To all our customers

Regarding the change of names mentioned in the document, such as Mitsubishi Electric and Mitsubishi XX, to Renesas Technology Corp.

The semiconductor operations of Hitachi and Mitsubishi Electric were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Mitsubishi Electric, Mitsubishi Electric Corporation, Mitsubishi Semiconductors, and other Mitsubishi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Note : Mitsubishi Electric will continue the business operations of high frequency & optical devices and power devices.

Renesas Technology Corp.
Customer Support Dept.
April 1, 2003

M32C/83 GROUP
DATA SHEET

REV.1.02

Mitsubishi Electric proprietary information

RENESAS
Renesas Technology Corp.

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

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 ($f(X_{IN})$=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, P8$_5$ is used as an input port)
	Multifunction timer Output Input Three-phase motor control output		 16 bits x 5 channels (TA0, TA1, TA2, TA3, TA4) 16 bits x 6 channels (TB0, TB1, TB2, TB3, TB4, TB5) 1 circuit
	Intelligent I/O Time measurement function Waveform generation function Bit modulation PWM Real-time port Communication function		4 groups 12 channels (group 0: 8 channels, group1: 4 channels) 28 channels (group 0: 4 channels, group1,2,3: 8 channels each) 16 channels (group 2,3: 8 channels each) 16 channels (group 2,3: 8 channels each) • Clock synchronous serial I/O, UART: 2 channels (group 0 and 1) • HDLC data processing : 2 channels (group 0 and 1) • Clock synchronous variable length serial I/O:1 channel (group 2) • IE bus[1] : 1 channel (group 2) • 8-bit or 16-bit clock synchronous serial I/O : 1 channel (group 3)
	Serial I/O		5 channels (UART0 to UART4) Clock synchronous,Clock asynchronous,IE Bus[1], I^{2}C Bus [2]
	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		$\overline{CAS}$-before-$\overline{RAS}$ 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 • Main clock generation circuit • Sub clock generation circuit The above circuits include an internal feedback resistance and external ceramic resonator/crystal oscillator. • Ring oscillator (for the main clock oscillator stop detect function) • PLL frequency synthesizer
Electric charact-eristics	Supply voltage		4.2 to 5.5V ($f(X_{IN})$=30MHz, VDC on) 3.0 to 5.5V ($f(X_{IN})$=20MHz, VDC on) 3.0 to 3.6V ($f(X_{IN})$=20MHz, VDC off)
	Power consumption		38mA ($f(X_{IN})$=30MHz with no wait,Vcc=5V) 26mA ($f(X_{IN})$=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^{o}C, –40 to 85^{o}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^{2}C Bus is a trademark of Koninklijke Philips Electronics N. V.

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

Item			Performance
CPU	Basic instructions		108 instructions
	Shortest instruction execution time		33 ns ($f(X_{IN})$=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, P8$_5$ is used as an input port)
	Multifunction timer Output Input Three-phase motor control output		 16 bits x 5 channels (TA0, TA1, TA2, TA3, TA4) 16 bits x 6 channels (TB0, TB1, TB2, TB3, TB4, TB5) 1 circuit
	Intelligent I/O Time measurement function Waveform generation function Bit modulation PWM Real time port Communication function		4 groups 5 channels (group 0: 3 channels, group1: 2 channels) 10 channels (group 0,3: 2 channels each, group1,2: 3 channels each) 5 channels (group 2 : 3 channels, group 3 : 2 channels) 5 channels (group 2 : 3 channels, group 3 : 2 channels) • Clock synchronous serial I/O, UART: 2 channels (group 0 and 1) • HDLC data processing : 2 channels (group 0 and 1) • Clock synchronous variable length serial I/O:1 channel (group 2) • IE bus^1 : 1 channel (group 2)
	Serial I/O		5 channels (UART0 to UART4) Clock synchronous,Clock asynchronous,IE Bus ^{1}, I^{2}C Bus ^{2}
	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		$\overline{CAS}$-before-$\overline{RAS}$ 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 • Main clock generation circuit • Sub clock generation circuit The above circuits include an internal feedback resistance and external ceramic resonator/crystal oscillator. • Ring oscillator (for the main clock oscillator stop detect function) • PLL frequency synthesizer
Electric charac-teristics	Supply voltage		4.2 to 5.5V ($f(X_{IN})$=30MHz, VDC on) 3.0 to 5.5V ($f(X_{IN})$=20MHz, VDC on) 3.0 to 3.6V ($f(X_{IN})$=20MHz, VDC off)
	Power consumption		38mA ($f(X_{IN})$=30MHz with no wait,Vcc=5V) 26mA ($f(X_{IN})$=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

