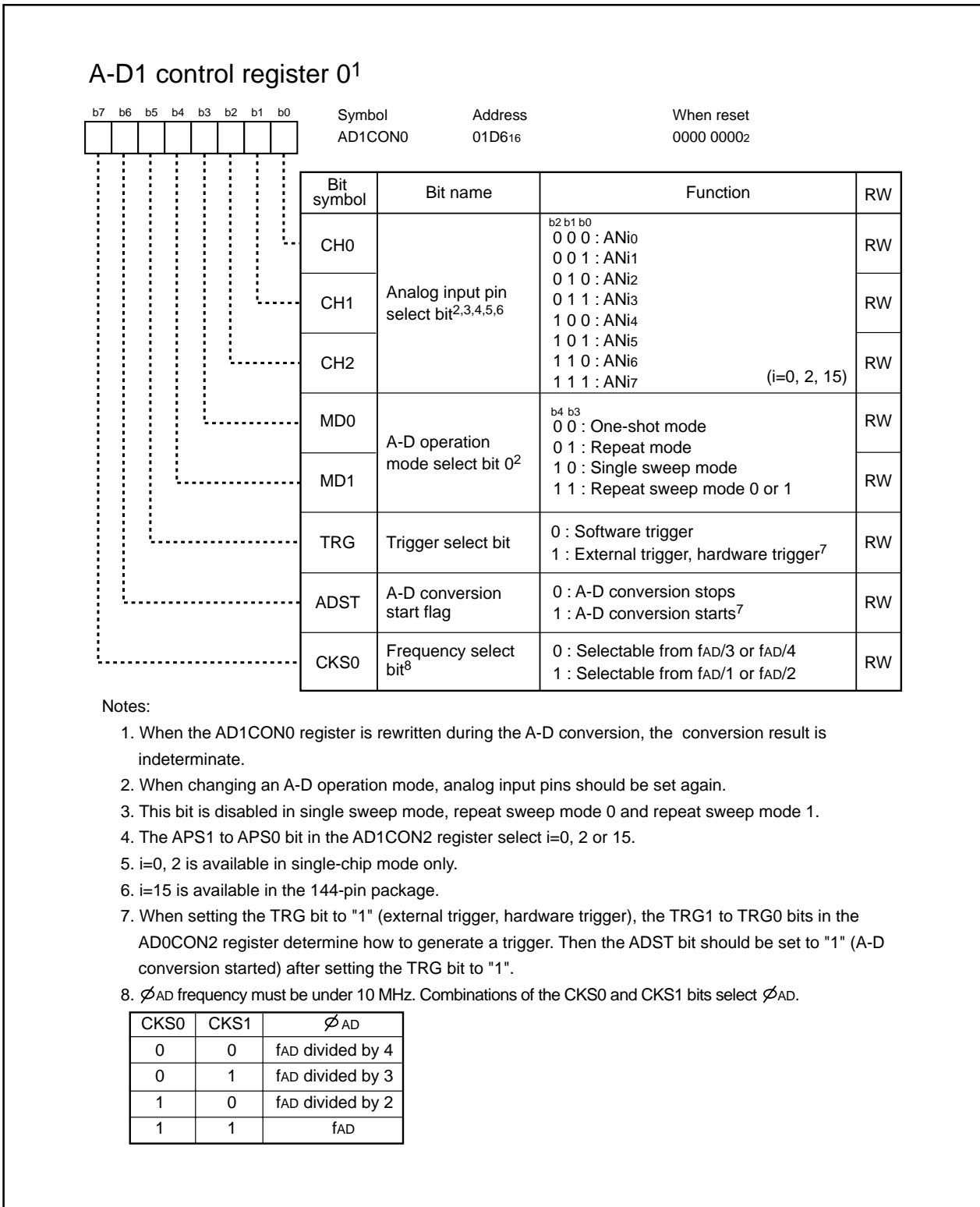
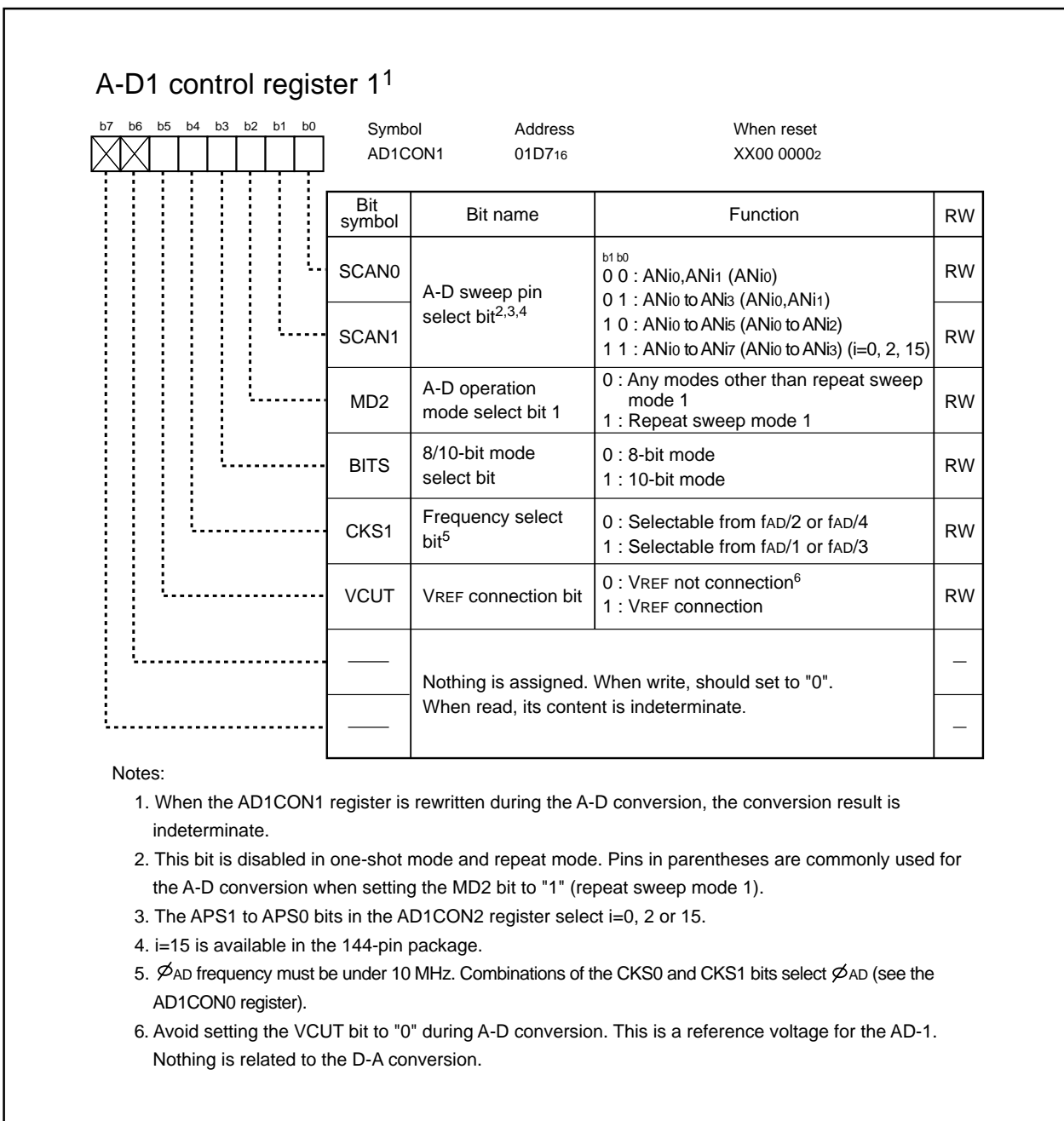


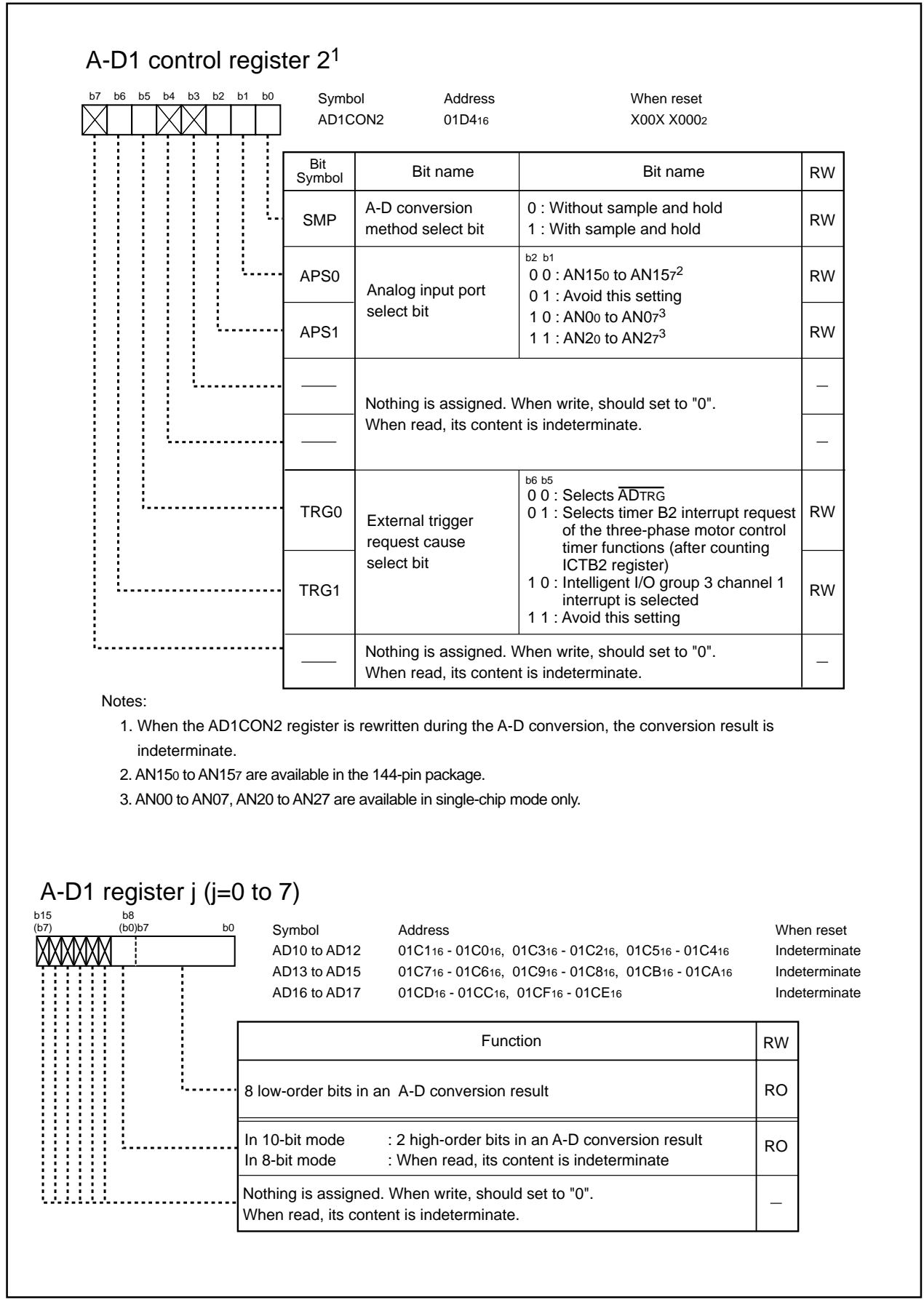
Figure 1.23.5. AD1CON0 Register

**A-D Converter**



**Figure 1.23.6. AD1CON1 Register**

**A-D Converter**



**Figure 1.23.7. AD1CON2 Register and AD10 to AD17 Registers**

## A-D Converter

## Mode Description

## (1) One-shot Mode

In one-shot mode, analog voltage that is input to a pin selected is converted to a digital form once. Table 1.23.4 lists specifications of one-shot mode.

Table 1.23.4. One-shot Mode Specifications

Item	Specification
Function	Analog voltage input to a pin that the CH2 to CH0 bits in the ADiCON0 register (i=0,1) select is converted to a digital form once.
Start condition	(a) When the TRG bit in the ADiCON0 register is set to "0" (software trigger) <ul style="list-style-type: none"> <li>The ADST bit in the ADiCON0 register is set to "1" (A-D conversion starts) by program</li> <li>The PST bit in the AD0CON2 register is set to "1" (A-D0 and A-D1 start the A-D conversion simultaneously) by program</li> </ul> (b) When the TRG bit is set to "1" (external trigger, hardware trigger) <ul style="list-style-type: none"> <li>The <math>\overline{\text{ADTRG}}</math> input pin changes "L" to "H" after the ADST bit is set to "1" by program</li> <li>One of the following interrupt is generated after the ADST bit is set to "1" by program <ul style="list-style-type: none"> <li>- Timer B2 interrupt request of three-phase motor control timer functions after the ICTB2 register completes counting</li> <li>- Intelligent I/O interrupt request <ul style="list-style-type: none"> <li>Group2 channel 1(A-D0), group3 channel 1 (A-D1)</li> </ul> </li> </ul> </li> </ul>
Stop condition	<ul style="list-style-type: none"> <li>A-D conversion is completed (When internal trigger is selected, the ADST bit is set to "0")</li> <li>The ADST bit is set to "0" (A-D conversion stops) by program</li> </ul>
Interrupt request generation timing	A-D conversion is completed
Input pin	Selectable from AN <sub>0</sub> to AN <sub>7</sub> , ANEX0 or ANEX1 Selectable from AN <sub>j0</sub> to AN <sub>j7</sub> (j = 0, 2, 15)
Reading of A-D converter result	Read the AD <sub>ik</sub> register (k=0 to 7) corresponding to a selected pin

## (2) Repeat Mode

In repeat mode, analog voltage that is input to a pin selected is repeatedly converted to a digital form. Table 1.23.5 lists specifications of repeat mode.

Table 1.23.5. Repeat Mode Specifications

Item	Specification
Function	Analog voltage input to pin that the CH2 to CH0 bits in the ADiCON0 register (i=0,1) select is repeatedly converted to a digital form
Start condition	Same as one-shot mode
Stop condition	The ADST bit is set to "0" (A-D conversion stops) by program
Interrupt request generation timing	Not generated
Input pin	Selectable from AN <sub>0</sub> to AN <sub>7</sub> , ANEX0 or ANEX1 Selectable from AN <sub>j0</sub> to AN <sub>j7</sub> (j = 0, 2, 15), ANEX0, ANEX1
Reading of A-D converter result	Read the AD <sub>ik</sub> registers (k = 0 to 7) corresponding to a selected pin

## A-D Converter

**(3) Single Sweep Mode**

In single sweep mode, analog voltage that is input to pins selected is converted one-by-one to a digital form. Table 1.23.6 lists specifications of single sweep mode.

**Table 1.23.6. Single Sweep Mode Specifications**

Item	Specification
Function	Analog voltage input to pins that the SCAN1 to SCAN0 bits in the ADiCON1 register ( $i = 0, 1$ ) select is converted one-by-one to a digital form.
Start condition	Same as one-shot mode
Stop condition	<ul style="list-style-type: none"> <li>A-D conversion is completed (When internal trigger is selected, the ADST bit is set to "0")</li> <li>The ADST bit is set to "0" (A-D conversion stops) by program</li> </ul>
Interrupt request generation timing	Sweeping is completed
Input pin	Selectable from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), or AN0 to AN7 (8 pins) Selectable from ANj0 to ANj1 (2 pins), ANj0 to ANj3 (4 pins), ANj0 to ANj5 (6 pins), or ANj0 to ANj7 (8 pins) ( $j = 0, 2, 15$ )
Reading of A-D converter result	Read the ADik register ( $k = 0$ to 7) corresponding to selected pins

**(4) Repeat Sweep Mode 0**

In repeat sweep mode 0, analog voltage that is input pins selected is repeatedly converted into a digital form. Table 1.23.7 lists specifications of repeat sweep mode 0.

**Table 1.23.7. Repeat Sweep Mode 0 Specifications**

Item	Specification
Function	Analog voltage input to pins that the SCAN1 to SCAN0 bits in the ADiCON0 register ( $i=0,1$ ) select is repeatedly converted into a digital form.
Start condition	Same as one-shot mode
Stop condition	The ADST bit in the ADiCON0 register is set to "0" by program
Interrupt request generation timing	Not generated
Input pin	Selectable from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), or AN0 to AN7 (8 pins) Selectable from ANj0 to ANj1 (2 pins), ANj0 to ANj3 (4 pins), ANj0 to ANj5 (6 pins), or ANj0 to ANj7 (8 pins) ( $j = 0, 2, 15$ )
Reading of A-D converter result	Read the ADik registers ( $k = 0$ to 7) corresponding to selected pins

**A-D Converter****(5) Repeat Sweep Mode 1**

In repeat sweep mode 1, analog voltage that is input to all pins is repeatedly converted to a digital form with putting emphasis on a pin or pins selected. Table 1.23.8 lists specifications of repeat sweep mode 1.

**Table 1. 23. 8. Repeat Sweep Mode 1 Specifications**

Item	Specification
Function	Analog voltage input to all pins is repeatedly converted to a digital form with putting emphasis on a pin or pins selected the SCAN1 to SCAN0 bits in the ADiCON1 register (i=0,1). e.g. When ANj0 is selected (j =none, 0, 2, 15), analog voltage is converted to a digital form as the following order : ANj0 → ANj1 → ANj0 → ANj2 → ANj0 → ANj3 ..... etc.
Start condition	Same as one-shot mode
Stop condition	The ADST bit in the ADiCON0 register is set to "0" (A-D conversion stops) by program
Interrupt request generation timing	Not generated
Input pin	ANj0 to ANj7
Pins to be put emphasis	Selectable from AN0 to AN1 (2 pins), AN0 to AN3 (4 pins), AN0 to AN5 (6 pins), or AN0 to AN7 (8 pins) Selectable from ANj0 to ANj1 (2 pins), ANj0 to ANj3 (4 pins), ANj0 to ANj5 (6 pins), or ANj0 to ANj7 (8 pins) (j = 0, 2, 15)
Reading of A-D converter result	Read the ADik registers (k = 0 to 7) corresponding to selected pins

**Function****(a) Resolution Select Function**

The BITS bit in the ADiCON1 register (i = 0,1) determines resolution. When setting the BITS bit to "1" (10-bit precision), an A-D conversion result is stored into bits 0 to 9 in the ADij register (j = 0 to 7). When setting the BITS bit to "0" (8-bit precision), an A-D conversion result is stored into bits 0 to 7 in the ADij register.

**(b) Sample and Hold Function**

When the SMP bit in the ADiCON2 register is set to "1" (sample and hold), conversion rate per pin increases. 28  $\emptyset$ AD cycles are achieved with an 8-bit resolution and 33  $\emptyset$ AD are with 10-bit resolution. The sample and hold function can be selected in all modes. The A-D conversion should be started after selecting whether the sample and hold function is to be used or not.

**(c) Trigger Select Function**

Combinations of the TRG bit in the ADiCON0 register (i=0,1) and the TRG1 to TRG0 bits in the ADiCON2 register determines a start trigger for the A-D conversion. Table 1.23.9 lists settings of the trigger select function.

## A-D Converter

Table 1.23.9. Trigger Select Function Setting

Bit and setting value		Trigger
ADiCON0 register	ADiCON2 register	
TRG = 0	-	Software trigger When the ADST bit in the ADiCON0 register is set to "1" by program, the A-Di starts the A-D conversion.
	-	Two-circuit simultaneous start When the PST bit in the AD0CON2 register is set to "1" by program, the A-D0 and A-D1 start the A-D conversion simultaneously. (Refer to "(d) Two-circuit simultaneously start" below.)
TRG = 1 <sup>1</sup>	TRG1 to TRG0 = 00 <sub>2</sub>	Falling edge of the $\overline{\text{ADTRG}}$ input signal
	TRG1 to TRG0 = 01 <sub>2</sub>	Timer B2 interrupt request of three-phase motor control timer functions (after the ICTB2 register completes counting)
	TRG1 to TRG0 = 10 <sub>2</sub>	Intelligent I/O interrupt request Group2 channel 1 (A-D0), Group3 channel 1 (A-D1)

i = 0, 1

## Notes

- 1: When the ADST bit is set to "1", A-Di starts A-D conversion at trigger generating.

**(d) Two-Circuit Simultaneous Start (Software Trigger)**

The A-D0 and A-D1 can start the A-D conversion simultaneously when the PST bit in the AD0CON2 register is set to "1" (two-circuit simultaneous start).

Avoid setting the PST bit to "1" while either A-D0 or A-D1 circuit is operating. Avoid setting the PST bit to "1" when TRG bit is set "1" (external trigger). When using the PST bit, avoid setting the ADST bit to "1" (A-D conversion starts).

**(e) Input Pin Replace Function**

When the ADS bit in the AD0CON2 register is set to "1" (channel replace enable), channels of the A-D0 can be replaced with channels of the A-D1 and vice versa.

AN<sub>j</sub> (j = 0 to 7) input is converted to a digital form in the A-D1 and the conversion result is stored into the AD1<sub>j</sub> register. AN<sub>0j</sub>, AN<sub>2j</sub> and AN<sub>15j</sub> inputs are converted to digital forms in the A-D0 and the conversion results are stored into the AD0<sub>j</sub> register. Both AD0CON0 register and AD1CON0 register and both AD0CON1 register and AD1CON1 register should be set the same value. The OPA1 to OPA0 bits in the AD0CON1 register are set to "00<sub>2</sub>" (ANEX0 and ANEX1 are not used).

**(f) Extended Analog Input Pins**

In one-shot mode and repeat mode, the ANEX0 and ANEX1 pins can be used as analog input pins. The OPA1 to OPA0 bits in the AD0CON1 register select pins to be used as analog input pins. The A-D conversion result for the ANEX0 input is stored into the AD00 register and the result for the ANEX1 is into the AD01 register.

**A-D Converter**

**(g) External Operation Amplifier (Op-Amp) Connection Mode**

In external op-amp connection mode, multiple analog voltage inputs can be amplified by one external op-amp, with ANEX0 and ANEX1 as extended analog input pins.

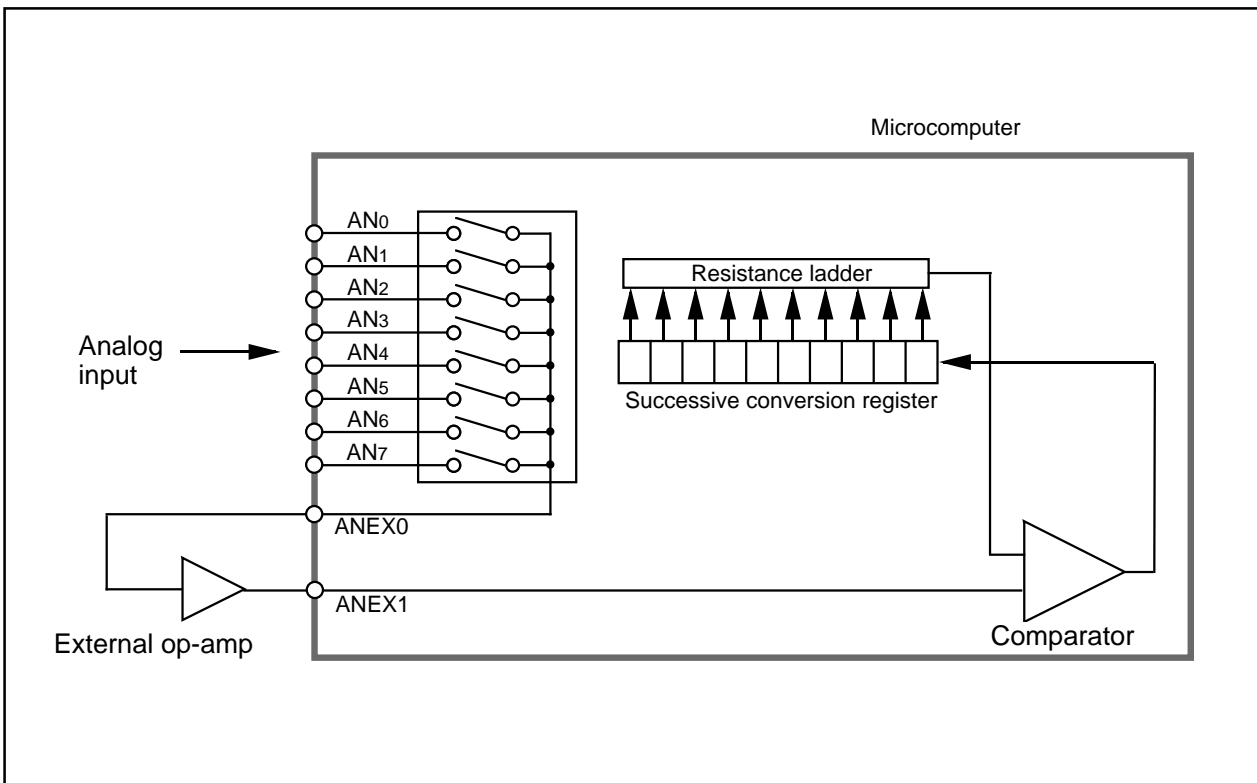
When the OPA1 to OPA0 bits in the ADCON1 register are set to "112" (external op-amp connection), voltage input to the AN0 to AN7 pins are output from the ANEX0 pin. This output should be amplified by the external op-amp to input to the ANEX1 pin.

Analog voltage input to the ANEX1 is converted to a digital form and the A-D conversion result is stored into the corresponding ADij register (i=0, 1, j=0 to 7). A-D conversion rate varies, depending on a response of the external op-amp. Avoid connecting the ANEX0 to ANEX1 pins directly.

Figure 1.23.8 shows an example of the external op-amp connection.

**Table 1. 23. 10. Extended Analog Input Pin Settings**

ADCON1 register		ANEX0 function	ANEX1 function
OPA1	OPA0		
0	0	Not used	Not used
0	1	Analog input to P95 (AN0)	Not used
1	0	Not used	Analog input to P96 (AN1)
1	1	Output to external op-amp	Input from external op-amp



**Figure 1.23.8. External Ope-amp Connection**

## A-D Converter

### (h) Function for Power Dissipation Reduction

When the A-D converter is not used, the VCUT bit in the ADiCON1 register ( $i=0,1$ ) allows a resistance ladder of the A-D converter to isolate it from the reference voltage input pin (VREF). Power dissipation is reduced by shutting off any current flow into the resistance ladder from the VREF pin.

When using the A-D converter, the VCUT bit should be set to "1" (VREF connection) before the ADST bit in the ADiCON0 register is set to "1" (A-D conversion starts). Avoid setting the ADST bit and VCUT bit to "1" simultaneously. Avoid setting the VCUT bit to "0" (no VREF connection) during the A-D conversion. The VCUT bit does not affect VREF of the D-A converter.

### (i) Analog Input Pin and External Sensor Equivalent Circuit

Figure 1.23.9 shows an example of the analog input pin and external sensor equivalent circuit.

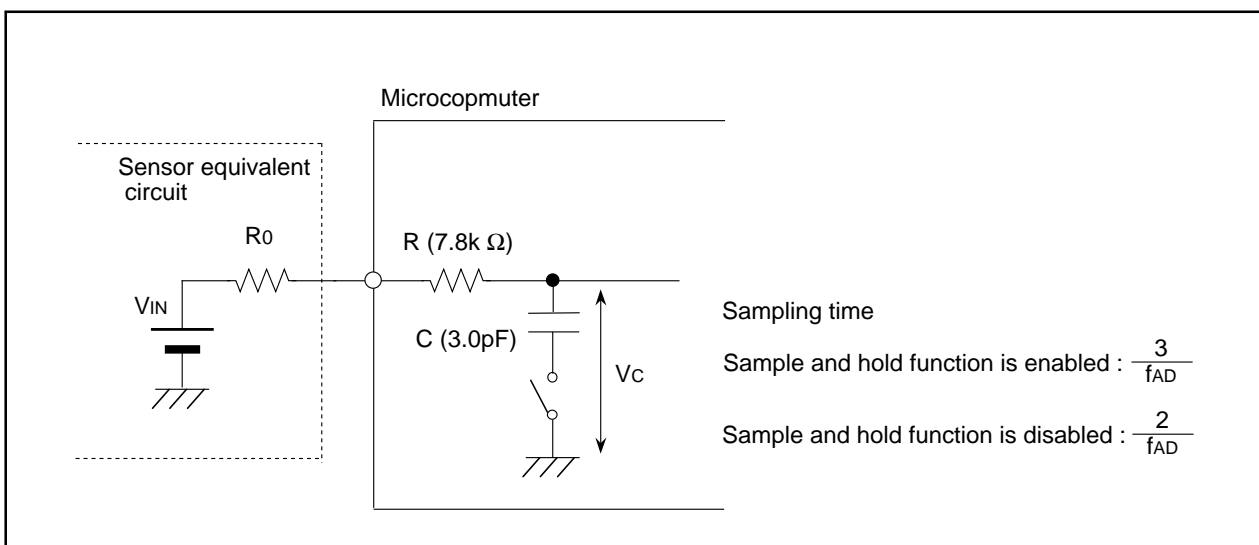


Figure 1.23.9. Analog Input Pin and External Sensor Equivalent Circuit

## Precaution

### (1) Read timing for A-D Conversion Result

If the A-D conversion is completed and the CPU reads the AD<sub>ij</sub> register ( $i=0,1, j=0$  to 7) when the A-D conversion result is stored into the AD<sub>ij</sub> register, wrong A-D conversion value is stored into the AD<sub>ij</sub> register. This event occurs when selecting the main clock divided or sub clock as the CPU clock source.

- **In One-shot or Single Sweep Mode**

Confirm that the A-D conversion is completed before reading the AD<sub>ij</sub> register. (The IR bit in the ADiC register indicates whether A-D conversion is completed or not).

- **In Repeat Mode or Repeat Sweep Mode 0 or 1**

Use the undivided main clock as the CPU clock.

## A-D Converter

---

### **(2) A-D conversion Result after forced-termination of A-D conversion**

When the ADST bit in the ADiCON0 register ( $i = 0, 1$ ) is set to "0" (A-D conversion stops) by program for a forced-termination during the A-D conversion, an A-D conversion result is indeterminate.

The AD<sub>ij</sub> register ( $j=0$  to 7), which does not perform the A-D<sub>i</sub> conversion, may also be indeterminate.

When A-D<sub>i</sub> is forcibly terminated, avoid using value of all AD<sub>ij</sub> registers.

When the ADS bit in the AD0CON2 register is set to "0" (channel replace disable) and either A-D0 or A-D1 is forced to terminate, another AD<sub>i</sub> as a survivor normally completes the A-D conversion. Values of the AD<sub>ij</sub> registers that does not perform the A-D conversion remain unchanged.

## D-A Converter

### D-A Converter

The D-A converter consists of two D-A converter circuit based on 8-bit R-2R method.

D-A conversion is performed when a value is written to corresponding the DA<sub>i</sub> registers (i=0,1). The DA<sub>i</sub>E bit in the DACON register should be set to output a D-A conversion result. The DA<sub>i</sub>E bit should be set to "1" (input enabled) to inhibit a pull-up of a corresponding port.

Output analog voltage (V) is calculated from a value n (n=decimal) that is set in the D-A register.

$$V = \frac{V_{REF} \times n}{256} \quad (n = 0 \text{ to } 255)$$

V<sub>REF</sub> : reference voltage (it is not related to the VCUT bit in the ADiCON1 register)

Table 1.24.1 lists specifications of the D-A converter. Table 1.24.2 lists pin settings of the DA0 and DA1 pins. Figure 1.24.1 shows a block diagram of the D-A converter. Figure 1.24.2 shows the D-A control register. Figure 1.24.3 shows a D-A converter equivalent circuit.

When the D-A converter is not used, the DA<sub>i</sub> register is set to "0016" and the DA<sub>i</sub>E bit is set to "0" (output disabled).

**Table 1.24.1. D-A Converter Specifications**

Item	Specification
Conversion method	R-2R method
Resolution	8 bits
Analog output pin	2 channels

**Table 1.24.2. Pin Settings**

Port	Function	Bit and setting value		
		PD9 register <sup>1</sup>	PS3 register <sup>1</sup>	PSL3 register
P93	DA0 output	PD9_3=0	PS3_3=0	PSL3_3=1
P94	DA1 output	PD9_4=0	PS3_4=0	PSL3_4=1

Notes:

1. The PD9 and PS3 registers should be set with subsequent to the instruction that the PRC2 bit in the PRCR register is set to "1" (write enable). Avoid generating an interrupt and operating a DMA transfer between the instruction to set the PRC2 bit to "1" and the instruction to write to the PD9 and PS3 registers.

## D-A Converter

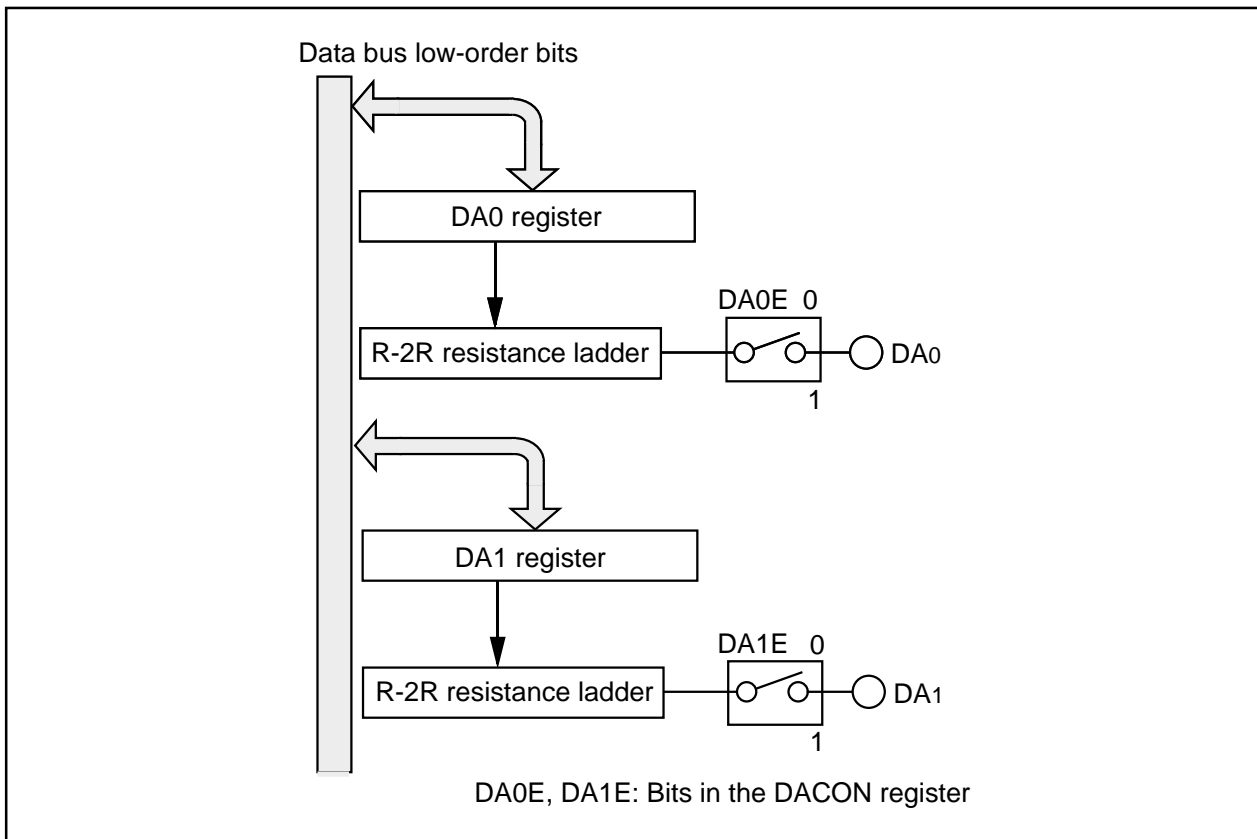


Figure 1.24.1. D-A Converter

D-A Converter

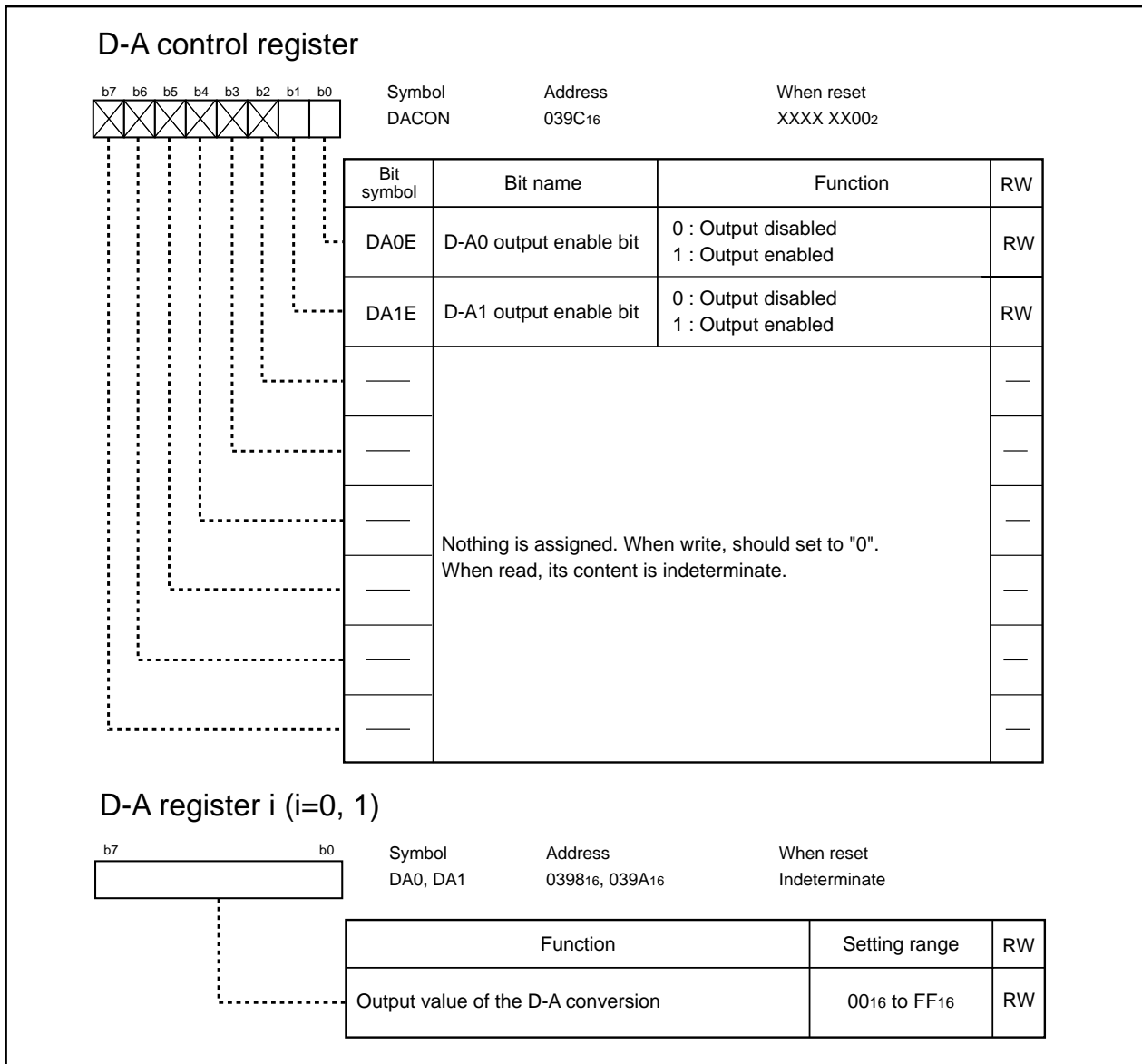


Figure 1.24.2. DACON Register, DA0 and DA1 Registers

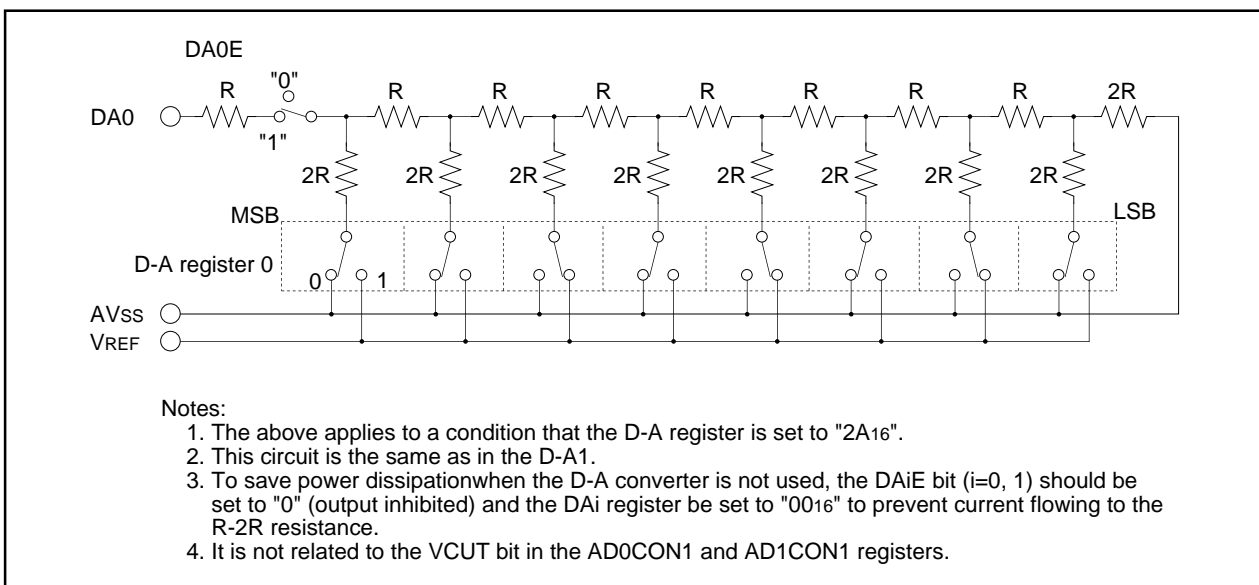


Figure 1.24.3. D-A converter Equivalent Circuit

## CRC calculation

# CRC Calculation

CRC (Cyclic Redundancy Check) calculation detects an error in data blocks. A generator polynomial of CRC\_CCITT ( $X^{16} + X^{12} + X^5 + 1$ ) generates CRC code.

The CRC code is a 16-bit code generated for a block of a given 8-bit data. The CRC code is set in the CRCD register whenever one byte-data is transferred to the CRCIN register after writing a default value into the CRCD register. CRC code generation for one byte-data is completed in two cycles.

Figure 1.25.1 shows a block diagram of a CRC circuit. Figure 1.25.2 shows registers related to CRC. Figure 1.25.3 shows an example of CRC calculation.

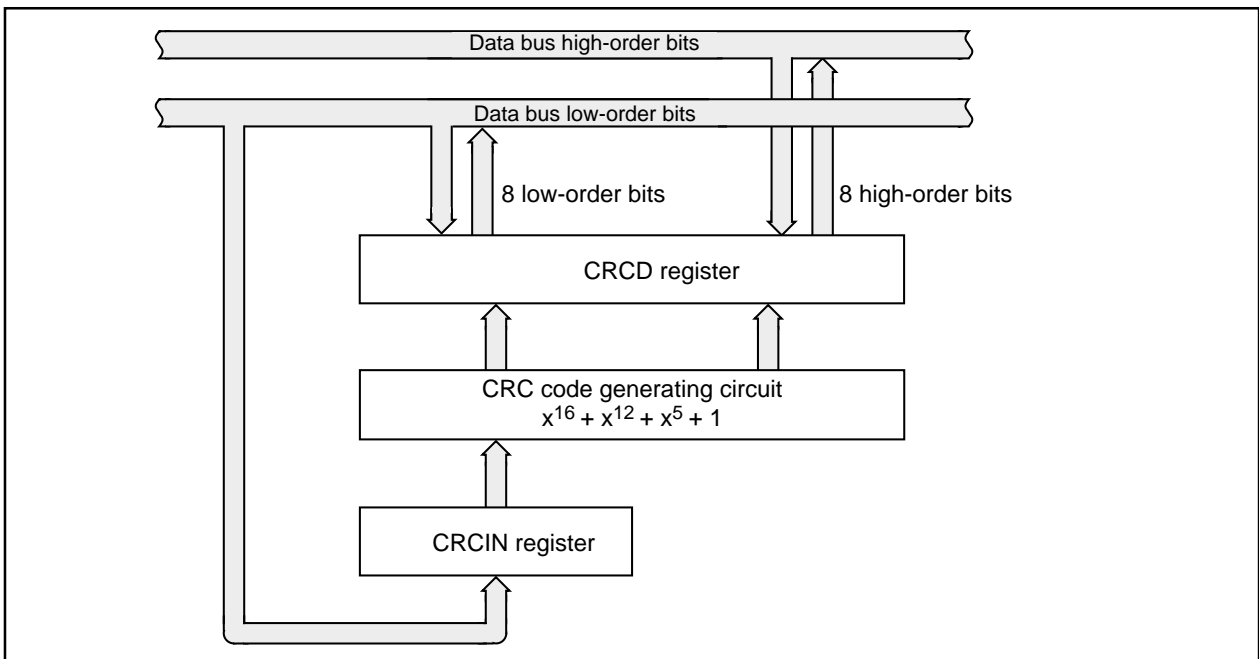


Figure 1.25.1. CRC Calculation Block Diagram

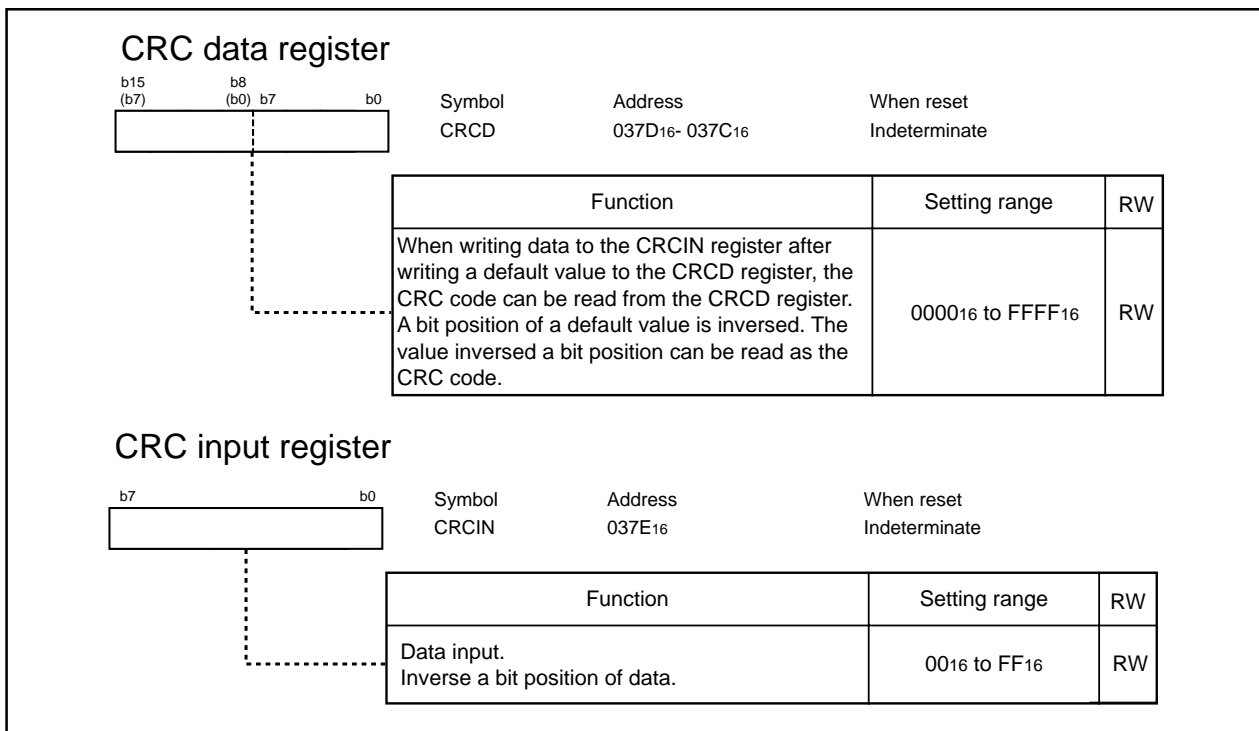


Figure 1.25.2. CRCD Register and CRCIN Register

## CRC calculation

### CRC calculation and setup procedure to generate CRC code for "80C416"

○ CRC calculation for M16C

CRC code : remainder of a division,  $\frac{\text{value that is inverted a bit position of a value written in CRCIN register}}{\text{generation polynomial}}$

Generation polynomial :  $X^{16} + X^{12} + X^5 + 1$  (1 0001 0000 0010 0001<sub>2</sub>)

○ Setting steps

(1) Inverse a bit position of "80C416" in byte units by program

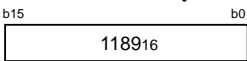
"8016" → "0116", "C416" → "2316"

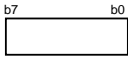
(2) Set 0000<sub>16</sub> (default value) →  CRCD register

(3) Set 01<sub>16</sub> → 

CRCIN register

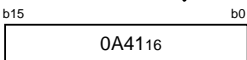
A bit position of the CRC code for "8016" (9188<sub>16</sub>) is inverted to "118916", which is stored into the CRCD register after 2 cycles.

 CRCD register

(4) Set 23<sub>16</sub> → 

CRCIN register

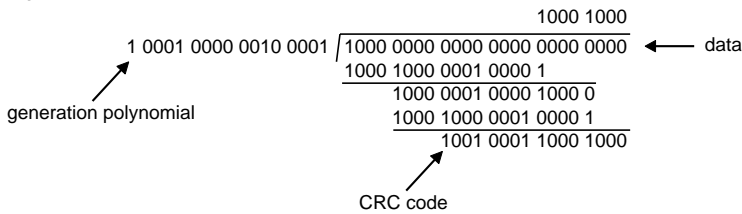
A bit position of the CRC code for "80C416" (8250<sub>16</sub>) is inverted to "0A4116", which is stored into the CRCD register after 2 cycles.

 CRCD register

○ Details of CRC calculation

As shown in (3) above, a bit position of a value set in the CRCIN register "0116" (00000001<sub>2</sub>) is inverted to become "10000000<sub>2</sub>".

Add "1000 0000 0000 0000 0000 0000<sub>2</sub>", as "10000000<sub>2</sub>" plus 16 digits, to "000016" as a default value of CRCD register to perform the modulo-2 division.



Modulo-2 Arithmetic is calculated on the law below.

0 + 0 = 0  
 0 + 1 = 1  
 1 + 0 = 1  
 1 + 1 = 0  
 - 1 = 1

"0001 0001 1000 1001<sub>2</sub>", inverted a bit position of "1001 0001 1000 1000<sub>2</sub> (9188<sub>16</sub>)" as a remainder, can be read from the CRCD register.

When going on to (4) above, "2316 (00100011<sub>2</sub>)" written in the CRCIN register is inverted to become "11000100<sub>2</sub>".

Add "1100 0100 0000 0000 0000 0000<sub>2</sub>", as "11000100<sub>2</sub>" plus 16 digits, to "1001 0001 1000 1000<sub>2</sub>" as a remainder of (3) left in the CRCD register to perform the modulo-2 division.

"0000 1010 0100 0001<sub>2</sub> (0A4116)", inverted a bit position of the remainder, can be read from CRCD register.

Figure 1.25.3. CRC Calculation

X-Y Conversion

# X-Y Conversion

X-Y conversion rotates a 16 x 16 matrix data by 90 degrees and inverse high-order bits and low-order bits of a 16-bit data. Figure 1.26.1 shows the XYC register.

The XiR register (i=0 to 15) and YjR register (j=0 to 15) are allocated to the same address. The XiR register is a write-only register, while the YjR register is a read-only register. The XiR and YjR registers should be accessed in 16-bit units from an even address. Operation cannot be guaranteed if the XiR and YiR registers are accessed in 8-bit units.

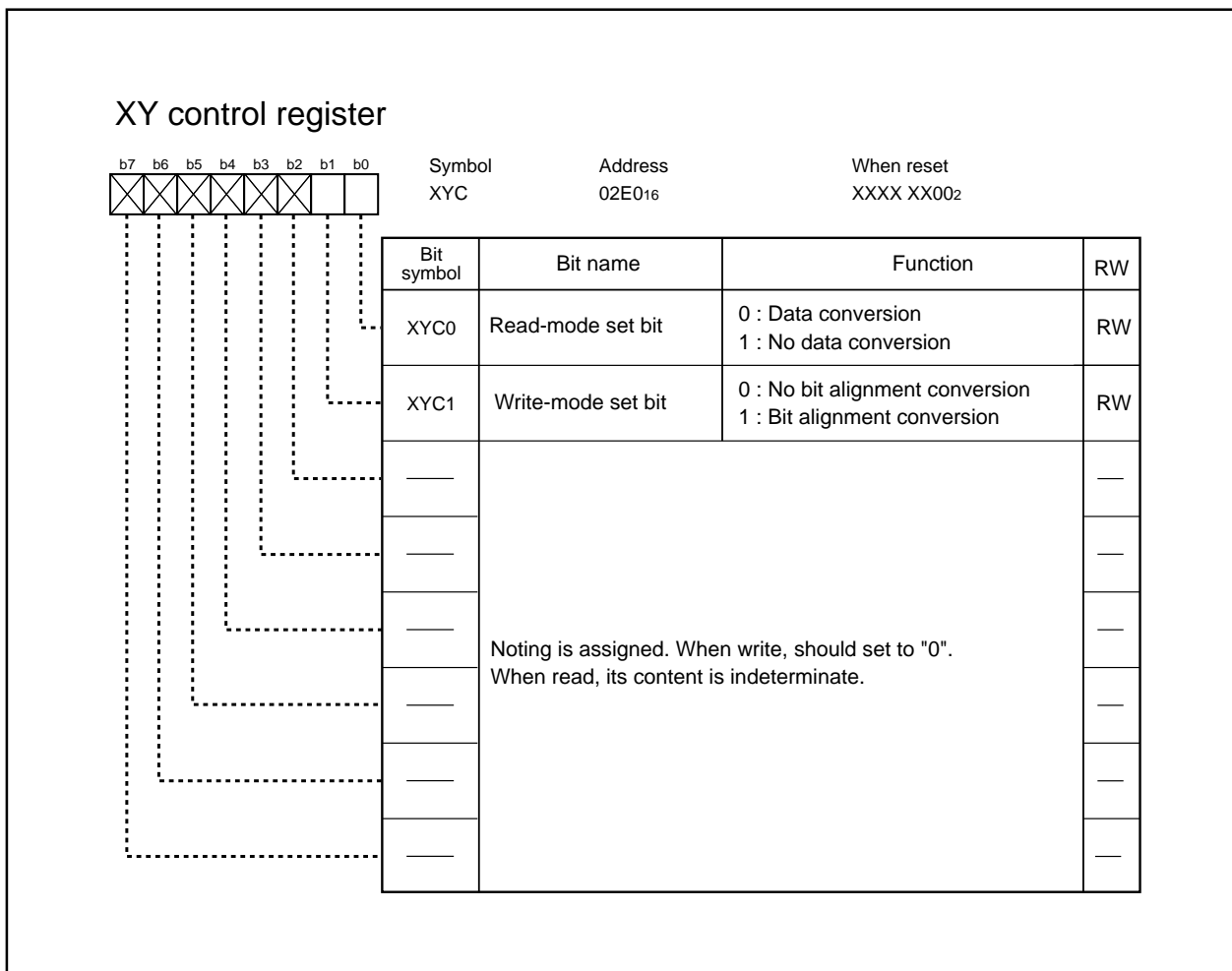


Figure 1.26.1. XYC Register

X-Y Conversion

The XYC0 bit in the XYC register determines how to read the YjR register.

When reading the YjR register with setting the XYC0 bit to "0" (data exchange), bit j in the X0R to X15R registers can be read simultaneously.

For example, when reading the Y0R register, bit 0 in the X0R register can be read by bit 0 in the Y0R register, bit 0 in the X1R register by bit 1 in the Y0R register, ..., bit 0 in the X14R register by bit 14 in the Y0R register and bit 0 in the X15R register by bit 15 in the Y0R register.

Figure 1.26.2 shows a conversion table when setting the XYC0 bit to "0". Figure 1.26.3 shows an example of the X-Y conversion.

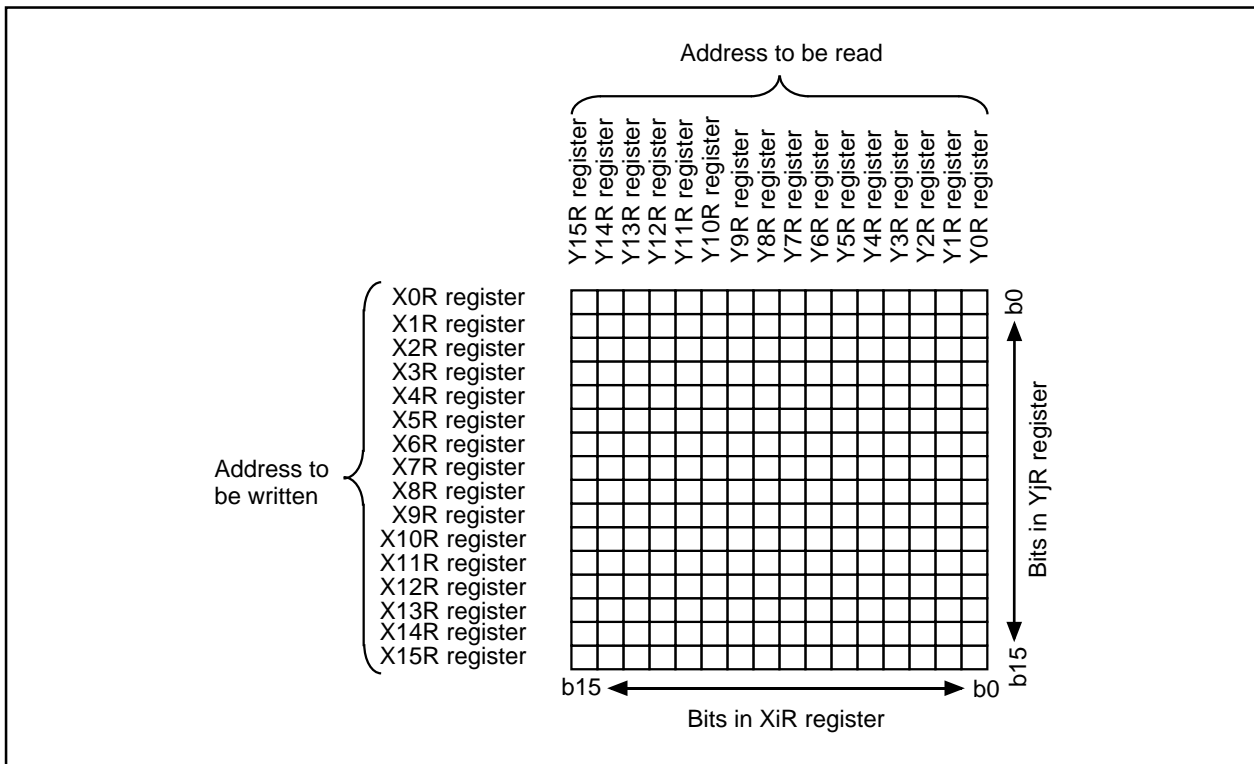


Figure 1.26.2. Conversion Table when Setting the XYC0 Bit to "0"

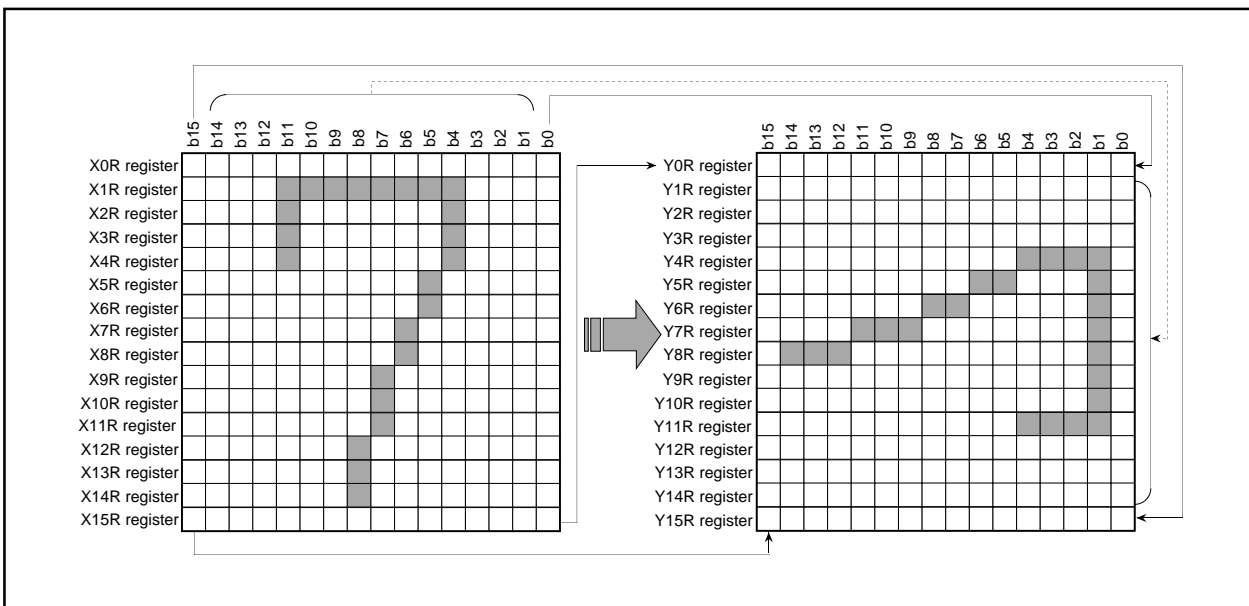


Figure 1.26.3. X-Y Conversion

X-Y Conversion

When reading the YjR register with setting the XYC0 bit in the XYC register to "1", a value written to the XiR register can be read directly. Figure 1.26.4 shows a conversion table when setting the XYC0 bit to "1."

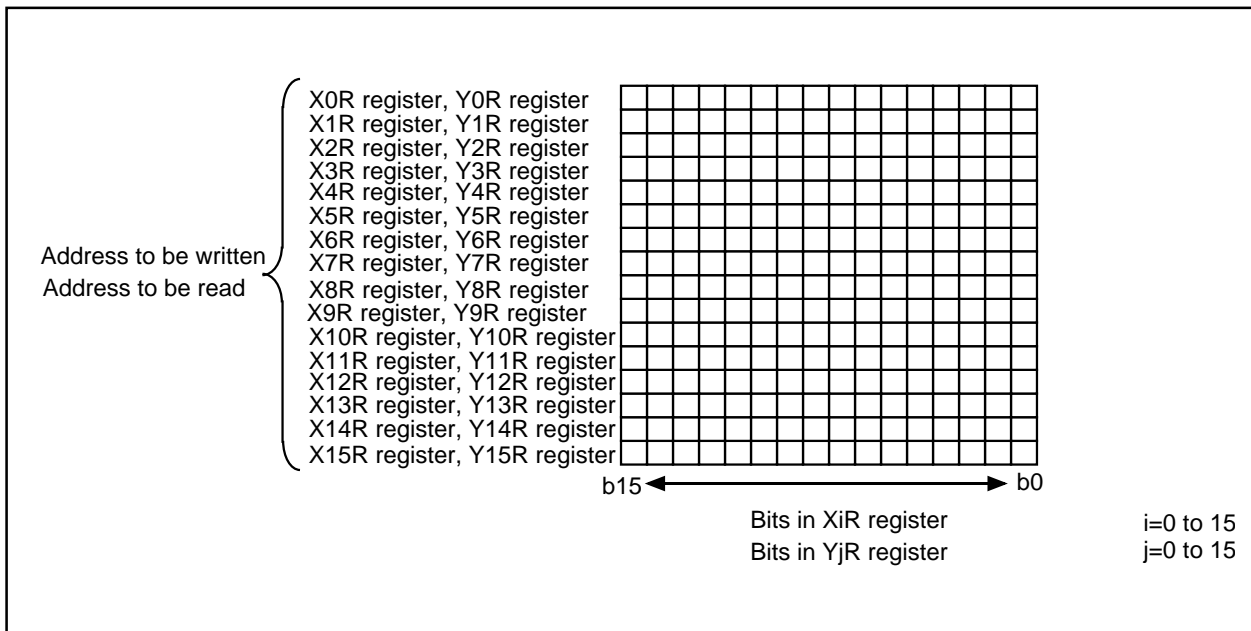


Figure 1.26.4. Conversion Table when Setting the XYC0 Bit to "1"

The XYC1 bit in the XYC register selects a bit alignment of value in the XiR register.

When writing to the Xi register with setting the XYC1 bit to "0" (no bit alignment conversion), a bit alignment is written without the conversion to inverse a bit alignment. When writing to the XiR register with setting the XYC1 bit to "1" (bit sequence replaced), a bit alignment is inverted to write.

Figure 1.26.5 shows a conversion table when setting the XYC1 bit to "1".

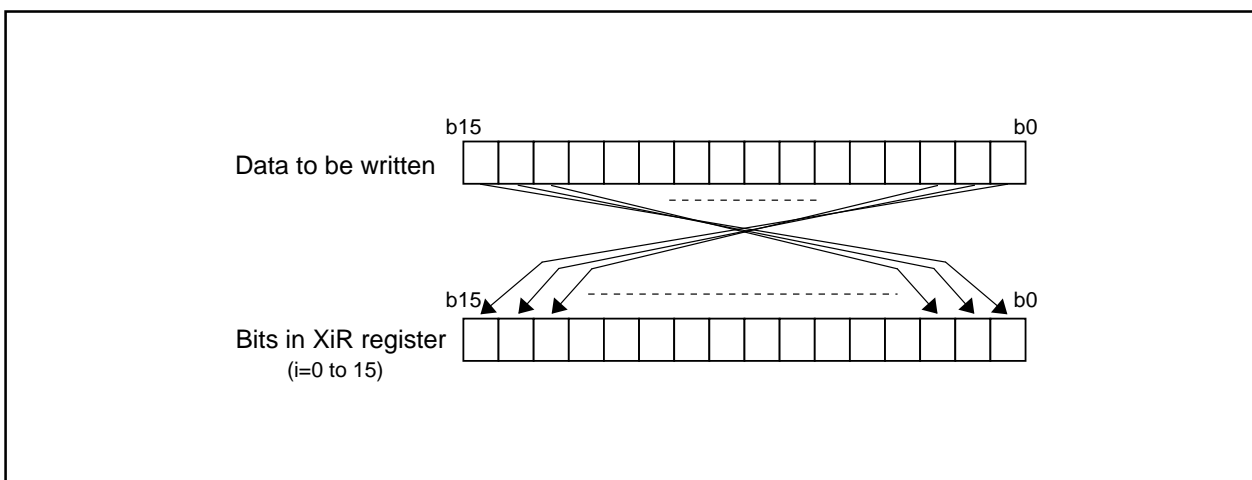


Figure 1.26.5. Conversion Table when Setting the XYC1 Bit to "1"

## DRAMC

## DRAMC

DRAM controller (DRAMC) controls DRAM space between 512K bytes and 8M bytes. Table 1.27.1 lists specifications of the DRAMC.

**Table 1.27.1. DRAMC Specifications**

Item	Specification
DRAM space	512KB, 1MB, 2MB, 4MB, 8MB
Bus control	2CAS/1W
Refresh	$\overline{\text{CAS}}$ before $\overline{\text{RAS}}$ refresh, Self refresh
Function mode	EDO, fast page mode
Wait	1 wait, 2 waits

Table 1.27.2 shows pins associated with DRAMC. Signals listed in Table 1.27.2 output when setting the AR2 to AR0 bits in the DRAMCONT register for the DRAM space and accessing DRAM. See Table 1.7.9 about  $\overline{\text{RAS}}$ ,  $\overline{\text{CASL}}$ ,  $\overline{\text{CASH}}$  and  $\overline{\text{DW}}$  signal operations. Figure 1.27.1 shows the DRAMCONT register and REFCNT register.

**Table 1.27.2. DRAMC-associated Pins**

Port	Bus for device access except DRAM <sup>1</sup>	Bus for DRAM access
P0	D0 to D7	D0 to D7
P1	D8 to D15	D8 to D15 <sup>2</sup>
P3	A8 to D15	MA0 to MA7
P40 to P44	A16 to D0	MA8 to MA12
P50	$\overline{\text{WRL}}$ / $\overline{\text{WR}}$	$\overline{\text{CASL}}$
P51	$\overline{\text{WRH}}$ / $\overline{\text{BHE}}$	$\overline{\text{CASH}}$
P52	$\overline{\text{RD}}$	$\overline{\text{DW}}$
P56	ALE	RAS

Notes:

1. This is an example of a separate bus and 16-bit data bus.
2. This bus is available when the DS2 bit in the DS register is set to "1" (16-bit data bus) and the PM02 bit in the PM register is also set to "1" ( $\overline{\text{RD}}$ / $\overline{\text{WRL}}$ / $\overline{\text{WRH}}$  in R/W mode).

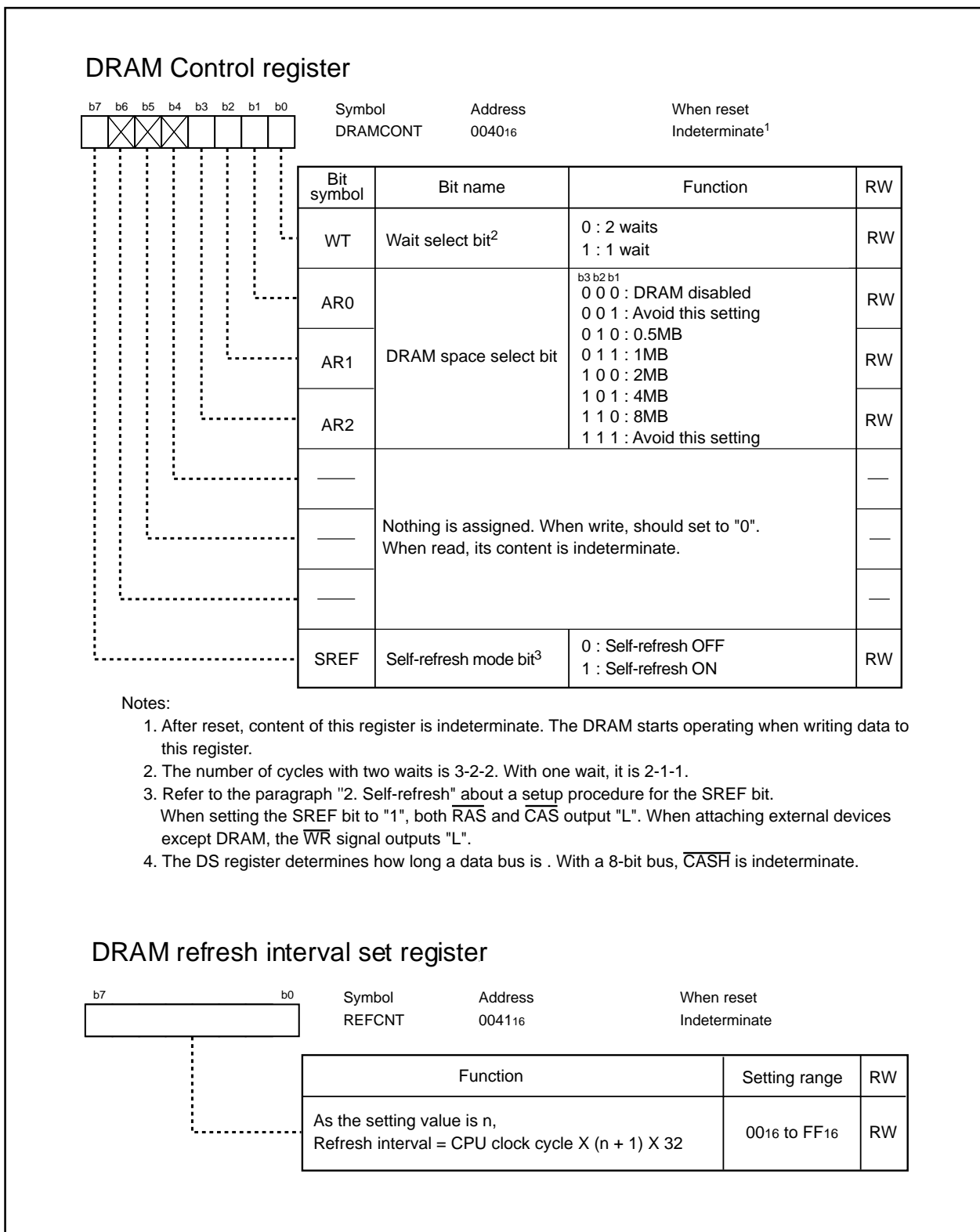


Figure 1.27.1 DRAMCONT Register and REFCNT Register

## DRAMC

DRAMC is unavailable when the PM11 to PM10 bits in the PM1 register is set to "112" (mode 3). The PM11 to PM10 bits should be set to "002," "012" or "102" (mode 0 to 2). For a 16-bit DRAM data bus, the PM02 bit in the PM0 register should be set to "1" ( $\overline{RD}/\overline{WRH}/\overline{WRL}$ ).

A wait time between DRAM power-on and memory operation and a necessary processing for feigned cycle for refresh vary depending on externally attached DRAM specifications.

## DRAMC Multiplexed Address Output

The DRAMC outputs signals, which is multiplexed row addresses and column addresses, to address bus A8 to A20. Figure 1.27.2 shows an output format for multiplexed addresses.

## Refresh

### 1. Refresh

Refresh method is  $\overline{CAS}$ -before- $\overline{RAS}$  refresh. The REFCNT register determines a refresh interval. Refresh signal is not output in a HOLD state.

A setting value of the REFCNT register is obtained from:

$$\text{Value of the REFCNT register (00}_{16}\text{ to FF}_{16}) = \text{refresh interval time} / (\text{CPU clock frequency} \times 32) - 1$$

### 2. Self-Refresh

The refresh signal stated above stops while the CPU stops in stop mode, etc. The DRAM self-refresh function can be activated by setting the self-refresh before the CPU is stopped. Setting and cancellation procedures for the self-refresh are as follows.

(1) setting for the self-refresh (1 wait, 4M bytes)

```

...
mov.b #00000001b,DRAMCONT ;Set the AR2 to AR0 bits to "0002" (DRAM disabled)
mov.b #10001011b,DRAMCONT ;Set the AR2 to AR0 bits again as soon as setting
                             the SREF bit to "1" (self-refresh on)
nop ;Execute the nop instruction twice
nop ;
...

```

(2) cancellation for the self-refresh (1 wait, 4M bytes)

```

...
mov.b #00000001b,DRAMCONT ;Set the AR2 to 0 bits to "0002" (self-refresh cancel-
                             lation) as soon as setting the SREF bit "0" (DRAM
                             disabled)
mov.b #00001011b,DRAMCONT ;Set the AR2 to AR0 bits again
mov.b 400h, 400h ;Disable to access just after cancellation. It is a feign
                  read in this example.
...

```

Both  $\overline{RAS}$  and  $\overline{CAS}$  are set to "L" with the self-refresh. When other devices except the DRAM are attached and the  $\overline{WR}$  signal is set to "L", any processings like setting the  $\overline{CS}$  to "H" should be done.

Figures 1.27.3 to 1.27.5 show a bus timing during a DRAM access.

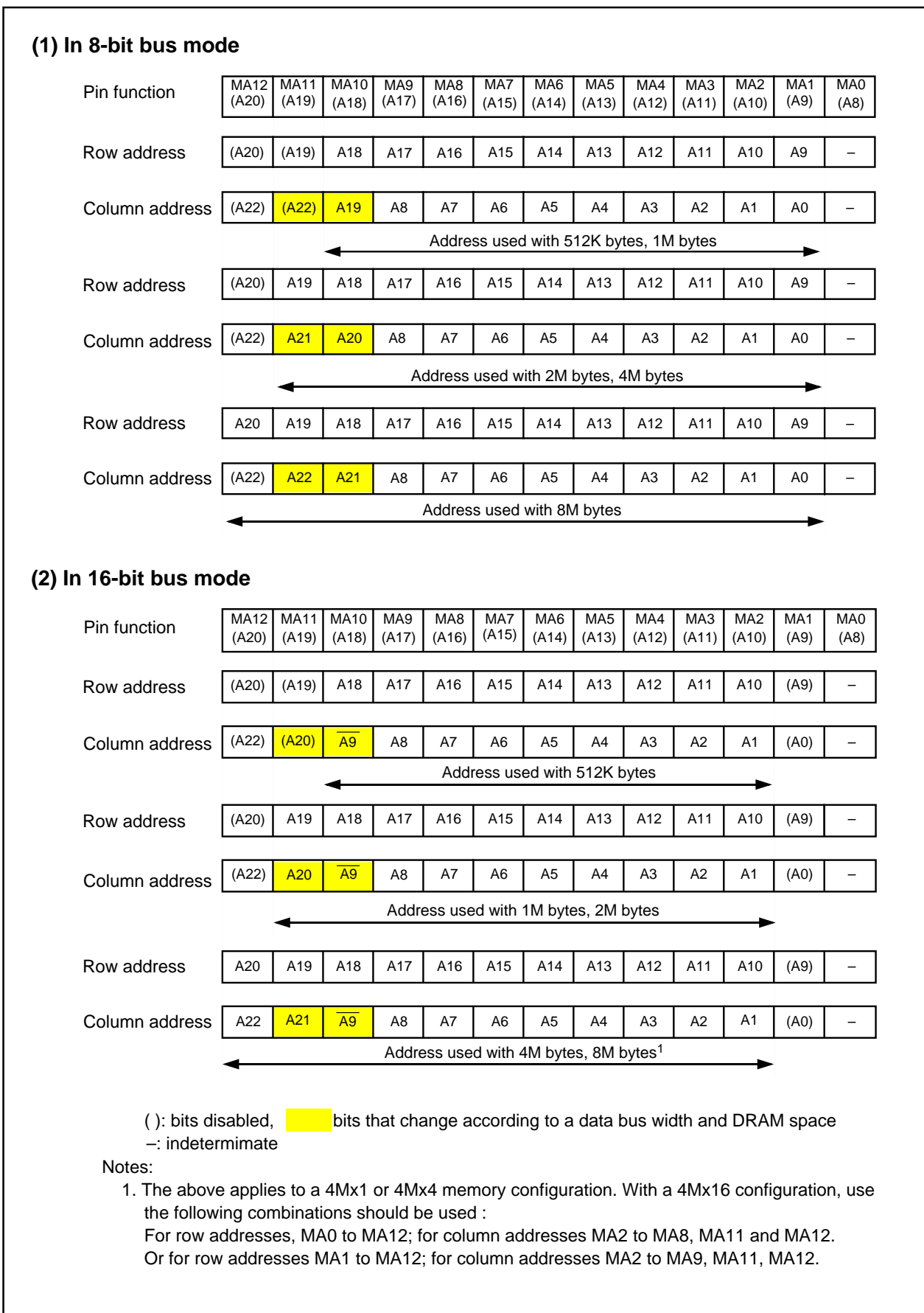
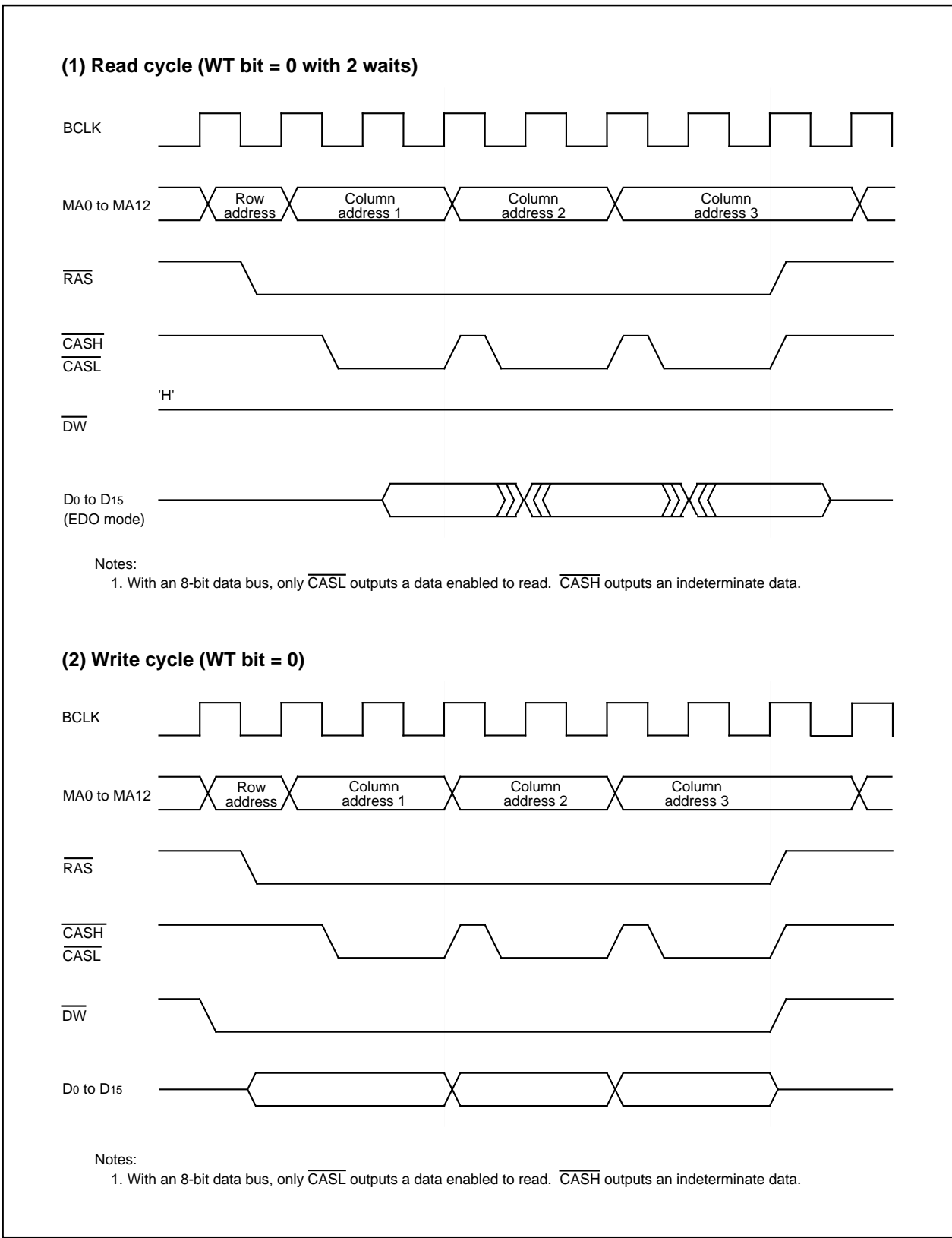


Figure 1.27.2. Address Multiplexed Output Pattern

**DRAMC**



**Figure 1.27.3. Bus Timing during DRAM Access (1)**

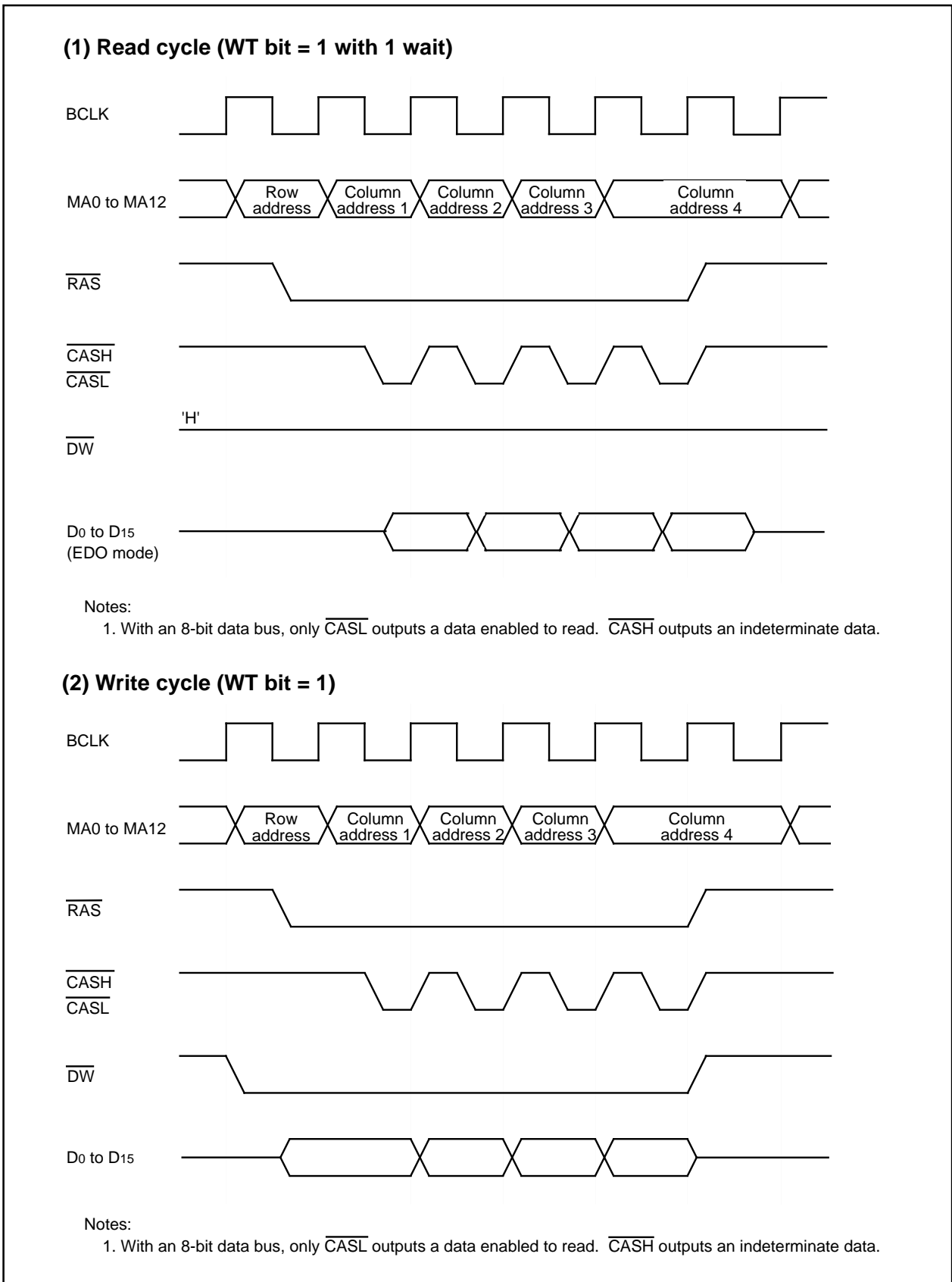
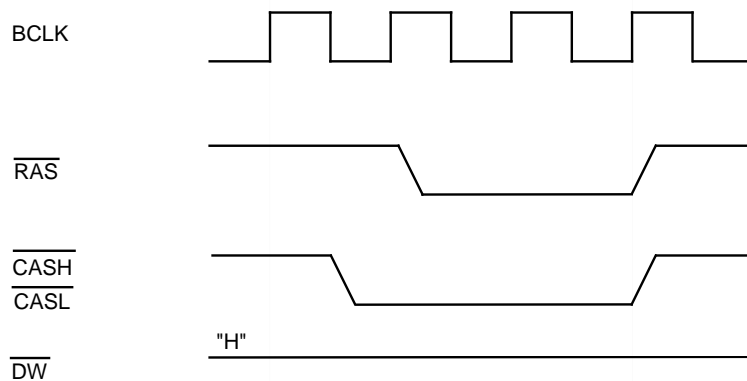


Figure 1.27.4. Bus Timing during DRAM Access (2)

DRAMC

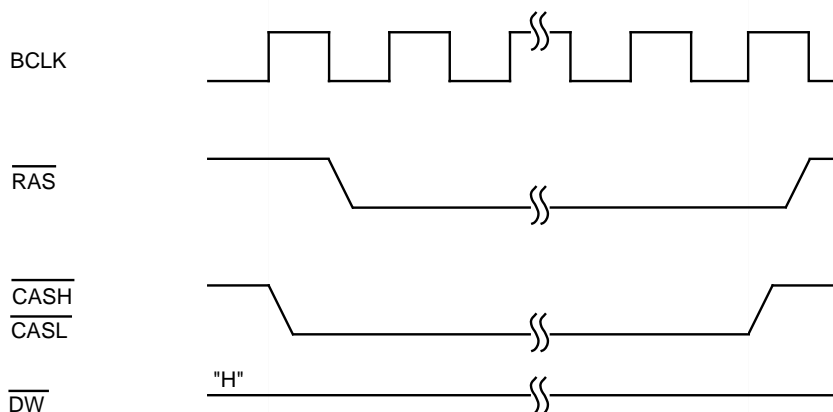
(1)  $\overline{\text{CAS}}$ -before- $\overline{\text{RAS}}$  refresh cycle



Notes:

1. With an 8-bit data bus, only  $\overline{\text{CASL}}$  outputs a data enabled to read.  $\overline{\text{CASH}}$  outputs an indeterminate data.

(1) Self-Refresh cycle



Notes:

1. With an 8-bit data bus, only  $\overline{\text{CASL}}$  outputs a data enabled to read.  $\overline{\text{CASH}}$  outputs an indeterminate data.

Figure 1.27.5. Bus Timing during DRAM Access (3)

## Programmable I/O Ports

87 programmable I/O ports from P0 and to P10 (excluding P85) are available in the 100-pin package and 123 programmable I/O ports from P0 and to P15 (excluding P85) are in the 144-pin package. Input and output for each port can be set in each pin with the direction register. Whether each block of 4 ports pulls up or not can be set. P85 is for input only and no pull-up for this port is allowed. The P8\_5 bit in the P8 register indicates an  $\overline{\text{NMI}}$  input level since P85 shares pins with  $\overline{\text{NMI}}$ .

Figures 1.28.1 to 1.28.4 show the programmable I/O port configurations.

Each pin functions as a programmable I/O port, an input/output for built-in peripheral device or a bus control pin.

To use the pins as inputs for built-in peripheral devices, refer to the description of each function for the setting. Refer to the section "Bus" when used as a bus control pin.

The following described registers are associated with the programmable I/O ports.

### Port Pi Direction Register (PDi Register, i=0 to 15)

Figure 1.28.5 shows the PDi register.

The PDi register selects whether the programmable I/O port is used for input or for output. Each bit in the PDi register corresponds one for one to each port.

In memory expansion and microprocessor mode, the bus control pins (A0 to A22,  $\overline{\text{A23}}$ , D0 to D15, MA0 to MA12,  $\overline{\text{CS0}}$  to  $\overline{\text{CS3}}$ ,  $\overline{\text{WRL/WR/CASL}}$ ,  $\overline{\text{WRH/BHE/CASH}}$ ,  $\overline{\text{RD/DW}}$ , BCLK/ALE/CLKOUT,  $\overline{\text{HLDA/ALE}}$ ,  $\overline{\text{HOLD}}$ , ALE/ $\overline{\text{RAS}}$ , and  $\overline{\text{RDY}}$ ) in the PDi register cannot be modified. No bit in the direction register for P85 is provided.

### Port Pi Register (Pi Register, i=0 to 15)

Figure 1.28.6 shows the Pi register.

The Pi register writes and reads data for input and output to and from an external device. The Pi register consists of a port latch to hold output data and a circuit to read pin states. Each bit in the Pi register corresponds one for one to each port.

In memory expansion and microprocessor mode, the bus control pins (A0 to A22,  $\overline{\text{A23}}$ , D0 to D15, MA0 to MA12,  $\overline{\text{CS0}}$  to  $\overline{\text{CS3}}$ ,  $\overline{\text{WRL/WR/CASL}}$ ,  $\overline{\text{WRH/BHE/CASH}}$ ,  $\overline{\text{RD/DW}}$ , BCLK/ALE/CLKOUT,  $\overline{\text{HLDA/ALE}}$ ,  $\overline{\text{HOLD}}$ , ALE/ $\overline{\text{RAS}}$ , and  $\overline{\text{RDY}}$ ) in the Pi register cannot be modified.

### Function Select Register A0 to A9 (PS0 to PS9 Registers)

Figures 1.28.7 to 1.28.11 show the PS0 to PS9 registers.

The P*Si* register (i=0 to 9) selects either I/O port or peripheral function output when I/O port shares pins with peripheral function output (excluding DA0 and DA1.)

Tables 1.28.3 to 1.28.12 list peripheral functions for port selected by the function select register. When multiple peripheral function outputs are assigned to a pin, the PSL0 to PSL3 registers should be set to select which function is used.

### Function Select Register B0 to B3 (PSL 0 to PSL3 Registers)

Figures 1.28.12 and 1.28.13 show the PSL0 to PSL3 registers.

When multiple peripheral function outputs are assigned to a pin, the PSL0 to PSL2 registers and the PSL3\_1 to PSL3\_2 and PSL3\_7 bits in the PSL3 register select which peripheral function output is used.

The PSL1 register and PSC register select which peripheral function is output from port P7.

The PSL0 to PSL3 registers are available when bits in the corresponding PSi register (i=0 to 9) are set for the peripheral functions.

Refer to the paragraph "Analog input and other peripheral function input" about the PSL3\_3 to PSL3\_6 bits in the PSL3 register.

### Function Select Register C (PSC Register)

Figure 1.28.14 shows the PSC register.

When three peripheral function outputs are assigned to a pin, the PSC\_0 to 4 and PSC\_6 bits select which peripheral function output is used.

The PSC\_0 to PSC\_4 and PSC\_6 bits are available when bits in the PS1 register for the corresponding pin are set to "1" (peripheral function selected) and bits in the PSL1 register are set to "0" (PSC register enabled).

Refer to the paragraph "Analog input and other peripheral function input" about the PSC\_7 bit.

### Pull-up Control Register 0 to 4 (PUR0 to PUR4 Registers)

Figures 1.28.15 to 1.28.16 show the PUR0 to PUR4 registers.

The PUR0 to PUR4 registers select whether each block of 4 pins pulls up or not. A port, with bits in the PUR0 to PUR4 registers to be set to "1" (pull-up) and the direction register to be set to "0" (input mode), is applied a pull-up.

Bits in the PUR0 to PUR1 registers in P0 to P5, running as a bus, should be set to "0" (no pull-up) in memory expansion and microprocessor mode. When using P0, P1, P40 to P43 as input ports in memory expansion and microprocessor mode, these ports can be applied a pull-up.

### Port Control Register (PCR Register)

Figure 1.28.17 shows the PCR register.

The PCR register selects either a CMOS port or an N-channel open drain as a port P1 output format. When setting the PCR0 bits to "1", port P1 cannot output in a perfect open drain since the P-channel in the CMOS port always remains turned off. A absolute maximum rating of the input voltage falls within the range from - 0.3 V to Vcc + 0.3 V.

When using port P1 as a data bus in memory expansion and microprocessor mode, the PCR0 bit should be set to "0". When using port P1 as a port in memory expansion and microprocessor mode, in the PCR0 bit can determines an output format.

### Input Function Select Register (IPS Register)

Figure 1.28.18 shows the IPS register.

The IPS0 to IPS1 and IPS3 to IPS6 bits select which pin is assigned for the intelligent I/O and CAN input functions.

Refer to the paragraph "Analog input and other peripheral function input" about the IPS2 bit.

### Analog Input and Other Peripheral Function Input

The PSL3\_3 to PSL3\_6 bits in the PSL3 register, the PSC\_7 bit in the PSC register and the IPS2 bit in the ISP register separates analog I/O from other peripheral functions. By setting the corresponding bits to "1" (analog I/O) to use the analog I/O (DA0, DA1, ANEX0, ANEX1, AN4 to AN7, AN150 to AN157), it prevents intermediate potential from being impressed to other peripheral functions. Impressed intermediate potential may cause to increase power consumption.

The corresponding bits should be set to "0" (except analog I/O) when not using analog I/O. Peripheral function input except the analog I/O is available when this bit is set to "0" and it is indeterminate when the bit is set to "1". When setting the PSC\_7 bit to "1", key input interrupt request remains unchanged regardless of  $\overline{KI0}$  to  $\overline{KI3}$  pin input levels change.