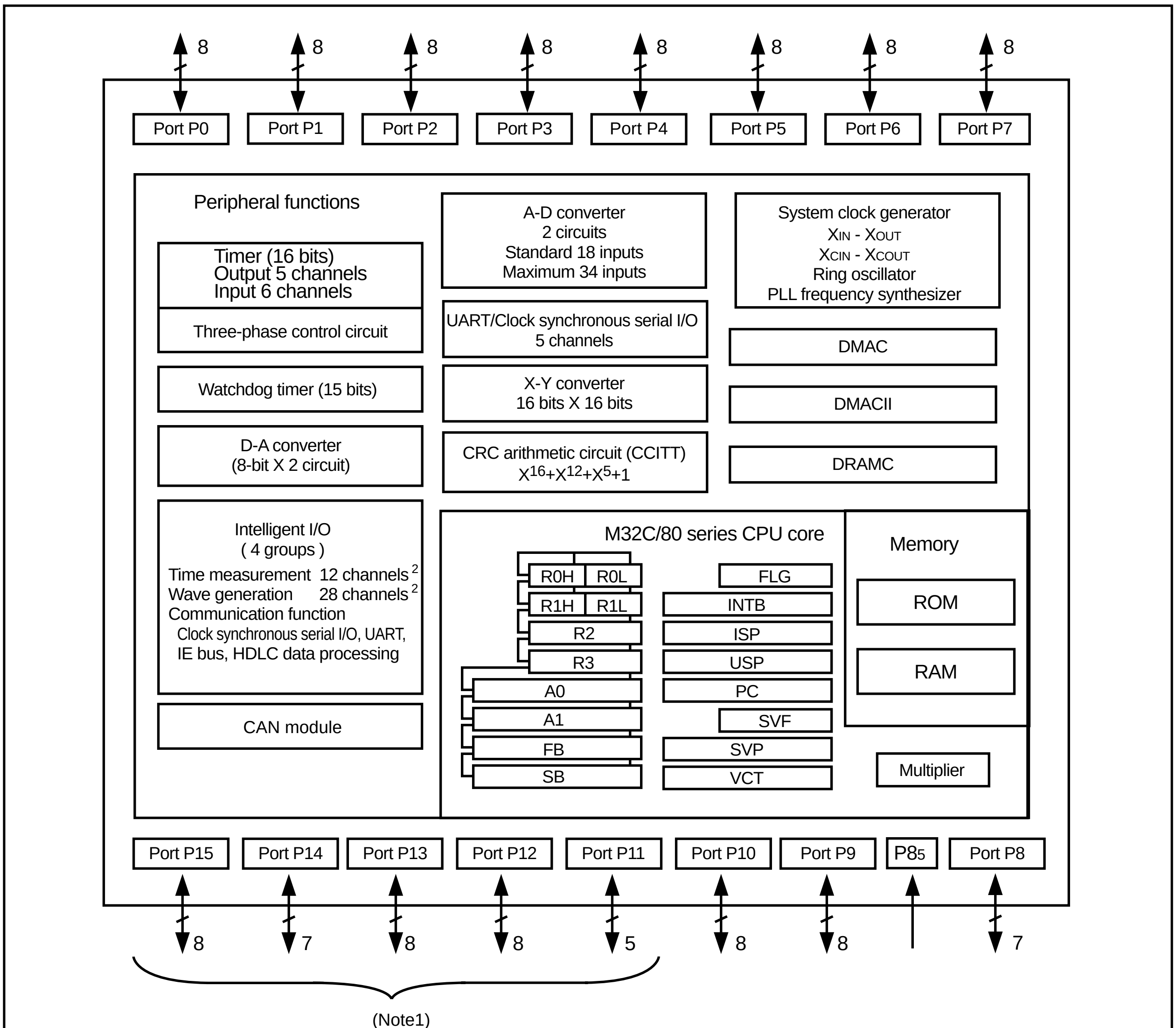
Notes :