Programmable I/O Port

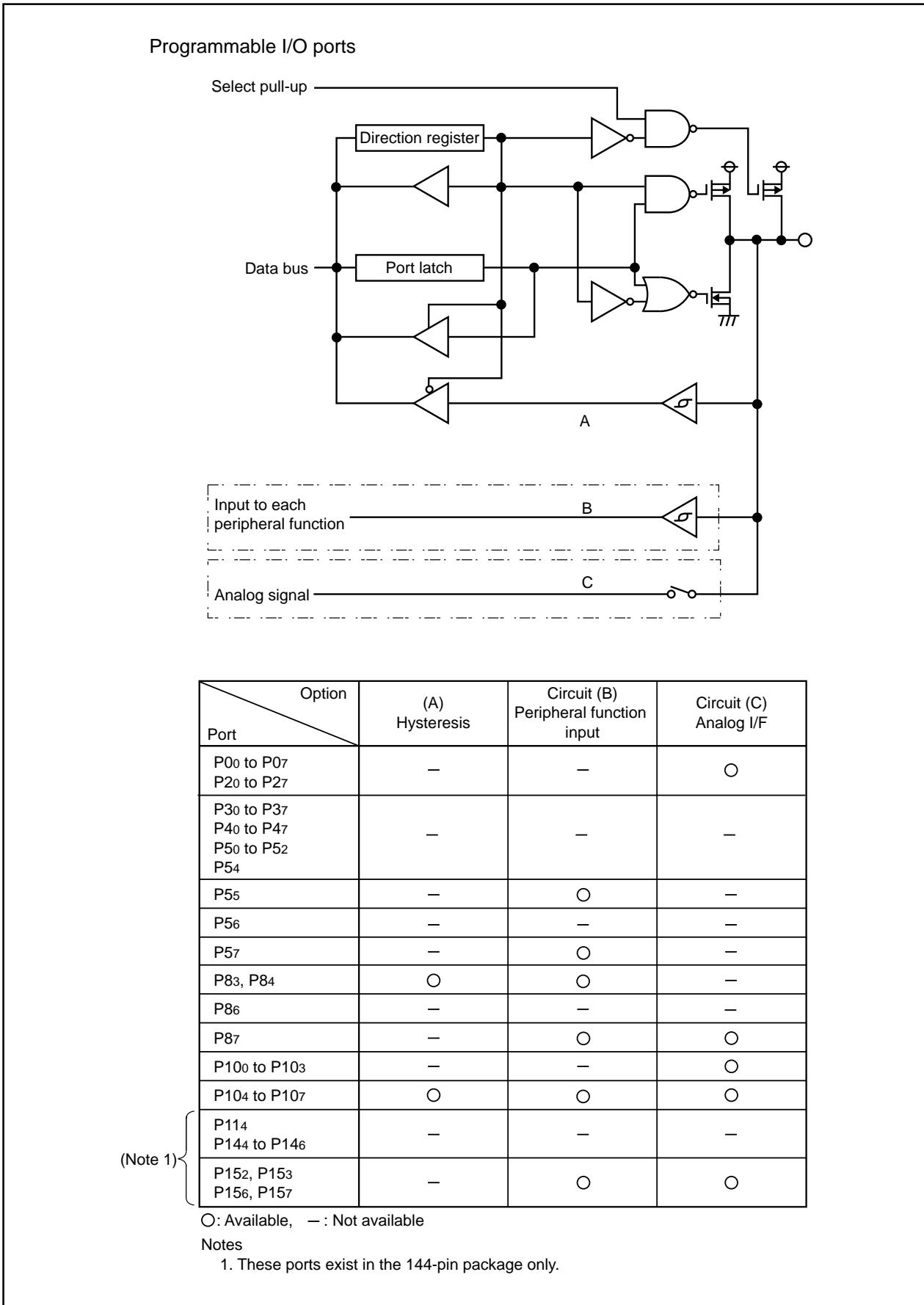
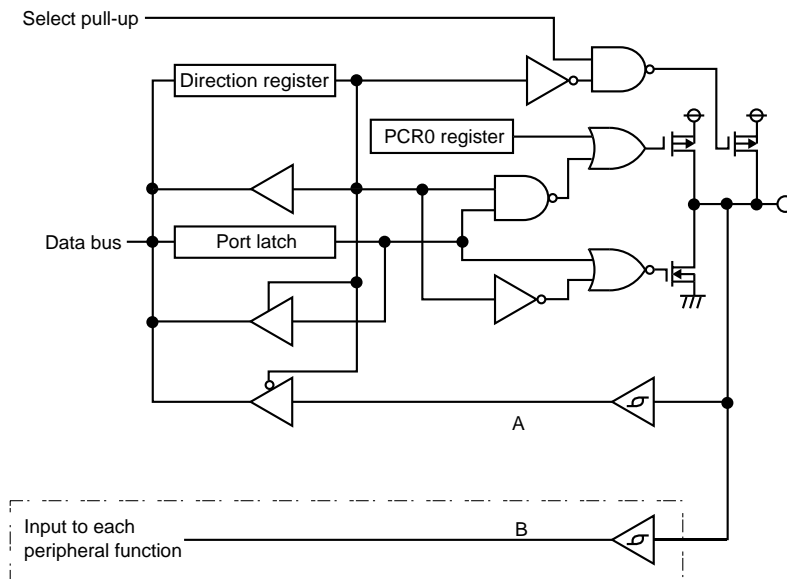


Figure 1.28.1. Programmable I/O Ports (1)

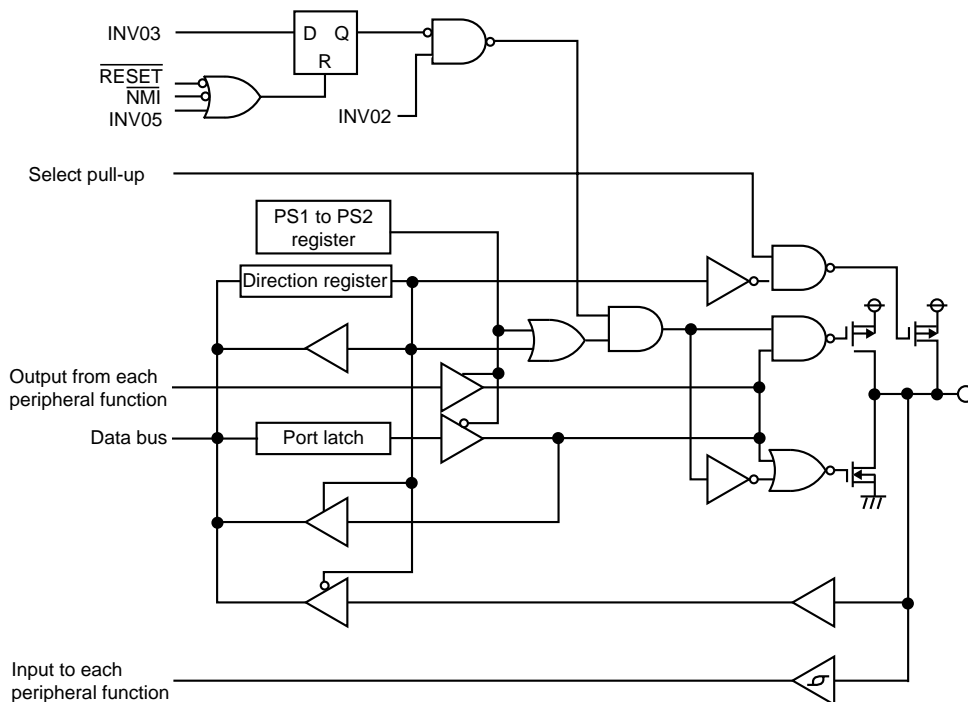
Programmable I/O Port

Programmable I/O ports with port control register



Port \ Option	(A) Hysteresis	Circuit (B) Peripheral function input
P10 to P14	—	—
P15 to P17	○	○

○ : Available, — : Not available

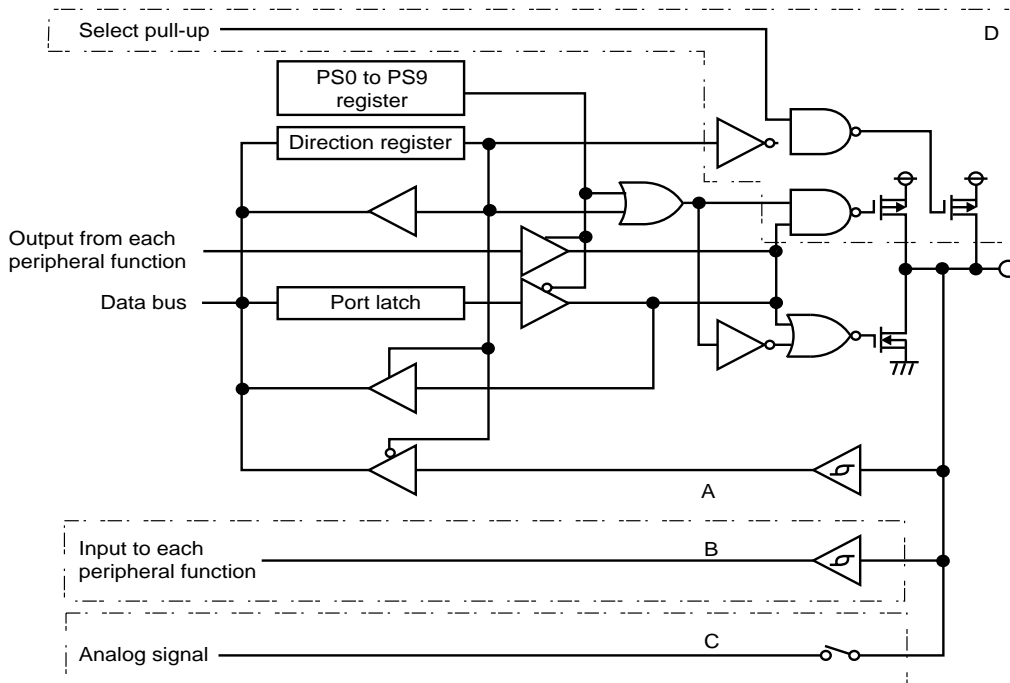


Port : P72, P73, P74, P75, P80, P81

Figure 1.28.2. Programmable I/O Ports (2)

Programmable I/O Port

Programmable I/O ports with function select register



Option	(A) Hysteresis	Circuit (B) Peripheral function input	Circuit (C) Analog I/F	Circuit (D)
Port				
P53	—	—	—	○
P60 to P67	—	○	—	○
P70, P71 (Note 1)	—	○	—	—
P76, P77	—	○	—	○
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 2)

○ : Available, — : Not available

Notes:

1. P70 and P71 output in N-channel open drain.
2. These ports are provided in the 144-pin package only.

Figure 1.28.3. Programmable I/O Ports (3)

Programmable I/O Port

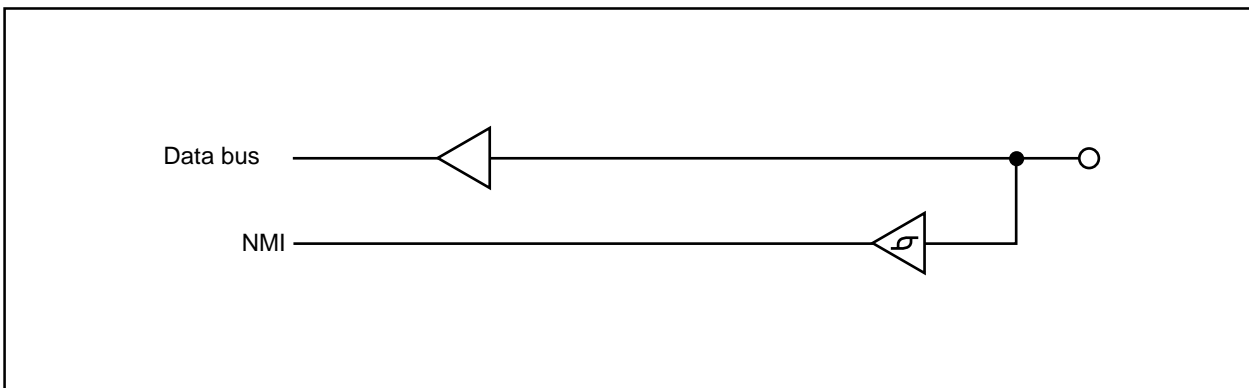


Figure 1.28.4. Programmable I/O Ports (4)

**Port Pi direction register (i=0 to 15)**

Symbol	Address	When reset
PD0 to PD5	03E2 <sub>16</sub> , 03E3 <sub>16</sub> , 03E6 <sub>16</sub> , 03E7 <sub>16</sub> , 03EA <sub>16</sub> , 03EB <sub>16</sub> <sup>2</sup>	00 <sub>16</sub>
PD6 to PD10	03C2 <sub>16</sub> , 03C3 <sub>16</sub> , 03C6 <sub>16</sub> <sup>4</sup> , 03C7 <sub>16</sub> <sup>1</sup> , 03CA <sub>16</sub> ,	00 <sub>16</sub>
PD11 to PD15	03CB <sub>16</sub> <sup>4</sup> , 03CE, 03CF <sub>16</sub> , 03D2 <sub>16</sub> <sup>4</sup> , 03D3 <sub>16</sub> <sup>3</sup>	00 <sub>16</sub>

Bit Symbol	Bit name	Function	RW
PDi_0	Port Pi0 direction register	0 : Input mode (Functions as input port) 1 : Output mode (Functions as output port)	RW
PDi_1	Port Pi1 direction register	0 : Input mode (Functions as input port) 1 : Output mode (Functions as output port)	RW
PDi_2	Port Pi2 direction register	0 : Input mode (Functions as input port) 1 : Output mode (Functions as output port)	RW
PDi_3	Port Pi3 direction register	0 : Input mode (Functions as input port) 1 : Output mode (Functions as output port)	RW
PDi_4	Port Pi4 direction register	0 : Input mode (Functions as input port) 1 : Output mode (Functions as output port)	RW
PDi_5	Port Pi5 direction register	0 : Input mode (Functions as input port) 1 : Output mode (Functions as output port)	RW
PDi_6	Port Pi6 direction register	0 : Input mode (Functions as input port) 1 : Output mode (Functions as output port)	RW
PDi_7	Port Pi7 direction register	0 : Input mode (Functions as input port) 1 : Output mode (Functions as output port)	RW

**Notes:**

- The PD9 and PS3 registers should be set with subsequent to the instruction that the PRC2 bit in the PRCR register is set to "1" (write enable). Avoid generating an interrupt and operating a DMA transfer between the instruction to set the PRC2 bit to "1" and the instruction to write to the PD9 and PS3 registers.
- In memory expansion and microprocessor mode, the bus control pins ( $A_0$  to  $A_{22}$ ,  $\bar{A}_{23}$ ,  $D_0$  to  $D_{15}$ ,  $MA_0$  to  $MA_{12}$ ,  $\overline{CS}_0$  to  $\overline{CS}_3$ ,  $\overline{WRL}/\overline{WR}/\overline{CASL}$ ,  $\overline{WRH}/\overline{BHE}/\overline{CASH}$ ,  $\overline{RD}/\overline{DW}$ ,  $\overline{BCLK}/\overline{ALE}/\overline{CLKOUT}$ ,  $\overline{HLD}/\overline{ALE}$ ,  $\overline{HOLD}$ ,  $\overline{ALE}/\overline{RAS}$  and  $\overline{RDY}$ ) of the direction register cannot be changed.
- The PD11 to PD15 registers should be set to "00<sub>16</sub>" in the 100-pin package.
- Nothing is assigned in the PD8\_5 bit in the PD8 register, the PD11\_5 to PD11\_7 in the PD11 register and the P14\_7 bit in the PD14 register. When write, these bits should be set to "0". When read, its content is indeterminate.

Figure 1.28.5. PD0 to PD15 Registers

Programmable I/O Port

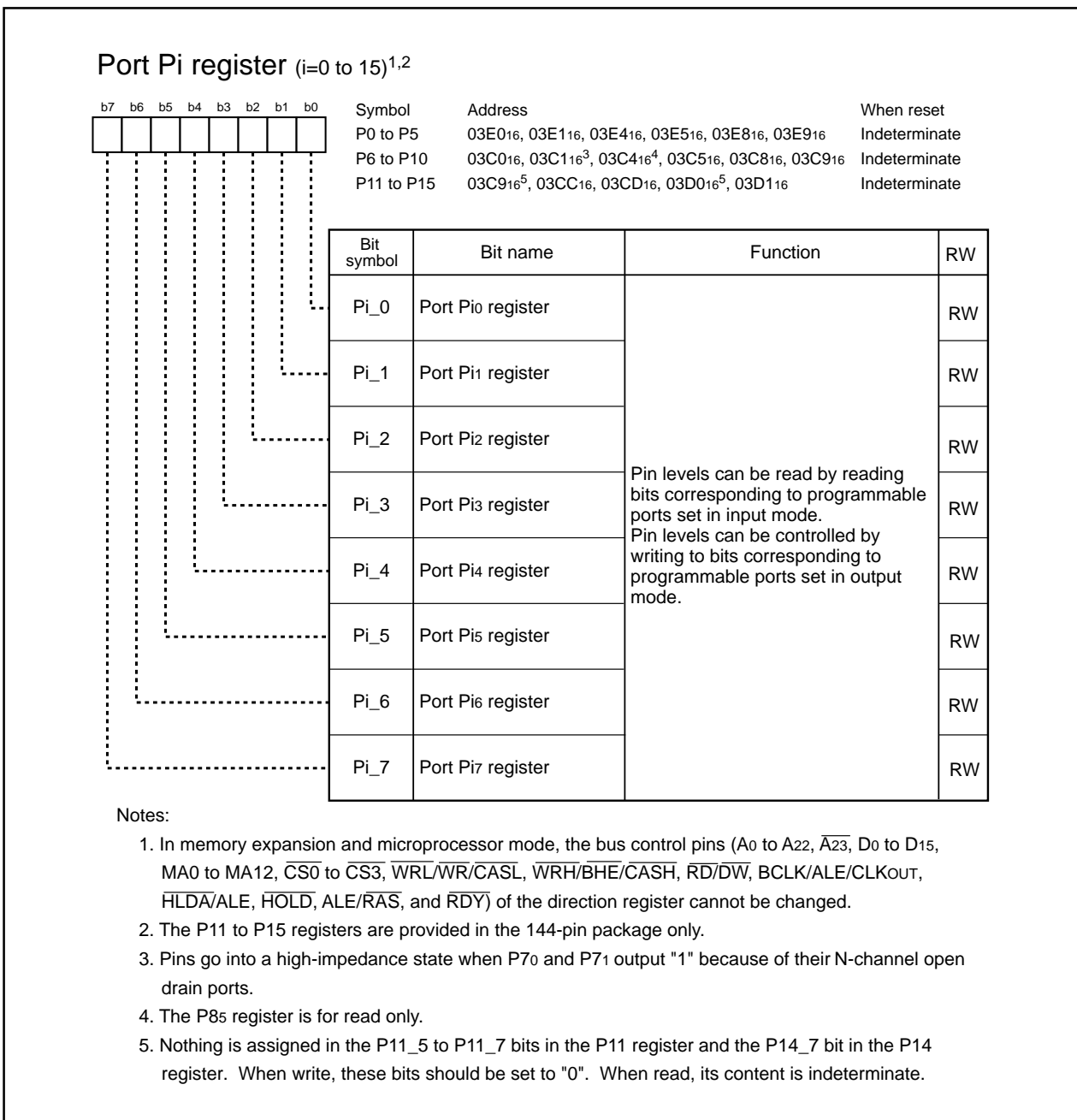


Figure 1.28.6. P0 to P15 Registers

Programmable I/O Port

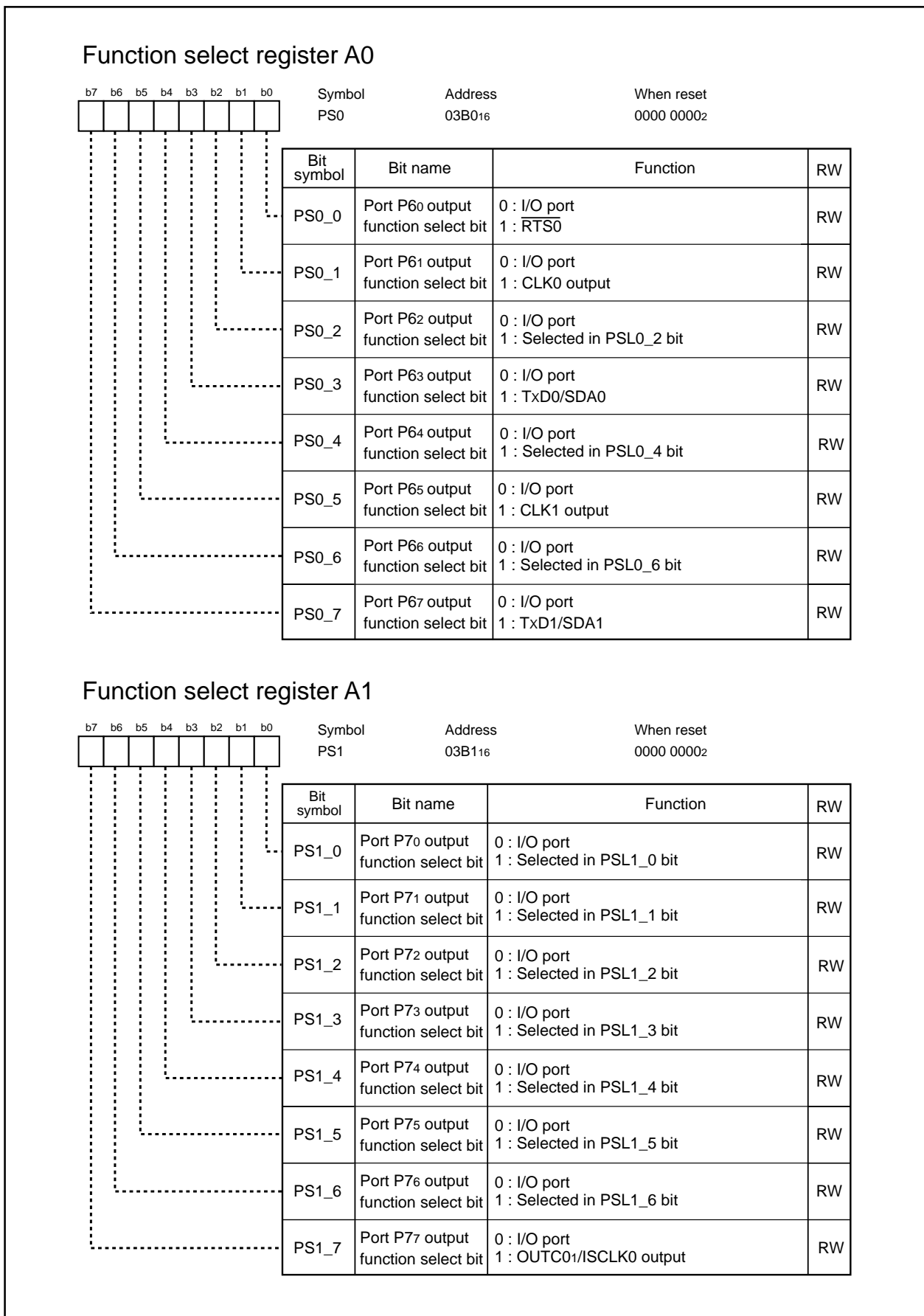


Figure 1.28.7. PS0 Register and PS1 Register

Programmable I/O Port

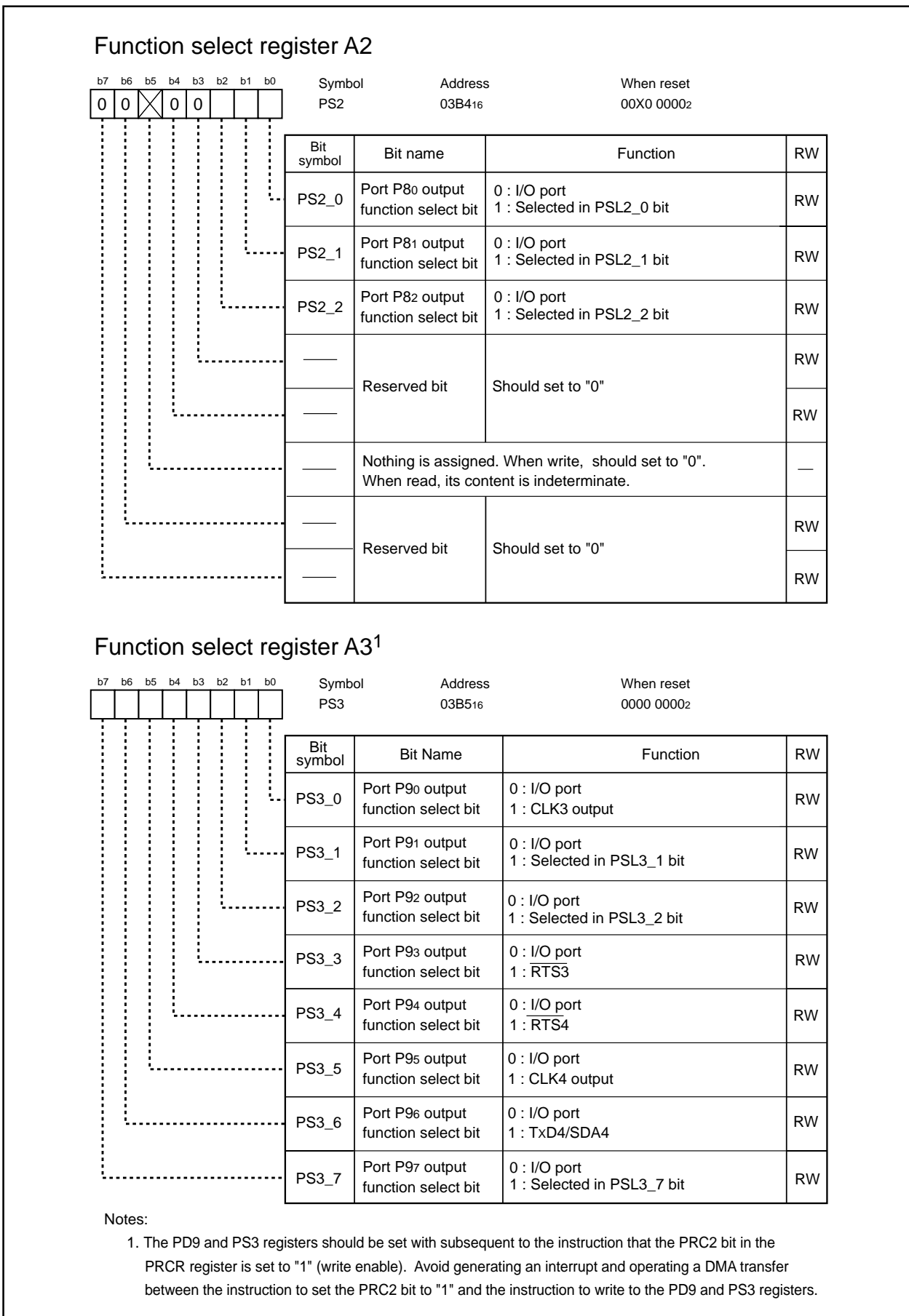


Figure 1.28.8. PS2 Register and PS3 Register

Programmable I/O Port

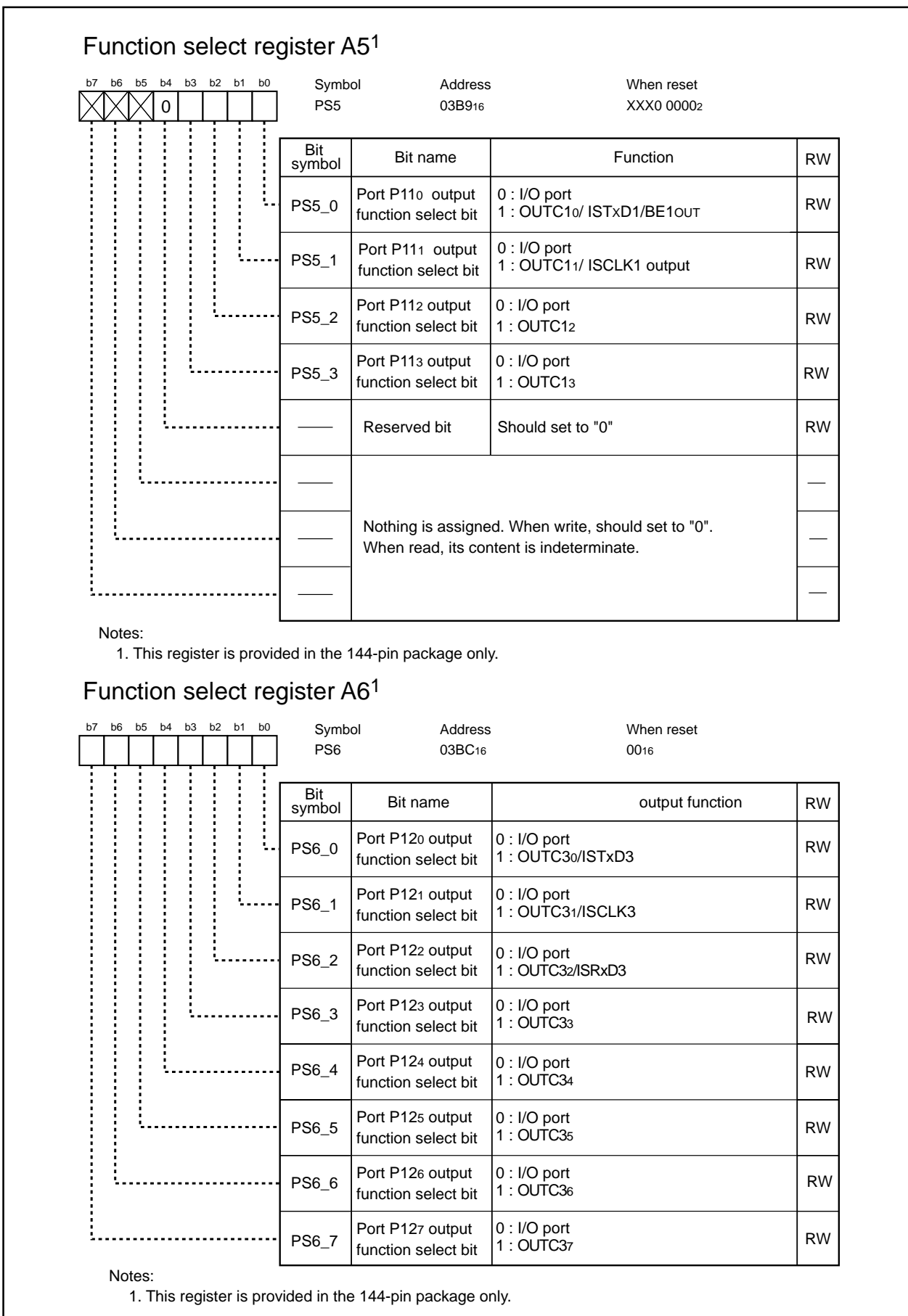
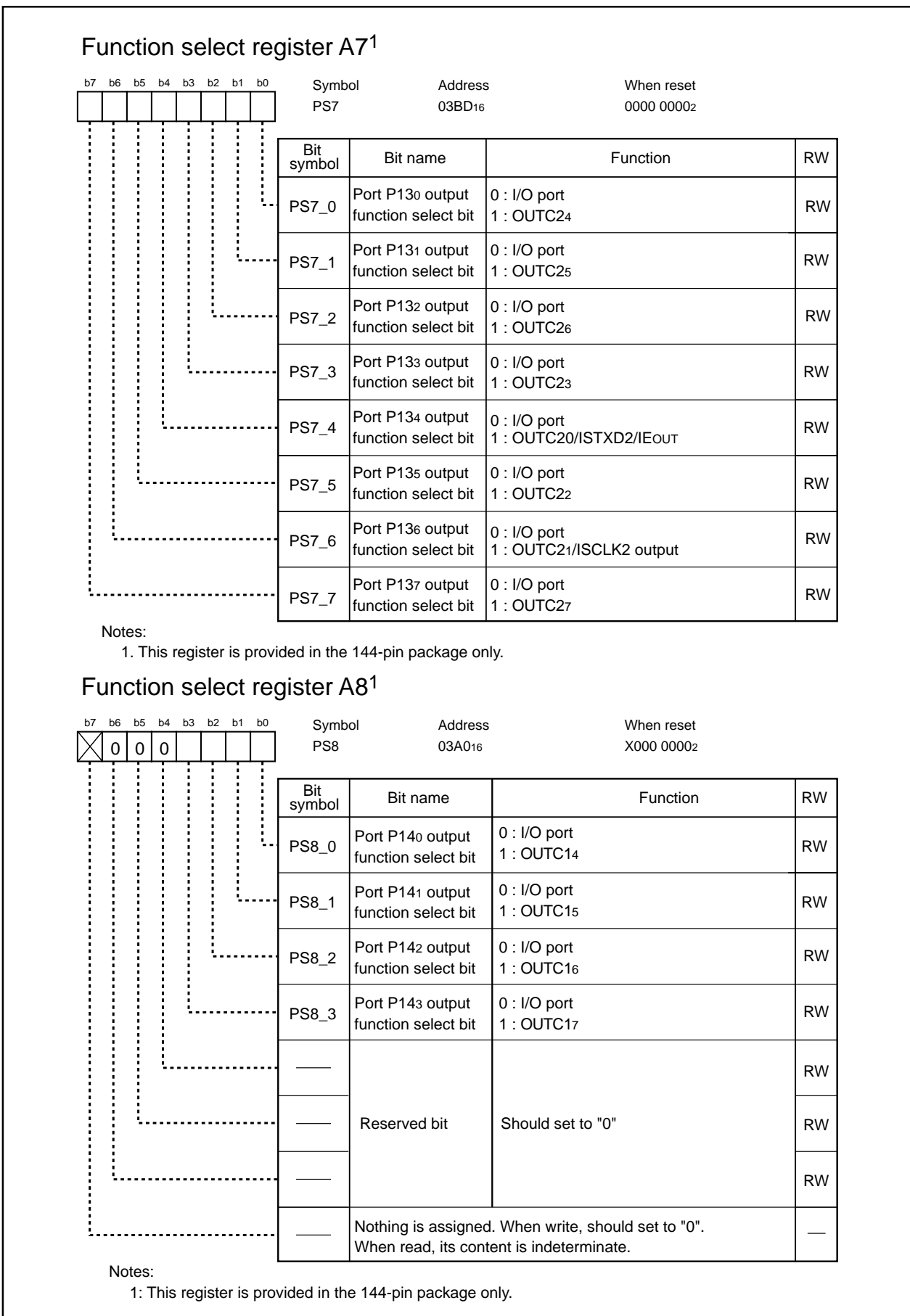


Figure 1.28.9. PS5 Register and PS6 Register

Programmable I/O Port



Programmable I/O Port

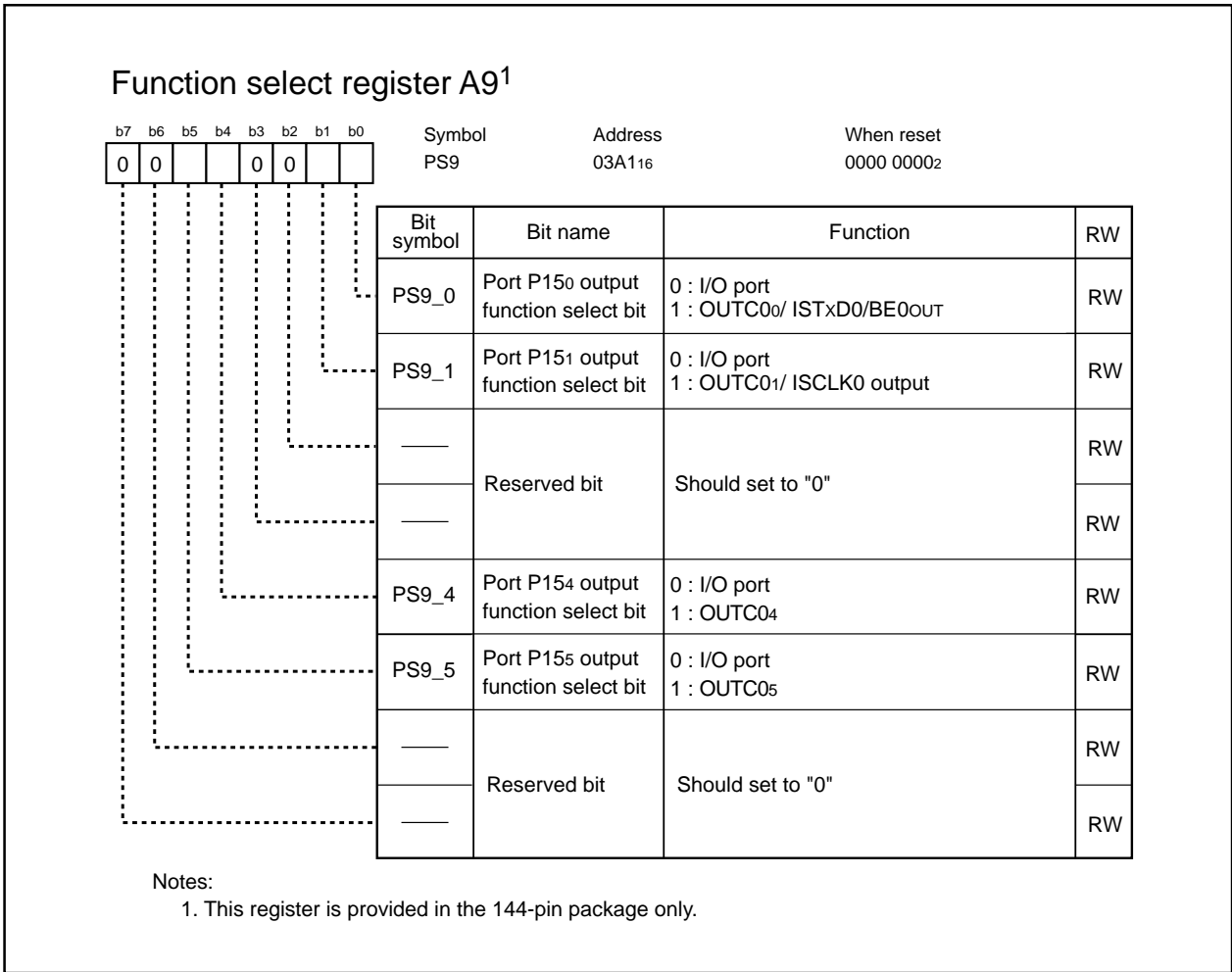


Figure 1.28.11. PS9 Register

Programmable I/O Port

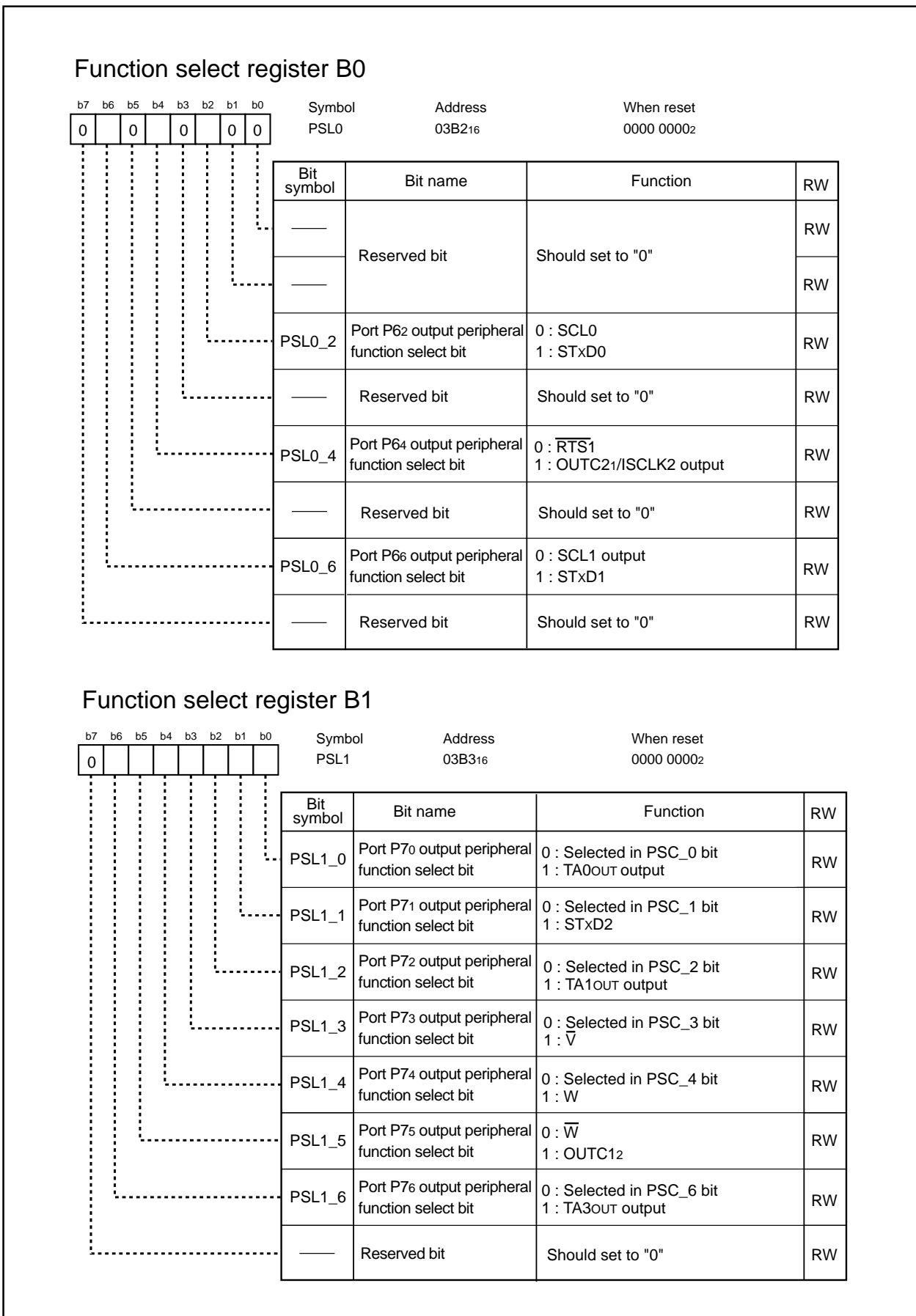
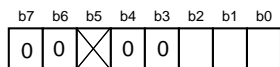


Figure 1.28.12. PSL0 Register and PSL1 Register

Programmable I/O Port

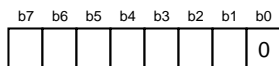
Function select register B2



Symbol: PSL2  
 Address: 03B6<sub>16</sub>  
 When reset: 00X0 0000<sub>2</sub>

Bit symbol	Bit name	Function	RW
PSL2_0	Port P80 output peripheral function select bit	0 : TA4OUT output 1 : U	RW
PSL2_1	Port P81 output peripheral function select bit	0 : $\bar{U}$ 1 : OUTC30/ISTxD3	RW
PSL2_2	Port P82 output peripheral function select bit	0 : OUTC32/ISRxD3 1 : CANOUT	RW
—	Reserved bit	Should set to "0"	RW
—			RW
—	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
—	Reserved bit	Should set to "0"	RW
—			RW

Function select register B3



Symbol: PSL3  
 Address: 03B7<sub>16</sub>  
 When reset: 0000 0000<sub>2</sub>

Bit symbol	Bit name	Function	RW
—	Reserved bit	Should set to "0"	RW
PSL3_1	Port P91 output peripheral function select bit	0 : SCL3 output 1 : STxD3	RW
PSL3_2	Port P92 output peripheral function select bit	0 : TxD3/SDA3 1 : OUTC20/ISTxD2/IEOUT	RW
PSL3_3	Port P93 output peripheral function select bit	0 : Expect DA0 1 : DA0	(Note 1) RW
PSL3_4	Port P94 output peripheral function select bit	0 : Expect DA1 1 : DA1	(Note 1) RW
PSL3_5	Port P95 output peripheral function select bit	0 : Expect ANEX0 1 : ANEX0	(Note 1) RW
PSL3_6	Port P96 output peripheral function select bit	0 : Expect ANEX1 1 : ANEX1	(Note 1) RW
PSL3_7	Port P97 output peripheral function select bit	0 : SCL4 output 1 : STxD4	RW

Notes:

- Although DA0, DA1, ANEX0 and ANEX1 can be used when this bit is set to "0", the power consumption may increase.

Figure 1.28.13. PSL2 Register and PSL3 Register

Programmable I/O Port

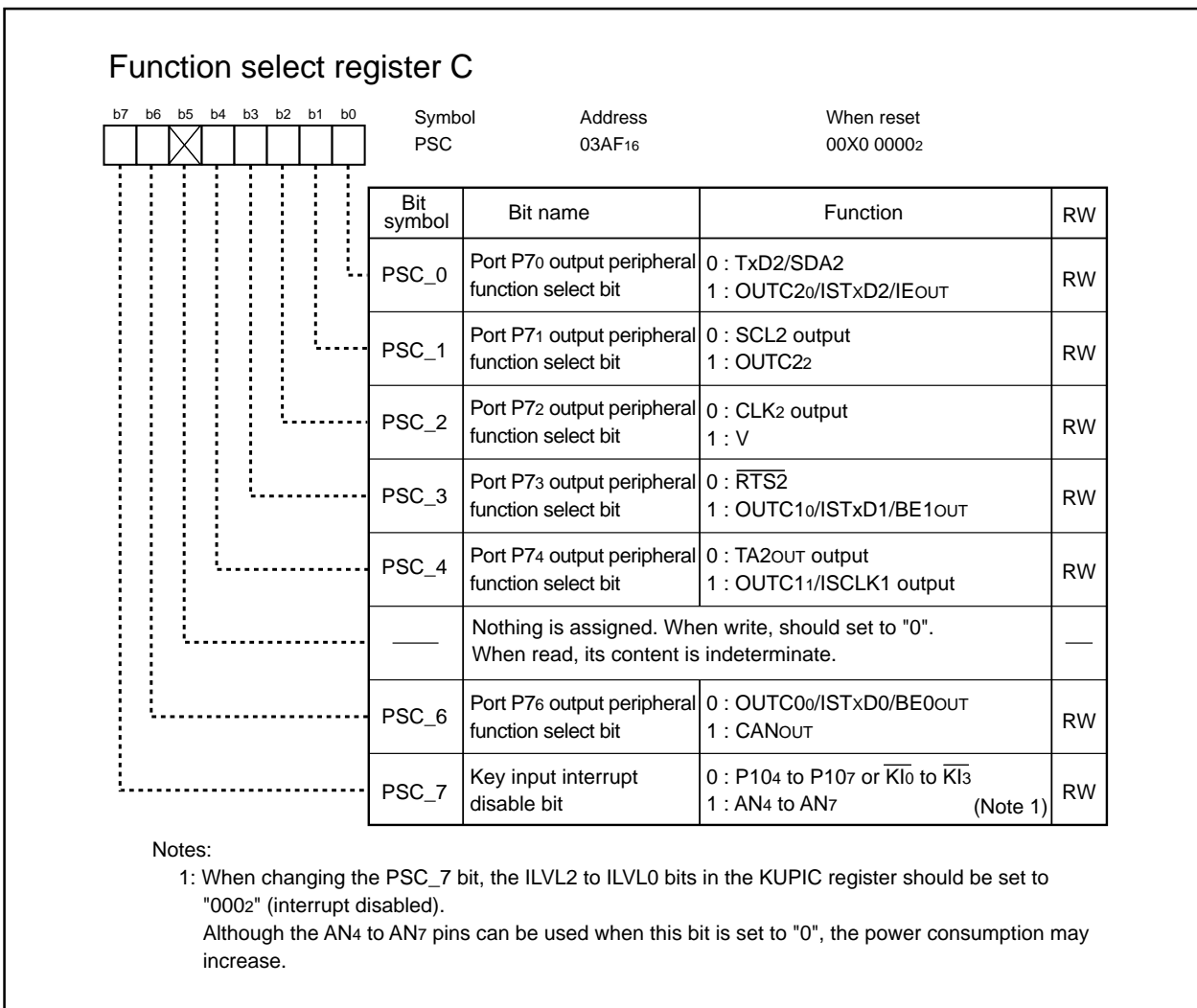


Figure 1.28.14. PSC Register

Programmable I/O Port

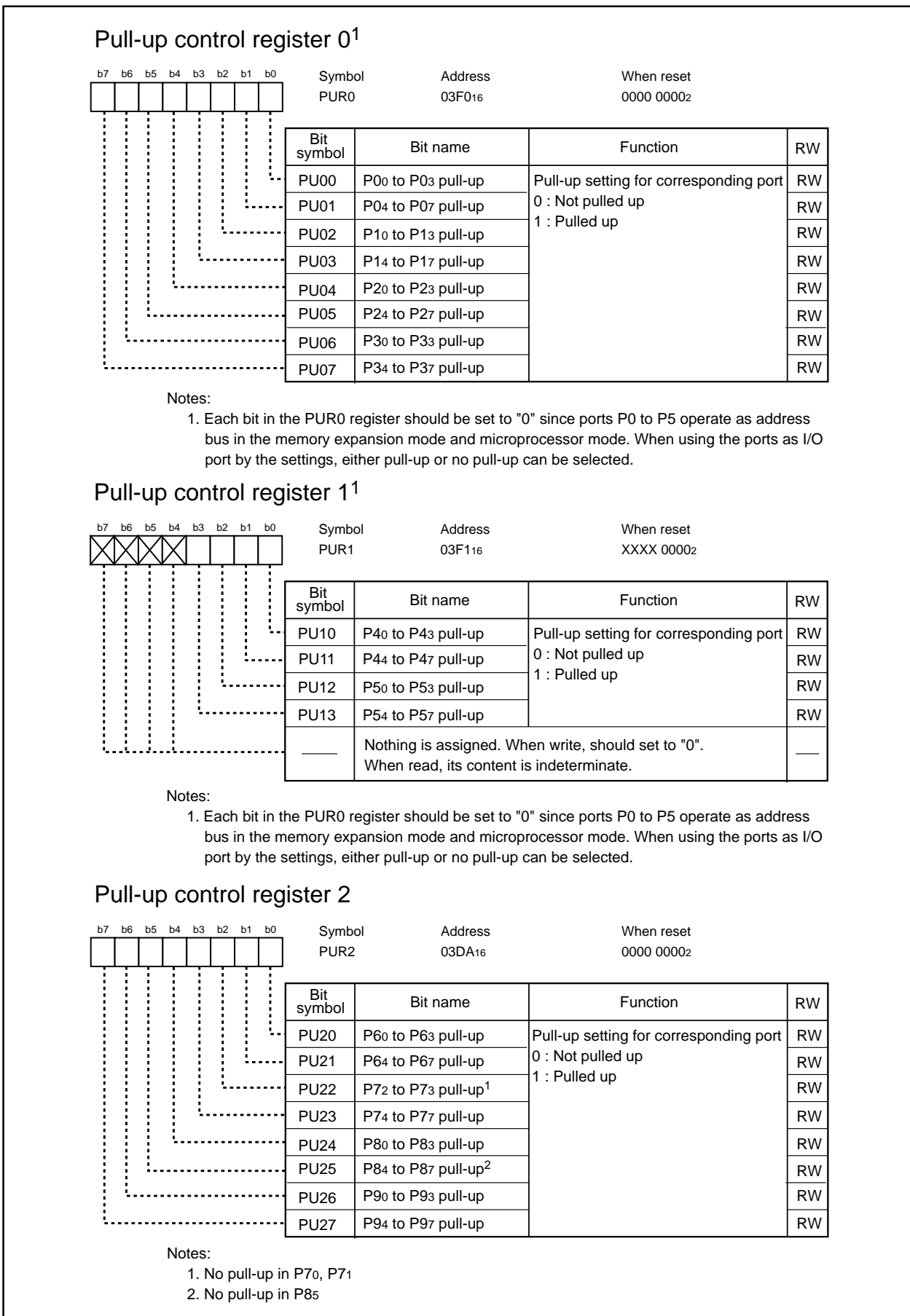
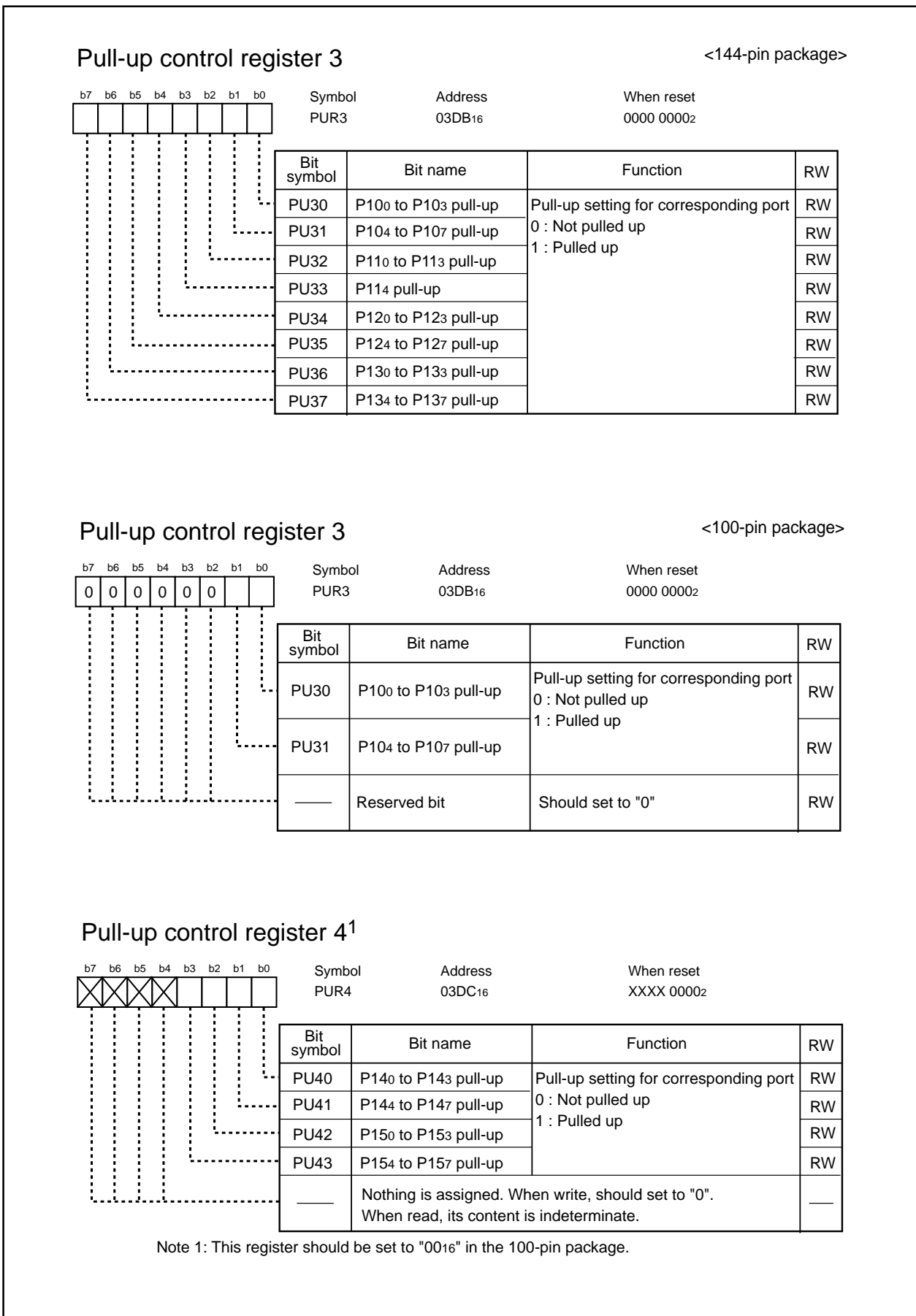


Figure 1.28.15. PUR0 Register, PUR1 Register and PUR2 Register

Programmable I/O Port



Programmable I/O Port

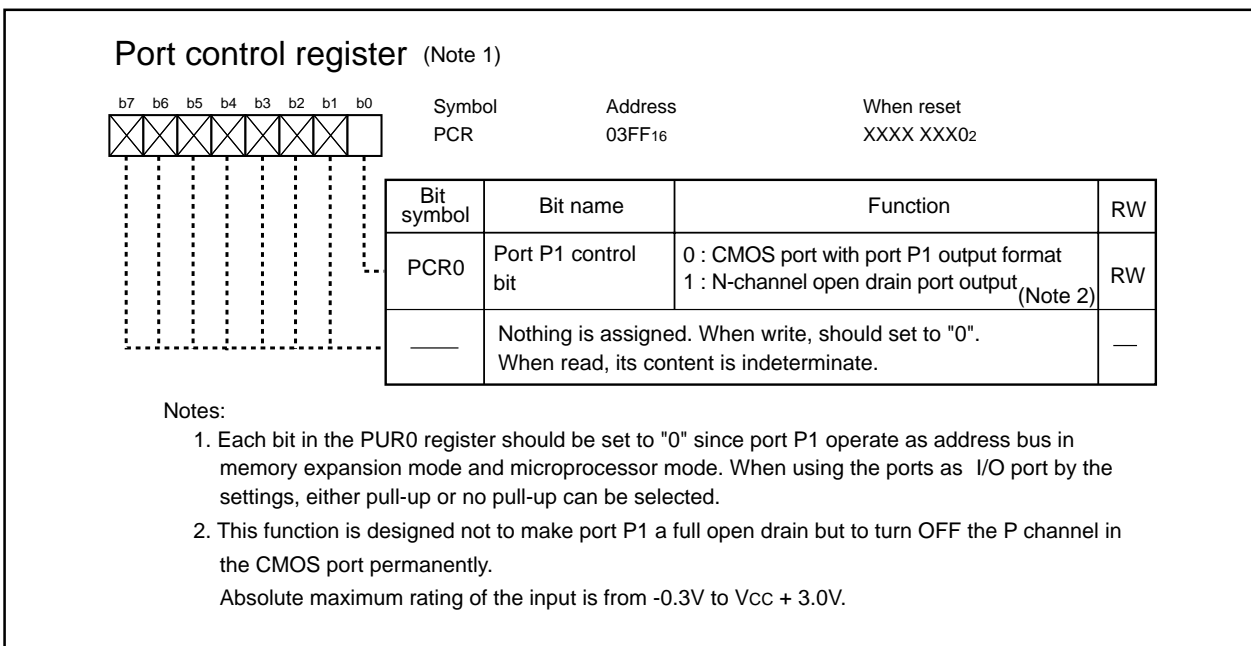


Figure 1.28.17. PCR Register

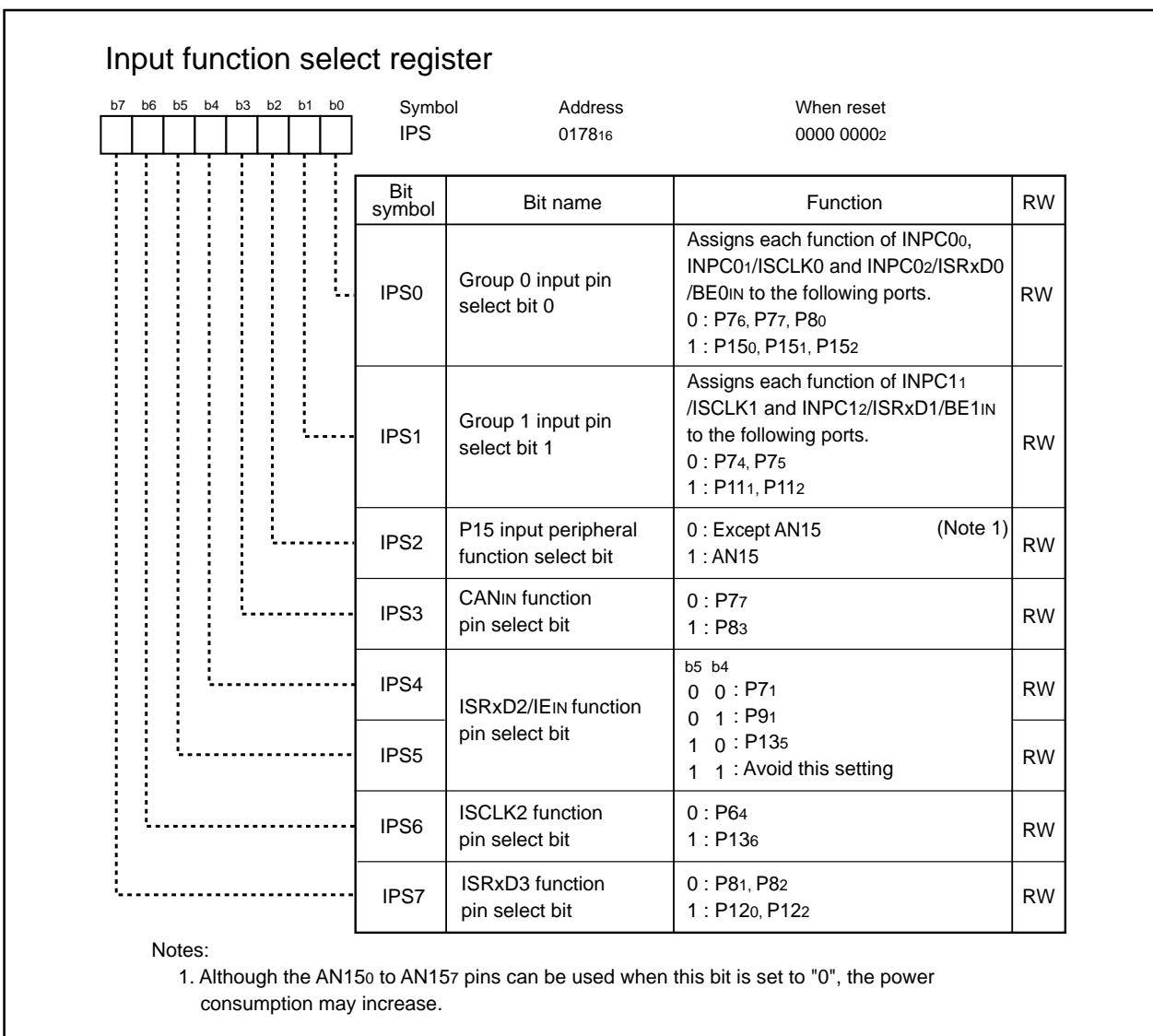


Figure 1.28.18. IPS Register

Programmable I/O Port

**Table 1.28.1. Unassigned Pin Handling in Single-chip Mode**

Pin name	How to handle
Ports P0 to P15 (excluding P85) <sup>1</sup>	After setting for input mode, connect each pin to Vss via a resistance (pull-down); or after setting for output mode, leave these pins open.
XOUT <sup>2</sup>	Open
NMI(P85)	Connect to Vcc via a resistance (pull-up)
AVcc	Connect to Vcc
AVss, VREF, BYTE	Connect to Vss

Notes:

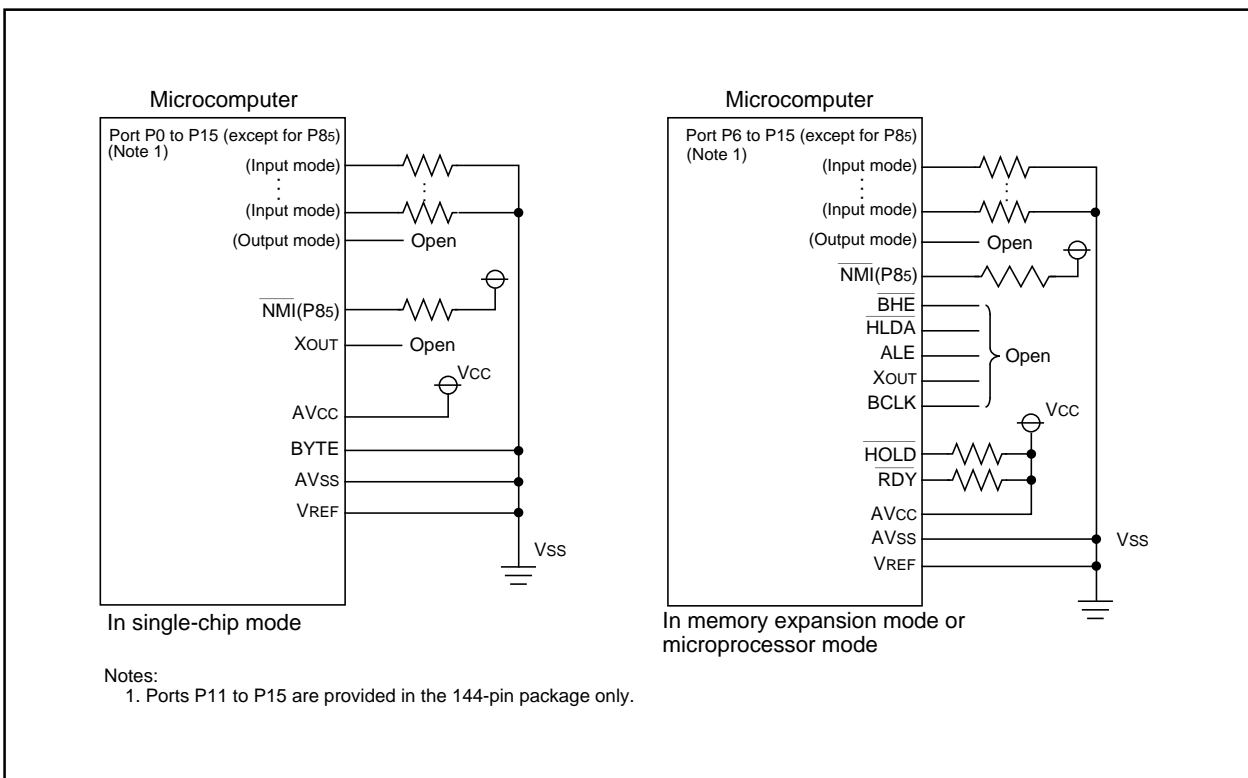
1. Ports P11 to P15 are provided in the 144-pin package only.
2. When the external clock is input to the XIN pin, the pin should be handled like the above.

**Table 1.28.2. Unassigned Pin Handling in Memory Expansion Mode and Microprocessor Mode**

Pin name	How to handle
Ports P6 to P15 (excluding P85) <sup>1</sup>	After setting for input mode, connect each pin to Vss via a resistance (pull-down); or after setting for output mode, leave these pins open.
BHE, ALE, HLDA, XOUT <sup>2</sup> , BCLK	Open
HOLD, RDY, NMI(P85)	Connect to Vcc via a resistance (pull-up)
AVcc	Connect to Vcc
AVss, VREF	Connect to Vss

Notes:

1. Ports P11 to P15 are provided in the 144-pin package only.
2. When the external clock is input to the XIN pin, the pin should be handled like the above.



**Figure 1.28.19. Unassigned Pin Handling**

## Programmable I/O Port

Table 1.28.3. Port P6 Peripheral Function Output Control

	PS0 register	PSL0 register
Bit 0	0: P60/CTS0/SS0 1: RTS0	Should set to "0"
Bit 1	0: P61/CLK0 input 1: CLK0 output	Should set to "0"
Bit 2	0: P62/RxD0/SCL0 input 0: Selected in PSL0 register	0: SCL0 output 1: STxD0
Bit 3	0: P63/SRxD0 1: TxD0/SDA0	Should set to "0"
Bit 4	0: P64/CTS1/SS1 1: Selected in PSL0 register	0: RTS1 1: OUTC21/ISCLK2 output
Bit 5	0: P65/CLK1 input 1: CLK1 output	Should set to "0"
Bit 6	0: P66/RxD1/SCL1 input 1: Selected in PSL0 register	0: SCL1 output 1: STxD1
Bit 7	0: P67/SRxD1 1: TxD1/SDA1	Should set to "0"

Table 1.28.4. Port P7 Peripheral Function Output Control

	PS1 register	PSL1 register	PSC register
Bit 0	0: P70/SRxD2/TA0OUT input 1: Selected in PSL1 register	0: Selected in PSC register 1: TA0OUT output	0: TxD2/SDA2 1: OUTC20/ISTxD2/IEOUT
Bit 1	0: P71/TB5IN/TA0IN/RxD2/ ISRxD2/IEIN/SCL2 input 1: Selected in PSL1 register	0: Selected in PSC register 1: STxD2	0: SCL2 output 1: OUTC22
Bit 2	0: P72/TA1OUT input/ CLK2 input 1: Selected in PSL1 register	0: Selected in PSC register 1: TA1OUT output	0: CLK2 output 1: V
Bit 3	0: P73/CTS2/SS2/TA1IN 1: Selected in PSL1 register	0: Selected in PSC register 1: V	0: RTS2 1: OUTC10/ISTxD1/BE1OUT
Bit 4	0: P74/INPC11/ISCK1 input/ TA2OUT input 1: Selected in PSL1 register	0: Selected in PSC register 1: W	0: TA2OUT output 1: OUTC11/ISCLK1 output
Bit 5	0: P75/TA2IN/INPC12/ISRxD1/ BE1IN 1: Selected in PSL1 register	0: $\bar{W}$ 1: OUTC12	Should set to "0"
Bit 6	0: P76/INPC00/TA3OUT input 1: Selected in PSL1 register	0: Selected in PSC register 1: TA3OUT output	0: OUTC00/ISTxD0/BE0OUT 1: CANOUT
Bit 7	0: P77/TA3IN/CANIN/ISCLK0 input/INPC01 1: OUTC01/ISCLK0	Should set to "0"	0: P104 to P107 or $\bar{K}I_0$ to $\bar{K}I_3$ 1: AN4 to AN7 (No relation to P77)

## Programmable I/O Port

**Table 1.28.5. Port P8 Peripheral Function Output Control**

	PS2 register	PSL2 register
Bit 0	0: P80/INPC02/ISRxD0/BE0IN /TA4OUT input 1: Selected in PSL2 register	0: TA4OUT output 1: U
Bit 1	0: P81/TA4IN 1: Selected in PSL2 register	0: $\bar{U}$ 1: OUTC30/ISTxD3
Bit 2	0: P82/ $\bar{INT0}$ 1: Selected in PSL2 register	0: OUTC32/ISRxD3 1: CANOUT
Bit 3 to 7	Should set to "0"	

**Table 1.28.6. Port P9 Peripheral Function Output Control**

	PS3 register	PSL3 register
Bit 0	0: P90/TB0IN/CLK3 input 1: CLK3 output	Should set to "0"
Bit 1	0: P91/TB1IN/RxD3/ISRxD2/SCL3 input/IEIN 1: Selected in PSL3 register	0: SCL3 output 1: STxD3
Bit 2	0: P92/TB2IN/SRxD3 1: Selected in PSL3 register	0: TxD3/SDA3 1: OUTC20/ISTxD2/IEOUT
Bit 3	0: P93/TB3IN/CTS3/SS3/DA0 output 1: RTS3	0: Except DA0 1: DA0
Bit 4	0: P94/TB4IN/CTS4/SS4/DA1 output 1: RTS4	0: Except DA1 1: DA1
Bit 5	0: P95/ANEX0/CLK4 input 1: CLK4 output	0: Except ANEX0 1: ANEX0
Bit 6	0: P96/SRxD4/ANEX1 1: TxD4/SDA4	0: Except ANEX1 1: ANEX1
Bit 7	0: P97/RxD4/ $\bar{ADTRG}$ /SCL4 input 1: Selected in PSL3 register	0: SCL4 output 1: STxD4

**Table 1.28.7. Port P10 Peripheral Function Output Control**

	PSC register
Bit 7	0: P104 to P107 or K10 to K13 1: AN4 to AN7

## Programmable I/O Port

**Table 1.28.8. Port P11 Peripheral Function Output Control**

	PS5 register
Bit 0	0: P110 1: OUTC10/ISTxD1/BE1OUT
Bit 1	0: P111/INPC11/ISCLK1 input 1: OUTC11/ISCLK1 output
Bit 2	0: P112/INPC12/ISRxD1/BEIN 1: OUTC12
Bit 3	0: P113 1: OUTC13
Bit 4 to 7	Should set to "0"

**Table 1.28.9. Port P12 Peripheral Function Output Control**

	PS6 register
Bit 0	0: P120 1: OUTC30/ISTxD3
Bit 1	0: P121/ISCLK3 input 1: OUTC31/ISCLK3 output
Bit 2	0: P122 1: OUTC32/ISRxD3
Bit 3	0: P123 1: OUTC33
Bit 4	0: P124 1: OUTC34
Bit 5	0: P125 1: OUTC35
Bit 6	0: P126 1: OUTC36
Bit 7	0: P127 1: OUTC37

## Programmable I/O Port

**Table 1.28.10. Port P13 Peripheral Function Output Control**

	PS7 register
Bit 0	0: P130 1: OUTC24
Bit 1	0: P131 1: OUTC25
Bit 2	0: P132 1: OUTC26
Bit 3	0: P133 1: OUTC23
Bit 4	0: P134 1: OUTC20/ISTxD2/IEOUT
Bit 5	0: P135/ISRxD2/IEIN 1: OUTC22
Bit 6	0: P136/ISCLK2 input 1: OUTC21/ISCLK2 output
Bit 7	0: P137 1: OUTC27

**Table 1.28.11. Port P14 Peripheral Function Output Control**

	PS8 register
Bit 0	0: P140 1: OUTC14
Bit 1	0: P141 1: OUTC15
Bit 2	0: P142/INPC16 1: OUTC16
Bit 3	0: P143/INPC17 1: OUTC17
Bit 4 to 7	Should set to "0"

**Table 1.28.12. Port P15 Peripheral Function Output Control**

	PS9 register
Bit 0	0: P150/INPC00/AN150 1: OUTC00/ISTxD0/BEOUT
Bit 1	0: P151/INPC01/AN151/ISCLK0 input 1: OUTC01/ISCLK0 output
Bit 2 to 3	Should set to "0"
Bit 4	0: P154/INPC04/AN154 1: OUTC04
Bit 5	0: P155/INPC05/AN155 1: OUTC05
Bit 6 to 7	Should set to "0"

## Usage Precaution

### Notes on the 100-pin Package

Set address space 03CB<sub>16</sub>, 03CE<sub>16</sub>, 03CF<sub>16</sub>, 03D2<sub>16</sub> and 3D3<sub>16</sub> to "FF<sub>16</sub>" as each default value in the 100-pin package. Set address space 03DC<sub>16</sub> to "00<sub>16</sub>" in the 100-pin package.

### HOLD Signal

When entering microprocessor or memory expansion mode from single-chip mode and using a  $\overline{\text{HOLD}}$  pin input, all PD4\_0 to PD4\_7 bits in the PD4 register and PD5\_0 to PD5\_2 bits in the PD5 register are set to "0" (input mode). Then the PM01 to PM00 bits in the PM0 register are set to "112" (microprocessor mode) or "102" (memory expansion mode).

When all PD4\_0 to PD4\_7 bits in the PD4 register and PD5\_0 to PD5\_2 bits in the PD5 register are set to "1" (output mode) and the PM01 to PM00 bits are set to "112" (microprocessor mode) or "102" (memory expansion mode), P40 to P47 ( $\overline{\text{A16}}$  to  $\overline{\text{A23}}$ ,  $\overline{\text{CS0}}$  to  $\overline{\text{CS3}}$ , MA8 to MA12) and P50 to P52 ( $\overline{\text{RD}}/\overline{\text{WR}}/\overline{\text{BHE}}$ ,  $\overline{\text{RD}}/\overline{\text{WRL}}/\overline{\text{WRH}}$ ,  $\overline{\text{CASL}}/\overline{\text{CASH}}/\overline{\text{DW}}$ ) are not placed in a high-impedance state regardless of "L" input to the  $\overline{\text{HOLD}}$  pin.

### Microprocessor Mode

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

When the CNVss pin is in "H", the microcomputer starts an operating in microprocessor mode after reset. Internal ROM space cannot be accessed even if the microcomputer enters memory expansion or single-chip mode after entering microprocessor mode.

### PLL Frequency Synthesizer

Make the supply voltage stable to use the PLL frequency synthesizer.

For ripple with the supply voltage 5V, keep below 10kHz as frequency, below 0.5V (peak to peak) as voltage fluctuation band and below 1V/mS as voltage fluctuation rate.

### Stop Mode and Wait Mode

- (1) When exiting stop mode by hardware reset, set the  $\overline{\text{RESET}}$  pin to "L" until a main clock oscillation is stabilized.
- (2) Insert at least four NOP instructions after the WAIT instruction or a instruction that the CM10 bit in the CM1 register is set to "1". When entering wait mode or stop mode, an instruction queue reads an instruction following the WAIT instruction and an instruction to set the CM10 bit to "1" (all clocks stopped). The next instruction may be executed before entering wait mode or stop mode, depending on a combination of the instruction and an execution timing.

### Interrupts

#### (1) ISP Setting

ISP is reset to "000000<sub>16</sub>" after reset. The microcomputer runs out of control if an interrupt is acknowledged before setting ISP. Set ISP before an interrupt is acknowledged. With the  $\overline{\text{NMI}}$  interrupt, Reset ISP at the beginning of program. All interrupts including the  $\overline{\text{NMI}}$  interrupt are acknowledged when executing the first instruction after reset. Set an even number in ISP to increase an operating rate for interrupt sequence.

## Usage Precaution

(2)  $\overline{\text{NMI}}$  Interrupt

- The  $\overline{\text{NMI}}$  interrupt cannot be obstructed. Connect (pull up) the  $\overline{\text{NMI}}$  pin to  $V_{\text{CC}}$  via a resistor if not used.
- A  $\overline{\text{NMI}}$  pin value can be read by the P8\_5 bit in the P8 register. Read this bit only when identifying pin levels after the  $\overline{\text{NMI}}$  interrupt is generated.
- Input at least two CPU clock cycles + 300ns as "L" width to the  $\overline{\text{NMI}}$  pin.

(3)  $\overline{\text{INT}}$  Interrupt• **Edge sense**

Input at least 250ns as "L" or "H" width to the  $\overline{\text{INT}}_i$  pins ( $i = 0$  to 5) regardless of the CPU clock.

• **Level sense**

Input at least one CPU clock cycle + 200ns as "L" or "H" width to the  $\overline{\text{INT}}_i$  pins. (At least 234ns when  $X_{\text{IN}} = 30\text{MHz}$  and no division.)

- When a polarity of the  $\overline{\text{INT}}_i$  pins is switched, the IR bit in the  $\text{INTiIC}$  register may be set to "1". Set the IR bit to "0" (no interrupt request) after switching.

Figure 1.9.16 shows a procedure to switch the  $\overline{\text{INT}}$  interrupt requests.

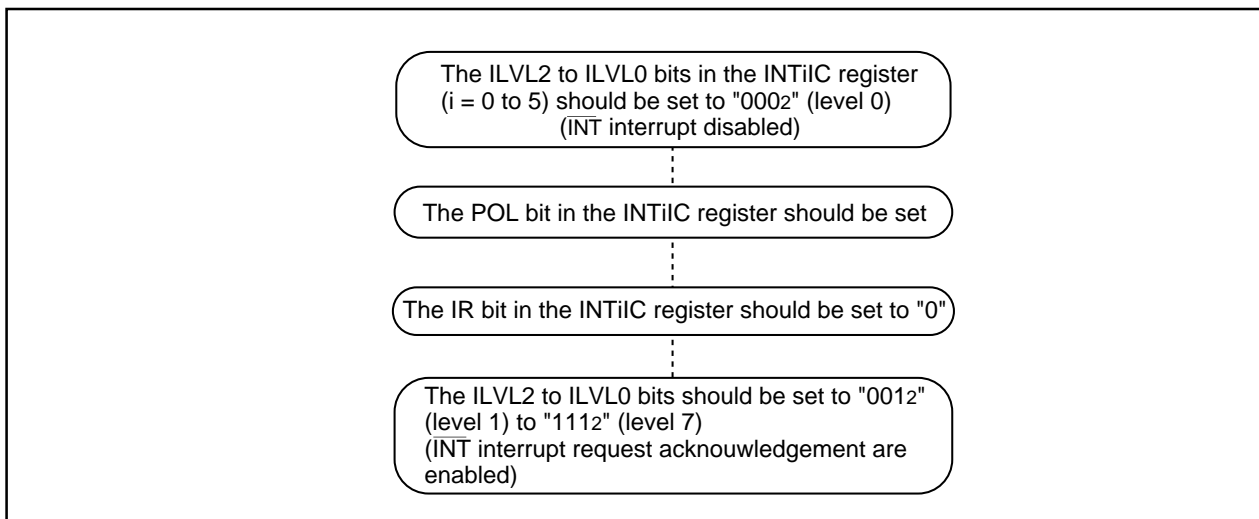


Figure 1.29.1. Switching Procedure for INT Interrupt

## (4) Changing Interrupt Control Register

Take the below procedure when changing the interrupt control register under an interrupt-inhibited condition.

• **Changing Bits Except Interrupt Request Bit**

An interrupt may be disabled to leave the IR bit unchanged in "0" when a corresponding interrupt is generated during an instruction executing. If that is a problem, use the below instructions to change the register.

AND, OR, BCLR, BSET

• **Changing Interrupt Request Bit**

When setting the IR bit to "0" (no interrupt request), the IR bit may remain unchanged in "1", depending on an used instruction. If that is a problem, use the below instruction to change the register.

MOV

## Usage Precaution

---

### (5) Changing IIOiR Register (i = 0 to 11)

When bits 1 to 7 in the IIOiR register are set to "0" (no interrupt request), use the below instructions to change the register.

AND, BCLR

## DMAC

(1) When setting registers associated with the DMAC, the MDi1 to MDi0 bits (i=0 to 3) in the DMDj register (j=0,1) corresponding to channel i should be set to "002" (DMA disabled). Then set the MDi1 to MDi0 bits to "012" (single transfer) or "112" (repeat transfer). This setting allows a DMA request of channels to be received.

(2) Avoid setting the DRQ bit in the DMiSL register to "0" (no request).

When a DMA request is generated with a channel disabled<sup>1</sup>, a DMA transfer is not operated and the DRQ bit is set to "0."

Notes :

1. This state means that the MDi1 to MDi0 bits is set to "002" or the DCTi register is set to "0000<sub>16</sub>" (the number of transfer=0).

(3) When a DMA transfer is operated by a software trigger, set the DSR and DRQ bits in the DMiSL register to "1" simultaneously.

e.g. OR.B #0A0h, DMiSL --- Set the DSR and DRQ bits to "1" simultaneously.

(4) With the DMA interrupt including other channels, avoid generating a DMA request of channel i when setting the DCTi register to "1" and the MDi1 to MDi0 bits of corresponding channel i to "012" or "112". Generate a DMA request of channel i after setting a DMA-associated register of channel i. Set the peripheral function that causes a DMA request to be generated after setting a DMA-associated register. If not fulfilling the above conditions (setting the  $\overline{\text{INT}}$  interrupt as a DMA request), avoid setting the DCTi register to "1".

## Timer A

The TAI<sub>s</sub> bit (i=1 to 4) in the TABSR register is set to "0" (stops counting) after reset. The TAI<sub>s</sub> bit should be set to "1" (starts counting) after setting the TAI register.

### 1. Timer Mode

The TAI register can indicate a value of the counter at any time while counting. The counter can be "FFFF<sub>16</sub>" while reloading. When setting a value in the TAI register while the counter stops, the setting value can be read until the counter starts counting.

### 2. Event Counter Mode

#### (1) Common

The TAI register can indicate a value of the counter at any time while counting. The counter can be "FFFF<sub>16</sub>" in underflow and "0000<sub>16</sub>" in overflow while reloading. When setting a value in the TAI register while the counter stops, the setting value can be read until the counter starts counting.

## Usage Precaution

---

### (2) For free-running operation

A value of the TAI register may be indeterminate when the counter starts. The counter may start at an indeterminate value regardless of a value that is set in the TAI register before the counter starts.

- No switching between the counter increment and the counter decrement

Set the TCK0 bit to "0" (reloading). Set the TAI register before starting the counter. Set a value in TAI register again between the counter starting and an underflow or overflow occurring. The TAI register is set to "0000<sub>16</sub>" to increment the counter and "FFFF<sub>16</sub>" to decrement the counter to get the same operation as when setting the TCK0 bit to "1" (free-running).

- When switching between the counter increment and the counter decrement

Set the TCK0 bit to "0" (reloading) before a count pulse is input. Set the TCK0 bit to "1" (free-running) after one pulse of a count pulse is input.

### 3. One-Shot Timer Mode

#### (1) When setting the TABSR register to "0" (stops counting), the followings occur:

- The counter stops and a content of the reload register is reloaded.
- The TAIOUT pin outputs "L".
- The IR bit in the TAIIC register is set to "1" (interrupt request) in one CPU clock cycle.

#### (2) Output in one-shot timer mode synchronizes with an internally generated count source. With an external trigger, one-cycle delay of a count source as maximum occurs between a trigger input to the TAIIN pin and an output in one-shot timer mode.

#### (3) The IR bit is set to "1" when the microcomputer enters any of the following mode:

- Entering one-shot timer mode after reset.
- Entering one-shot timer mode from timer mode.
- Entering one-shot timer mode from event counter mode.

With the timer Ai interrupt, set the IR bit to "0" after entering one of modes listed above.

#### (4) When a trigger occurs while counting, the counter is decremented once after a second trigger occurs. Then a value of the reload register is reloaded to the counter and the counter runs. To generate a trigger while counting, generate a second trigger in at least one timer count source cycle after the first trigger occurs.

### 4. Pulse Width Modulation Mode

#### (1) The IR bit is set to "1" when the microcomputer enters any of the following mode:

- Entering PWM mode after reset.
- Entering PWM mode from timer mode.
- Change PWM mode from event counter mode.

With the timer Ai interrupt, set the IR bit to "0" after entering one of modes listed above.

#### (2) When setting the TAI S register to "0" (stops counting) while a PWM pulse is output, the followings occur:

- The counter stops.
- If the TAIOUT pin is output "H", output level is in "L" and the IR bit is set to "1".
- If the TAIOUT pin is output "L", both output level and the IR bit remains unchanged.

## Timer B

The TBiS bits in the TABSR and TBSR registers are set to "0" (stops counting) after reset. Set the TBiS bit to "1" after setting the TBi register.

## Usage Precaution

---

### 1. Timer Mode and Event Counter Mode

The TBi register ( $i=0$  to 5) can indicate a value of the counter at any time while counting. The counter can be "FFFF16" while reloading. When setting a value in the TBi register while the counter stops, the setting value can be read until the counter starts counting.

### 2. Pulse Period Measurement Mode and Pulse Width Measurement Mode

- (1) When changing the MR1 to MR0 bits in the TBiMR register after the counter starts, the IR bit in the TBiC register may be set to "1" (interrupt request).
- (2) When the counter starts and the microcomputer detects the first valid edge, an indeterminate counter value is transferred to the reload register. At this time, a timer Bi interrupt request is not generated.
- (3) A value of the counter is indeterminate when the counter starts. The MR3 bit in the TBiMR register may be set to "1" (overflow) before a valid edge is input. A timer Bi interrupt request may be generated.
- (4) To set the MR3 bit to "0" (no overflow), set the TBiMR register in at least one count source cycle after setting the TBiS bit to "1" (overflow).

## Intelligent I/O

To start the base timer, set either the BTiS bit ( $i=0$  to 3) in the B TSR register or the BTS bit in the GiBCR1 register. If both are set to "1," set both bits to "0" for the counter stop.

## A-D Converter

- (1) Avoid setting the ADiCON0 ( $i=0, 1$ ) (except the ADST bit), ADiCON1 and ADiCON2 registers during the A-D conversion.
- (2) When the VCUT bit in the ADiCON1 register changes "0" to "1", wait in 1  $\mu$ s or longer before starting the A-D conversion.
- (3) To change A-D operation mode, set the CH2 to CH0 bits in the ADiCON0 register and the SCAN1 to SCAN0 bits in the ADiCON1 register again to determine analog input pins.
- (4) In one-shot or single sweep mode, the IR bit in the ADiC register indicates whether the A-D conversion is completed. Verify the IR bit in the ADiC register before reading the ADij register ( $j=0$  to 7).
- (5) In repeat mode, repeat sweep mode 0 or repeat sweep mode 1, use the undivided main clock as the CPU clock.
- (6) When the A-D conversion is terminated by program during the A-D conversion, a result of the A-D conversion is indeterminate. The A-D registers, which does not operate for A-D conversion, may be indeterminate. Avoid using all values of the A-D register when the A-D conversion is terminated by program during the A-D conversion.
- (7) When  $f(X_{IN})$  is faster than 10 MHz, keep the  $\emptyset$  AD frequency 10 MHz or less by dividing  $f(X_{IN})$ .

## Programmable I/O Ports

The P72 to P75, P80 and P81 pins contain the forced-terminated function of three-phase PWM output. Three-phase motor control timer functions and the  $\overline{NMI}$  pin affect these pins above when setting these pins for output functions (port output, timer output, three-phase PWM output, serial I/O output and intelligent I/O output).

Table 1.29.1 lists setting values in the INV00 register, the  $\overline{NMI}$  pin input level and output pin states.

## Usage Precaution

Table 1.29.1. INV00 Register and  $\overline{\text{NMI}}$  Pin

Setting Value of INV00 register		input level to $\overline{\text{NMI}}$ pin	States of P72 to P75, P80, and P81 pin (When setting an output pin)
INV02 bit	INV03 bit		
0 (not used three-phase motor control timer function)	—	—	Output functions selected in the PS1, PSL1, PSC, PS2 and PSL2 registers
1 (used three-phase motor control timer function)	0 (three-phase PWM output disabled)	—	High-impedance
	1 (three-phase PWM output enabled) <sup>1</sup>	H	Output functions selected in the PS1, PSL1, PSC, PS2, and PSL2 registers
		L (forced-terminated)	High-impedance

Notes :

1. The INV03 bit is set to "0" after input "L" to the  $\overline{\text{NMI}}$  pin.

## Noise

To reduce noise, connect a bypass capacitor (approximately 0.1 $\mu$ F) between the Vcc and Vss pins with short wiring and thicker circuit trace.

## Reducing Power Consumption

- (1) When not using the A-D convertor, set the VCUT bit in the ADiCON1 register to "0" (no VREF connection). When using the A-D convertor, wait at least 1  $\mu$ s to start the A-D convertor after setting the VCUT bit to "1" (VREF connection).
- (2) To use AN4 (P104) to AN7 (P107), set the PSC\_7 bit in the PSC register to "1" (key input interrupt disabled).  
When setting the PSC\_7 bit to "1," the key input interrupt is not available. Nothing can be input to port pins even if the direction registers in P104 to P107 are set for input. An input result becomes indeterminate.
- (3) To use ANEX0 and ANEX1, set the PSL3\_5 bit in the PSL3 register to "1" (ANEX0 used) and the PSL3\_6 bit in the PSL3 register to "1" (ANEX1 used).  
When setting the PSL3\_5 and PSL3\_6 bits to "1", nothing can be input to port pins even if the direction registers in P104 to P107 are set for input. An input result becomes indeterminate. Also, it is unavailable to input the peripheral function except ANEX0 and ANEX1.
- (4) When not using the A-D convertor, set the DAj bit (j=0, 1) in the DACON register to "0" (input disabled) and the DAj register to "00<sub>16</sub>".
- (5) When using the D-A convertor, set the PSL3\_3 bit to "1" (D-A0) the PSL3\_4 bit to "1" (except DA-0)  
When setting the PSL3\_3 and PSL3\_4 bits to "1", nothing can be input to port pins even if the direction registers in P104 to P107 are set for input. An input result becomes indeterminate. Also, it is unavailable to input the peripheral function.

## Register Settings

Table 1.29.2 lists registers including indeterminate bits when read. Set immediate values in these registers. When altering a present value in the register for the next value to be rewritten, write the present value into the register and also RAM. Then write the next value in the register after writing it into RAM.

## Usage Precaution

Table 1.29.2. Register Having Indeterminate Bits when Read

Register	Address	Register	Address
WDT5 register	000E <sub>16</sub>	U2BRG register	0339 <sub>16</sub>
G0RI register	00EC <sub>16</sub>	U2TB register	033B <sub>16</sub> , 033A <sub>16</sub>
G1RI register	012C <sub>16</sub>	UDF register	0344 <sub>16</sub>
G2TB register	016D <sub>16</sub> , 016C <sub>16</sub>	TA0 register <sup>1</sup>	0347 <sub>16</sub> , 0346 <sub>16</sub>
U4BRG register	02F9 <sub>16</sub>	TA1 register <sup>1</sup>	0349 <sub>16</sub> , 0348 <sub>16</sub>
U4TB register	02FB <sub>16</sub> , 02FA <sub>16</sub>	TA2 register <sup>1</sup>	034B <sub>16</sub> , 034A <sub>16</sub>
TA11 register	0303 <sub>16</sub> , 0302 <sub>16</sub>	TA3 register <sup>1</sup>	034D <sub>16</sub> , 034C <sub>16</sub>
TA21 register	0305 <sub>16</sub> , 0304 <sub>16</sub>	TA4 register <sup>1</sup>	034F <sub>16</sub> , 034E <sub>16</sub>
TA41 register	0307 <sub>16</sub> , 0306 <sub>16</sub>	U0BRG register	0369 <sub>16</sub>
DTT register	030C <sub>16</sub>	U0TB register	036B <sub>16</sub> , 036A <sub>16</sub>
ICTB2 register	030D <sub>16</sub>	U1BRG register	02E9 <sub>16</sub>
U3BRG register	0329 <sub>16</sub>	U1TB register	02EB <sub>16</sub> , 02EA <sub>16</sub>
U3TB register	032B <sub>16</sub> , 032A <sub>16</sub>	AD0CON2 register	0394 <sub>16</sub>

Notes :

1. In one-shot timer mode and pulse width modulation mode only.

### Resetting CNVss Pin with "H"

Once the CNVss pin is in "H" and the microcomputer enters microprocessor mode after reset, an access to the internal ROM is unavailable even if entering memory expansion or single-chip mode.

### Low-Voltage Operation

VDC (voltage down converter) steps down externally provided power supply to 3.3V as the internal operating voltage. When the power supply is in 3.3V, separate VDC to reduce power consumption. Figure 1.29.2 shows a procedure how to separate VDC.

Take the procedure shown on Figure 1.29.2 with the CPU clock divided by eight as soon as reset. Set the VDC0 register (0001B<sub>16</sub>) to "0F<sub>16</sub>". With the power supply above 3.3V, avoid setting the VDC0 register.

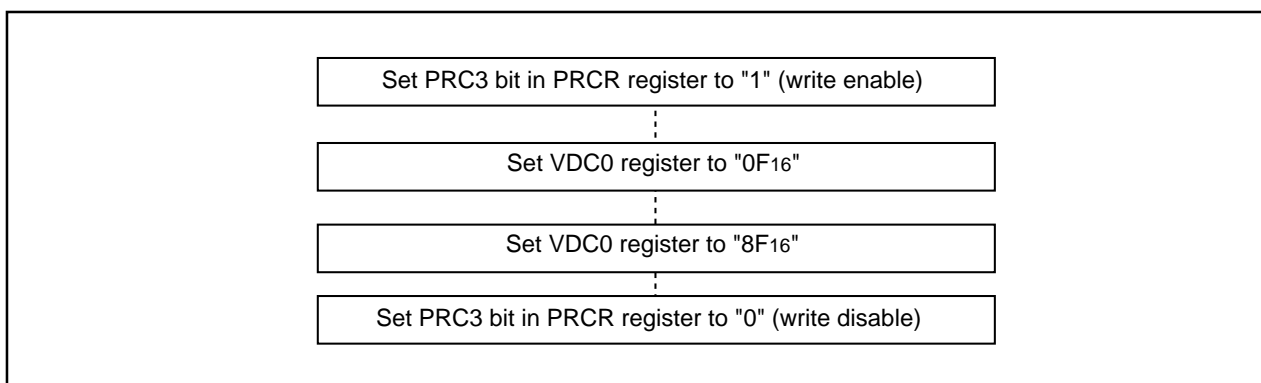


Figure 1.29.2. VDC Separating Procedure

## Flash Memory Version

## Flash Memory Version

## Flash Memory Performance

The flash memory version has the same function, except a built-in flash memory, as the mask ROM version. In the flash memory version, the flash memory can perform in three rewrite modes : CPU rewrite mode, standard serial I/O mode and parallel I/O mode.

Table 1.30.1 lists specifications of the flash memory version. (See Tables 1.1.1 and 1.1.2 for items that are not listed in Table 1.30.1.)

Table 1.30.1. Flash Memory Version Specifications

Item		Specification
Supply voltage		4.2V to 5.5V (f(XIN)=30MHz, with no wait) 3.0V to 5.5V (f(XIN)=20MHz, with no wait)
Voltage required for program and erasure		VDC ON : 4.2 to 5.5V, VDC OFF : 3.0 to 3.6V CPU clock=12.5MHz with 1 wait, CPU clock=6.25MHz with no wait
Flash memory operating mode		3 modes (parallel I/O, standard serial I/O, CPU rewrite)
Erasable block to be split	User ROM space	See Figure 1.30.1
	Boot ROM space	1 block (8K bytes) <sup>1</sup>
Method for program		In 256-byte unit as a page
Method for erasure		All erasure, block erasure
Method to control program and erasure		Program and erasure are controlled by software command
Method to protect		Protected for each block by a lock bit
Number of commands		8 commands
Number of program and erasure		100 times
Data holding period		10 years
ROM code protection		Parallel I/O and standard serial I/O modes are supported

Notes :

1. The boot ROM space contains a program to control programming and erasing in standard I/O mode with shipment.  
The boot ROM space is rewritten in parallel I/O mode only.

Table 1.30.2. Flash Memory Rewrite Mode

Flash memory rewrite mode	CPU rewrite mode	Standard serial I/O mode	Parallel I/O mode
Function	User ROM space is rewritten by the CPU executing software commands	User ROM space is rewritten by using a dedicated serial writer Standard serial I/O mode 1: clock synchronous serial I/O Standard serial I/O mode 2: clock asynchronous serial I/O	Boot ROM space and user ROM space are rewritten by using a dedicated parallel writer
Space to be rewritten	User ROM space	User ROM space	User ROM space Boot ROM space
Operating mode	Single-chip mode Memory expansion mode Boot mode	Boot mode	Parallel I/O mode
ROM writer	-	Serial writer	Parallel writer

Flash Memory Version

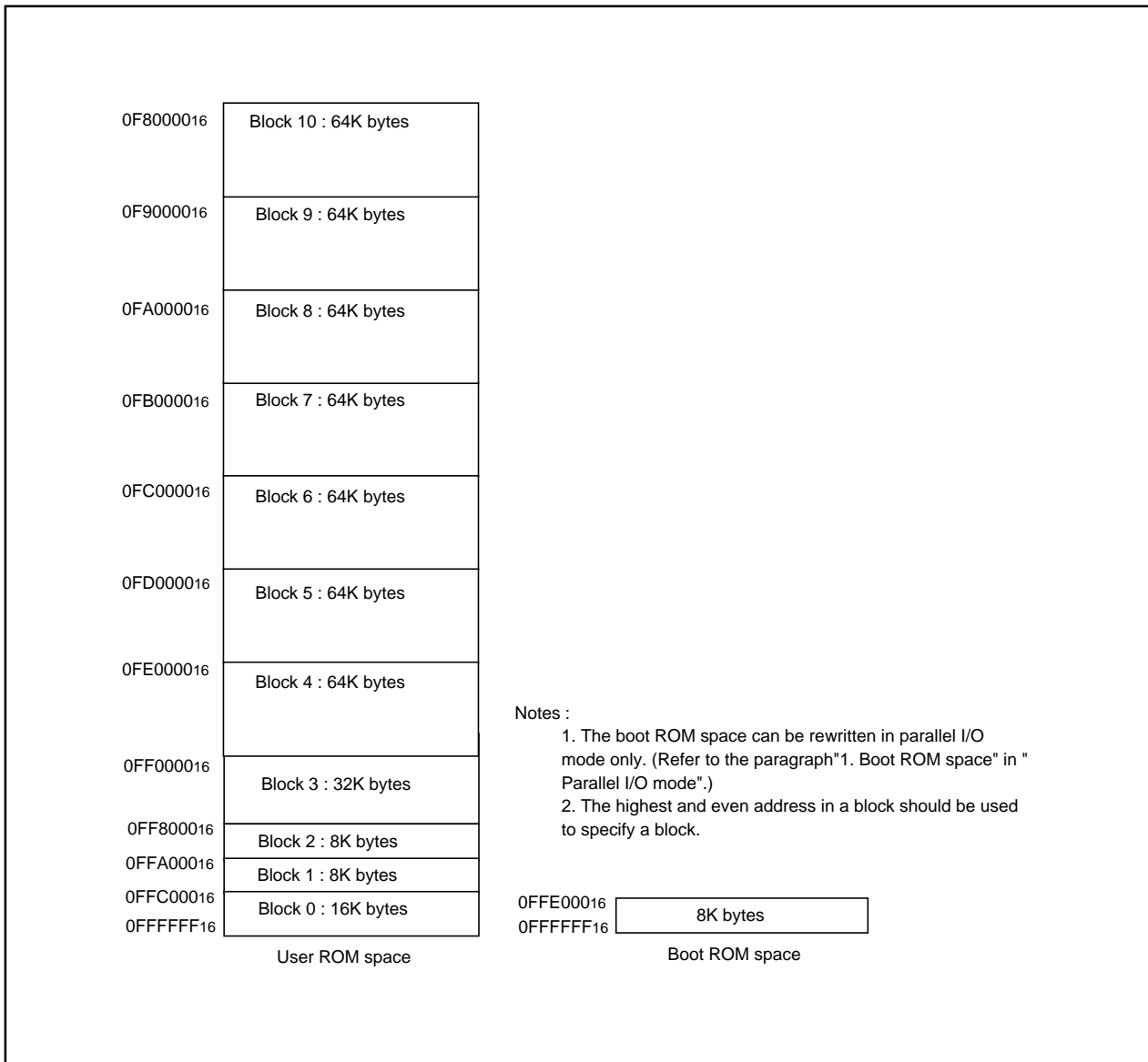
**1. Memory Map**

The flash memory version has a 8K-byte boot ROM space and an user ROM space which stores a microcomputer operating program in single-chip or memory expansion mode.

Figure 1.30.1 shows a block diagram of the flash memory.

The user ROM space is divided into several blocks, each of which can be individually protected (locked) against programming or erasure. The user ROM space can be rewritten in CPU rewrite, standard serial I/O and parallel I/O modes.

Addresses of the boot ROM space overlap partial addresses of the user ROM space. The boot ROM space can be rewritten in parallel I/O mode only (Refer to the paragraph "Parallel I/O mode"). When resetting with "H" input to the CNVss pin and the P50 pin and "L" to the P55 pin, a program in the boot ROM space is executed (Refer to the paragraph "Boot mode"). When resetting with "L" input to the CNVss pin, a program in the user ROM space is executed and the boot ROM space cannot be read.



**Figure 1.30.1. Flash Memory Block Diagram**

## 2. Boot Mode

The microcomputer enters boot mode when resetting with "H" input to the CNVss and P50 pins and "L" to the P55 pin. A program in the boot ROM space is executed after reset.

In boot mode, the FMR05 bit switches from the boot ROM space to the user ROM space and vice versa. The boot ROM space contains a program to control programming and erasing in standard serial I/O mode with shipment. (Refer to the paragraph "Standard serial I/O mode".)

The boot ROM space can be rewritten in parallel I/O mode. The boot ROM space can be rewritten for each system if any rewrite control programs used in CPU rewrite mode are written in the boot ROM space.

## 3. Functions to Prevent Flash Memory from Rewriting

The flash memory version has a built-in ROM code protect function for parallel I/O mode and a built-in ID code check function for standard I/O mode to prevent the flash memory from reading or rewriting.