1. IE Bus is a trademark of NEC Corporation.
2. I^{2}C Bus is a trademark of Koninklijke Philips Electronics N. V.

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 interruput, timer, serial I/O, D-A converter, DMAC, CRC calculation circuit, A-D converter, DRAMC, intelligent I/O and I/O ports.



Notes :
1. Ports P11 to P15 are provided in the 144-pin package only.
2. For the 144-pin package only.

Figure 1.1.1. M32C/83 Group Block Diagram

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

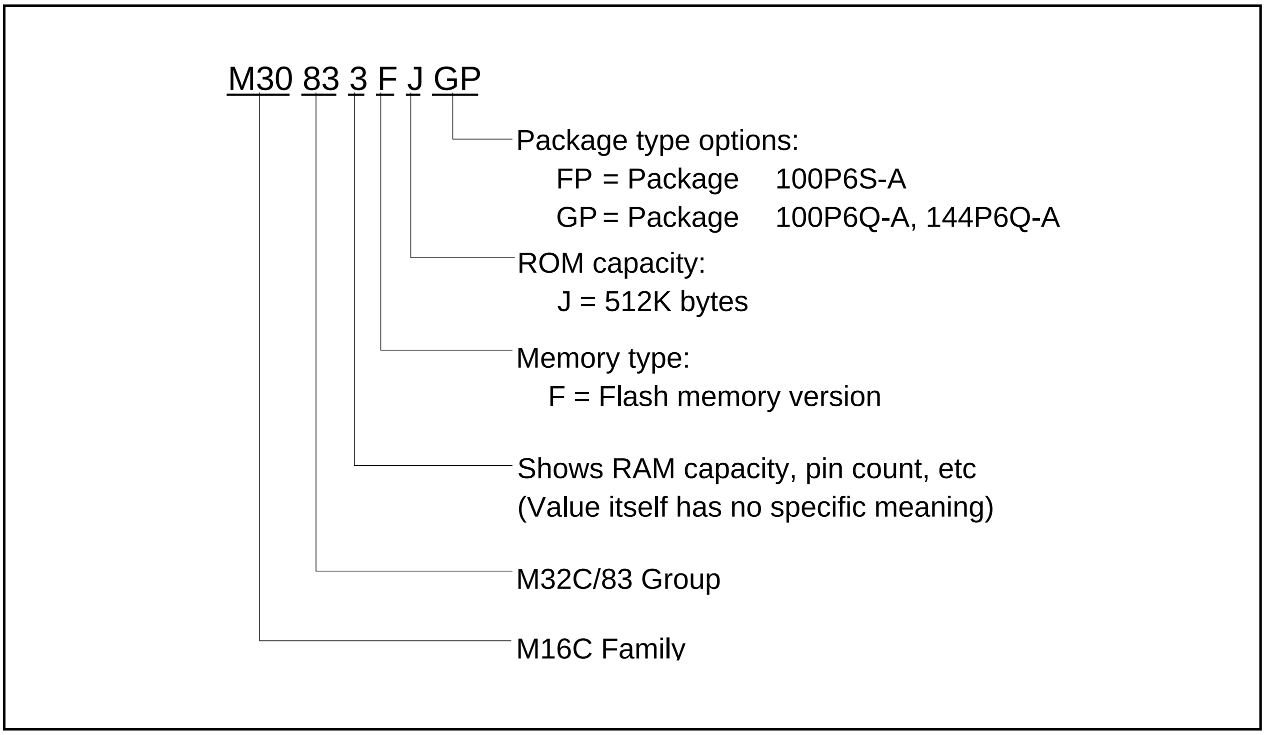
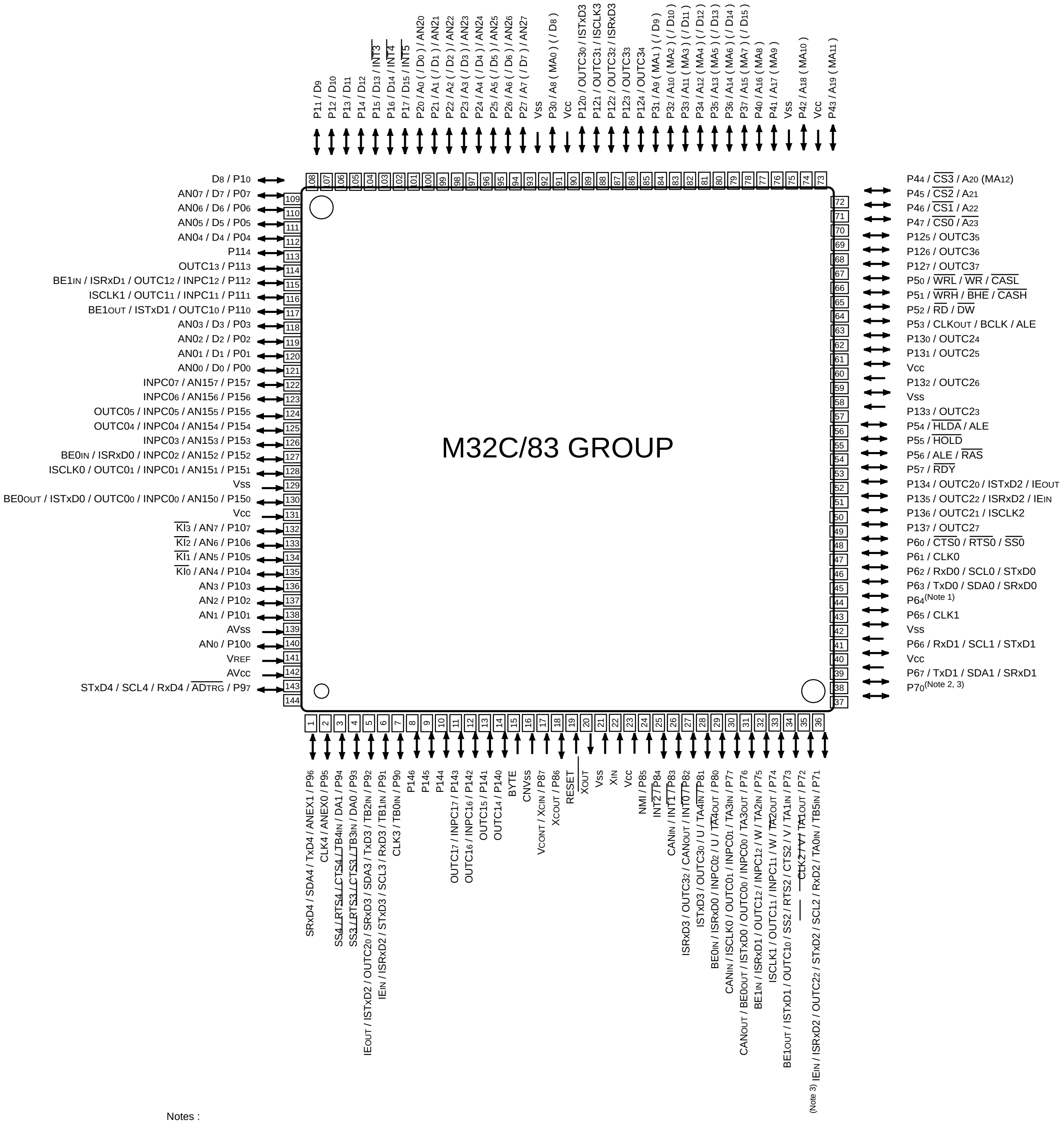