### • ROM Code Protect Function

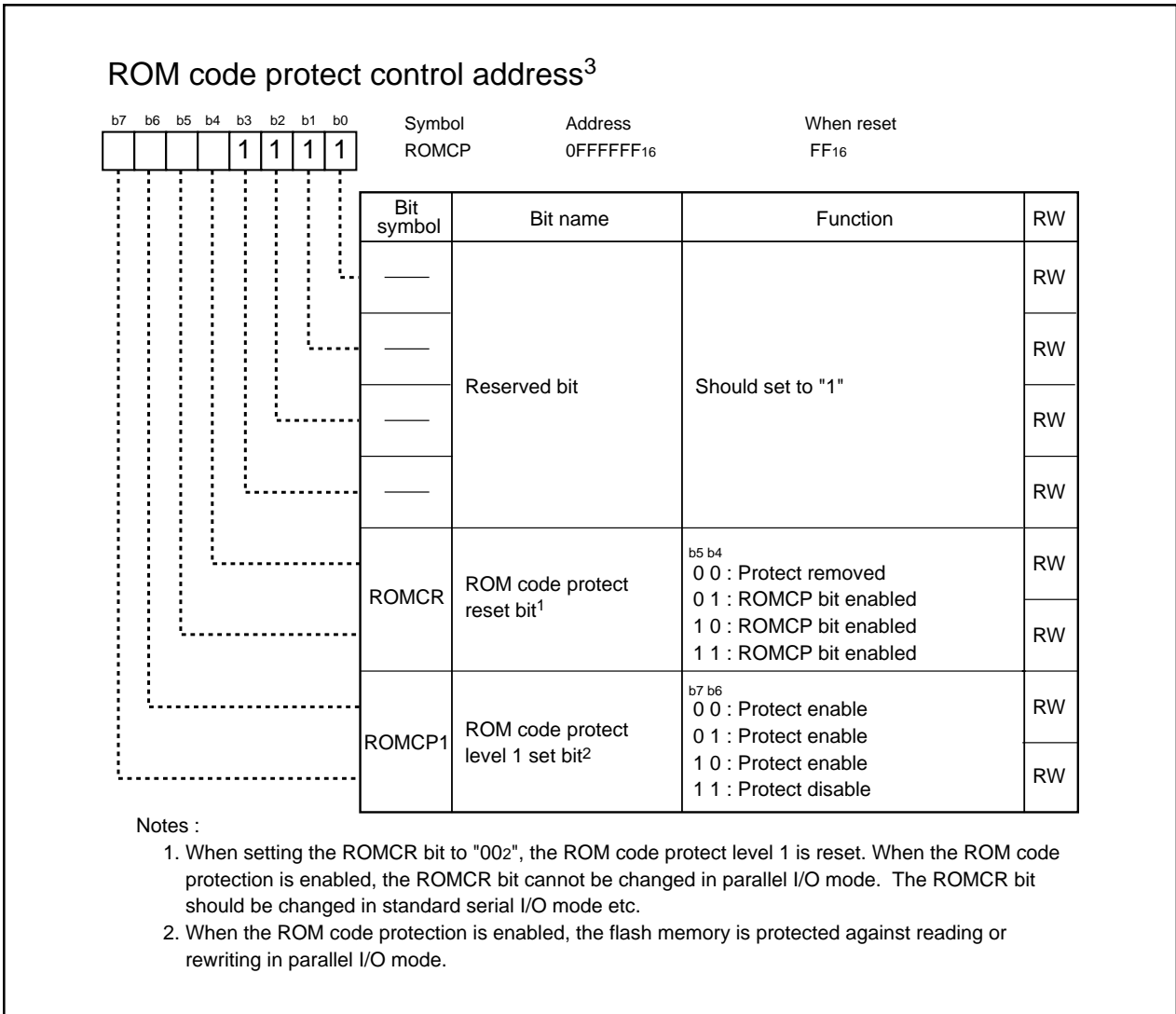
The ROM code protect function prevents the flash memory from reading and rewriting in parallel I/O mode. Figure 1.30.2 shows the ROMCP register. The ROMCP register is located on the user ROM space.

The ROMCP1 bit consists of two bits. When setting one of 2 bits or both bits to "0" by program, the ROM code protect function is enabled to prevent the flash memory from reading and rewriting .

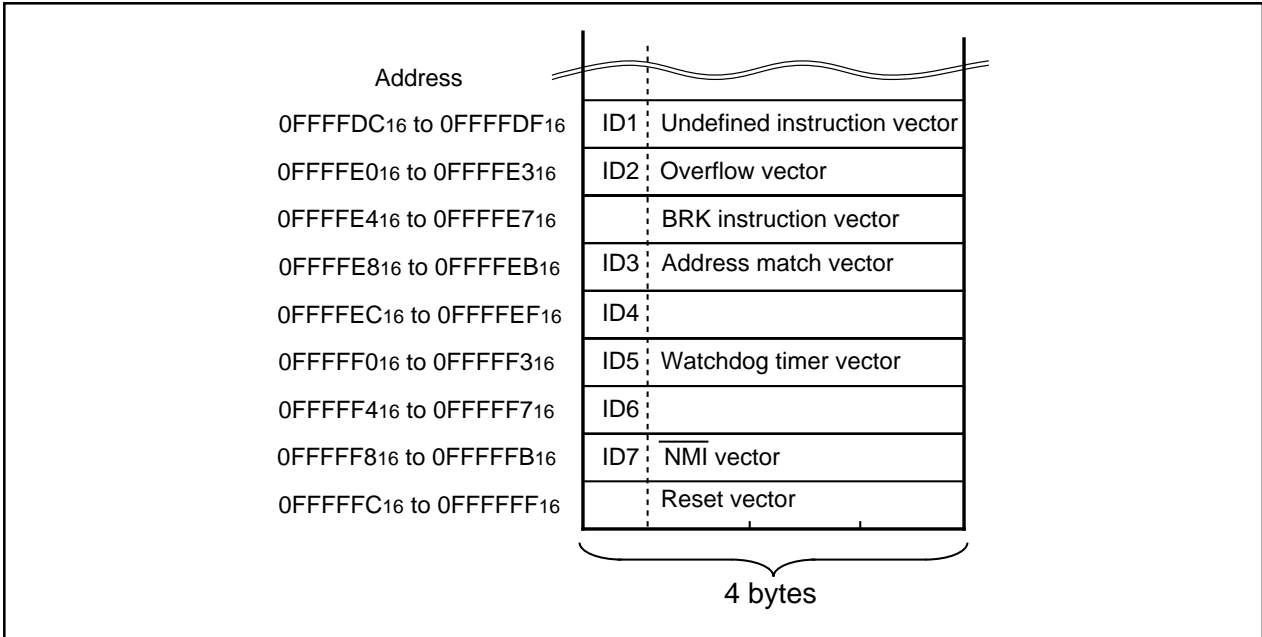
When setting the ROMCR bits to "002" (ROM code protection is removed), the flash memory can be read or rewritten. Once the ROM code protect function is enabled, the ROMCR bits cannot be changed in parallel I/O mode. The ROMCR bit should be rewritten in standard serial I/O mode or other modes.

### • ID Code Check Function

The ID code check function is available in standard serial I/O mode. When the flash memory is not in a blank, the ID code check function determines whether an ID code sent from external devices matches an ID code in the flash memory. If the ID codes does not match, commands from external devices are not accepted. ID code as a 8-bit data is stored into addresses 0FFFFDF<sub>16</sub>, 0FFFFE3<sub>16</sub>, 0FFFFEB<sub>16</sub>, 0FFFFEF<sub>16</sub>, 0FFFFF3<sub>16</sub>, 0FFFFF7<sub>16</sub> and 0FFFFFB<sub>16</sub>. A program with ID code set in these addresses should be written to the flash memory.



**Figure 1.30.2. ROMCP Register**



**Figure 1.30.3. Address for ID code stored**

## Flash Memory Version

**CPU Rewrite Mode**

In CPU rewrite mode, the user ROM space can be rewritten when the CPU executes the software commands. Without a ROM writer or the like, the user ROM space can be rewritten with the microcomputer mounted on board.

A rewrite control program should be rewritten to the user ROM space or the boot ROM space beforehand. Program on the flash memory is not executed in CPU rewrite mode. After transferring the control program to another space (internal RAM, etc) except the flash memory, it should be executed in the direction space. CPU rewrite mode can be entered when the microcomputer is in single-chip mode, memory expansion mode and boot mode.

In CPU rewrite mode, the software command, shown on Table 1.30.3, can be used. Refer to the paragraph "5. Software command" about detail of each command.

Commands and data should be read from or written to even addresses in the user ROM space by 16 bits.

When writing a command code, 8 high-order bits (D15 to D8) are ignored.

**Table 1.30.3. Software commands**

Software command	First bus cycle			Second bus cycle			Third bus cycle		
	Mode	Address	Data (D15 to D0)	Mode	Address	Data (D15 to D0)	Mode	Address	Data (D15 to D0)
Read array	Write	X	xxFF <sub>16</sub>						
Read status register	Write	X	xx70 <sub>16</sub>	Read	X	SRD			
Clear status register	Write	X	xx50 <sub>16</sub>						
Page program	Write	X	xx41 <sub>16</sub>	Write	WA	WD	Write	WA+2	WD
Block erase	Write	X	xx20 <sub>16</sub>	Write	BA	xxD0 <sub>16</sub>			
All unlock block erase	Write	X	xxA7 <sub>16</sub>	Write	X	xxD0 <sub>16</sub>			
Lock bit program	Write	X	xx77 <sub>16</sub>	Write	BA	xxD0 <sub>16</sub>			
Read lock bit status	Write	X	xx71 <sub>16</sub>	Read	BA	D <sub>6</sub>			

SRD : Data of the status register (D7 to D0)

WA : Address to be written (increment A<sub>7</sub> to A<sub>0</sub> by 2 from "00<sub>16</sub>" to "FE<sub>16</sub>")

WD : Data to write (16 bits)

BA : Highest-order address of a block (A<sub>0</sub> = 0)

D<sub>6</sub> : Lock bit (D<sub>6</sub> = 1: unlock, D<sub>6</sub> = 0: locked)

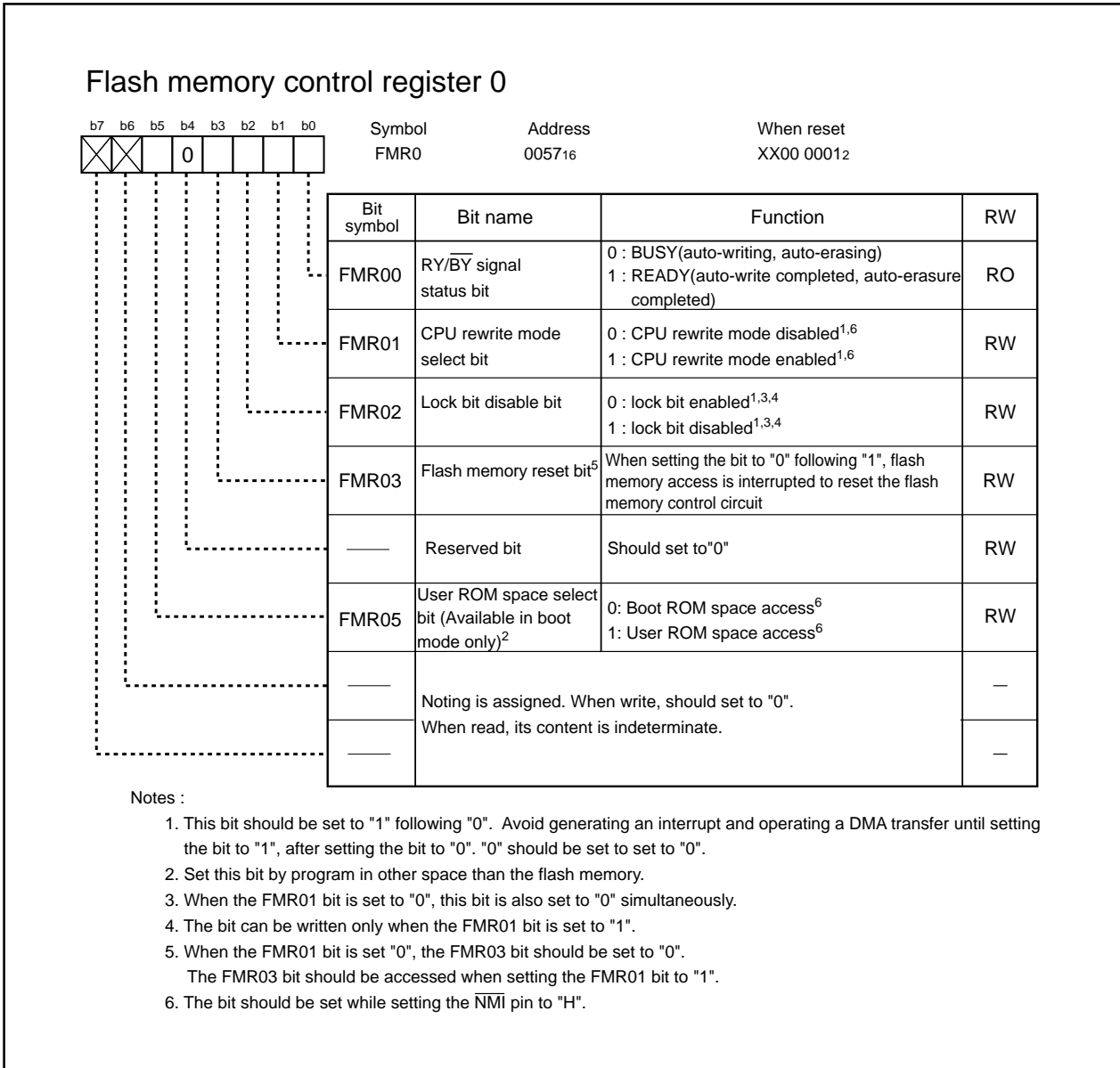
X : Any even address in user ROM space

xx : 8 high-order bits of command code (ignored)

Flash Memory Version

**1. Flash Memory Control Register 0 (FMR0 Register)**

Figure 1.30.4 shows the FMR0 register.



**Figure 1.30.4. FMR0 register**

• **FMR00 Bit**

The FMR00 bit indicates the write status machine (WSM) operation state during auto-write and auto-erasure. The FMR00 bit is set to "0" during auto-write or auto-erasure and set to "1" when auto-write is completed or auto-erasure is completed. The FMR00 bit changes while executing the page program, block erase, all unlock block erase or lock bit program command. The FMR00 bit indicates whether auto-write or auto-erasure is completed. The FMR00 bit is changed by the above commands only.

**• FMR01 Bit**

Commands can be accepted when setting the FMR01 bit to "1" (CPU rewrite mode). When setting the FMR01 bit to "1", it should be set to "1" following "0". When setting the FMR01 bit to "0", it should be set to "0".

CPU rewrite mode is entered by setting the FMR01 bit to "1" and programs on the flash memory cannot be executed. An instruction written to this bit should be executed on another space (internal RAM, etc) except the flash memory.

If executing a command for CPU rewrite mode in boot mode, the FMR05 bit should be set to "1" (user ROM space access).

**• FMR02 Bit**

The lock bit set for each block can be disabled when setting the FMR02 bit to "1". (Refer to the paragraph "3. Data protect function".) The lock bit is enabled when setting the FMR02 bit to "0". When setting the FMR01 bit to "1", the FMR02 bit can be set. When setting the FMR02 bit to "1", it should be set to "1" following "0". When setting the FMR02 bit to "0", it should be set to "0".

The FMR02 bit does not change a lock bit state but disables a lock bit function. When executing the block erase command or all unlock block erase command with setting the FMR02 bit to "1", a lock bit state changes "0" (locked) to "1" (unlocked) after the command is completed.

**• FMR03 Bit**

When setting the FMR03 bit to "0" following "1", access to user ROM space is interrupted to reset the flash memory control circuit. The flash memory enters read array mode after reset. The FMR00 bit is set to "1" (READY) and the SRD register is set to "8016". (Refer to the paragraph "2. Status register".) When the FMR03 bit resets the flash memory control circuit during auto-write or auto-erasure, auto-write or auto-erasure is interrupted. Data in the block is disabled.

When setting the FMR03 bit to "0", it should be set to "0" following "1".

**• FMR05 Bit**

The FMR05 bit switches from the boot ROM space to the user ROM space in boot mode. This bit should be set to "0" to access (read) the boot ROM space or "1" (user ROM space access) to access (read, write or erase) the user ROM space. An instruction written to the FMR05 bit should be executed in another space (internal RAM, etc) except the flash memory.

The user ROM space is accessed (read), regardless of the FMR05 bit, in other modes except boot mode.

## 2. Status Register (SRD register)

The write state machine (WSM) in the flash memory controls programming and erasing of the flash memory. The SRD register indicates whether WSM operates normally and whether program and erasure are completed properly or not. Refer to the paragraph "6. Full status check" about each error.

Table 1.30.4 lists the SRD register.

The SRD register can be read by the read status command (Refer to the paragraph "5. Software command").

**Table 1.30.4. SRD register**

Symbol	Status name	Definition	
		0	1
SR0 (D0)	Reserved bit	-	-
SR1 (D1)	Reserved bit	-	-
SR2 (D2)	Reserved bit	-	-
SR3 (D3)	Block status after program	Normally completed	Error (excessive write error)
SR4 (D4)	Program status	Normally completed	Error (program error)
SR5 (D5)	Erasure status	Normally completed	Error (erase error)
SR6 (D6)	Reserved bit	-	-
SR7 (D7)	Write state machine (WSM) status	BUSY	READY

D7 to D0 : These data bus are read when executing a read status register command.

### • Block Status after Program (SR3)

When the page program command is completed with an excessive write error, the SR3 bit is set to "1". When executing the clear status command, the SR3 bit is set to "0". After reset or after setting the FMR03 bit to "0" following "1", the SR3 bit is set to "0".

### • Program Status (SR4)

When executing the page program command or lock bit program command with a programming error, the SR4 bit is set to "1". When executing the clear status command, the SR4 bit is set to "0". After reset or after setting the FMR03 bit to "0" following "1", the SR4 bit is set to "0".

### • Erasure Status (SR5)

When executing the block erase command or all unlock block erase command with an erasure error, the SR5 bit is set to "1". When executing the clear status command, the SR5 bit is set to "0". After reset or after setting the FMR03 bit to "0" following "1", the SR5 bit is set to "0".

### • Write State Machine (WSM) Status (SR7)

The SR7 bit indicates WSM operation status. The SR7 bit is set to "0" during auto-write or auto-erasure and to "1" while auto-write or auto-erasure is completed. The SR7 bit changes while executing the page program, block erase, all unlock block erase or lock bit program command. The SR7 bit changes with the above commands only. After reset or after setting the FMR03 bit to "0" following "1", the SR7 bit is set to "0".

The FMR00 bit indicates WSM status. The FMR00 bit indicates whether auto-write or auto-erasure is completed.

### 3. Data Protect Function

Each block in the flash memory has a nonvolatile lock bit. The lock bit is enabled when the FMR02 bit is set to "0" (lock bit enabled). Each lock bit can determine whether each block is allowed to program and erase. This helps to prevent data from programming or erasing accidentally or mistakenly. Block status with the lock bit is as follows.

- When setting the lock bit to "0", a block is locked (Disable to program and erase)
- When setting the lock bit to "1", a block is not locked (Enable to program and erase)

The lock bit is set to "0" (locked) when executing the lock bit program command. The lock bit is set to "1" (unlocked) when erasing a block. The lock bit cannot be set to "1" by command.

Lock bit status can be read by the read lock bit status command.

Function of the lock bit is disabled when setting the FMR02 bit to "1" to have all blocks unlocked. (Each lock bit does not change.) Function of the lock bit is enabled when setting the FMR02 bit to "0". (Lock bit is remained.)

When executing the block erase command or all unlock block erase command with setting the FMR02 bit to "1", a target block or all blocks are erased regardless of a lock bit status. The lock bit for each block is set to "1" after the command is completed.

Refer to the paragraph "5. Software command" about each command.

## Flash Memory Version

#### 4. How to Enter and Exit CPU Rewrite Mode

Figure 1.30.5 shows how to enter and exit CPU rewrite mode.

A program on the flash memory cannot be executed in CPU rewrite mode. A rewrite control program should be executed on another space except the flash memory (internal RAM, etc) after transferring the program to that space.

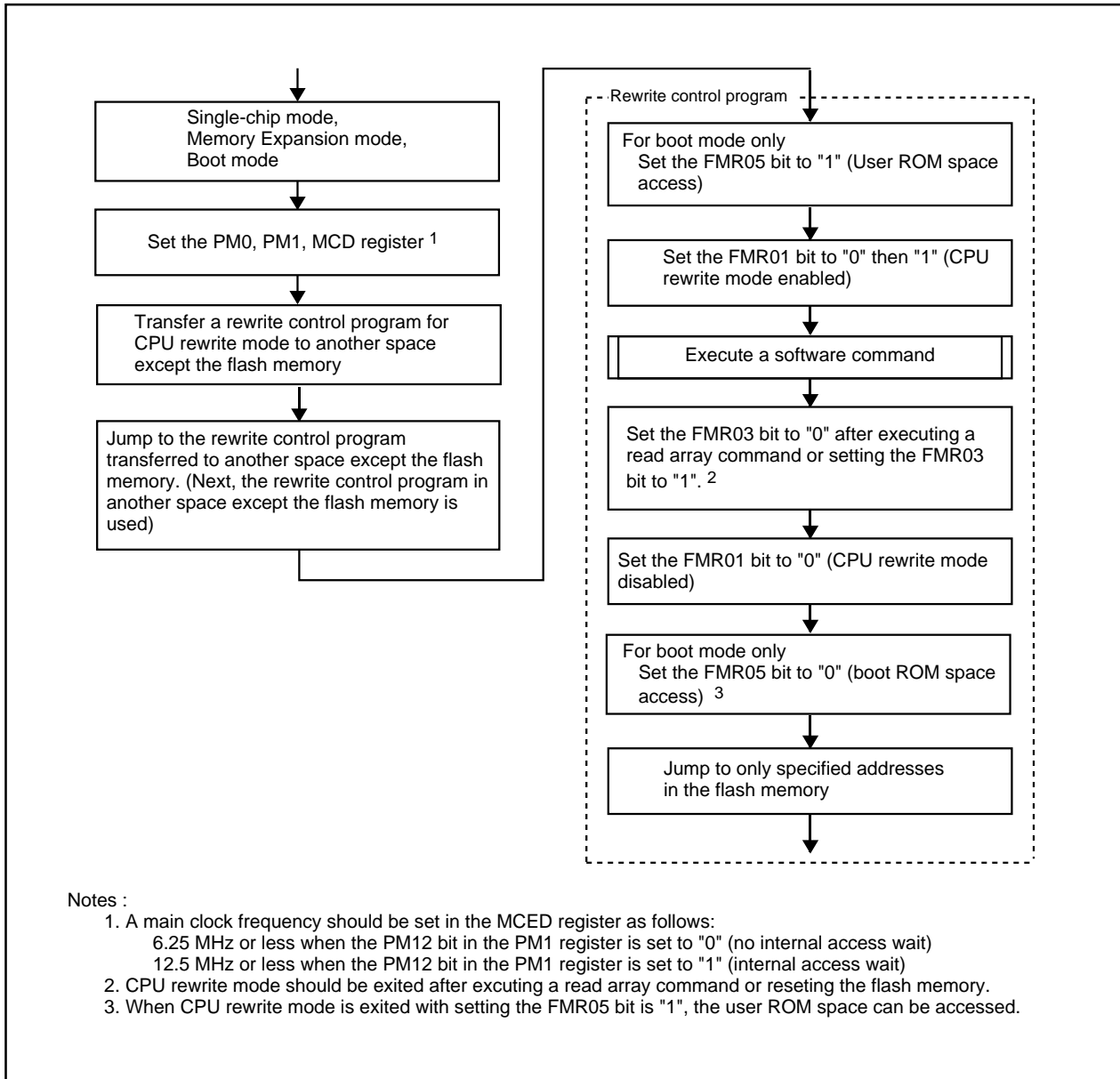


Figure 1.30.5. Setting and reset CPU rewrite mode

## 5. Software Commands

Data and the software command should be read from or write to even addresses in the user ROM space by 16 bits. While writing a command code, 8 high-order bits (D15 to D8) are disabled.

### • Read Array

The read array command reads the flash memory.

Command code "xxFF16" should be written in the first bus cycle to enter read array mode. Content of a specified address can be read after the next bus cycle.

The microcomputer remains unchanged in read array mode until another command is written.

### • Read Status Register

The read status register command reads the SRD register.

Command code "xx7016" should be written in the first bus cycle to read the SRD register in the second bus cycle. (See Table 1.30.4.) An even address in the user ROM space should be read.

### • Clear Status Register

The clear status register command sets the SRD register to "0".

Command code "xx5016" should be written in the first bus cycle to set the SR3 to SR5 bits in the SRD register to "0" (see Table 1.30.4).

### • Page Program

The page program command executes a program by 128 words (256 bytes).

Command code "xx4116" should be written in the first bus cycle and data by 16 bits be written between the 2nd bus cycle and 129th bus cycle. 8 low-order bits (A7 to A0) of an address to be written should be incremented by 2 from "0016" to "FF16".

Auto-write starts, or data is programmed and verified, after writing 128 words data. Avoid accessing the flash memory or executing the next command while auto-write is in progress.

The FMR00 bit in the FMR0 register determines whether auto-write is completed.

The SRD register indicates an auto-write log after auto-write is completed. (Refer to the paragraph "6. Full Status Check".)

Figure 1.30.6 shows a flow chart of the page programming. Programming should be started after erasing a page already programmed (erasing in a block). If programmed to a page already programmed, no program error occurs but the page is indeterminate.

The lock bit can prevent each block from programming. (Refer to the paragraph "3. Data Protect Function".)

Flash Memory Version

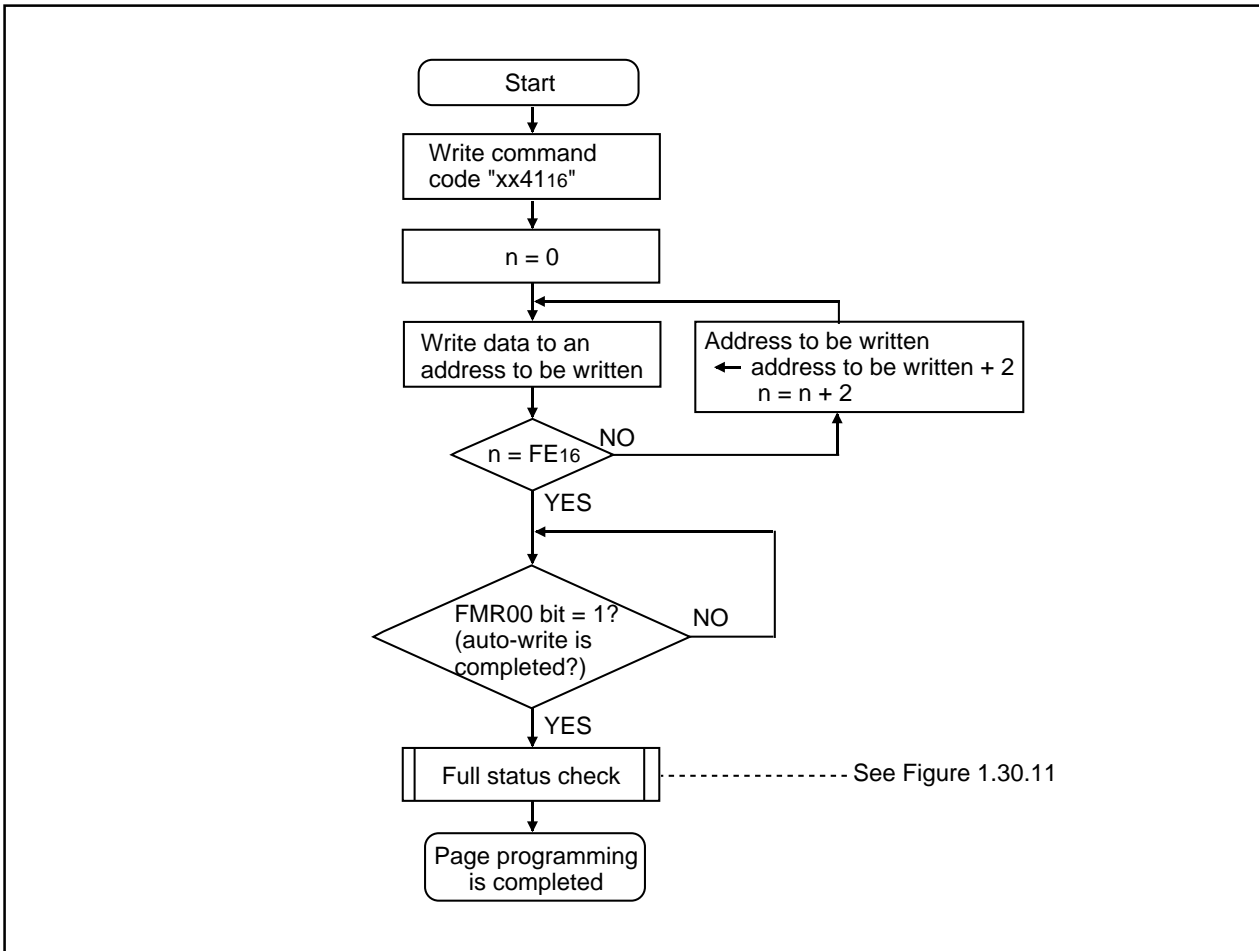


Figure 1.30.6. Page Program Command

## Flash Memory Version

**• Block Erase**

The block erase command erases blocks by block.

Command code "xx20<sub>16</sub>" should be written in the first bus cycle and "xxD0<sub>16</sub>" be written to highest-order address in a block erased ( $A_0 = 0$ ) in the second bus cycle to erase a specified block automatically (erase and verify). Avoid accessing the flash memory or executing the next command while auto-erasure is in progress.

The FMR00 bit in the FMR0 register determines whether auto-erasure is completed.

The SDR register indicates an auto-erasure log after auto-erasure is completed. (Refer to the paragraph "6. Full Status Check".)

Figure 1.30.7 shows a flow chart of the block erasure.

The lock bit can prevent each block from erasing. (See "3. Data Protect Function".)

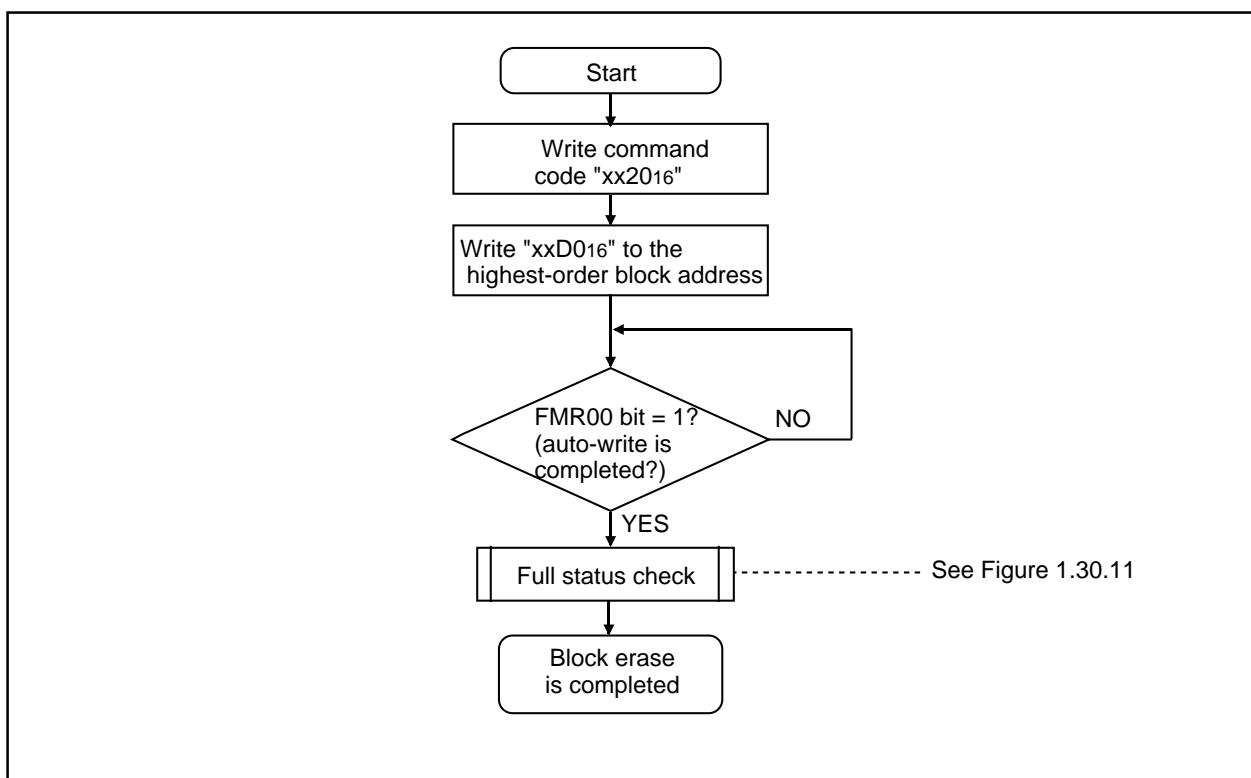


Figure 1.30.7. Block Erase command

Flash Memory Version

• **All Unlock Block Erase**

The all unlock block erase command erases all blocks.

Command code "xxA716" should be written in the first bus cycle and "xxD016" be written in the second bus cycle to start erasing all blocks automatically (erase and verify). Avoid accessing the flash memory or executing the next command while auto-erasure is in progress.

The FMR00 bit in the FMR0 register determines whether auto-erasure is completed.

The SDR register indicates an auto-erasure log after auto-erasure is completed. (Refer to the paragraph "6. Full Status Check".)

Figure 1.30.8 shows a flow chart of the all unlock block erase.

The lock bit can prevent each block from erasing. (Refer to the paragraph "3.Data Protect Function".)

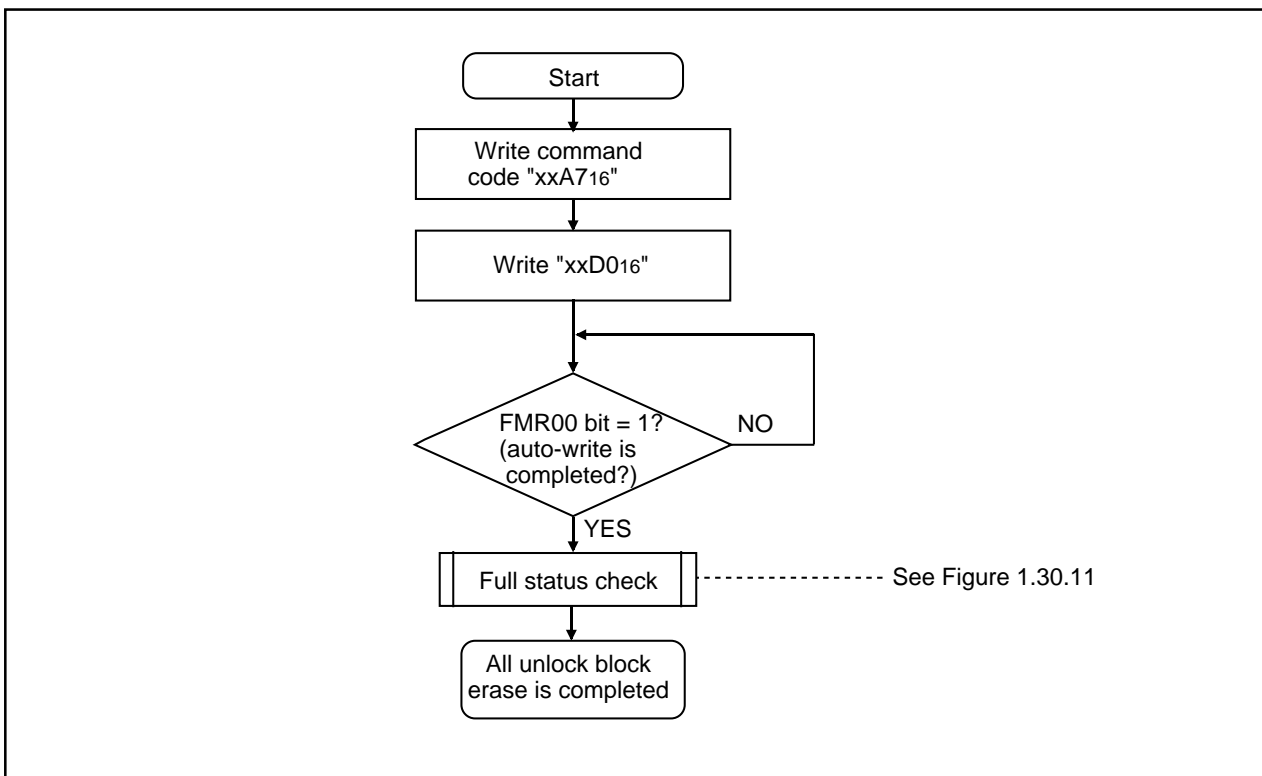


Figure 1.30.8. All Unlock Block Erase Command

## Flash Memory Version

**• Lock Bit Program**

The lock bit program command sets the lock bit in a specified block to "0" (locked).

Command code "xx7716" should be written in the first bus cycle and "xxD016" be written to the highest-order address of a block ( $A_0 = 0$ ) in the second bus cycle to set the lock bit in the specified block to "0".

Avoid accessing the flash memory or executing the next command while auto-erase is in progress.

The FMR00 bit in the FMR0 register determines whether auto-erase is completed.

The SDR register indicates auto-erase log after auto-erase is completed. (Refer to the paragraph "6. Full Status Check".)

Figure 1.30.9 shows a flow chart of the lock bit program.

Refer to the paragraph "3. Data Protect Function" about how to set the lock bit to "0" and how to function the lock bit.

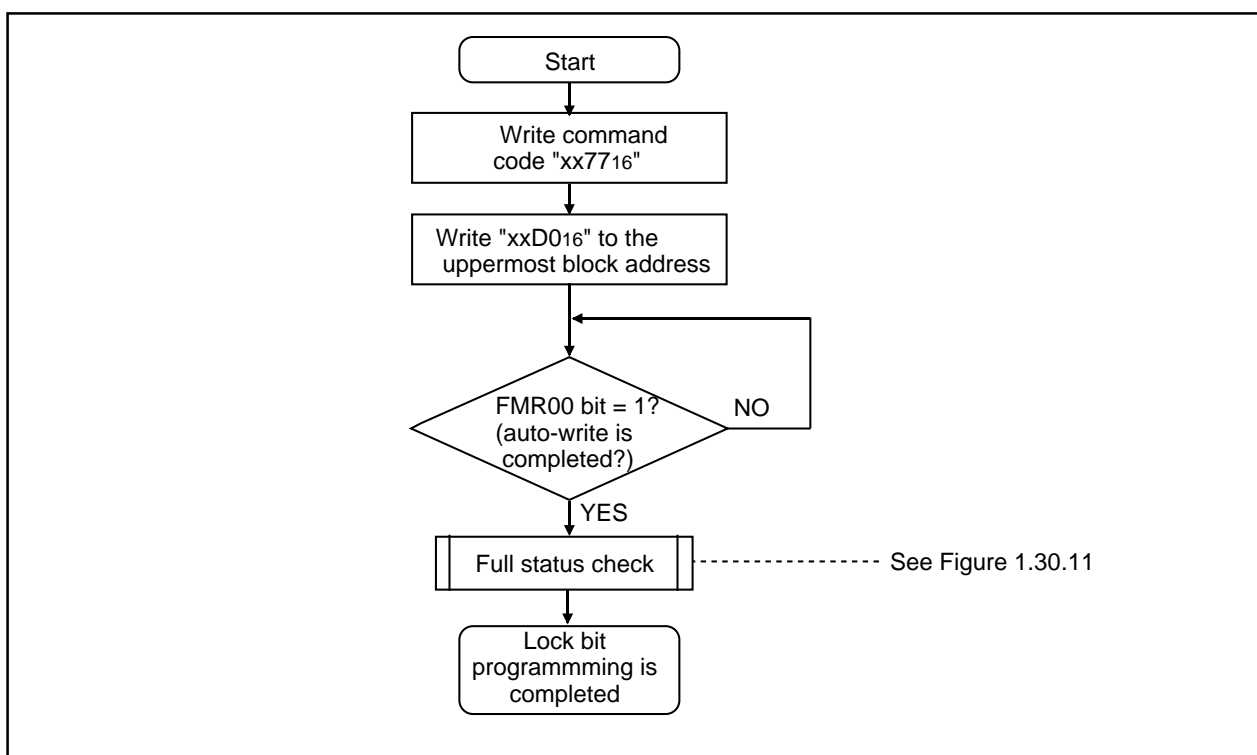


Figure 1.30.9. Lock Bit Program Command

Flash Memory Version

• **Lock Bit Read Status**

The lock bit read status command reads a lock bit status in a specified block.

Command code "xx7116" should be written in the first bus cycle and "xxD016" be written to the highest-order address of a block ( $A_0 = 0$ ) in the second bus cycle to read a lock bit status in the specified block out to data bus (D6).

Figure 1.30.10 shows a flow chart of the lock bit read program.

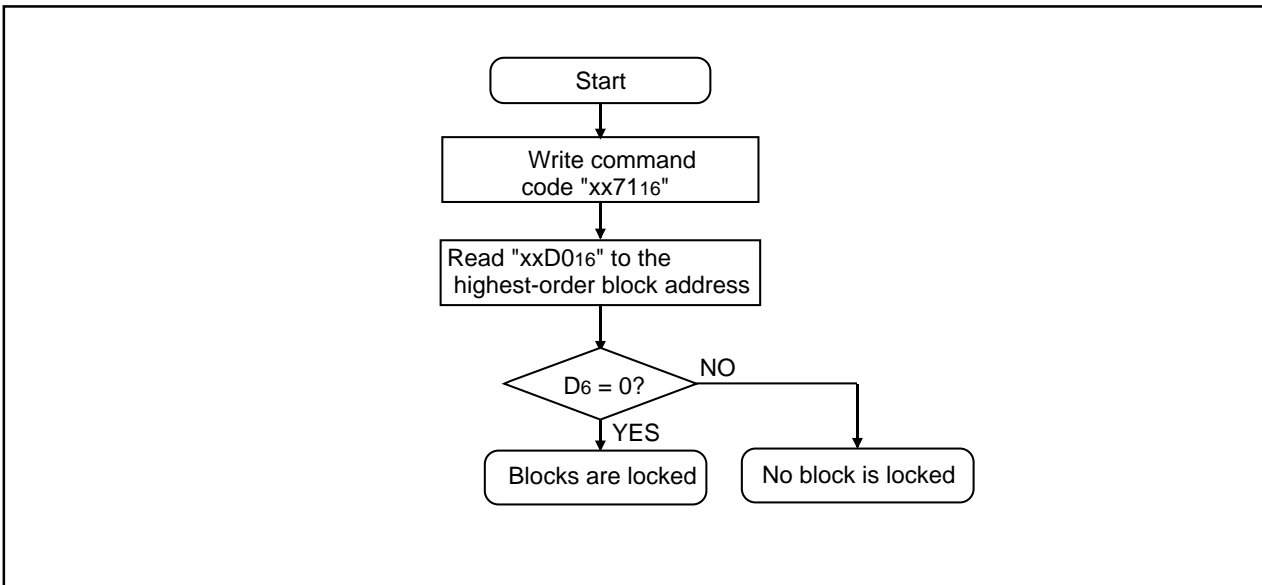


Figure 1.30.10. Lock Bit Read Status Command

## 6. Full Status Check

If an error occurs when a programming or erasure is completed, the SR3 to SR5 bits in the SRD register are set to "1" to indicate each error. When a programming or erasure is completed, a programming or erasure log can be verified by checking these status (full status check).

Table 1.30.5 lists errors and the SRD register (SR3 to SR5). Figure 1.30.11 shows a flow chart of the full status check and how to handle each error.

**Table 1.30.5. Error and SRD Register**

SRD register			Error	Error occurs
SR5	SR4	SR3		
1	1	0	Command sequence error	<ul style="list-style-type: none"> <li>When writing an incorrect command</li> <li>When writing data without valid values ("xxD016" or "xxFF16") in the second bus cycle of the lock bit program, block erase or all unlock block erase command<sup>1</sup></li> </ul>
1	0	0	Erase error	<ul style="list-style-type: none"> <li>When executing the block erase command on a locked block<sup>2</sup></li> <li>When executing the block erase or the all unlock block erase commands on an unlock block and erasing a block incorrectly</li> </ul>
0	1	0	Program error	<ul style="list-style-type: none"> <li>When executing the page program command on a page in a locked block<sup>2</sup></li> <li>When executing the page program command on a page in an unlocked block and programming incorrectly</li> <li>When executing the lock bit program command and programming incorrectly</li> </ul>
0	0	1	Excessive write error	When writing excessively after the page program command is executed

Notes :

1. When writing command code "xxFF16" in the second bus cycle of these commands, the microcomputer enters read array mode. A command code written in the first bus cycle is disabled.
2. When setting the FMR02 bit to "1" (lock bit disabled), no error occurs under this conditions.

Flash Memory Version

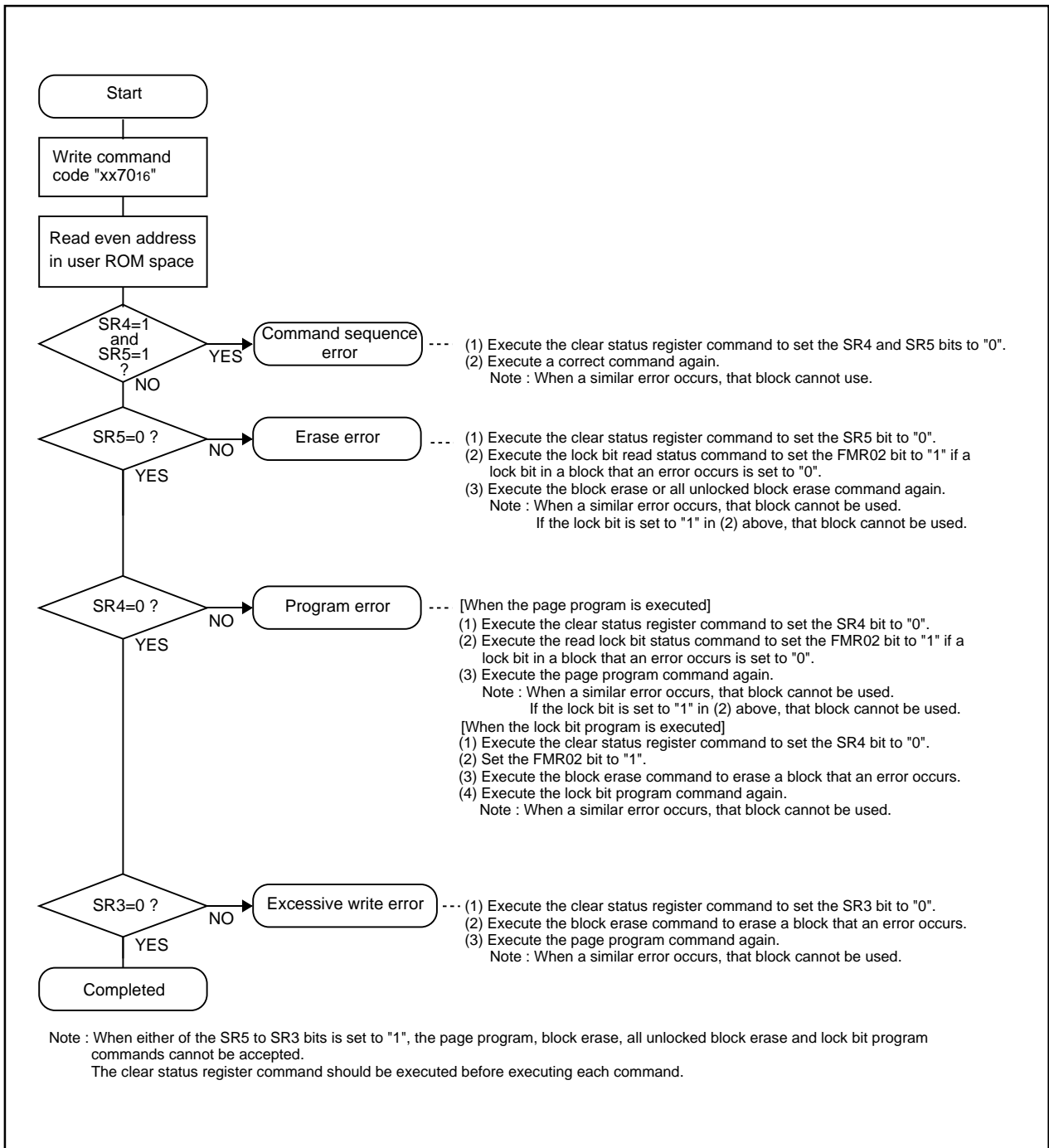


Figure 1.30.11. Full Status Check and Handling Procedure for Each Errors

## 7. Precautions in CPU Rewrite Mode

### • Operating Speed

Before entering CPU rewrite mode, main clock frequency that the MCD register determines should be following:

6.25 MHz or less when the PM12 bit in the PM1 register is set to "0" (no internal access wait)

12.5 MHz or less when the PM12 bit in the PM1 register is set to "1" (internal access wait)

### • Unusable Instructions

In CPU rewrite mode, a program on the flash memory cannot be executed and also interrupt vectors on the flash memory cannot be read. The rewrite control program should be executed after transferring the program to another space except the flash memory. (See Figure 1.30.5.)

The UND instruction, INTO instruction, JMPS instruction, JSRS instruction and BRK instruction, which access the flash memory, cannot be used.

### • Interrupts

(1) Any interrupts which has a vector in a relocatable vector table can be used by transferring vectors into the RAM space.

(2) The watchdog timer and  $\overline{\text{NMI}}$  interrupts can be used since the FMR01 bit is forced to change to "0" (CPU rewrite mode disabled) when the interrupts are generated. It is required that a destination address for the interrupt is set in a fixed vector table and that an interrupt program is available. The rewrite operation is terminated when the  $\overline{\text{NMI}}$  or watchdog timer interrupt is generated. Rewrite the FMR01 bit again after an interrupt service routine is completed.

(3) The address match interrupt, which access internal data in the flash memory, cannot be used.

### • Reading and Writing Commands and Data

Commands and data should be read from or written to even addresses in the user ROM space by 16 bits.

### • Reset

Reset is always enabled.

### • Access Disable

The FMR01 bit and FMR05 bit should be written in another space except the flash memory.

### • How to Access

The FMR01 bit and FMR02 bit should be set to "1" following "0" if setting these bits to "1". Avoid an interrupt and DMA transfer before setting these bits to "1" following "0". The FMR01 bit should be set to "1" after "H" input to the P85/ $\overline{\text{NMI}}$  pin if setting the bit to "1".

### • Rewrite the User ROM Space

When a supply voltage drops while rewriting a block which stores a rewrite control program in CPU rewrite mode, the flash memory rewrite control program cannot be rewritten properly. The flash memory may be rewritten incorrectly after that.

The user ROM space should be rewritten in standard serial I/O mode or parallel I/O mode.

## Flash Memory Version

---

### Standard Serial I/O Mode

In standard serial I/O mode, the user ROM space can be rewritten by a serial writer for the M32C/83 group with the microcomputer mounted on board. For more information about a serial writer, contact your serial writer manufacturer. Refer to the user's manual included with your serial writer about how to use.

Standard serial I/O mode has the following modes.

- Standard serial I/O mode 1 (clock synchronous)
- Standard serial I/O mode 2 (clock asynchronous)

#### 1. Pin Function

Table 1.30.6 lists pin functions (flash memory standard serial I/O mode). Figures 1.30.12 to 1.30.14 show pin connections in standard serial I/O mode.

#### 2. ID Code Check Function

The ID code check function is used in standard serial I/O mode. When the flash memory is not in a blank, the ID code check function determines whether an ID code sent from external devices matches an ID code written to the flash memory. If both ID codes do not match, commands from external devices are not accepted. ID code as a 8-bit data is stored into addresses 0FFFFDF<sub>16</sub>, 0FFFFE3<sub>16</sub>, 0FFFFEB<sub>16</sub>, 0FFFFEF<sub>16</sub>, 0FFFFF3<sub>16</sub>, 0FFFFF7<sub>16</sub> and 0FFFFFB<sub>16</sub>. A program set with ID code in these addresses should be written to the flash memory.

## Flash Memory Version

Table 1.30.6. Pin Functions (Flash memory standard serial I/O mode)

Pin	Name	I/O	Description
Vcc Vss	Power input	I	Apply 4.2V to 5.5V to Vcc pin Apply 0 V to Vss pin
CNVss	Mode select CNVss	I	Connect CNVss to Vcc pin
RESET	Reset input	I	Reset input pin. While $\overline{\text{RESET}}$ pin is set to "L", input 20 cycle or longer clock to XIN pin
XIN XOUT	Clock input Clock output	I O	Connect a ceramic resonator or quartz oscillator between XIN and XOUT pins. To input an externally generated clock, input it to XIN pin. Leave XOUT pin open.
BYTE	BYTE input	I	Connect BYTE to Vcc or Vss
AVcc AVss	Analog power supply input	I	Connect AVss to Vss Connect AVcc to Vcc
VREF	Reference voltage input	I	Reference voltage pin for A-D converter
P00 to P07	Input port P0	I	Input "H" or "L" or open
P10 to P17	Input port P1	I	Input "H" or "L" or open
P20 to P27	Input port P2	I	Input "H" or "L" or open
P30 to P37	Input port P3	I	Input "H" or "L" or open
P40 to P47	Input port P4	I	Input "H" or "L" or open
P50	$\overline{\text{CE}}$ input	I	Input "H"
P55	EPM input	I	Input "L"
P51 to P54, P56, P57	Input port P5	I	Input "H" or "L" or open
P60 to P63	Input port P6	I	Input "H" or "L" or open
P64	BUSY output	O	Standard serial mode 1: BUSY signal output pin Standard serial mode 2: Monitors to check a program operation
P65	SCLK input	I	Standard serial mode 1: Serial clock input pin Standard serial mode 2: Input "L"
P66	Data input RxD	I	Serial data input pin
P67	Data output TxD	O	Serial data output pin <sup>1</sup>
P70 to P77	Input port P7	I	Input "H" or "L" or open
P80 to P84, P86, P87	Input port P8	I	Input "H" or "L" or open
P85	NMI input	I	Connect P85 to Vcc
P90 to P97	Input port P9	I	Input "H" or "L" or open
P100 to P107	Input port P10	I	Input "H" or "L" or open
P110 to P114	Input port P11	I	Input "H" or "L" or open <sup>2</sup>
P120 to P127	Input port P12	I	Input "H" or "L" or open <sup>2</sup>
P130 to P137	Input port P13	I	Input "H" or "L" or open <sup>2</sup>
P140 to P146	Input port P14	I	Input "H" or "L" or open <sup>2</sup>
P150 to P157	Input port P15	I	Input "H" or "L" or open <sup>2</sup>

## Notes :

1. In standard serial I/O mode 1, "L" should be input to the TxD pin while the  $\overline{\text{RESET}}$  pin is set to "L". This pin should be connected to Vss via a resistor. This pin becomes a data output pin after reset. A pull-down resistance value should be adjusted in the system to avoid affecting data transfers.
2. These pins are provided in the 144-pin package.

Flash Memory Version

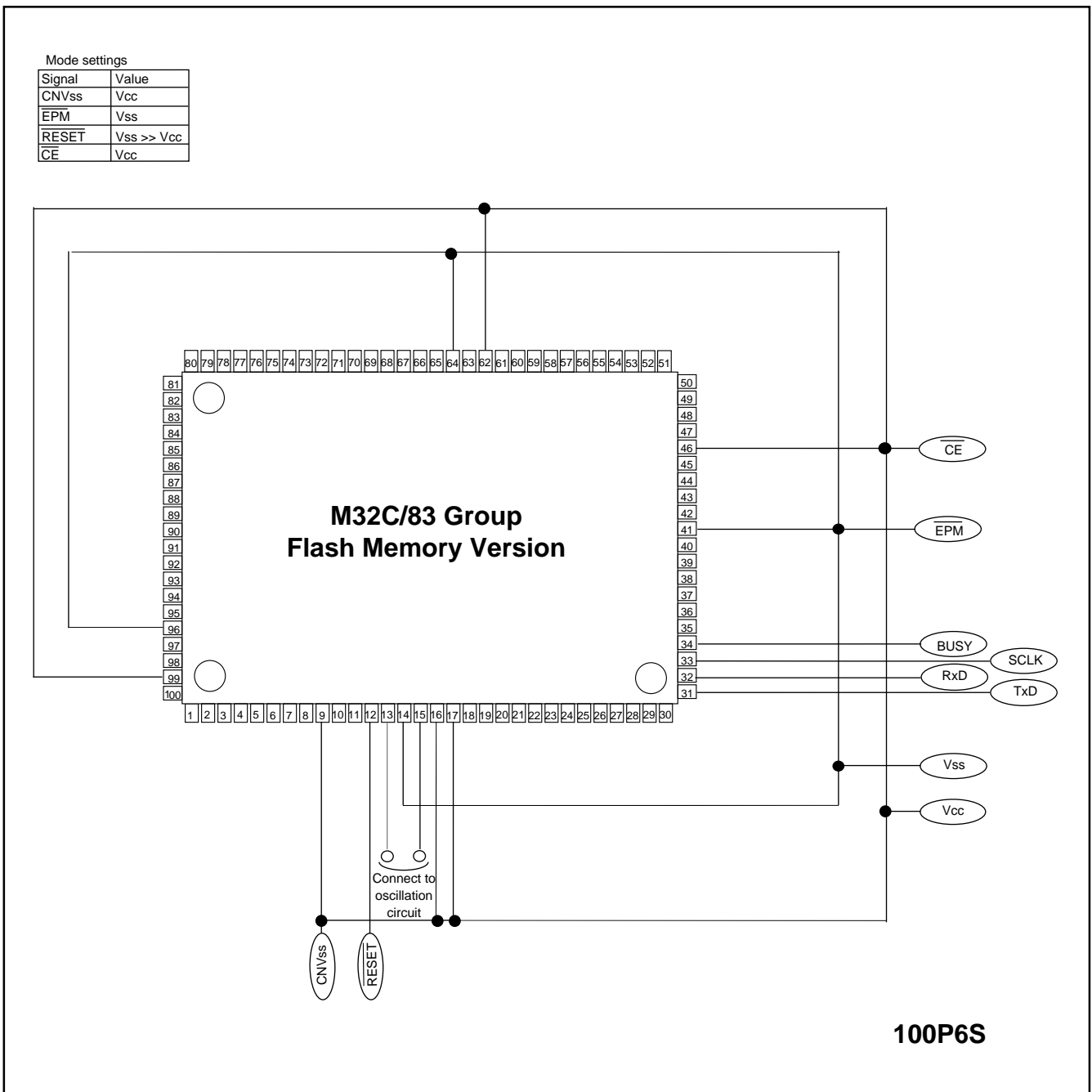


Figure 1.30.12. Pin Connections in Standard Serial I/O Mode (1)

Flash Memory Version

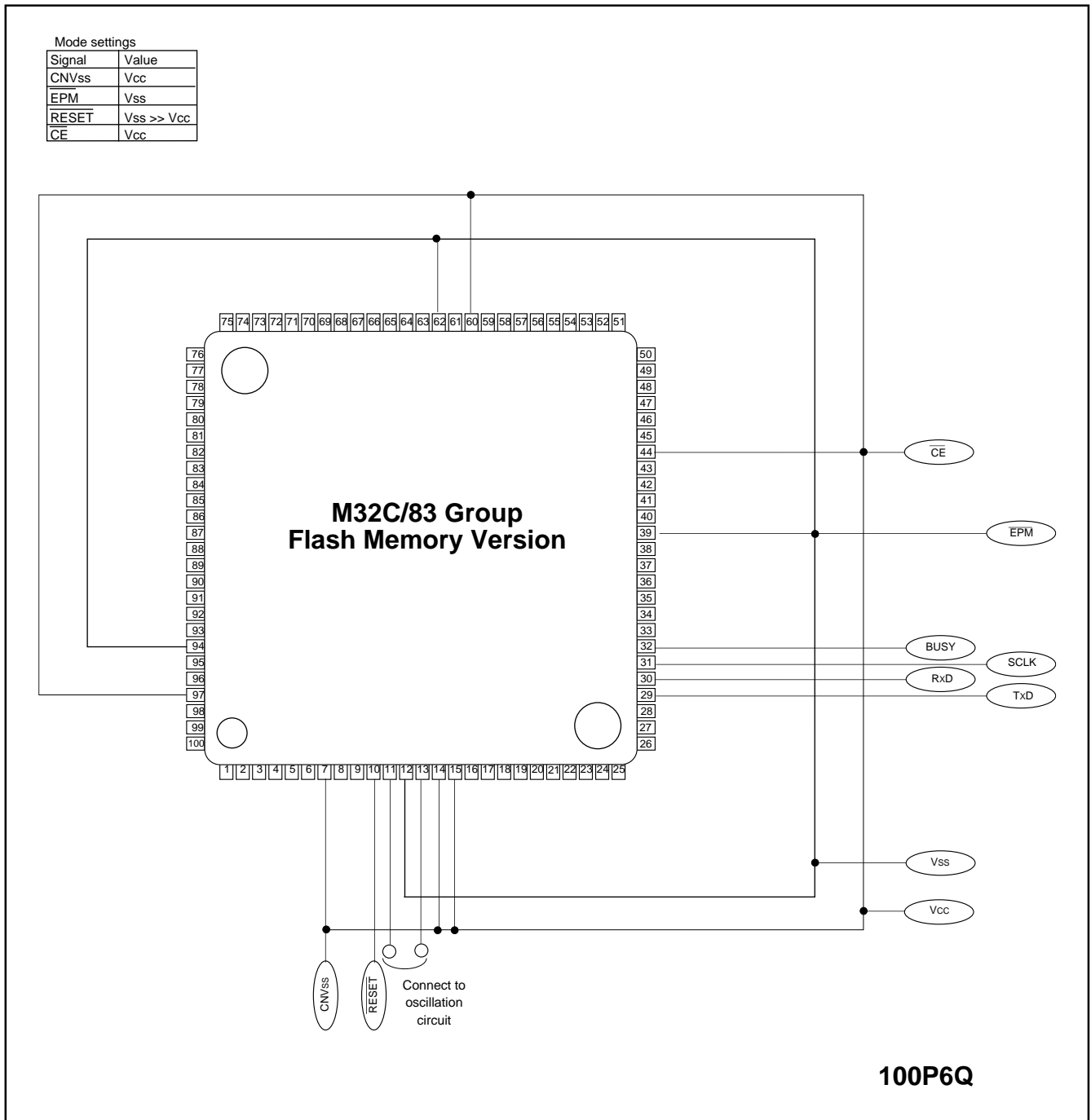


Figure 1.30.13. Pin Connections in Standard Serial I/O Mode (2)

Flash Memory Version

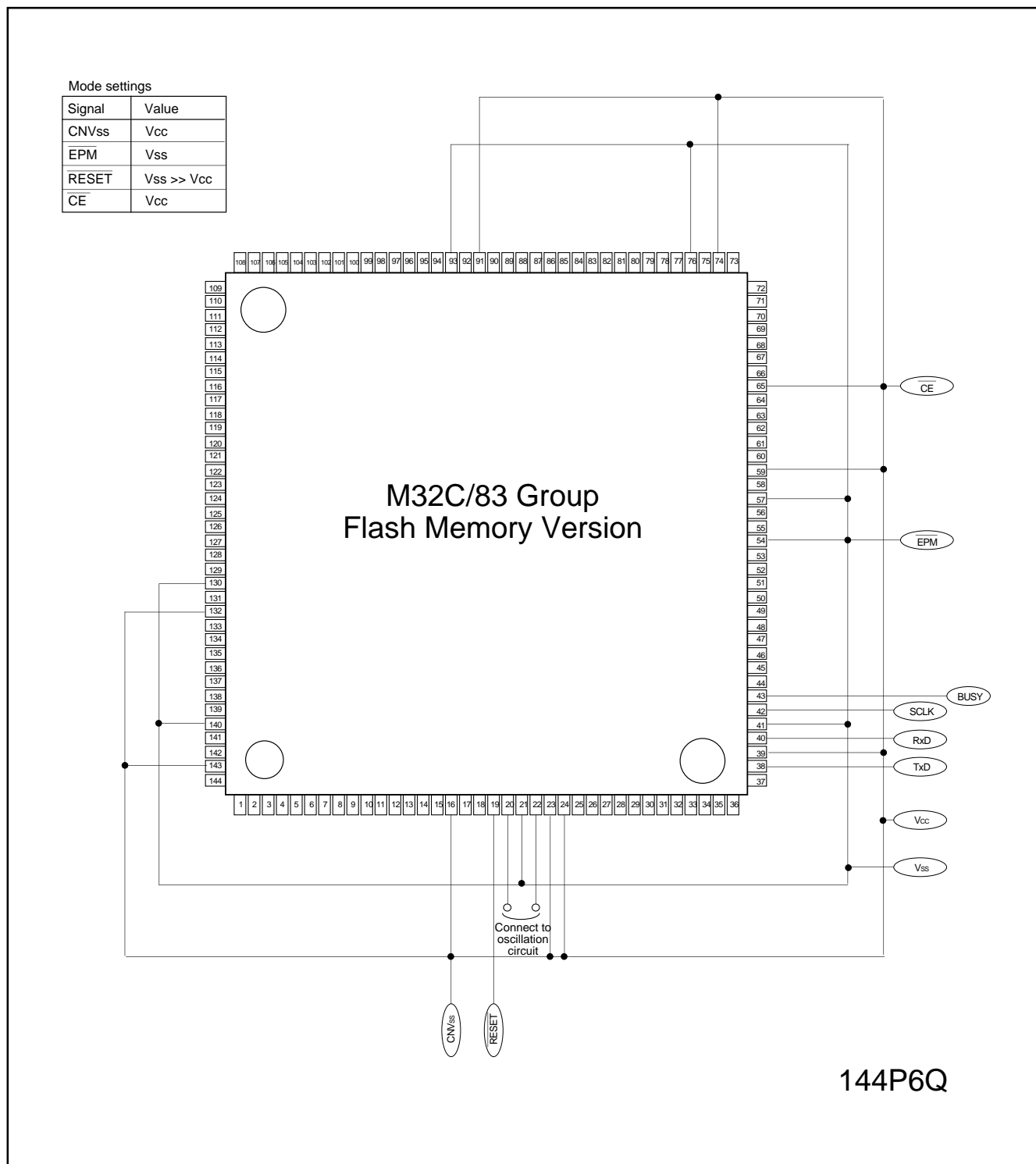


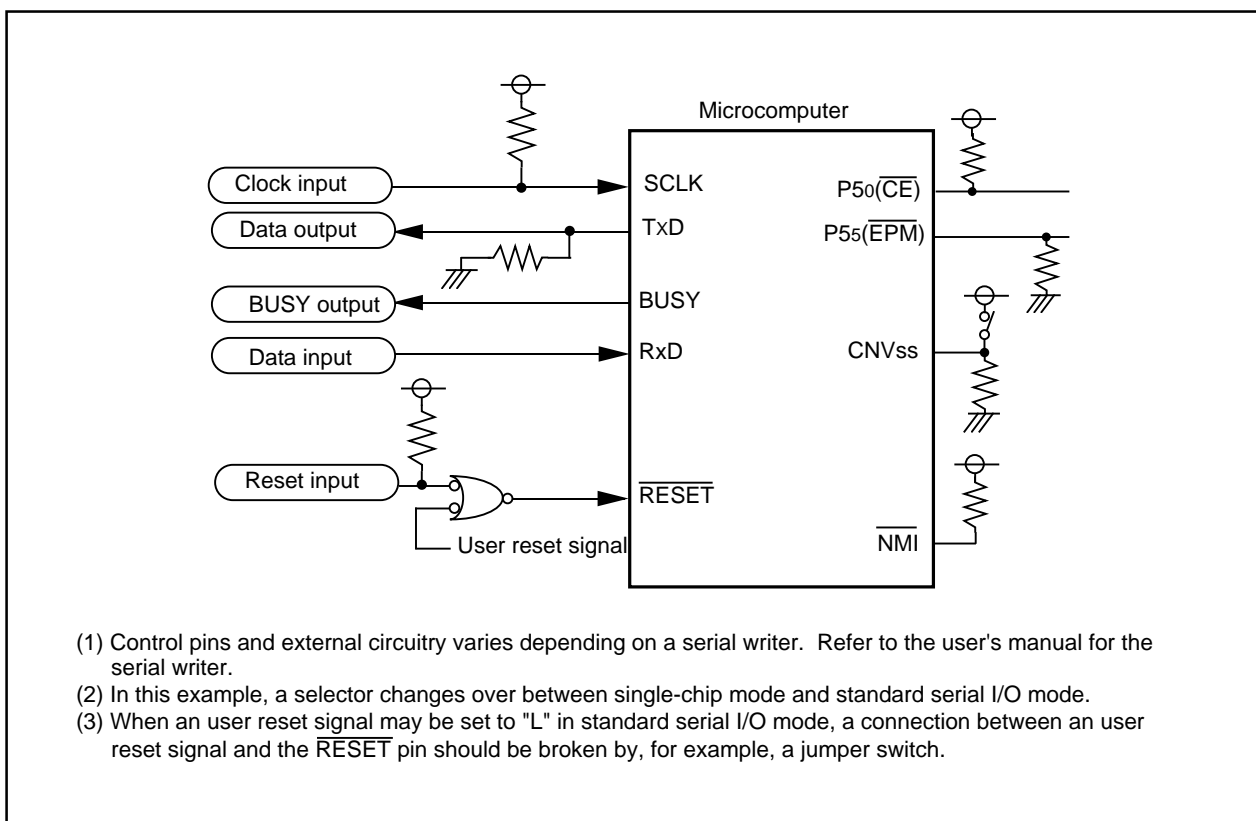
Figure 1.30.14. Pin Connections in Standard Serial I/O Mode (3)

### Precautions in Standard Serial I/O Mode

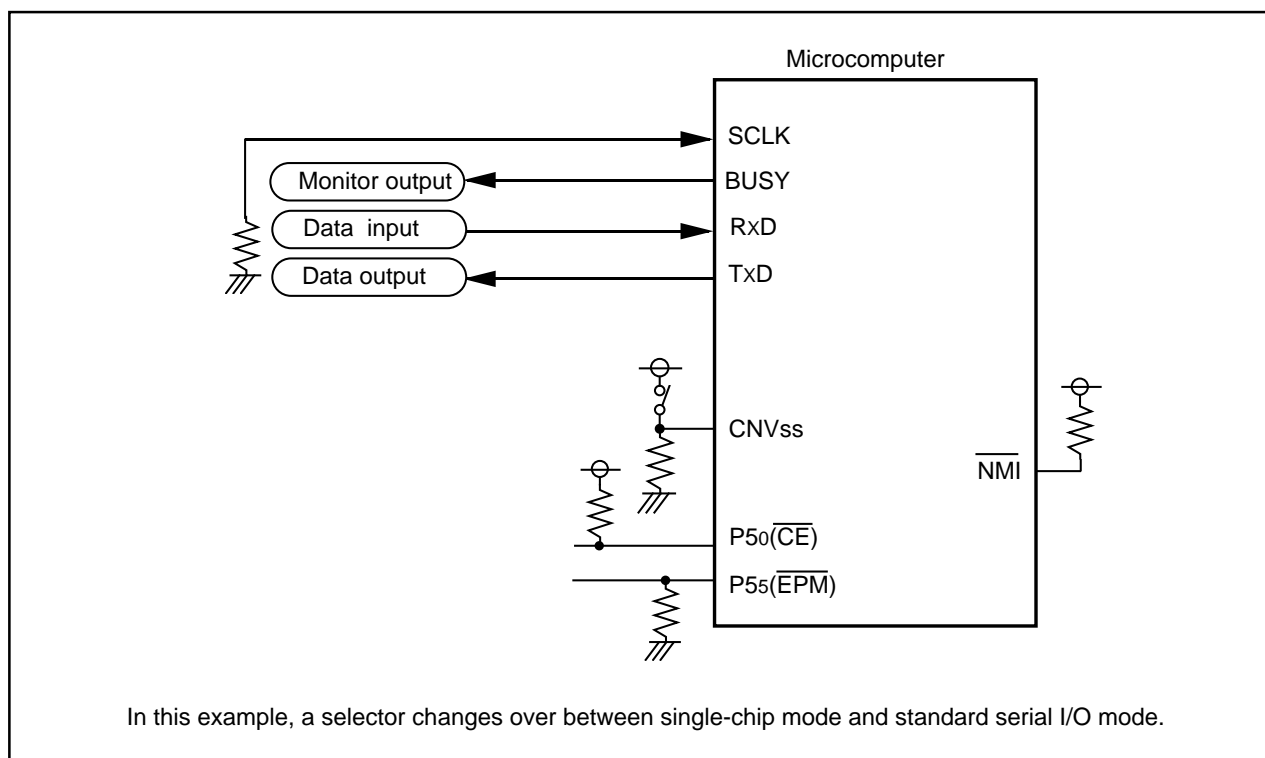
- When rewriting the boot ROM space in parallel I/O mode, serial I/O mode cannot be used.
- When an user reset signal may be set to "L" in serial I/O mode, a connection between an user reset signal and the  $\overline{\text{RESET}}$  pin should be broken by a jumper switch, for example.

### Example of Circuit Application in the Standard Serial I/O Mode

Figure 1.30.15 shows an example of a circuit application in standard serial I/O mode 1. Figure 1.30.16 shows an example of a circuit application in standard serial I/O mode 2. Refer to the user's manual for serial writer to handle pins controlled by a serial writer.



**Figure 1.30.15. Circuit Application in Standard Serial I/O Mode 1**



**Figure 1.30.16. Circuit Application in Standard Serial I/O Mode 2**

## Parallel I/O Mode

In parallel I/O mode, the user ROM space and the boot ROM space can be rewritten by a parallel writer for the M32C/83 Group. For more information about a parallel writer, contact your parallel writer manufacturer. Refer to the user's manual included with your parallel writer about how to use.

### 1. Boot ROM Space

The boot ROM space occupies 8K bytes as one block. Rewrite control program in standard serial I/O mode is written in the boot ROM space with shipment from Mitsubishi Electric.

In parallel I/O mode, the boot ROM space is located on addresses 0FFE000<sub>16</sub> to 0FFFFFF<sub>16</sub>. When rewriting the boot ROM space, rewrite this address range only. (Avoid accessing other than addresses 0FFE000<sub>16</sub> to 0FFFFFF<sub>16</sub>.)

### 2. ROM Code Protect Function

The ROM code protect function prevents the flash memory from reading and rewriting in parallel I/O mode. Figure 1.30.2 shows the ROMCP register. The ROMCP register is located on the user ROM space.

The ROMCP1 bit consists of two bits. When setting one of 2 bits or both bits to "0" by program, the ROM code protect function is enabled to prevent the flash memory from reading and rewriting.

When setting the ROMCR bits to "002" (ROM code protection removed), the flash memory can be read or rewritten. Once the ROM code protect function is enabled, the ROMCR bits cannot be changed in parallel I/O mode. The ROMCR bit should be rewritten in standard serial I/O mode or other modes.

### 3. Precautions on Parallel I/O Mode

When rewriting the boot ROM space in parallel I/O mode, standard serial I/O mode cannot be used. (Refer to the paragraph "Standard serial I/O mode".)

Electrical Characteristics (V<sub>cc</sub> = 5V)

## Electrical Characteristics

Table 1.31.1. Absolute Maximum Ratings

Symbol	Parameter	Condition	Value	Unit
V <sub>cc</sub>	Supply voltage	V <sub>cc</sub> =AV <sub>cc</sub>	-0.3 to 6.0	V
AV <sub>cc</sub>	Analog supply voltage	V <sub>cc</sub> =AV <sub>cc</sub>	-0.3 to 6.0	V
V <sub>i</sub>	Input voltage	RESET, CNV <sub>ss</sub> , 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 <sup>1</sup> , VREF, XIN	-0.3 to V <sub>cc</sub> +0.3	V
		P70, P71	-0.3 to 6.0	V
V <sub>o</sub>	Output voltage	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 <sup>1</sup> , VREF, XIN	-0.3 to V <sub>cc</sub> +0.3	V
		P70, P71	-0.3 to 6.0	V
P <sub>d</sub>	Power consumption	T <sub>opr</sub> =25°C	500	mW
T <sub>opr</sub>	Operating ambient temperature		-20 to 85	°C
T <sub>stg</sub>	Storage temperature		-65 to 150	°C

Notes :

1. Ports P11 to P15 are provided in the 144-pin package.

Electrical Characteristics (V<sub>CC</sub> = 5V)Table 1.31.2. Recommended Operating Conditions (V<sub>CC</sub> = 3.0V to 5.5V at Topr = -20 to 85°C)

Symbol	Parameter	Standard			Unit	
		Min	Typ	Max		
V <sub>CC</sub>	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	
AV <sub>CC</sub>	Analog supply voltage		V <sub>CC</sub>		V	
V <sub>SS</sub>	Supply voltage		0		V	
AV <sub>SS</sub>	Analog supply voltage		0		V	
V <sub>IH</sub>	Input "H" voltage	P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P72-P77, P80-P87 <sup>3</sup> , P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 <sup>4</sup> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	0.8V <sub>CC</sub>		V <sub>CC</sub>	V
		P70, P71	0.8V <sub>CC</sub>		6.0	V
		P00-P07, P10-P17 (In single-chip mode)	0.8V <sub>CC</sub>		V <sub>CC</sub>	V
		P00-P07, P10-P17 (In memory expansion and microprocessor modes)	0.5V <sub>CC</sub>		V <sub>CC</sub>	V
V <sub>IL</sub>	Input "L" voltage	P20-P27, P30-P37, P40-P47, P50-P57, P60-P67, P70-P77, P80-P87 <sup>3</sup> , P90-P97, P100-P107, P110-P114, P120-P127, P130-P137, P140-P146, P150-P157 <sup>4</sup> , X <sub>IN</sub> , RESET, CNV <sub>SS</sub> , BYTE	0		0.2V <sub>CC</sub>	V
		P00-P07, P10-P17 (In single-chip mode)	0		0.2V <sub>CC</sub>	V
		P00-P07, P10-P17 (In memory expansion and microprocessor modes)	0		0.16V <sub>CC</sub>	V
I <sub>OH(peak)</sub>	Peak output "H" current <sup>2</sup>	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 <sup>4</sup>			-10.0	mA
I <sub>OH(avg)</sub>	Average output "H" current <sup>1</sup>	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 <sup>4</sup>			-5.0	mA
I <sub>OL(peak)</sub>	Peak output "L" current <sup>2</sup>	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 <sup>4</sup>			10.0	mA
I <sub>OL(avg)</sub>	Average output "L" current <sup>1</sup>	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 <sup>4</sup>			5.0	mA
f(X <sub>IN</sub> )	Main clock input frequency	VDC-ON	V <sub>CC</sub> =4.2 to 5.5V	0	30	MHz
			V <sub>CC</sub> =3.0 to 4.2V	0	20	MHz
		VDC-pass through	V <sub>CC</sub> =3.0 to 3.6V	0	20	MHz
f(X <sub>CIN</sub> )	Sub clock oscillation frequency			32.768		kHz

## Notes :

- Output current is averaged within 100ms.
- Total I<sub>OL</sub> (peak) for ports P0, P1, P2, P86, P87, P9, P10, P11, P14 and P15 must be 80mA and below. Total I<sub>OH</sub> (peak) for ports P0, P1, P2, P86, P87, P9, P10, P11, P14 and P15 must be -80mA and below. Total I<sub>OL</sub> (peak) for ports P3, P4, P5, P6, P7, P80 to P84, P12 and P13 must be 80mA and below. Total I<sub>OH</sub> (peak) for ports P3, P4, P5, P6, P72 to P77, P80 to P84, P12 and P13 must be -80mA and below.
- V<sub>IH</sub> and V<sub>IL</sub> reference for P87 applies to use P87 as a programmable input ports. It do not apply to use P87 as X<sub>CIN</sub>.
- Ports P11 to P15 are provided in the 144-pin package only.

Electrical Characteristics (Vcc = 5V)

Table 1.31.3. Electrical Characteristics (VCC=4.2 to 5.5V, VSS=0V

at Topr= -20 to 85°C, f(XIN)=30MHZ unless otherwise specified)

VCC = 5V

Symbol	Parameter	Condition	Standard			Unit	
			Min	Typ	Max		
VOH	Output "H" 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 <sup>1</sup>	IOH=-5mA	3.0			V
		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 <sup>1</sup>	IOH=-200µA	4.7			V
		XOUT	IOH=-1mA	3.0			V
		XCOUT	No load applied		3.3		V
VOL	Output "L" 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 <sup>1</sup>	IOL=5mA			2.0	V
		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 <sup>1</sup>	IOL=200µA			0.45	V
		XOUT	IOL=1mA			2.0	V
		XCOUT	No load applied		0		V
VT+-VT-	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
		RESET		0.2		1.8	V
IiH	Input "H" 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 <sup>1</sup> , XIN, RESET, CNVss, BYTE	Vi=5V			5.0	µA
IiL	Input "L" 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 <sup>1</sup> , XIN, RESET, CNVss, BYTE	Vi=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 <sup>1</sup>	Vi=0V	30	50	167	kΩ
RfXIN	Feedback resistance	XIN			1.5		MΩ
RfXCIN	Feedback resistance	XCIN			10		MΩ
VRAM	RAM retention voltage	VDC-ON		2.5			V
Icc	Supply current	Measurement condition: In single-chip mode, output pins are open and other pins are connected to Vss.	f(XIN)=30MHz, square wave, no division		38	54	mA
			f(XCIN)=32kHz, with a wait Topr=25°C		470		µA
			When clock is stopped Topr=25°C		0.4	20	µA

Notes :

1. Ports P11 to P15 are provided in the 144-pin package.

Electrical Characteristics ( $V_{CC} = 5V$ ) $V_{CC} = 5V$ **Table 1.31.4. A-D Conversion Characteristics ( $V_{CC} = AV_{CC} = V_{REF} = 4.2$  to  $5.5V$ ,  $V_{SS} = AV_{SS} = 0V$  at  $T_{opr} = -20$  to  $85^{\circ}C$ ,  $f(X_{IN}) = 30MHz$  unless otherwise specified)**

Symbol	Parameter	Measurement condition	Standard			Unit
			Min	Typ	Max	
-	Resolution	$V_{REF} = V_{CC}$			10	Bits
INL	Integral nonlinearity error	$V_{REF} = V_{CC} = 5V$	AN0 to AN7		$\pm 3$	LSB
			ANEX0, ANEX1 External op-amp connection mode		$\pm 7$	LSB
DNL	Differential nonlinearity error				$\pm 1$	LSB
-	Offset error				$\pm 3$	LSB
-	Gain error				$\pm 3$	LSB
$R_{LADDER}$	Ladder resistance	$V_{REF} = V_{CC}$	8		40	$k\Omega$
$t_{CONV}$	Conversion time(10bit)		3.3			$\mu s$
$t_{CONV}$	Conversion time(8bit)		2.8			$\mu s$
$t_{SAMP}$	Sampling time		0.3			$\mu s$
$V_{REF}$	Reference voltage		2		$V_{CC}$	V
$V_{IA}$	Analog input voltage		0		$V_{REF}$	V

Notes :

1. Divide  $f(X_{IN})$ , if exceeding 10 MHz, to keep  $\emptyset AD$  frequency on 10 MHz or less.

**Table 1.31.5. D-A Conversion Characteristics ( $V_{CC} = V_{REF} = 4.2$  to  $5.5V$ ,  $V_{SS} = AV_{SS} = 0V$  at  $T_{opr} = -20$  to  $85^{\circ}C$ ,  $f(X_{IN}) = 30MHz$  unless otherwise specified)**

Symbol	Parameter	Measurement condition	Standard			Unit
			Min	Typ	Max	
-	Resolution				8	Bits
-	Absolute accuracy				1.0	%
$t_{su}$	Setup time				3	$\mu s$
$R_O$	Output resistance		4	10	20	$k\Omega$
$I_{VREF}$	Reference power supply input current	(Note 1)			1.5	mA

Notes :

1. This applies when using one D-A converter and setting the D-A register of unused D-A converter to "0016".  
Ladder resistance in A-D converter is excluded.  
 $I_{VREF}$  is sent even if  $V_{REF}$  is unconnected at the ADiCON1 register ( $i=0,1$ ).

Electrical Characteristics (V<sub>CC</sub> = 5V)V<sub>CC</sub> = 5VTiming Requirements (V<sub>CC</sub> = 4.2 to 5.5V, V<sub>SS</sub> = 0V at Topr = -20 to 85°C unless otherwise specified)

Table 1.31.6. External Clock Input

Symbol	Parameter	Standard		Unit
		Min	Max	
t <sub>c</sub>	External clock input cycle time	33		ns
t <sub>w(H)</sub>	External clock input "H" pulse width	13		ns
t <sub>w(L)</sub>	External clock input "L" pulse width	13		ns
t <sub>r</sub>	External clock rising-edge time		5	ns
t <sub>f</sub>	External clock falling-edge time		5	ns

Table 1.31.7. Memory Expansion and Microprocessor Modes

Symbol	Parameter	Standard		Unit
		Min	Max	
t <sub>ac1(RD-DB)</sub>	Data input access time (RD standard, no wait)		(Note 1)	ns
t <sub>ac1(AD-DB)</sub>	Data input access time (AD standard, CS standard, no wait)		(Note 1)	ns
t <sub>ac2(RD-DB)</sub>	Data input access time (RD standard, with wait)		(Note 1)	ns
t <sub>ac2(AD-DB)</sub>	Data input access time (AD standard, CS standard, with wait)		(Note 1)	ns
t <sub>ac3(RD-DB)</sub>	Data input access time (RD standard, when accessing multiplex bus space)		(Note 1)	ns
t <sub>ac3(AD-DB)</sub>	Data input access time (AD standard, CS standard, when accessing multiplex bus space)		(Note 1)	ns
t <sub>ac4(RAS-DB)</sub>	Data input access time (RAS standard, DRAM access)		(Note 1)	ns
t <sub>ac4(CAS-DB)</sub>	Data input access time (CAS standard, DRAM access)		(Note 1)	ns
t <sub>ac4(CAD-DB)</sub>	Data input access time (CAD standard, DRAM access)		(Note 1)	ns
t <sub>su(DB-BCLK)</sub>	Data input setup time	26		ns
t <sub>su(RDY-BCLK)</sub>	RDY input setup time	26		ns
t <sub>su(HOLD-BCLK)</sub>	HOLD input setup time	30		ns
t <sub>h(RD-DB)</sub>	Data input hold time	0		ns
t <sub>h(CAS-DB)</sub>	Data input hold time	0		ns
t <sub>h(BCLK-RDY)</sub>	RDY input hold time	0		ns
t <sub>h(BCLK-HOLD)</sub>	HOLD input hold time	0		ns
t <sub>d(BCLK-HLDA)</sub>	HLDA output delay time		25	ns

## Notes :

1. A value can be obtained from the following expressions, according to BCLK frequency.  
Insert a wait or use lower operation frequency f(BCLK) if a calculated value is negative.

$$t_{ac1(RD-DB)} = \frac{10^9}{f(BCLK) \times 2} - 35 \quad [ns]$$

$$t_{ac1(AD-DB)} = \frac{10^9}{f(BCLK)} - 35 \quad [ns]$$

$$t_{ac2(RD-DB)} = \frac{10^9 \times m}{f(BCLK) \times 2} - 35 \quad [ns] \quad (m=3 \text{ when 1 wait, } m=5 \text{ and when 2 waits and } m=7 \text{ when 3 waits})$$

$$t_{ac2(AD-DB)} = \frac{10^9 \times n}{f(BCLK)} - 35 \quad [ns] \quad (n=2 \text{ when 1 wait, } n=3 \text{ when 2 waits and } n=4 \text{ when 3 waits})$$

$$t_{ac3(RD-DB)} = \frac{10^9 \times m}{f(BCLK) \times 2} - 35 \quad [ns] \quad (m=3 \text{ when 2 waits and } m=5 \text{ when 3 waits})$$

$$t_{ac3(AD-DB)} = \frac{10^9 \times n}{f(BCLK) \times 2} - 35 \quad [ns] \quad (n=5 \text{ when 2 waits and } n=7 \text{ when 3 waits})$$

$$t_{ac4(RAS-DB)} = \frac{10^9 \times m}{f(BCLK) \times 2} - 35 \quad [ns] \quad (m=3 \text{ when 1 wait and } m=5 \text{ when 2 waits})$$

$$t_{ac4(CAS-DB)} = \frac{10^9 \times n}{f(BCLK) \times 2} - 35 \quad [ns] \quad (n=1 \text{ when 1 wait and } n=3 \text{ when 2 waits})$$

$$t_{ac4(CAD-DB)} = \frac{10^9 \times l}{f(BCLK)} - 35 \quad [ns] \quad (l=1 \text{ when 1 wait and } l=2 \text{ when 2 waits})$$

Electrical Characteristics ( $V_{CC} = 5V$ ) $V_{CC} = 5V$ **Timing Requirements** $(V_{CC} = 4.2$  to  $5.5V$ ,  $V_{SS} = 0V$  at  $T_{opr} = -20$  to  $85^{\circ}C$  unless otherwise specified)**Table 1.31.8. Timer A Input (Count Source 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 "H" pulse width	40		ns
$t_w(TAL)$	TAiIN input "L" pulse width	40		ns

**Table 1.31.9. Timer A Input (Gate Input in Timer Mode)**

Symbol	Parameter	Standard		Unit
		Min	Max	
$t_c(TA)$	TAiIN input cycle time	400		ns
$t_w(TAH)$	TAiIN input "H" pulse width	200		ns
$t_w(TAL)$	TAiIN input "L" pulse width	200		ns

**Table 1.31.10. 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 "H" pulse width	100		ns
$t_w(TAL)$	TAiIN input "L" pulse width	100		ns

**Table 1.31.11. Timer A Input (External Trigger Input in Pulse Width Modulation Mode)**

Symbol	Parameter	Standard		Unit
		Min	Max	
$t_w(TAH)$	TAiIN input "H" pulse width	100		ns
$t_w(TAL)$	TAiIN input "L" pulse width	100		ns

**Table 1.31.12. Timer A Input (Counter Increment/decrement 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 "H" pulse width	1000		ns
$t_w(UPL)$	TAiOUT input "L" pulse width	1000		ns
$t_{su}(UP-TIN)$	TAiOUT input setup time	400		ns
$t_h(TIN-UP)$	TAiOUT input hold time	400		ns

Electrical Characteristics ( $V_{CC} = 5V$ ) $V_{CC} = 5V$ 

## Timing Requirements

 $(V_{CC} = 4.2$  to  $5.5V$ ,  $V_{SS} = 0V$  at  $T_{opr} = -20$  to  $85^{\circ}C$  unless otherwise specified)

Table 1.31.13. Timer B Input (Count Source Input in eEvent 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 "H" pulse width (counted on one edge)	40		ns
$t_{w(TBL)}$	TBiIN input "L" 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 "H" pulse width (counted on both edges)	80		ns
$t_{w(TBL)}$	TBiIN input "L" pulse width (counted on both edges)	80		ns

Table 1.31.14. 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 "H" pulse width	200		ns
$t_{w(TBL)}$	TBiIN input "L" pulse width	200		ns

Table 1.31.15. 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 "H" pulse width	200		ns
$t_{w(TBL)}$	TBiIN input "L" pulse width	200		ns

Table 1.31.16. A-D trigger Input

Symbol	Parameter	Standard		Unit
		Min	Max	
$t_{c(AD)}$	$\overline{ADTRG}$ input cycle time (trigger available at minimum and above)	1000		ns
$t_{w(ADL)}$	$\overline{ADTRG}$ input "L" pulse width	125		ns

Table 1.31.17. Serial I/O

Symbol	Parameter	Standard		Unit
		Min	Max	
$t_{c(CK)}$	CLKi input cycle time	200		ns
$t_{w(CKH)}$	CLKi input "H" pulse width	100		ns
$t_{w(CKL)}$	CLKi input "L" 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 1.31.18. External Interrupt  $\overline{INTi}$  Input

Symbol	Parameter	Standard		Unit
		Min	Max	
$t_{w(INH)}$	$\overline{INTi}$ input "H" pulse width	250		ns
$t_{w(INL)}$	$\overline{INTi}$ input "L" pulse width	250		ns

Electrical Characteristics ( $V_{CC} = 5V$ ) $V_{CC} = 5V$ 

## Switching Characteristics

( $V_{CC} = 4.2$  to  $5.5V$ ,  $V_{SS} = 0V$  at  $T_{opr} = -20$  to  $85^{\circ}C$  unless otherwise specified)

Table 1.31.19. Memory Expansion Mode And Microprocessor Mode (without wait)

Symbol	Parameter	Measurement condition	Standard		Unit
			Min	Max	
$t_d(BCLK-AD)$	Address output delay time	Figure 1.31.1		18	ns
$t_h(BCLK-AD)$	Address output hold time (BCLK standard)		-3		ns
$t_h(RD-AD)$	Address output hold time (RD standard)		0		ns
$t_h(WR-AD)$	Address output hold time (WR standard)		(Note1)		ns
$t_d(BCLK-CS)$	Chip-select output delay time			18	ns
$t_h(BCLK-CS)$	Chip-select output hold time (BCLK standard)		-3		ns
$t_h(RD-CS)$	Chip-select output hold time (RD standard)		0		ns
$t_h(WR-CS)$	Chip-select output hold time (WR standard)		(Note1)		ns
$t_d(BCLK-ALE)$	ALE signal output delay time			18	ns
$t_h(BCLK-ALE)$	ALE signal output hold time		-2		ns
$t_d(BCLK-RD)$	RD signal output delay time			18	ns
$t_h(BCLK-RD)$	RD signal output hold time		-5		ns
$t_d(BCLK-WR)$	WR signal output delay time			18	ns
$t_h(BCLK-WR)$	WR signal output hold time		-3		ns
$t_d(DB-WR)$	Data output delay time (WR standard)		(Note1)		ns
$t_h(WR-DB)$	Data output hold time (WR standard)		(Note1)		ns
$t_w(WR)$	WR output width		(Note1)		ns

## Notes :

1. A value can be obtained from the following expressions, according to BCLK frequency.

$$t_d(DB-WR) = \frac{10^9}{f_{(BCLK)}} - 20 \quad [ns]$$

$$t_h(WR-DB) = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [ns]$$

$$t_h(WR-AD) = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [ns]$$

$$t_h(WR-CS) = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [ns]$$

$$t_w(WR) = \frac{10^9}{f_{(BCLK)} \times 2} - 15 \quad [ns]$$

Electrical Characteristics (V<sub>CC</sub> = 5V)V<sub>CC</sub> = 5V

## Switching Characteristics

(V<sub>CC</sub> = 4.2 to 5.5V, V<sub>SS</sub> = 0V at Topr = -20 to 85°C unless otherwise specified)Table 1.31.20. Memory Expansion Mode and Microprocessor Mode  
(With a Wait, Accessing External Memory)

Symbol	Parameter	Measurement condition	Standard		Unit
			Min	Max	
td(BCLK-AD)	Address output delay time	Figure 1.31.1		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)		(Note1)		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)		(Note1)		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)		(Note1)		ns
th(WR-DB)	Data output hold time (WR standard)		(Note1)		ns
tw(WR)	WR output width		(Note1)		ns

## Notes :

1. A value can be obtained from the following expressions, according to BCLK frequency.

$$td(DB - WR) = \frac{10^9 \times n}{f(BCLK)} - 20 \quad [ns] \quad (n=1 \text{ when 1 wait } n=2 \text{ when 2 waits and } n=3 \text{ when 3 waits})$$

$$th(WR - DB) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$th(WR - AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$th(WR - CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$tw(WR) = \frac{10^9 \times n}{f(BCLK) \times 2} - 15 \quad [ns] \quad (n=1 \text{ when 1 wait } n=3 \text{ when 2 waits and } n=5 \text{ when 3 waits})$$

## Electrical Characteristics (Vcc = 5V)

Vcc = 5V

## Switching Characteristics

(Vcc = 4.2 to 5.5V, Vss = 0V at Topr = -20 to 85°C unless otherwise specified)

**Table 1.31.21. Memory Expansion Mode and Microprocessor Mode**  
**(With a Wait, Accessing External Memory, Multiplex Bus Space Selected)**

Symbol	Parameter	Measurement condition	Standard		Unit
			Min	Max	
td(BCLK-AD)	Address output delay time	Figure 1.31.1		18	ns
th(BCLK-AD)	Address output hold time (BCLK standard)		-3		ns
th(RD-AD)	Address output hold time (RD standard)		(Note 1)		ns
th(WR-AD)	Address output hold time (WR standard)		(Note 1)		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 1)		ns
th(WR-CS)	Chip-select output hold time (WR standard)		(Note 1)		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 1)		ns
th(WR-DB)	Data output hold time (WR standard)		(Note 1)		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 1)		ns
th(ALE-AD)	ALE signal output hold time (address standard)		(Note 1)		ns
tdz(RD-AD)	Address output high-impedance start time			8	ns

## Notes :

1. A value can be obtained from the following expressions, according to BCLK frequency.

$$th(RD - AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$th(WR - AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$th(RD - CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$th(WR - CS) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$td(DB - WR) = \frac{10^9 \times m}{f(BCLK) \times 2} - 25 \quad [ns] \quad (m=3 \text{ when 2 waits and } m=5 \text{ when 3 waits})$$

$$th(WR - DB) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

$$td(AD - ALE) = \frac{10^9}{f(BCLK) \times 2} - 20 \quad [ns]$$

$$th(ALE - AD) = \frac{10^9}{f(BCLK) \times 2} - 10 \quad [ns]$$

Electrical Characteristics (V<sub>CC</sub> = 5V)V<sub>CC</sub> = 5V

## Switching Characteristics

(V<sub>CC</sub> = 4.2 to 5.5V, V<sub>SS</sub> = 0V at Topr = -20 to 85°C unless otherwise specified)

**Table 1.31.22. Memory Expansion Mode and Microprocessor Mode**  
**(With a Wait, Accessing External Memory, DRAM Space Selected)**

Symbol	Parameter	Measurement condition	Standard		Unit
			Min	Max	
td(BCLK-RAD)	Row address output delay time	Figure 1.31.1		18	ns
th(BCLK-RAD)	Row address output hold time (BCLK standard)		-3		ns
td(BCLK-CAD)	Column address output delay time			18	ns
th(BCLK-CAD)	Column address output hold time (BCLK standard)		-3		ns
th(RAS-RAD)	Row address output hold time after RAS output		(Note 1)		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 1)		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 1)		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 1)		ns

## Notes :

1. A value can be obtained from the following expressions, according to BCLK frequency.

$$t_{h(RAS - RAD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 13 \quad [\text{ns}]$$

$$t_{RP} = \frac{10^9}{f_{(BCLK)} \times 2} \times 3 - 20 \quad [\text{ns}]$$

$$t_{su(DB - CAS)} = \frac{10^9}{f_{(BCLK)}} - 20 \quad [\text{ns}]$$

$$t_{su(CAS - RAS)} = \frac{10^9}{f_{(BCLK)} \times 2} - 13 \quad [\text{ns}]$$

Electrical Characteristics (Vcc = 5V)

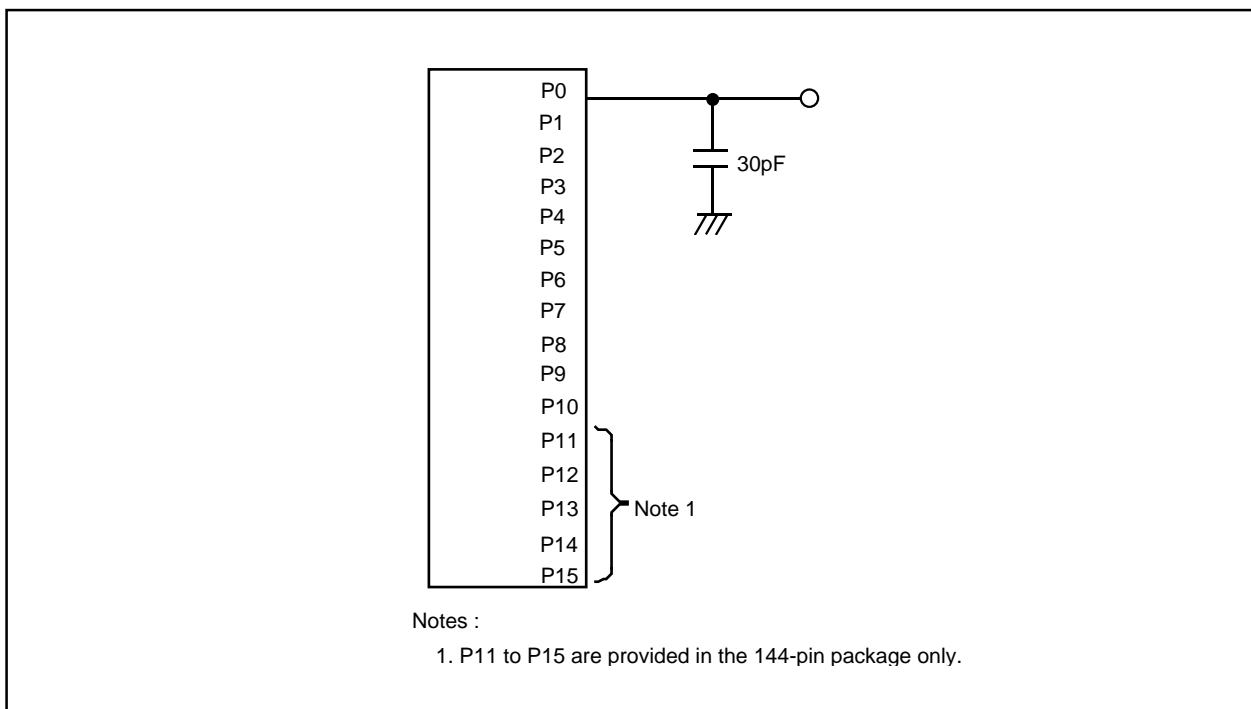
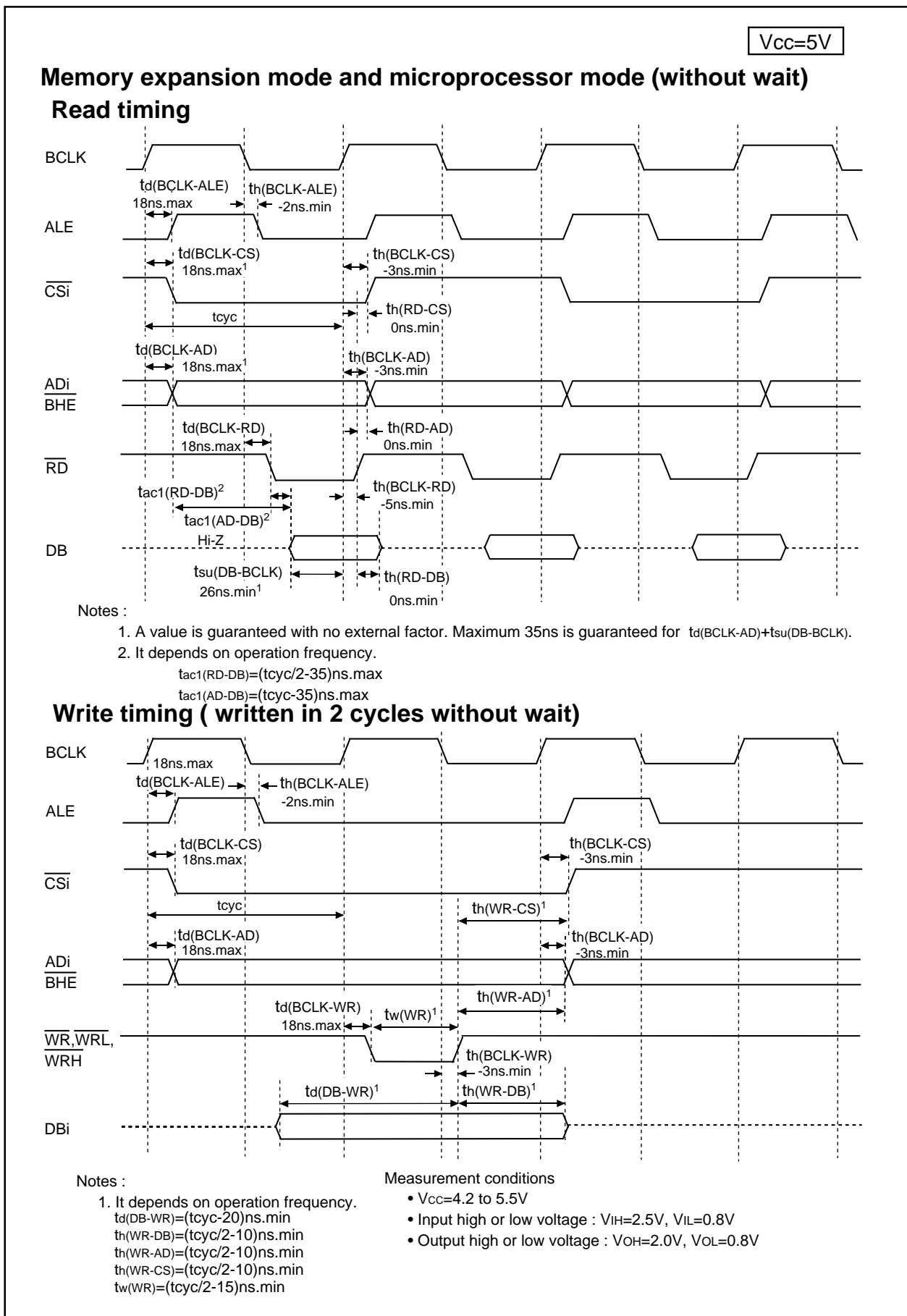


Figure 1.31.1. Ports P0 to P15 Measurement Circuit

**Electrical Characteristics (Vcc = 5V)**



**Figure 1.31.2. Vcc=5V Timing Diagram (1)**

Electrical Characteristics (Vcc = 5V)

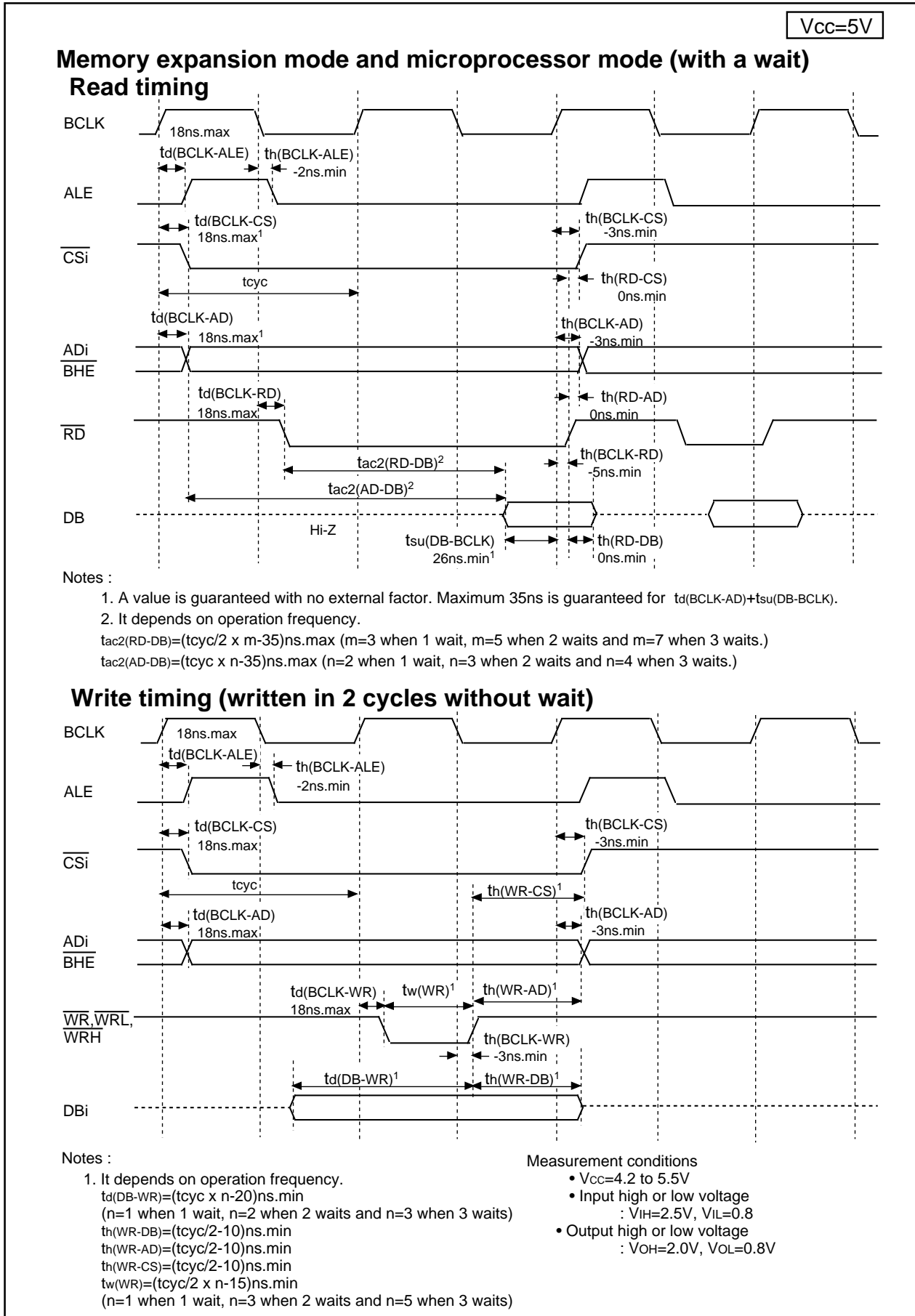


Figure 1.31.3. Vcc=5V Timing Diagram (2)

Electrical Characteristics (Vcc = 5V)

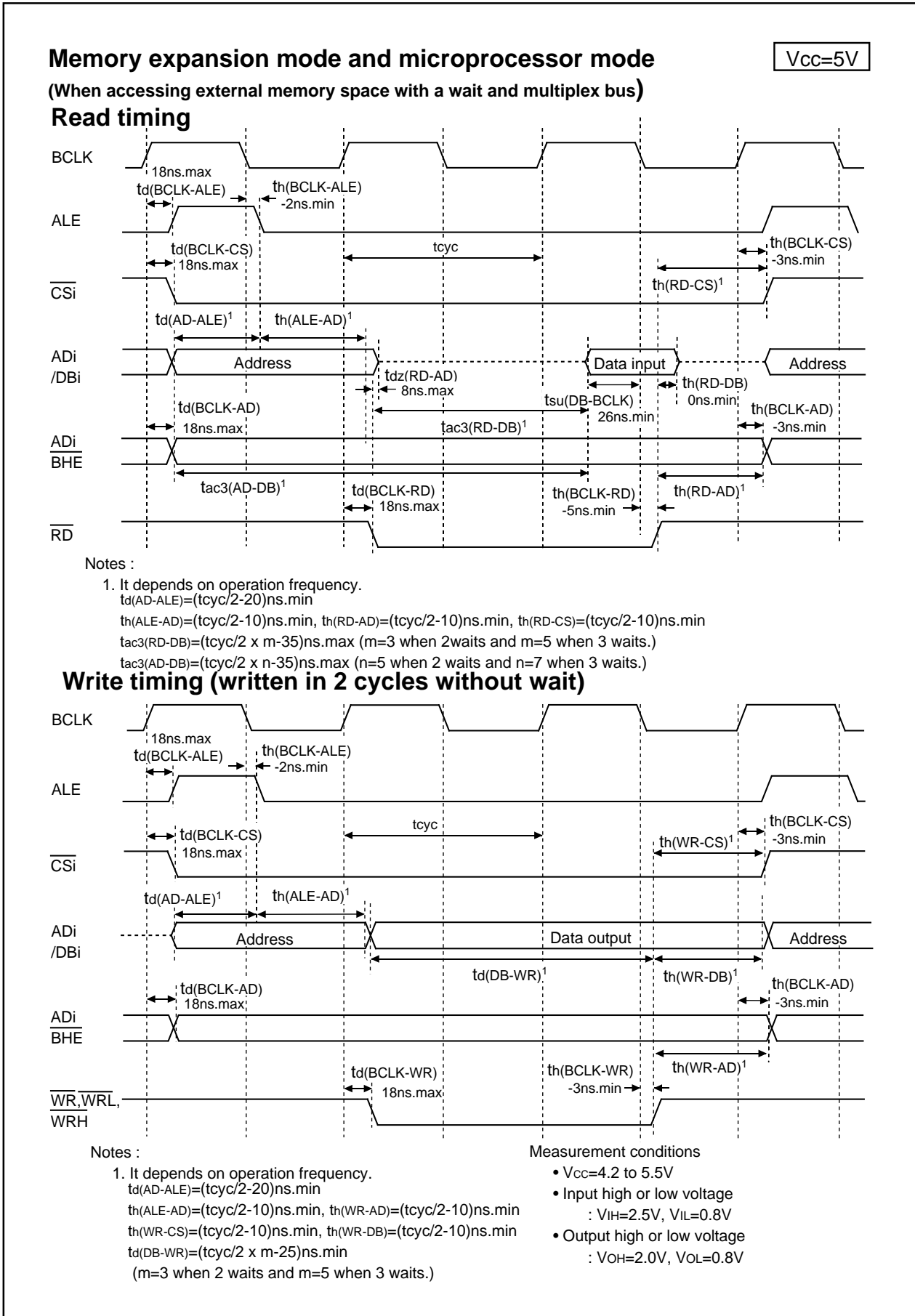


Figure 1.31.4. Vcc=5V Timing Diagram (3)

Electrical Characteristics (Vcc = 5V)

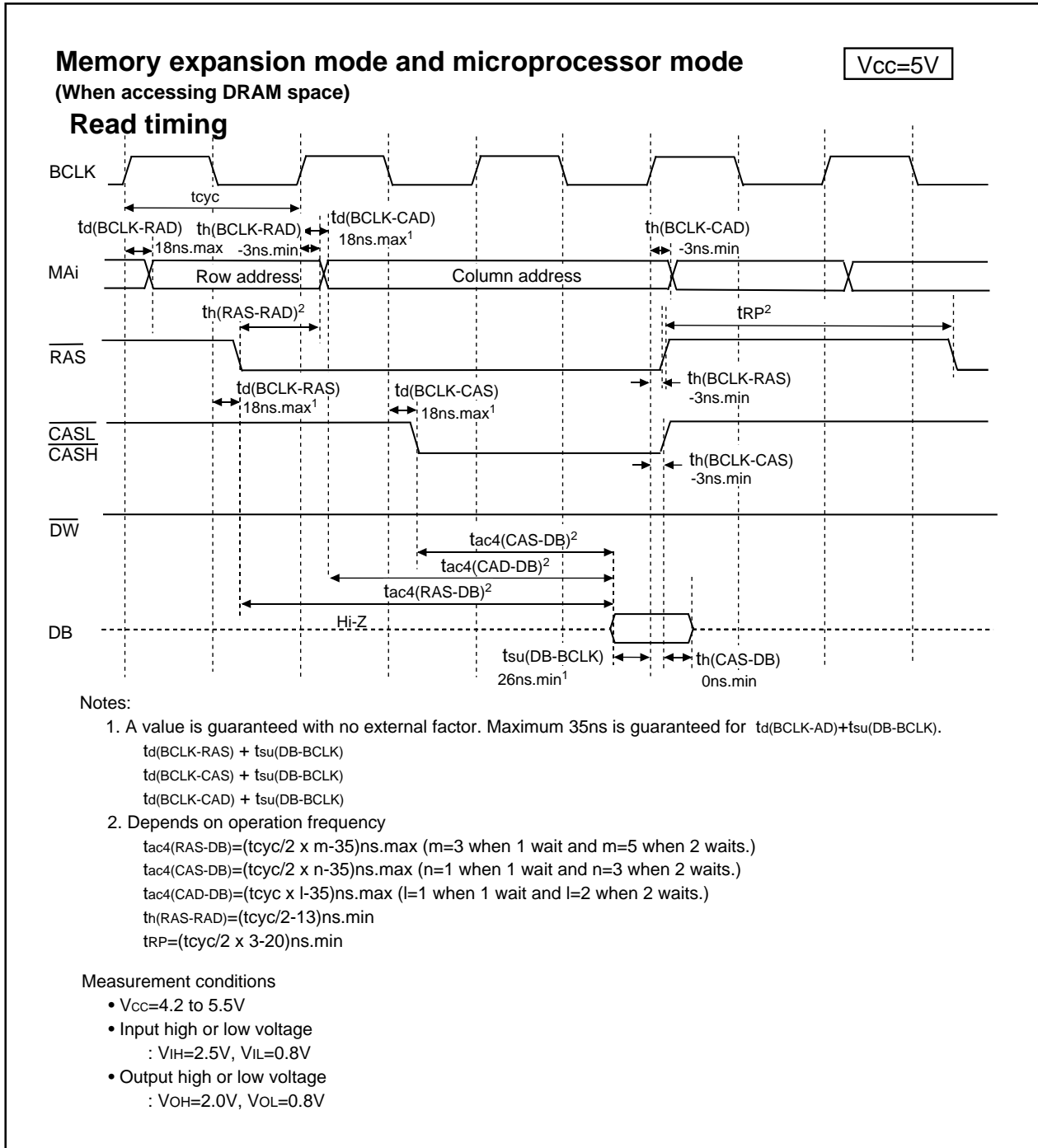


Figure 131.5. Vcc=5V Timing Diagram (4)

Electrical Characteristics (Vcc = 5V)

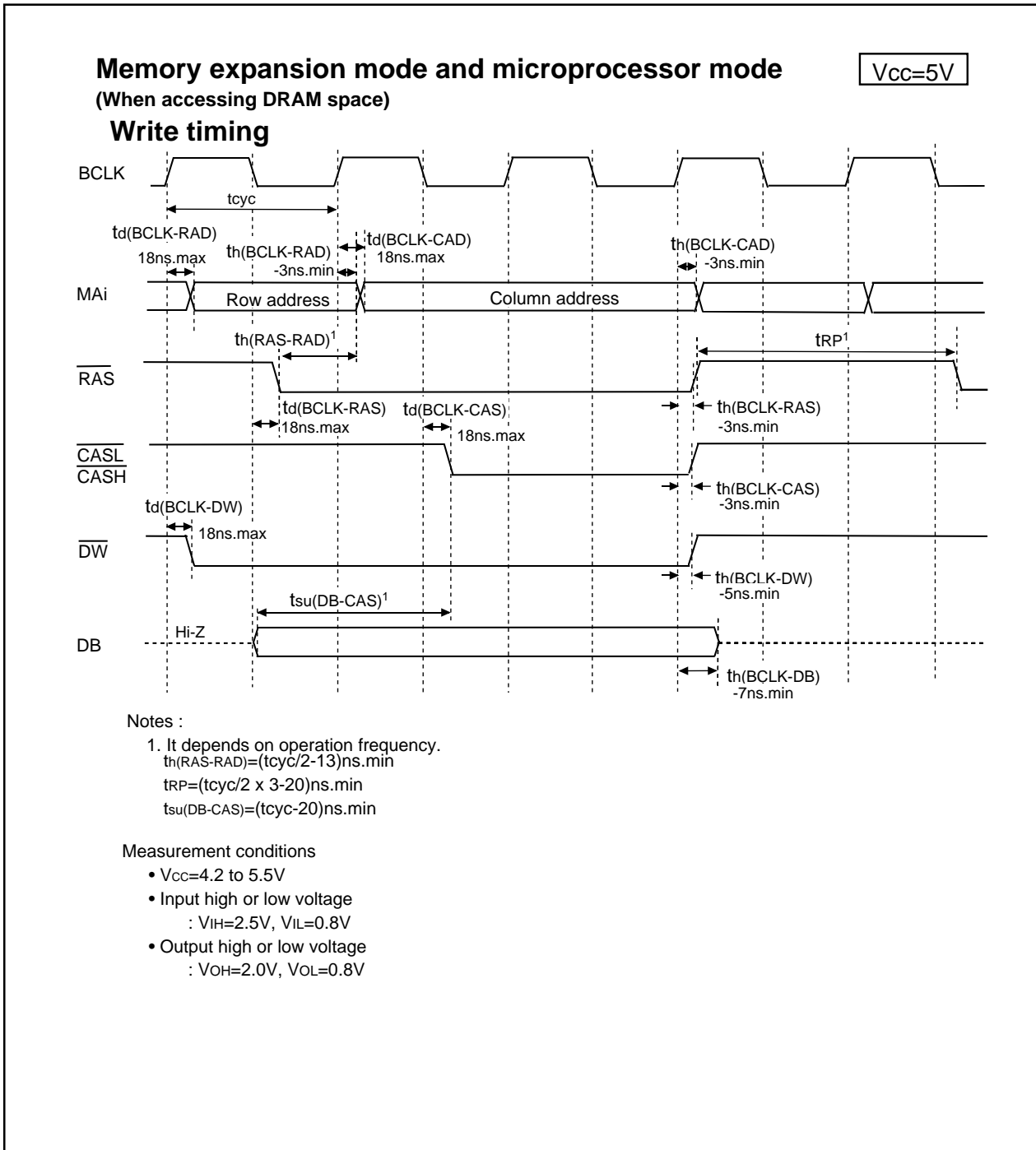


Figure 1.31.6. Vcc=5V Timing Diagram (5)

Electrical Characteristics (V<sub>CC</sub> = 5V)

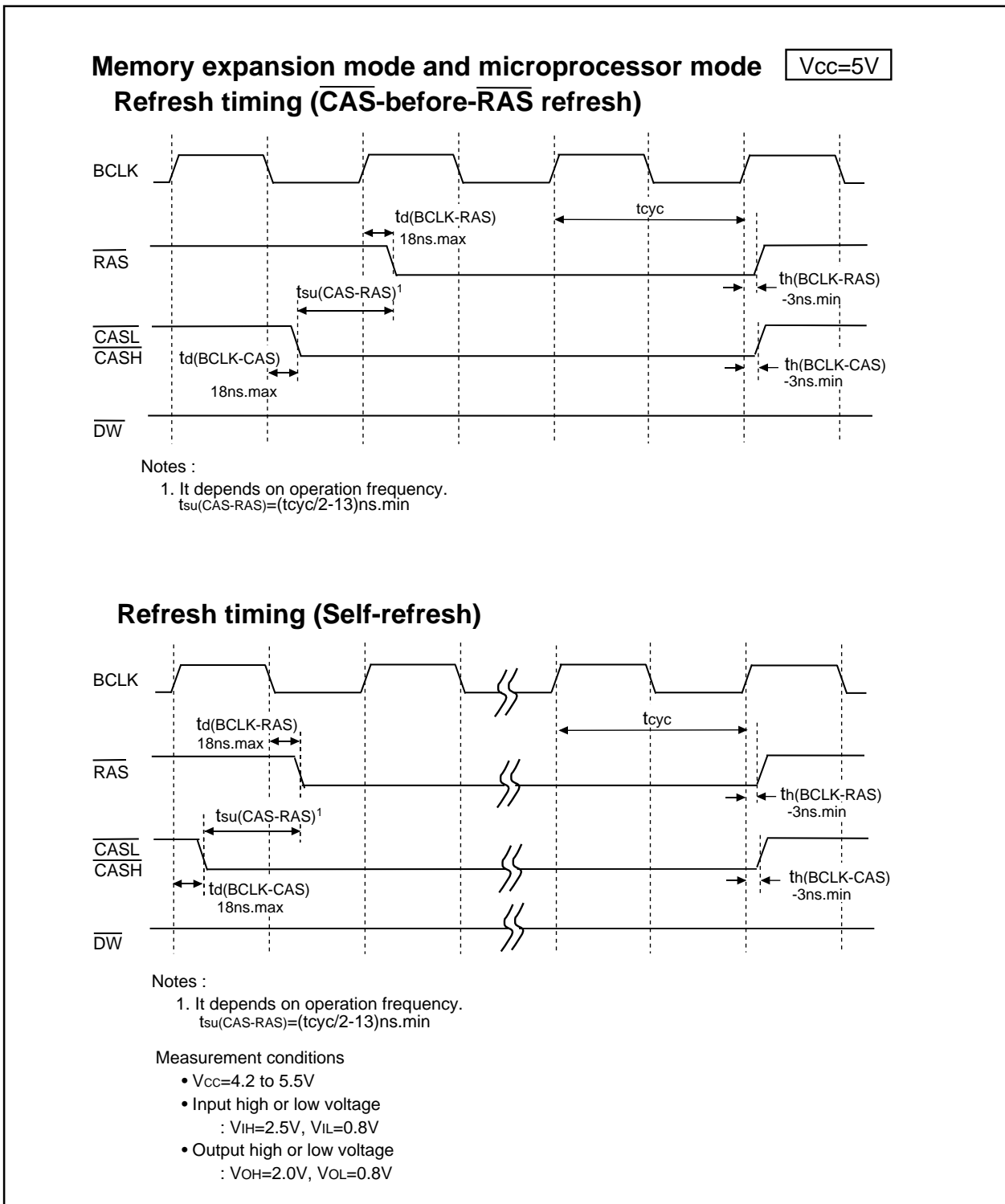


Figure 1.31.7. V<sub>CC</sub>=5V Timing Diagram (6)

Electrical Characteristics ( $V_{CC} = 5V$ )

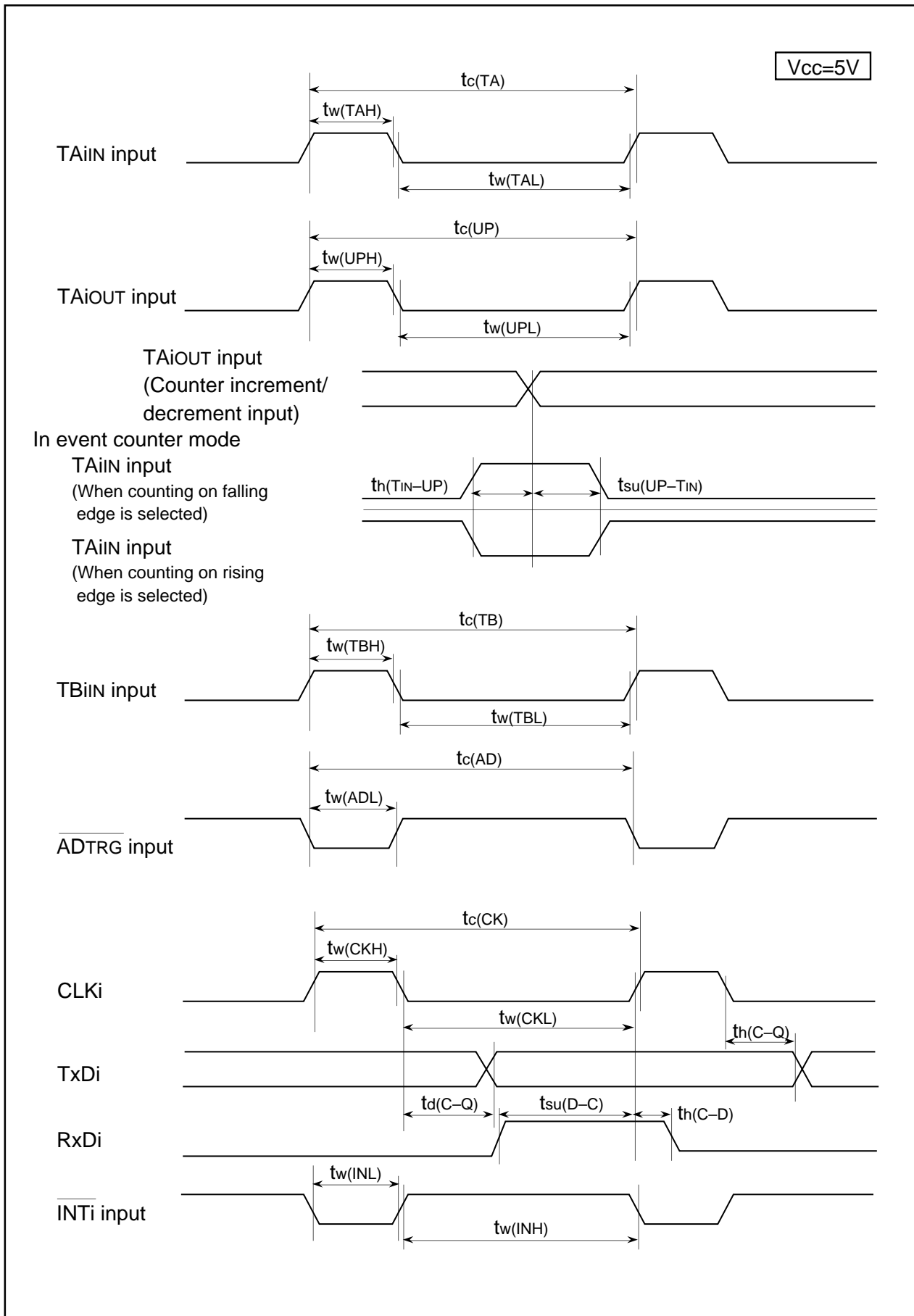


Figure 1.31.8.  $V_{CC}=5V$  Timing Diagram (7)

Electrical Characteristics (Vcc = 5V)

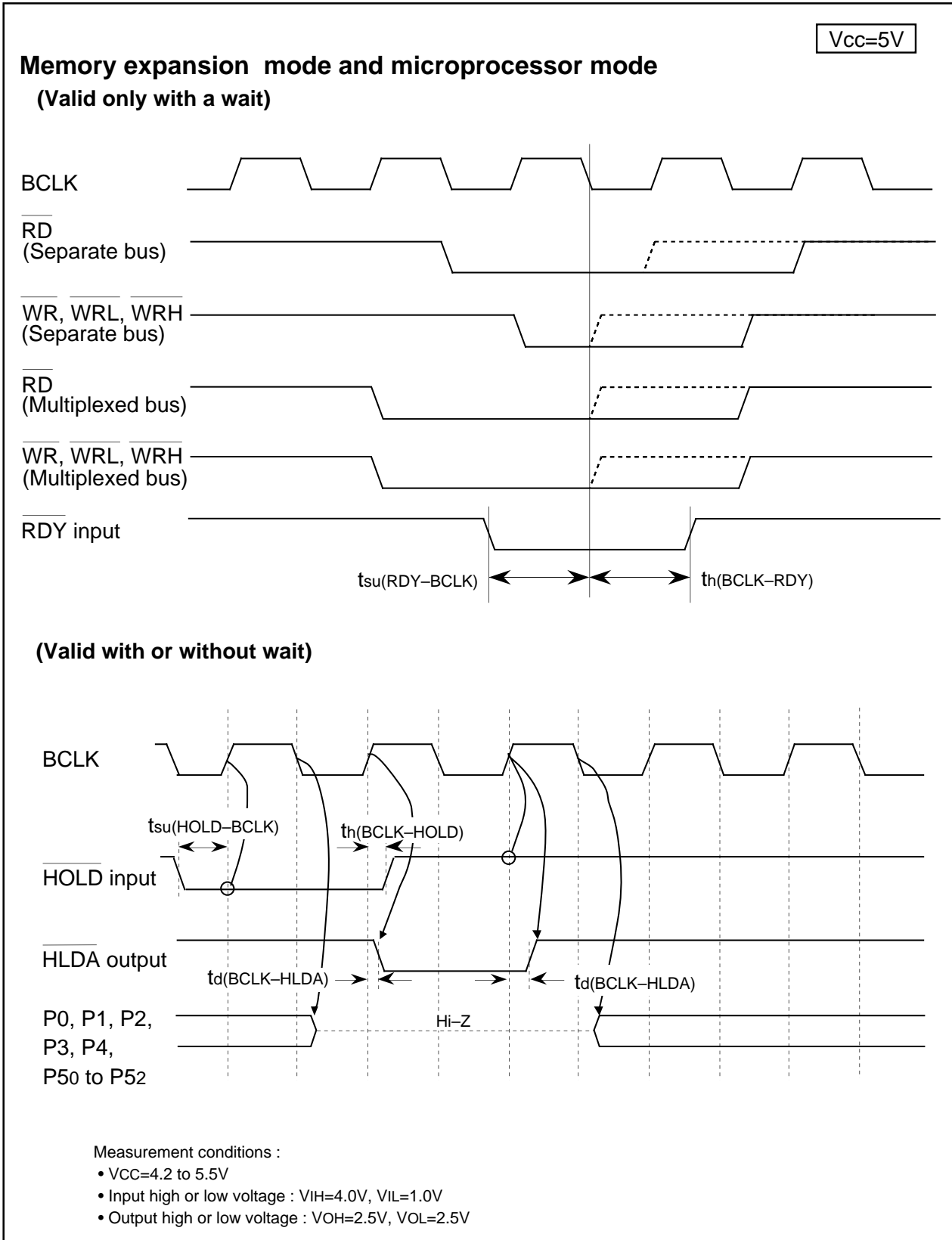


Figure 1.31.9. Vcc=5V Timing Diagram (8)

Electrical Characteristics (Vcc = 3.3V)

VCC = 3.3V

**Table 1.31.23. Electrical Characteristics (VCC=3.0 to 3.6V, VSS=0V at Topr = -20 to 85°C, f(XIN)=20MHZ unless otherwise specified)**

Symbol	Parameter		Condition	Standard			Unit
				Min	Typ	Max	
VOH	Output "H" 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 <sup>1</sup>	IOH=-1mA	2.7			V
		XOUT	IOH=-0.1mA	2.7			V
		XCOUT	No load applied		3.3		V
VOL	Output "L" 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 <sup>1</sup>	IOL=1mA			0.5	
		XOUT	IOL=0.1mA			0.5	V
		XCOUT	No load applied		0		V
VT+-VT-	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
		RESET	0.2		1.8	V	
IiH	Input "H" 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 <sup>1</sup> , XIN, RESET, CNVss, BYTE	Vi=3V			4.0	μA
IiL	Input "L" 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 <sup>1</sup> , XIN, RESET, CNVss, BYTE	Vi=0V			-4.0	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 <sup>1</sup>	Vi=0V	66	120	500	kΩ
RfXIN	Feedback resistance	XIN			3.0		MΩ
RfXCIN	Feedback resistance	XCIN			20.0		MΩ
VRAM	RAM retention voltage	VDC-ON		2.5			V
		VDC-pass through		2.0			V
Icc	Power supply current	Measuring condition: In single-chip mode, the output pins are open and other pins are connected to Vss.	f(XIN)=20MHz, square wave, no division		26	38	mA
			f(XCIN)=32kHz, with a wait, VDC-pass through, Topr=25°C		5.0		μA
			f(XCIN)=32kHz, with a wait, VDC-ON, Topr=25°C		340		mA
			When clock is stopped Topr=25°C	0.4	20	μA	

Notes :

1. Ports P11 to P15 are provided in the 144-pin package only.

Electrical Characteristics ( $V_{CC} = 3.3V$ ) $V_{CC} = 3.3V$ **Table 1.31.24. A-D Conversion Characteristics ( $V_{CC} = AV_{CC} = V_{REF} = 3.0$  to  $3.6V$ ,  $V_{SS} = AV_{SS} = 0V$  at  $T_{opr} = -20$  to  $85^{\circ}C$ ,  $f(X_{IN}) = 20MHz$  unless otherwise specified)**

Symbol	Parameter		Measurement condition	Standard			Unit
				Min	Typ	Max	
-	Resolution		$V_{REF} = V_{CC}$			10	Bits
ISL	Integral nonlinearity error	No S&H function(8-bit)	$V_{CC} = V_{REF} = 3.3V$			$\pm 2$	LSB
DSL	Differential nonlinearity error	No S&H function(8-bit)				$\pm 1$	LSB
-	Offset error	No S&H function(8-bit)				$\pm 2$	LSB
-	Gain error	No S&H function(8-bit)				$\pm 2$	LSB
$R_{LADDER}$	Ladder resistance		$V_{REF} = V_{CC}$	8		40	k $\Omega$
$t_{CONV}$	Conversion time(8bit)			9.8			$\mu s$
$V_{REF}$	Reference voltage			3.0		$V_{CC}$	V
$V_{IA}$	Analog input voltage			0		$V_{REF}$	V

S&amp;H: Sample and hold

Notes :

1. Divide  $f(X_{IN})$ , if exceeding 10MHz, to keep  $\emptyset AD$  frequency on 10 MHz or less.

**Table 1.31.25. D-A Conversion Characteristics ( $V_{CC} = V_{REF} = 3.0$  to  $3.6V$ ,  $V_{SS} = AV_{SS} = 0V$  at  $T_{opr} = -20$  to  $85^{\circ}C$ ,  $f(X_{IN}) = 20MHz$  unless otherwise specified)**

Symbol	Parameter		Measurement condition	Standard			Unit
				Min	Typ	Max	
-	Resolution					8	Bits
-	Absolute accuracy					1.0	%
$t_{su}$	Setup time					3	$\mu s$
$R_o$	Output resistance			4	10	20	k $\Omega$
$I_{VREF}$	Reference power supply input current		(Note 1)			1.0	mA

Notes :

1. This applies when using one D-A converter and setting the D-A register of unused D-A converter to "0016". Ladder resistance in A-D converter is excluded.  
 $I_{VREF}$  is sent even if  $V_{REF}$  is unconnected to the ADiCON1 register ( $i=0,1$ ).

Electrical Characteristics (V<sub>CC</sub> = 3.3V)V<sub>CC</sub> = 3.3VTiming Requirements (V<sub>CC</sub> = 3.0 to 3.6V, V<sub>SS</sub> = 0V at Topr = -20 to 85°C unless otherwise specified)

Table 1.31.26. External Clock Input

Symbol	Parameter	Standard		Unit
		Min	Max	
t <sub>c</sub>	External clock input cycle time	50		ns
t <sub>w(H)</sub>	External clock input "H" pulse width	22		ns
t <sub>w(L)</sub>	External clock input "L" pulse width	22		ns
t <sub>r</sub>	External clock rising-edge time		5	ns
t <sub>f</sub>	External clock falling-edge time		5	ns

Table 1.31.27. Memory Expansion and Microprocessor Modes

Symbol	Parameter	Standard		Unit
		Min	Max	
t <sub>ac1</sub> (RD-DB)	Data input access time (RD standard, without wait)		(Note 1)	ns
t <sub>ac1</sub> (AD-DB)	Data input access time (AD standard, CS standard, without wait)		(Note 1)	ns
t <sub>ac2</sub> (RD-DB)	Data input access time (RD standard, with a wait)		(Note 1)	ns
t <sub>ac2</sub> (AD-DB)	Data input access time (AD standard, CS standard, with a wait)		(Note 1)	ns
t <sub>ac3</sub> (RD-DB)	Data input access time (RD standard, when accessing multiplex bus space)		(Note 1)	ns
t <sub>ac3</sub> (AD-DB)	Data input access time (AD standard, CS standard, when accessing multiplex bus space)		(Note 1)	ns
t <sub>ac4</sub> (RAS-DB)	Data input access time (RAS standard, DRAM access)		(Note 1)	ns
t <sub>ac4</sub> (CAS-DB)	Data input access time (CAS standard, DRAM access)		(Note 1)	ns
t <sub>ac4</sub> (CAD-DB)	Data input access time (CAD standard, DRAM access)		(Note 1)	ns
t <sub>su</sub> (DB-BCLK)	Data input setup time	30		ns
t <sub>su</sub> (RDY-BCLK)	RDY input setup time	40		ns
t <sub>su</sub> (HOLD-BCLK)	HOLD input setup time	60		ns
t <sub>h</sub> (RD-DB)	Data input hold time	0		ns
t <sub>h</sub> (CAS-DB)	Data input hold time	0		ns
t <sub>h</sub> (BCLK-RDY)	RDY input hold time	0		ns
t <sub>h</sub> (BCLK-HOLD)	HOLD input hold time	0		ns
t <sub>d</sub> (BCLK-HLDA)	HLDA output delay time		25	ns

## Notes :

1. A value can be obtained from the following expressions, according to BCLK frequency.

Insert a wait or use lower operation frequency f(BCLK) if a calculated value is negative.

$$t_{ac1}(RD-DB) = \frac{10^9}{f(BCLK) \times 2} - 35 \quad [ns]$$

$$t_{ac1}(AD-DB) = \frac{10^9}{f(BCLK)} - 35 \quad [ns]$$

$$t_{ac2}(RD-DB) = \frac{10^9 \times m}{f(BCLK) \times 2} - 35 \quad [ns] \quad (m=3 \text{ when 1 wait, } m=5 \text{ when 2 waits and } m=7 \text{ when 3 waits.})$$

$$t_{ac2}(AD-DB) = \frac{10^9 \times n}{f(BCLK)} - 35 \quad [ns] \quad (n=2 \text{ when 1 wait, } n=3 \text{ when 2 waits and } n=4 \text{ when 3 waits.})$$

$$t_{ac3}(RD-DB) = \frac{10^9 \times m}{f(BCLK) \times 2} - 35 \quad [ns] \quad (m=3 \text{ when 2 waits and } m=5 \text{ when 3 waits.})$$

$$t_{ac3}(AD-DB) = \frac{10^9 \times n}{f(BCLK) \times 2} - 35 \quad [ns] \quad (n=5 \text{ when 2 waits and } n=7 \text{ when 3 waits.})$$

$$t_{ac4}(RAS-DB) = \frac{10^9 \times m}{f(BCLK) \times 2} - 35 \quad [ns] \quad (m=3 \text{ when 1 wait and } m=5 \text{ when 2 waits.})$$

$$t_{ac4}(CAS-DB) = \frac{10^9 \times n}{f(BCLK) \times 2} - 35 \quad [ns] \quad (n=1 \text{ when 1 wait and } n=3 \text{ when 2 waits.})$$

$$t_{ac4}(CAD-DB) = \frac{10^9 \times l}{f(BCLK)} - 35 \quad [ns] \quad (l=1 \text{ when 1 wait and } l=2 \text{ when 2 waits.})$$

Electrical Characteristics ( $V_{CC} = 3.3V$ ) $V_{CC} = 3.3V$ **Timing Requirements****( $V_{CC} = 3.0$  to  $3.6V$ ,  $V_{SS} = 0V$  at  $T_{op} = -20$  to  $85^{\circ}C$  unless otherwise specified)****Table 1.31.28. Timer A Input (Count Source 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 "H" pulse width	40		ns
$t_{w(TAL)}$	TAiIn input "L" pulse width	40		ns

**Table 1.31.29. Timer A Input (Gate Input in Timer Mode)**

Symbol	Parameter	Standard		Unit
		Min	Max	
$t_{c(TA)}$	TAiIn input cycle time	400		ns
$t_{w(TAH)}$	TAiIn input "H" pulse width	200		ns
$t_{w(TAL)}$	TAiIn input "L" pulse width	200		ns

**Table 1.31.30. 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 "H" pulse width	100		ns
$t_{w(TAL)}$	TAiIn input "L" pulse width	100		ns

**Table 1.31.31. Timer A Input (External Trigger Input in Pulse Width Modulation Mode)**

Symbol	Parameter	Standard		Unit
		Min	Max	
$t_{w(TAH)}$	TAiIn input "H" pulse width	100		ns
$t_{w(TAL)}$	TAiIn input "L" pulse width	100		ns

**Table 1.31.32. Timer A Input (Counter Increment/decrement 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 "H" pulse width	1000		ns
$t_{w(UPL)}$	TAiOUT input "L" pulse width	1000		ns
$t_{su(UP-TiN)}$	TAiOUT input setup time	400		ns
$t_{h(TiN-UP)}$	TAiOUT input hold time	400		ns

Electrical Characteristics ( $V_{CC} = 3.3V$ ) $V_{CC} = 3.3V$ **Timing Requirements****( $V_{CC} = 3.0$  to  $3.6V$ ,  $V_{SS} = 0V$  at  $T_{opr} = -20$  to  $85^{\circ}C$  unless otherwise specified)****Table 1.31.33. Timer B input (Count Source 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 "H" pulse width (counted on one edge)	40		ns
$t_{w(TBL)}$	TBiIn input "L" 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 "H" pulse width (counted on both edges)	80		ns
$t_{w(TBL)}$	TBiIn input "L" pulse width (counted on both edges)	80		ns

**Table 1.31.34. 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 "H" pulse width	200		ns
$t_{w(TBL)}$	TBiIn input "L" pulse width	200		ns

**Table 1.31.35. 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 "H" pulse width	200		ns
$t_{w(TBL)}$	TBiIn input "L" pulse width	200		ns

**Table 1.31.36. A-D Trigger Input**

Symbol	Parameter	Standard		Unit
		Min	Max	
$t_{c(AD)}$	$\overline{ADTRG}$ input cycle time (trigger available at minimum and above)	1000		ns
$t_{w(ADL)}$	$\overline{ADTRG}$ input "L" pulse width	125		ns

**Table 1.31.37. Serial I/O**

Symbol	Parameter	Standard		Unit
		Min	Max	
$t_{c(CK)}$	CLKi input cycle time	200		ns
$t_{w(CKH)}$	CLKi input "H" pulse width	100		ns
$t_{w(CKL)}$	CLKi input "L" 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 1.31.38. External Interrupt  $\overline{INTi}$  input**

Symbol	Parameter	Standard		Unit
		Min	Max	
$t_{w(INH)}$	$\overline{INTi}$ input "H" pulse width	250		ns
$t_{w(INL)}$	$\overline{INTi}$ input "L" pulse width	250		ns

Electrical Characteristics (V<sub>CC</sub> = 3.3V)V<sub>CC</sub> = 3.3V

## Switching Characteristics

(V<sub>CC</sub> = 3.0 to 3.6V, V<sub>SS</sub> = 0V at Topr = -20 to 85°C, unless otherwise specified)

Table 1.31.39. Memory Expansion and Microprocessor Modes (without a wait)

Symbol	Parameter	Measurement condition	Standard		Unit
			Min	Max	
td(BCLK-AD)	Address output delay time	Figure 1.31.1		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 1)		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 1)		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 1)		ns
th(WR-DB)	Data output hold time (WR standard)		(Note 1)		ns
tw(WR)	Write pulse width	(Note 1)		ns	

## Notes :

1. A value can be obtained from the following expressions, according to BCLK frequency.

$$td(DB - WR) = \frac{10^9}{f_{(BCLK)}} - 20 \quad [\text{ns}]$$

$$th(WR - DB) = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$th(WR - AD) = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$th(WR - CS) = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$tw(WR) = \frac{10^9}{f_{(BCLK)} \times 2} - 15 \quad [\text{ns}]$$

Electrical Characteristics (V<sub>CC</sub> = 3.3V)V<sub>CC</sub> = 3.3V

## Switching Characteristics

(V<sub>CC</sub> = 3.0 to 3.6V, V<sub>SS</sub> = 0V at Topr = -20 to 85°C unless otherwise specified)

**Table 1.31.40. Memory Expansion and Microprocessor Modes**  
(with a wait, accessing external memory)

Symbol	Parameter	Measurement condition	Standard		Unit
			Min	Max	
td(BCLK-AD)	Address output delay time	Figure 1.31.1		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 1)		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 1)		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 1)		ns
th(WR-DB)	Data output hold time (WR standard)		(Note 1)		ns
tw(WR)	Write pulse width		(Note 1)		ns

## Notes :

1. A value can be obtained from the following expressions, according to BCLK frequency.

$$td(DB - WR) = \frac{10^9 \times n}{f_{(BCLK)}} - 20 \quad [\text{ns}] \quad (n=1 \text{ when 1 wait, } n=2 \text{ when 2 waits and } n=3 \text{ when 3 waits.})$$

$$th(WR - DB) = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$th(WR - AD) = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$th(WR - CS) = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$tw(WR) = \frac{10^9 \times n}{f_{(BCLK)} \times 2} - 15 \quad [\text{ns}] \quad (n=1 \text{ when 1 wait, } n=3 \text{ when 2 waits and } n=5 \text{ when 3 waits.})$$

Electrical Characteristics (V<sub>CC</sub> = 3.3V)V<sub>CC</sub> = 3.3V

## Switching Characteristics

(V<sub>CC</sub> = 3.0 to 3.6V, V<sub>SS</sub> = 0V at T<sub>opr</sub> = -20 to 85°C unless otherwise specified)

**Table 1.31.41. Memory Expansion and Microprocessor Modes**  
(with a wait, accessing external memory, multiplex bus space selected)

Symbol	Parameter	Measurement condition	Standard		Unit
			Min	Max	
t <sub>d</sub> (BCLK-AD)	Address output delay time	Figure 1.31.1		18	ns
t <sub>h</sub> (BCLK-AD)	Address output hold time (BCLK standard)		0		ns
t <sub>h</sub> (RD-AD)	Address output hold time (RD standard)		(Note 1)		ns
t <sub>h</sub> (WR-AD)	Address output hold time (WR standard)		(Note 1)		ns
t <sub>d</sub> (BCLK-CS)	Chip-select output delay time			18	ns
t <sub>h</sub> (BCLK-CS)	Chip-select output hold time (BCLK standard)		0		ns
t <sub>h</sub> (RD-CS)	Chip-select output hold time (RD standard)		(Note 1)		ns
t <sub>h</sub> (WR-CS)	Chip-select output hold time (WR standard)		(Note 1)		ns
t <sub>d</sub> (BCLK-RD)	RD signal output delay time			18	ns
t <sub>h</sub> (BCLK-RD)	RD signal output hold time		-3		ns
t <sub>d</sub> (BCLK-WR)	WR signal output delay time			18	ns
t <sub>h</sub> (BCLK-WR)	WR signal output hold time		0		ns
t <sub>d</sub> (DB-WR)	Data output delay time (WR standard)		(Note 1)		ns
t <sub>h</sub> (WR-DB)	Data output hold time (WR standard)		(Note 1)		ns
t <sub>d</sub> (BCLK-ALE)	ALE signal output delay time (BCLK standard)			18	ns
t <sub>h</sub> (BCLK-ALE)	ALE signal output hold time (BCLK standard)		-2		ns
t <sub>d</sub> (AD-ALE)	ALE signal output delay time (address standard)		(Note 1)		ns
t <sub>h</sub> (ALE-AD)	ALE signal output hold time (address standard)		(Note 1)		ns
t <sub>dz</sub> (RD-AD)	Address output high-impedance start time			8	ns

## Notes :

1. A value can be obtained from the following formula, according to BCLK frequency.

$$t_{h(RD-AD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$t_{h(WR-AD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$t_{h(RD-CS)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$t_{h(WR-CS)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$t_{d(DB-WR)} = \frac{10^9 \times m}{f_{(BCLK)} \times 2} - 25 \quad [\text{ns}] \quad (m=3 \text{ when 2 waits and } m=5 \text{ when 3 waits.})$$

$$t_{h(WR-DB)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$t_{d(AD-ALE)} = \frac{10^9}{f_{(BCLK)} \times 2} - 20 \quad [\text{ns}]$$

$$t_{h(ALE-AD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

Electrical Characteristics (V<sub>CC</sub> = 3.3V)V<sub>CC</sub> = 3.3V

## Switching Characteristics

(V<sub>CC</sub> = 3.0 to 3.6V, V<sub>SS</sub> = 0V at Topr = -20 to 85°C unless otherwise specified)

**Table 1.31.42. Memory Expansion and Microprocessor Modes**  
(with a wait, accessing external memory, DRAM space selected)

Symbol	Parameter	Measurement condition	Standard		Unit
			Min	Max	
t <sub>d</sub> (BCLK-RAD)	Row address output delay time	Figure 1.31.1		18	ns
t <sub>h</sub> (BCLK-RAD)	Row address output hold time (BCLK standard)		0		ns
t <sub>d</sub> (BCLK-CAD)	Column address output delay time			18	ns
t <sub>h</sub> (BCLK-CAD)	Column address output hold time (BCLK standard)		0		ns
t <sub>h</sub> (RAS-RAD)	Row address output hold time after RAS output		(Note 1)		ns
t <sub>d</sub> (BCLK-RAS)	RAS output delay time (BCLK standard)			18	ns
t <sub>h</sub> (BCLK-RAS)	RAS output hold time (BCLK standard)		0		ns
t <sub>RP</sub>	RAS "H" hold time		(Note 1)		ns
t <sub>d</sub> (BCLK-CAS)	CAS output delay time (BCLK standard)			18	ns
t <sub>h</sub> (BCLK-CAS)	CAS output hold time (BCLK standard)		0		ns
t <sub>d</sub> (BCLK-DW)	Data output delay time (BCLK standard)			18	ns
t <sub>h</sub> (BCLK-DW)	Data output hold time (BCLK standard)		-3		ns
t <sub>su</sub> (DB-CAS)	CAS after DB output setup time		(Note 1)		ns
t <sub>h</sub> (BCLK-DB)	DB signal output hold time (BCLK standard)		-7		ns
t <sub>su</sub> (CAS-RAS)	CAS output setup time before RAS output (refresh)		(Note 1)		ns

## Notes :

1. A value can be obtained from the following expressions, according to BCLK frequency.

$$t_{h(RAS - RAD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 13 \quad [\text{ns}]$$

$$t_{RP} = \frac{10^9 \times 3}{f_{(BCLK)} \times 2} - 20 \quad [\text{ns}]$$

$$t_{su(DB - CAS)} = \frac{10^9}{f_{(BCLK)}} - 20 \quad [\text{ns}]$$

$$t_{su(CAS - RAS)} = \frac{10^9}{f_{(BCLK)} \times 2} - 13 \quad [\text{ns}]$$

Electrical Characteristics (V<sub>CC</sub> = 3.3V)

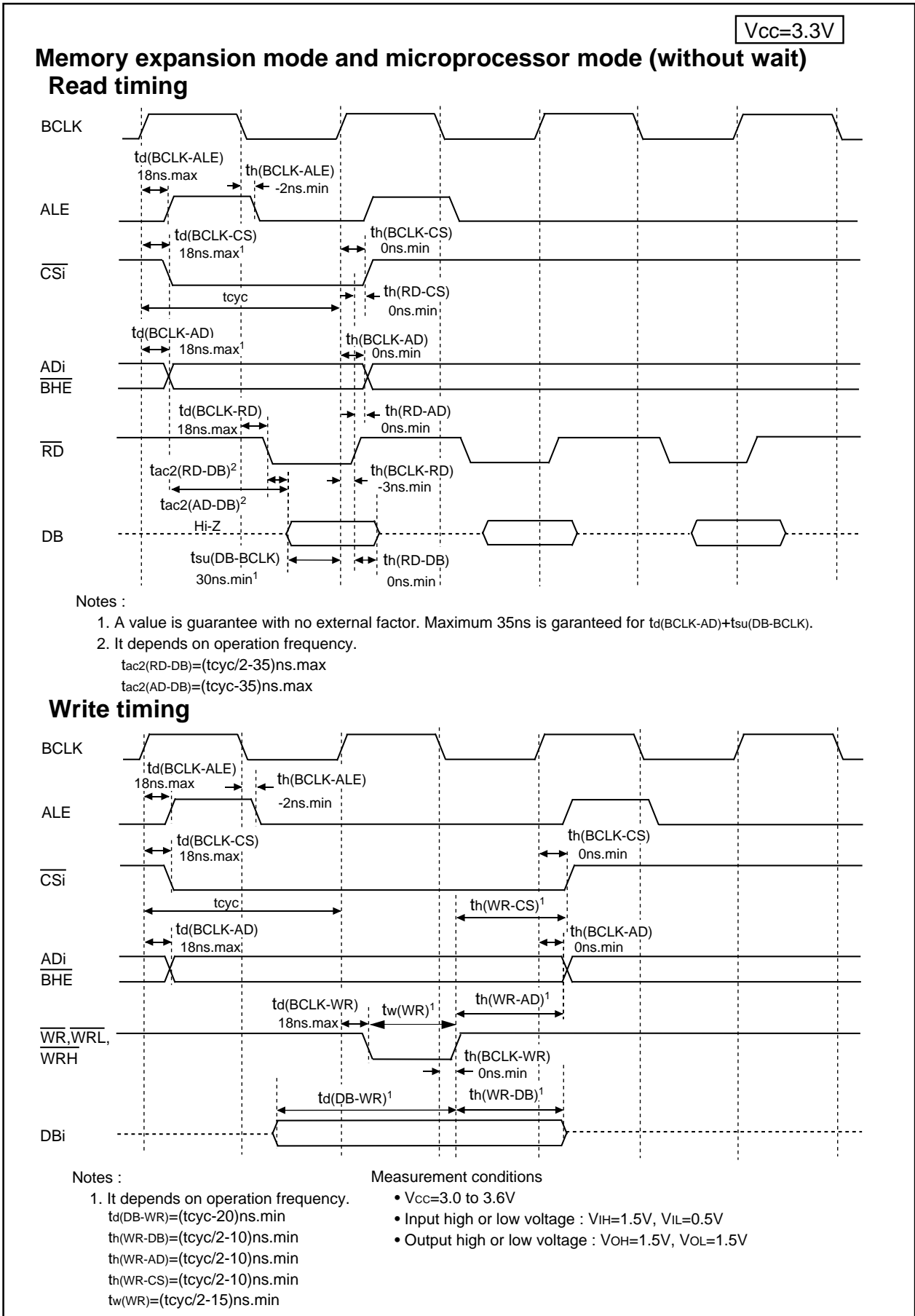


Figure 1.31.10. V<sub>CC</sub>=3.3V Timing Diagram (1)

Electrical Characteristics (V<sub>cc</sub> = 3.3V)

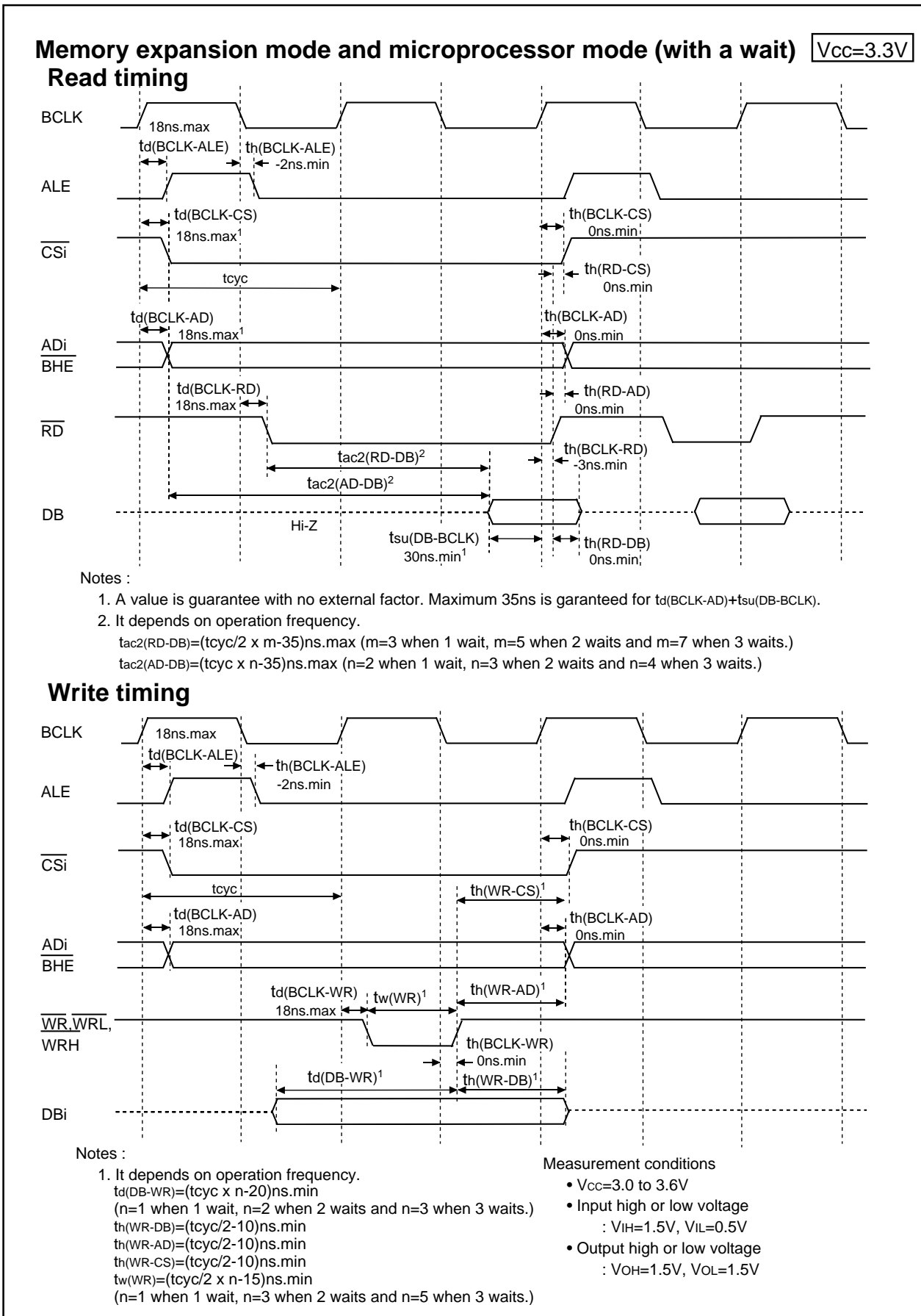


Figure 1.31.11. V<sub>cc</sub>=3.3V Timing Diagram (2)

Electrical Characteristics (V<sub>CC</sub> = 3.3V)

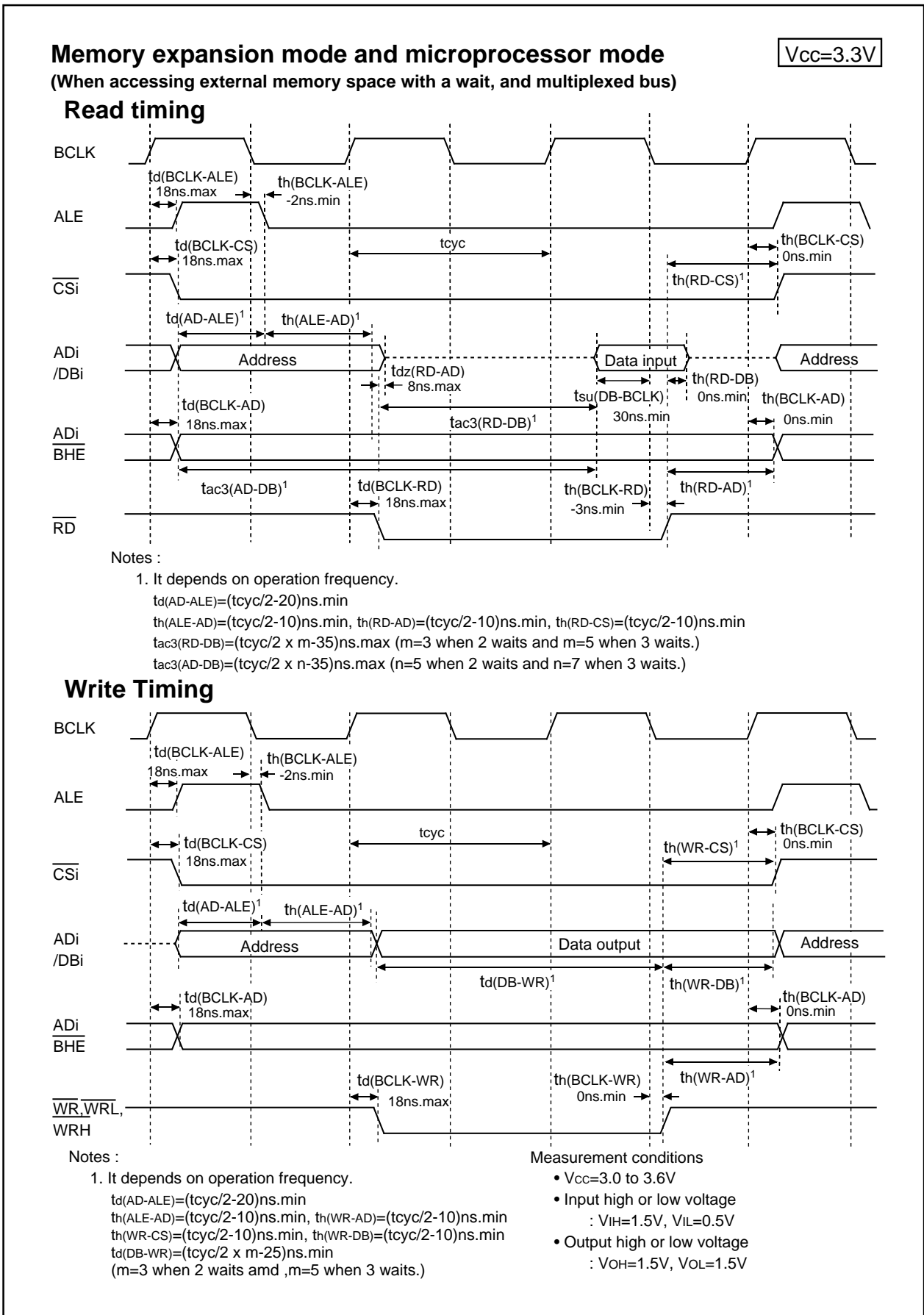


Figure 1.31.12. V<sub>CC</sub>=3.3V Timing Diagram (3)

Electrical Characteristics (V<sub>CC</sub> = 3.3V)

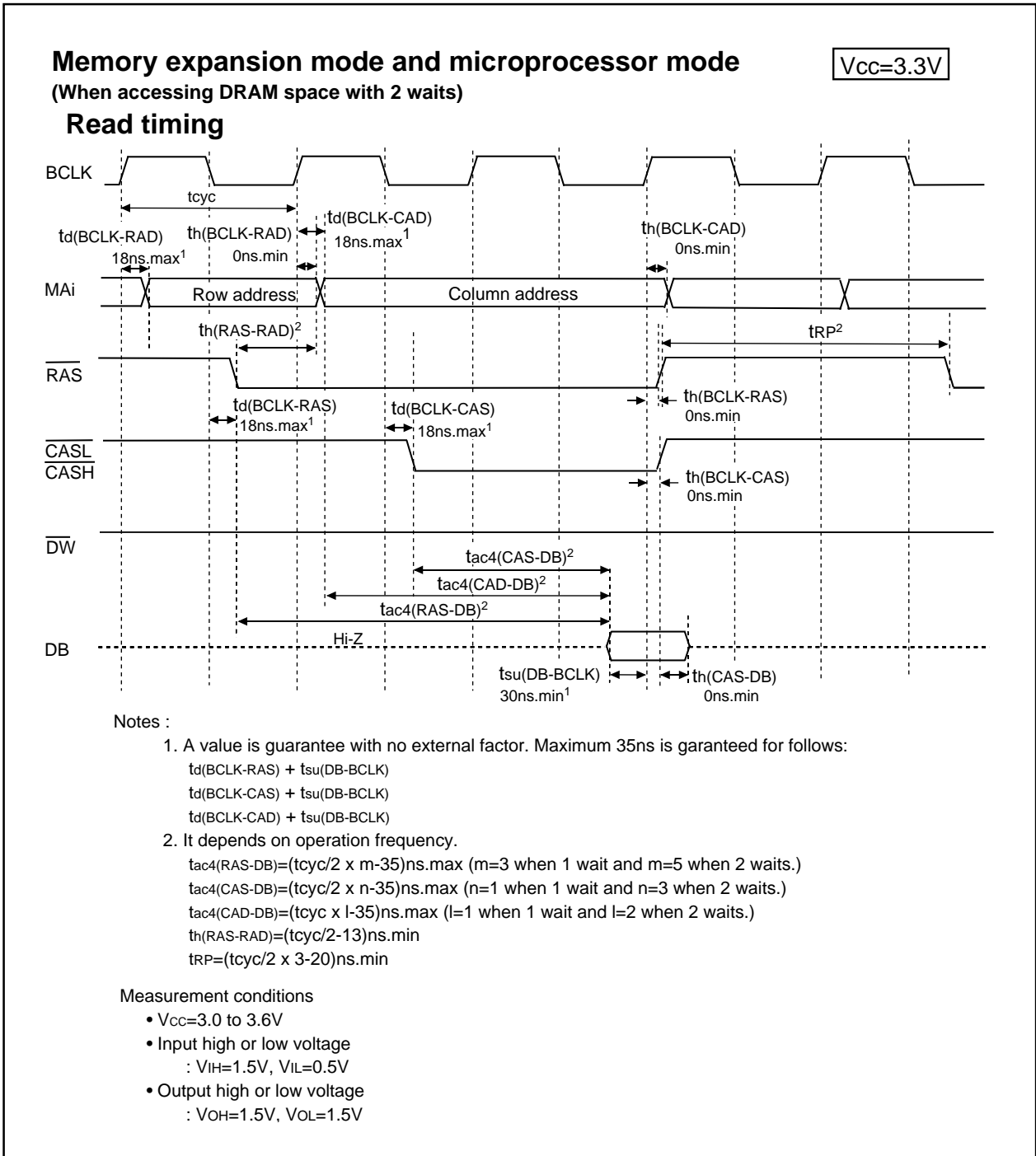


Figure 1.31.13. V<sub>CC</sub>=3.3V Timing Diagram (4)

Electrical Characteristics (V<sub>CC</sub> = 3.3V)

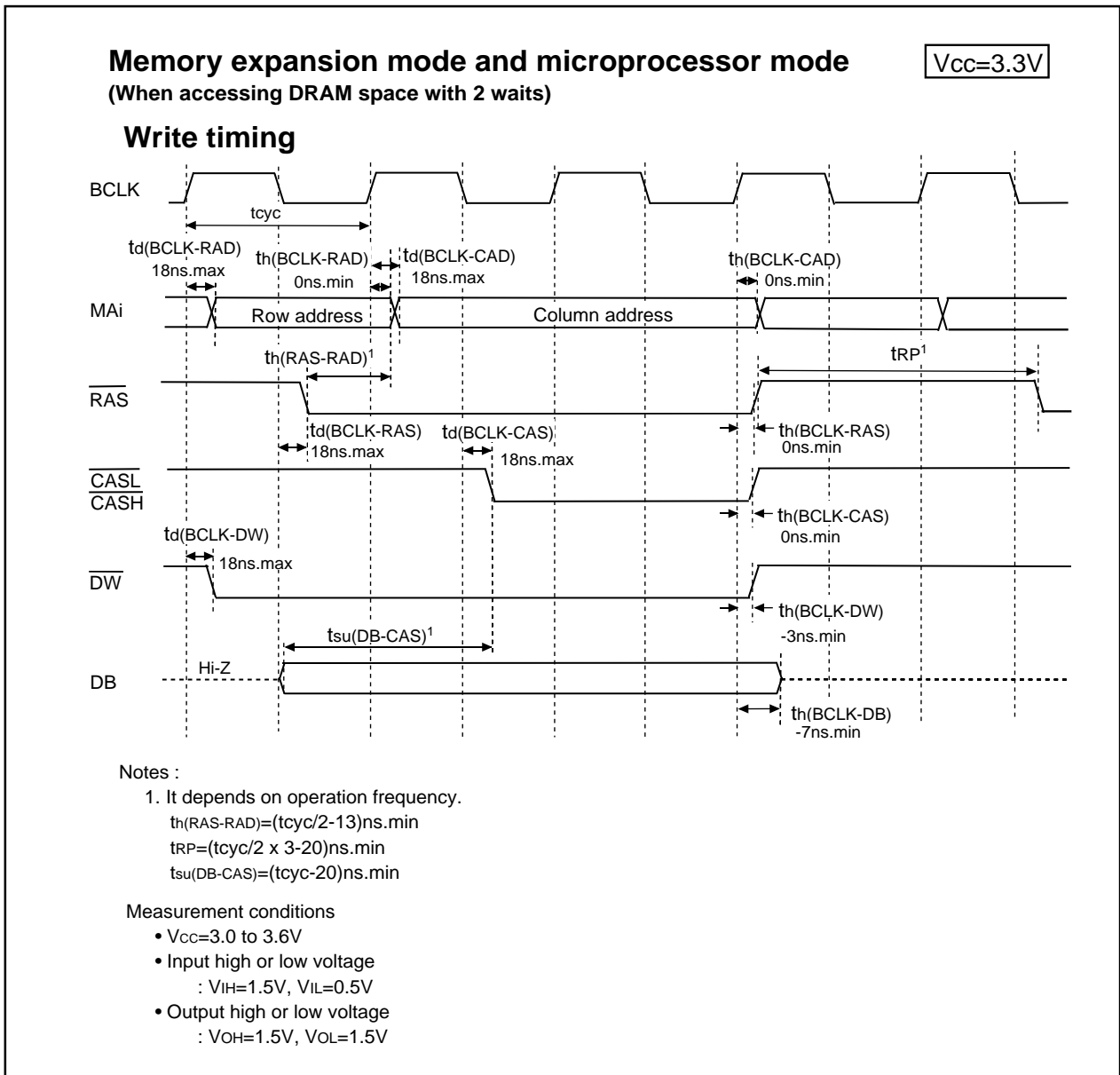


Figure 1.31.14. V<sub>CC</sub>=3.3V Timing Diagram (5)

Electrical Characteristics ( $V_{CC} = 3.3V$ )

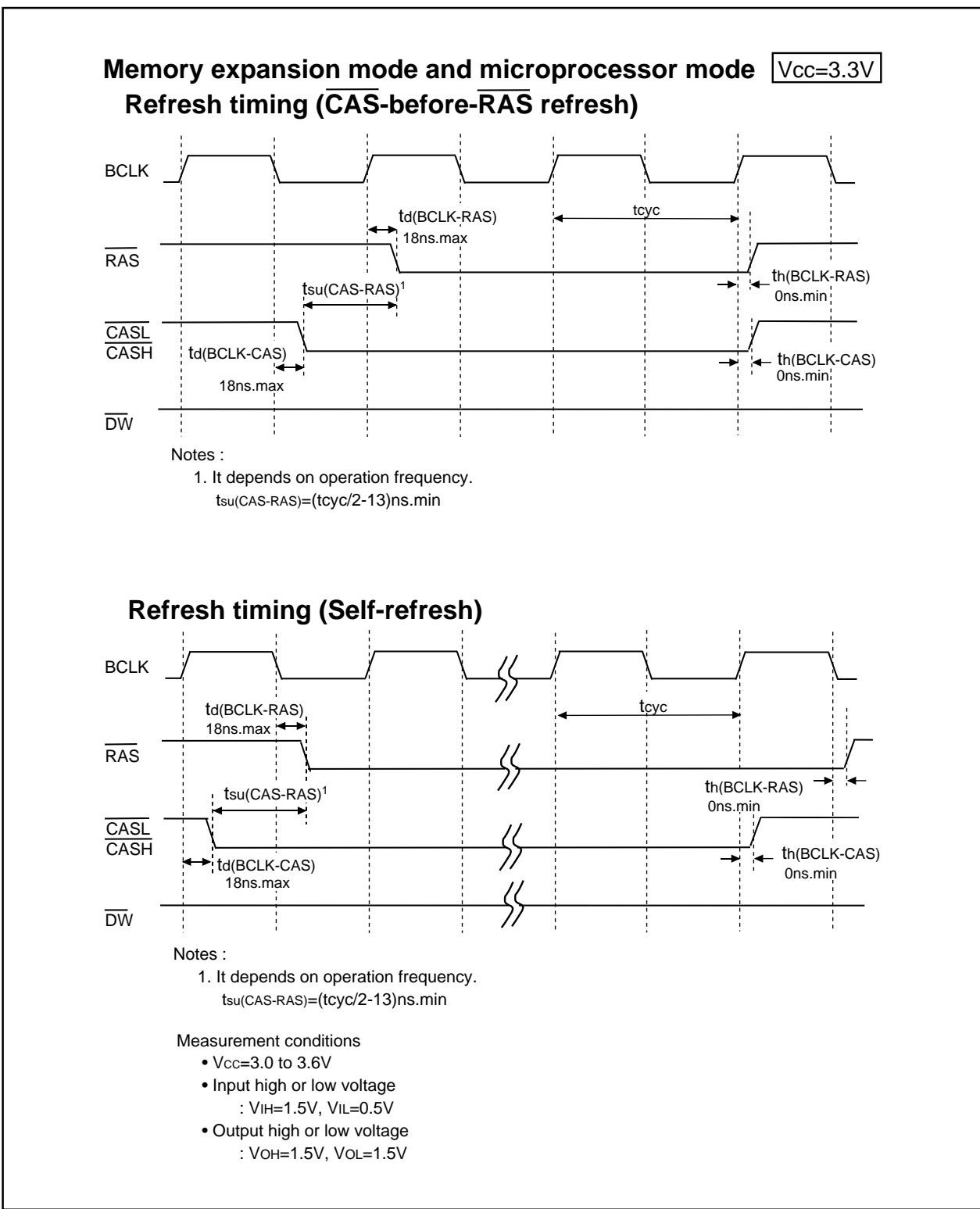


Figure 1.31.15.  $V_{CC}=3.3V$  Timing Diagram (6)

Electrical Characteristics ( $V_{CC} = 3.3V$ )

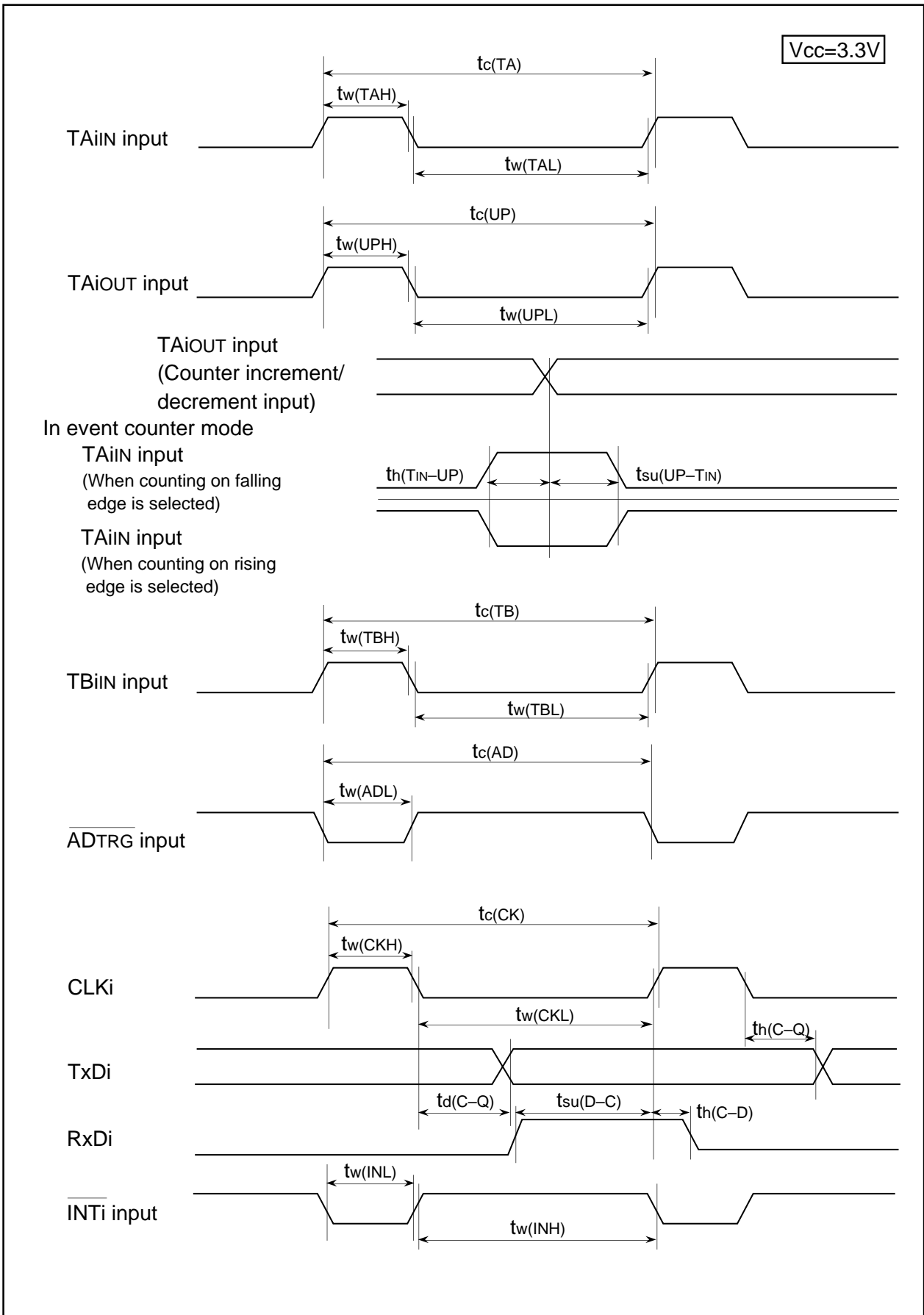


Figure 1.31.16.  $V_{CC}=3.3V$  Timing Diagram (7)

Electrical Characteristics ( $V_{CC} = 3.3V$ )

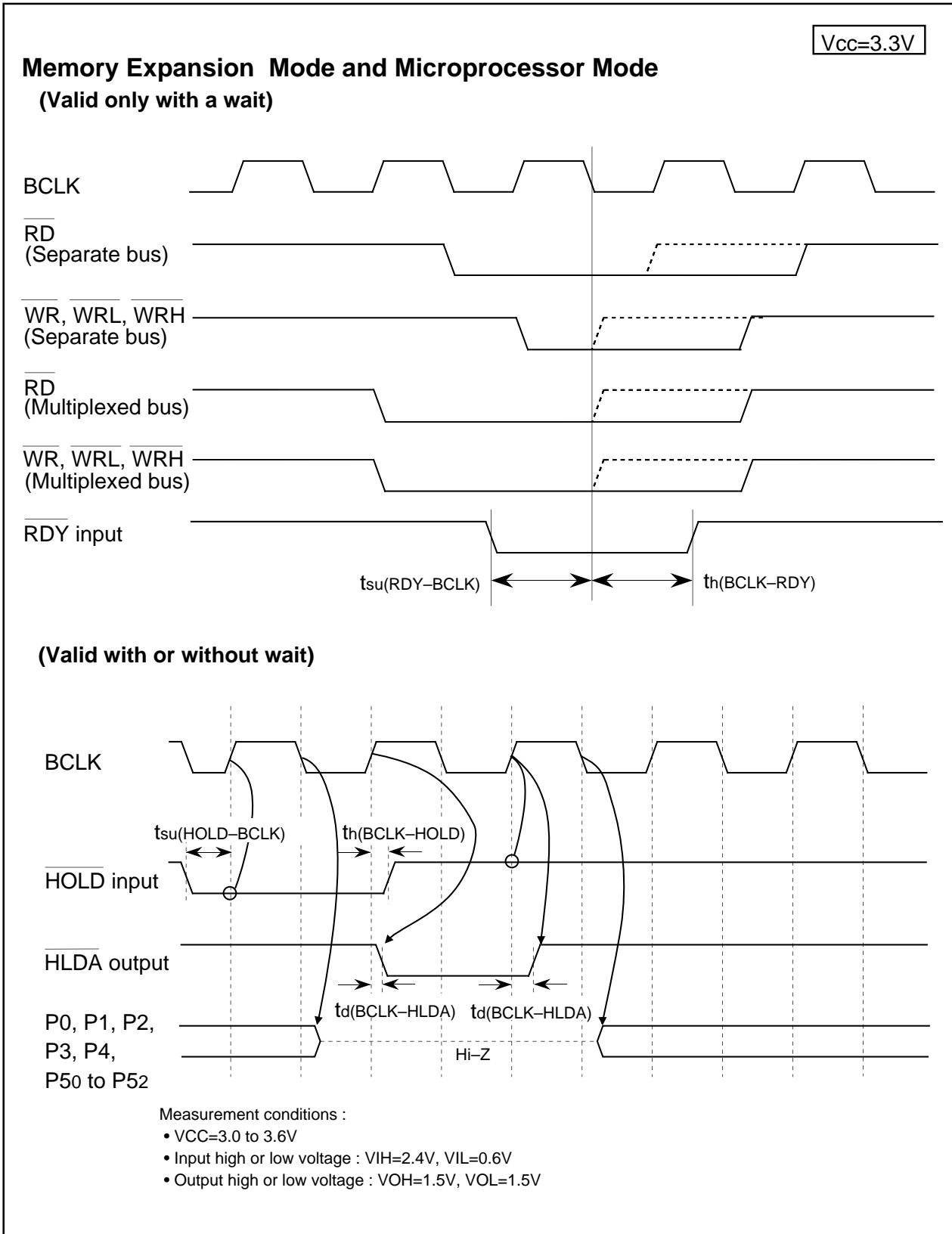


Figure 1.31.17.  $V_{CC}=3.3V$  Timing Diagram (8)

# Register Index

## A

A0 21  
A1 21  
AD00 to AD07 315  
AD0CON0 313  
AD0CON1 314  
AD0CON2 315  
AD10 to AD17 318  
AD1CON0 316  
AD1CON1 317  
AD1CON2 318  
AIER 106

## B

BTSR 244

## C

C0AFS 230  
C0BPR 210  
C0CONR 207  
C0CTLR0 198  
C0CTLR1 201  
C0EIMKR 214  
C0EISTR 215  
C0GMR0 217  
C0GMR1 218  
C0GMR2 219  
C0GMR3 220  
C0GMR4 221  
C0IDR 206  
C0LMAR0 217  
C0LMAR1 218  
C0LMAR2 219  
C0LMAR3 220  
C0LMAR4 221  
C0LMBR0 217  
C0LMBR1 218  
C0LMBR2 219  
C0LMBR3 220  
C0LMBR4 221  
C0MCTL0 to C0MCTL15 222

C0REC 210  
C0SBS 225  
C0SIMKR 213  
C0SISTR 211  
C0SLOT0\_0 226  
C0SLOT0\_1 226  
C0SLOT0\_2 227  
C0SLOT0\_3 227  
C0SLOT0\_4 228  
C0SLOT0\_5 228  
C0SLOT0\_6 to C0SLOT0\_13 229  
C0SLOT0\_14 229  
C0SLOT0\_15 229  
C0SLOT1\_0 226  
C0SLOT1\_1 226  
C0SLOT1\_2 227  
C0SLOT1\_3 227  
C0SLOT1\_4 228  
C0SLOT1\_5 228  
C0SLOT1\_6 to C0SLOT1\_13 229  
C0SLOT1\_14 229  
C0SLOT1\_15 229  
C0SLPR 202  
C0STR 203  
C0TEC 209  
C0TSR 209  
CM0 67, 114  
CM1 68  
CM2 70  
CPSRF 71  
CRCD 329  
CRCIN 329

## D

DA0 to DA1 328  
DACON 328  
DCT0 to DCT3 120  
DM0SL to DM3SL 117  
DMA0 to DMA3 121  
DMD0 to DMD1 119  
DRA0 to DRA3 121

## Register Index

---

DRAMCONT 335  
DRC0 to DRC3 120  
DS 52  
DSA0 to DSA3 121  
DTT 168

### F

FB 21  
FMR0 377

### G

G0BCR0 to G3BCR0 240  
G0BCR1 and G1BCR1 241  
G0BT to G3BT 240  
G0CMP0 to G0CMP3 280  
G0CR to G1CR 275  
G0EMR to G1EMR 277  
G0ERC to G1ERC 278  
G0ETC to G1ETC 277  
G0FE to G3FE 249  
G0FS to G1FS 249  
G0IRF to G1IRF 279  
G0MR to G1MR 276  
G0MSK0 to G0MSK1 280  
G0PO0 to G0PO7 248  
G0POCR0 to G0POCR7 246  
G0RB to G1RB 276  
G0RCRC to G1RCRC 280  
G0RI to G1RI 274  
G0TB to G1TB 279  
G0TCRC to G1TCRC 280  
G0TM0 to G0TM7 246  
G0TMCR0 to G0TMCR7 245  
G0TO to G1TO 274  
G0TPR6 to G0TPR7 245  
G1CMP0 to G1CMP3 280  
G1MSK0 to G1MSK1 280  
G1PO0 to G1PO7 248  
G1POCR0 to G1POCR7 246  
G1TM0 to G1TM7 246  
G1TMCR0 to G1TMCR7 245  
G1TPR6 to G1TPR7 245  
G2BCR1 242  
G2CR 292

G2MR 292  
G2PO0 to G2PO7 248  
G2POCR0 to G2POCR7 247  
G2RB 291  
G2RTP to G3RTP 250  
G2TB 291  
G3BCR1 243  
G3CR 303  
G3FLG 304  
G3MK4 to G3MK7 248  
G3MR 303  
G3PO0 to G3PO7 248  
G3POCR0 to G3POCR7 247  
G3RB 302  
G3TB 302

### I

ICTB 169  
IDB0 to IDB1 168  
IEAR 293  
IECR 293  
IERIF 294  
IETIF 294  
IFSR 104, 183  
IIO0IE to IIO11IE 109  
IIO0IR to IIO11IR 108  
INTB 21  
Interrupt Control 95, 96  
INVC0 166  
INVC1 167  
IPS 359  
ISP 21

### M

MCD 69

### O

ONSF 140

### P

P0 to P15 348  
PC 21  
PCR 359  
PD0 to PD15 347  
PLC0 72

**Register Index**

---

PLC1	73	TA1MR, TA2MR, TA4MR	171
PLV	72	TABSR	139, 155, 170
PM0	49	TB0 to TB5	154
PM1	50	TB0MR to TB5MR	155, 157, 159, 161
PRCR	87	TB2	170
PS0	349	TB2MR	171
PS1	349	TB2SC	169
PS2	350	TBSR	156
PS3	350	TCSPR	71, 141
PS5	351	TRGSR	141, 170
PS6	351		
PS7	352	<b>U</b>	
PS8	352	U0BRG to U4BRG	177
PS9	353	U0C0 to U4C0	178
PSC	356	U0C1 to U4C1	179
PSL0	354	U0MR to U4MR	177
PSL1	354	U0RB to U4RB	176
PSL2	355	U0SMR to U4SMR	179
PSL3	355	U0SMR2 to U4SMR2	180
PUR0	357	U0SMR3 to U4SMR3	181
PUR1	357	U0SMR4 to U4SMR4	182
PUR2	357	U0TB to U4TB	176
PUR3	358	UDF	140
PUR4	358	USP	21
		<b>V</b>	
<b>R</b>		VCT	21
R0	21	VDC0	371
R1	21		
R2	21	<b>W</b>	
R3	21	WCR	58
REFCNT	335	WDC	113
RLVL	97, 127	WDTS	113
RMAD0 to RMAD3	106	<b>X</b>	
ROMCP	375	X0R to X15R	331
		XYC	331
<b>S</b>		<b>Y</b>	
SB	21	Y0R to Y15R	331
SVF	21		
SVP	21		
<b>T</b>			
TA0 to TA4	138		
TA0MR to TA4MR	139, 144, 147, 150, 152		
TA1, TA2, TA4, TA11, TA21, TA41	169		

REVISION HISTORY

M32C/83 GROUP DATA SHEET

Rev.	Date	Description	
		Page	Summary
1.01	12/2002	All	<b>Full-fledged revision</b> <ul style="list-style-type: none"> <li>• Modify the notation system of registers and bits</li> </ul>
		23	<b>Reset</b> <ul style="list-style-type: none"> <li>• Delete the figure "Device's internal status after a reset is cleared".</li> </ul>
		65	<b>System Clock</b> <ul style="list-style-type: none"> <li>• Modify the figure "Clock Generation Circuit".</li> <li>• Add descriptions about the 'PLL clock'.</li> <li>• Modify the figure "Status Transition".</li> </ul>
		88	<b>Interrupt</b> <ul style="list-style-type: none"> <li>• Modify the figure "Intelligent I/O Interrupt and CAN Interrupt".</li> <li>• Add tables 'registers to be used and settings'.</li> <li>• Change symbols of the bits in the interrupt request register.</li> <li>• Change symbols of the bits in the interrupt enable register.</li> </ul>
		137	<b>Timer A</b> <ul style="list-style-type: none"> <li>• Modify the figure "Timer A Configuration".</li> <li>• Add tables 'registers to be used and settings'.</li> </ul>
		154	<b>Timer B</b> <ul style="list-style-type: none"> <li>• Modify the figure "Timer B Configuration".</li> <li>• Add tables 'registers to be used and settings'.</li> </ul>
		163	<b>Three-Phase Control Timer Function</b> <ul style="list-style-type: none"> <li>• Change the bit name, the 'INV17bit' in the INVC1 register to reserved bit.</li> </ul>
		174	<b>Serial I/O</b> <ul style="list-style-type: none"> <li>• Modify the figure "UARTi Block Diagram".</li> <li>• Add the table 'registers to be used and settings' in each mode.</li> <li>• Add distributions about the 'clock-divided synchronous function (GCI mode)'.</li> <li>• Add descriptions about the 'bus conflict detect function (IE mode)'.</li> </ul>
		264	<b>Intelligent I/O</b> <ul style="list-style-type: none"> <li>• Modify the figure "Intelligent I/O Group 0 Block Diagram".</li> <li>• Modify the figure "Intelligent I/O Group 1 Block Diagram".</li> <li>• Modify the figure "Intelligent I/O Group 2 Block Diagram".</li> <li>• Modify the figure "Intelligent I/O Group 3 Block Diagram".</li> <li>• Add the table 'registers and settings' associated with each function and mode.</li> <li>• Add a bit function of 'the BCK0 to BCK1 bit in the G0BCR0 to G3BCR0 register'.                      -Group 0 and 1</li> <li>• Add descriptions about the 'HDLC data processing mode'. -Group 0 and 1</li> <li>• Add distributions about the 'IEBus mode'. -Group2</li> <li>• Add descriptions about the '8-bit and 16-bit clock synchronous serial I/O function'                      -Group3</li> </ul>
		338	<b>A-D Convertor</b> <ul style="list-style-type: none"> <li>• Modify the figure "A-D Convertor Block Diagram".</li> <li>• Add the table 'pin settings'.</li> </ul>
		355	<b>D-A Convertor</b> <ul style="list-style-type: none"> <li>• Add the table 'pin settings'.</li> </ul>
		394	<b>Usage Precaution</b> <ul style="list-style-type: none"> <li>• Add descriptions about the 'PLL synthesizer'.</li> <li>• Add descriptions about the 'Timer A' and 'Timer B'.</li> <li>• Add descriptions about the 'Low-Voltage Operation'.</li> </ul>

REVISION HISTORY

M32C/83 GROUP DATA SHEET

Rev.	Date	Description	
		Page	Summary
1.02	1/2003	2-3 3	<p><b>Overview</b></p> <ul style="list-style-type: none"> <li>• Add -40 to 85°C to 'Operating ambient temperature' row in Table 1.1.1 and 1.1.2.</li> <li>• Delete 8-bit or 16-bit clock synchronous serial I/O:1 channel (group3) on 'Peripheral function' row in Table 1.1.2.</li> </ul>
		33	<p><b>SFR</b></p> <ul style="list-style-type: none"> <li>• Modify 00?0 X0002 to 0000 X0002 on 'value after RESET' column on '017B16' row.</li> </ul>
		78	<p><b>System Clock</b></p> <ul style="list-style-type: none"> <li>• Modify 0 to 1 on 'PLC00' column and '10MHz' row in Table 1.8.2.</li> </ul>
		78	<ul style="list-style-type: none"> <li>• Modify the PLC02 to PLC0 bits and the PLC05 to PLC04 bits to the PLC0 register in the third step in Figure 1.8.13.</li> </ul>
		80	<ul style="list-style-type: none"> <li>• Modify 1 to 0 on 'CM00' column and 'BCLK output' row in Table 1.8.5.</li> </ul>
		117	<p><b>DMAC</b></p> <ul style="list-style-type: none"> <li>• Add the note 3 in Figure 1.11.2.</li> </ul>
		141	<p><b>Timer</b></p> <ul style="list-style-type: none"> <li>• Modify TA4 and TA1 to TA0 and TA2 on the TA1TGL and TA1TGH in the top figure of Table 1.14.5.</li> <li>• Modify TA4 and TA1 to TA1 and TA3 on the TA2TGL and TA2TGH in the top figure of Table 1.14.5.</li> <li>• Modify TA4 and TA1 to TA2 and TA4 on the TA3TGL and TA3TGH in the top figure of Table 1.14.5.</li> <li>• Modify TA4 and TA1 to TA3 and TA0 on the TA4TGL and TA4TGH in the top figure of Table 1.14.5.</li> </ul>
		186	<p><b>Serial I/O</b></p> <ul style="list-style-type: none"> <li>• Modify PD7_0=0 to PD7_2=0 on 'PD7 register' column and 'CLK2 input' row in Table 1.18.4.</li> </ul>
		192	<ul style="list-style-type: none"> <li>• Modify PD7_0=0 to PD7_2=0 on 'PD7 register' column and 'CLK2 input' row in Table 1.19.4.</li> </ul>
		206	<ul style="list-style-type: none"> <li>• Modify a function discription on 'UiRRM' row in Table 1.20.9.</li> </ul>
		207	<ul style="list-style-type: none"> <li>• Modify PD7_2=0 to PD7_0=0 on 'PD7 register' column and 'SRxD2 input' row in Table 1.20.11.</li> <li>• Modify PD7_0=0 to PD7_2=0 on 'PD7 register' column and 'CLK2 input' row in Table 1.20.11.</li> </ul>
		216	<ul style="list-style-type: none"> <li>• Modify PS3_4=0 to PS3_5=0 on 'PS3 register' column and 'CLK4 input' row in Table 1.20.23.</li> </ul>
		226	<p><b>CAN Module</b></p> <ul style="list-style-type: none"> <li>• Modify PSL2_2=0 to PSL2_1=0 on 'PSL1 and PSL2 registers' column and 'P82' row in Table 1.21.2.</li> </ul>
		296	<p><b>Intelligent I/O</b></p> <ul style="list-style-type: none"> <li>• Modify Setting value of the GiPO0 register to Setting value of the GiPOk register as n and m on the second figure in Figure 1.22.26.</li> </ul>
		304	<ul style="list-style-type: none"> <li>• Modify RxD to ISRxD on 'IPOL' row and TxD to ISTxD on 'OPOL' row in Figure 1.22.33.</li> </ul>
		315	<ul style="list-style-type: none"> <li>• Modify IPS=1 to IPS1=1 on IPS registers column and 'P112' row in Table 1.22.26.</li> </ul>
		317	<ul style="list-style-type: none"> <li>• Modify TCRCRC to TCRCE on 'CRC' row in Table 1.22.28.</li> <li>• Delete SIOiTR and SIOiRR and add SRTiR in note 3 in Table 1.22.28.</li> </ul>
		320	<ul style="list-style-type: none"> <li>• Modify IER to OER in note 1 in the second figure of Figure 1.22.42.</li> </ul>

REVISION HISTORY

M32C/83 GROUP DATA SHEET

Rev.	Date	Description	
		Page	Summary
1.02	1/2003	324 334  364 385 388  390 391 393  394  398 400  429	<ul style="list-style-type: none"> <li>• Modify SIOiTR to SIO2TR and SIO5RR to SIO2RR in Table 1.22.30 and 1.22.36.</li> <li>• Modify GiCR to G3CR in Table 1.22.41.</li> </ul> <p><b>DRAMC</b></p> <ul style="list-style-type: none"> <li>• Modify SRDF to SREF in note 3 in Figure 1.27.1.</li> <li>• Modify IOUTC10 to OUTC10 on 'PSC_3' row in Figure 1.28.14.</li> <li>• Modify P0 to P5 to P1 in note 1 in Table 1.28.17.</li> </ul> <p><b>Programable I/O Port</b></p> <ul style="list-style-type: none"> <li>• Modify INPC1 to INPC11 on 'PS1 register' column and 'Bit 4' row in Table 1.28.4.</li> <li>• Modify INPC0 to INPC02 on 'PS2 register' column and 'Bit 0' row in table 1.28.5.</li> <li>• Modify ISCLK input to ISCLK0 input on 'Bit 1' row in table 1.28.12.</li> </ul> <p><b>Usage Precaution</b></p> <ul style="list-style-type: none"> <li>• Modify PM0 to PM00 in "HOLD Signal"</li> <li>• Modify all SP to ISP in (1) SP Setting of "Interrupts".</li> <li>• Modify all TAI to TBI in 1. Timer Mode and Event Counter Mode of "Timer B".</li> <li>• Modify the CAN module to the microcomputer in "Resetting CNVSS Pin with H".</li> <li>• Delete a discription of 'Difference between Flash Memory version and Masked ROM'</li> </ul> <p><b>Electric Caractistics</b></p> <ul style="list-style-type: none"> <li>• Modify IOH=5mA to IOL=5mA on 'VOL' row and 'Mesurement Condition' column in Table 1.31.3.</li> </ul>

MITSUBISHI SEMICONDUCTORS  
M32C/83 Group DATA SHEET REV. 1.02

---

December 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