Figure 1.1.2. Product Numbering System

Pin Assignments and Descriptions

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


M32C/83 GROUP
108 D8 / P10
109 AN07 / D7 / P07
110 AN06 / D6 / P06
111 AN05 / D5 / P05
112 AN04 / D4 / P04
113 P114
114 OUTC13 / P113
115 BE1IN / ISRxD1 / OUTC12 / INPC12 / P112
116 ISCLK1 / OUTC11 / INPC11 / P111
117 BE1OUT / ISTxD1 / OUTC10 / P110
118 AN03 / D3 / P03
119 AN02 / D2 / P02
120 AN01 / D1 / P01
121 AN00 / D0 / P00
122 INPC07 / AN157 / P157
123 INPC06 / AN156 / P156
124 OUTC05 / INPC05 / AN155 / P155
125 OUTC04 / INPC04 / AN154 / P154
126 INPC03 / AN153 / P153
127 BE0IN / ISRxD0 / INPC02 / AN152 / P152
128 ISCLK0 / OUTC01 / INPC01 / AN151 / P151
129 Vss
130 BE0OUT / ISTxD0 / OUTC00 / INPC00 / AN150 / P150
131 Vcc
132 KI3 / AN7 / P107
133 KI2 / AN6 / P106
134 KI1 / AN5 / P105
135 KI0 / AN4 / P104
136 AN3 / P103
137 AN2 / P102
138 AN1 / P101
139 AVss
140 AN0 / P100
141 VREF
142 AVcc
143 STxD4 / SCL4 / RxD4 / ADTRG / P97
144
107 P11 / D9
106 P12 / D10
105 P13 / D11
104 P14 / D12
103 P15 / D13 / INT3
102 P16 / D14 / INT4
101 P17 / D15 / INT5
100 P20 / A0 (/ D0) / AN20
99 P21 / A1 (/ D1) / AN21
98 P22 / A2 (/ D2) / AN22
97 P23 / A3 (/ D3) / AN23
96 P24 / A4 (/ D4) / AN24
95 P25 / A5 (/ D5) / AN25
94 P26 / A6 (/ D6) / AN26
93 P27 / A7 (/ D7) / AN27
92 Vss
91 P30 / A8 (MA0) (/ D8)
90 Vcc
89 P120 / OUTC30 / ISTxD3
88 P121 / OUTC31 / ISCLK3
87 P122 / OUTC32 / ISRxD3
86 P123 / OUTC33
85 P124 / OUTC34
84 P31 / A9 (MA1) (/ D9)
83 P32 / A10 (MA2) (/ D10)
82 P33 / A11 (MA3) (/ D11)
81 P34 / A12 (MA4) (/ D12)
80 P35 / A13 (MA5) (/ D13)
79 P36 / A14 (MA6) (/ D14)
78 P37 / A15 (MA7) (/ D15)
77 P40 / A16 (MA8)
76 P41 / A17 (MA9)
75 Vss
74 P42 / A18 (MA10)
73 Vcc
72 P43 / A19 (MA11)
P44 / CS3 / A20 (MA12)
71 P45 / CS2 / A21
70 P46 / CS1 / A22
69 P47 / CS0 / A23
68 P125 / OUTC35
67 P126 / OUTC36
66 P127 / OUTC37
65 P50 / WRL / WR / CASL
64 P51 / WRH / BHE / CASH
63 P52 / RD / DW
62 P53 / CLKOUT / BCLK / ALE
61 P130 / OUTC24
60 P131 / OUTC25
59 Vcc
58 P132 / OUTC26
57 Vss
56 P133 / OUTC23
55 P54 / HLDA / ALE
54 P55 / HOLD
53 P56 / ALE / RAS
52 P57 / RDY
51 P134 / OUTC20 / ISTxD2 / IEOUT
50 P135 / OUTC22 / ISRxD2 / IEIN
49 P136 / OUTC21 / ISCLK2
48 P137 / OUTC27
47 P60 / CTS0 / RTS0 / SS0
46 P61 / CLK0
45 P62 / RxD0 / SCL0 / STxD0
44 P63 / TxD0 / SDA0 / SRxD0
43 P64(Note 1)
42 P65 / CLK1
41 Vss
40 P66 / RxD1 / SCL1 / STxD1
39 Vcc
38 P67 / TxD1 / SDA1 / SRxD1
37 P70(Note 2, 3)
1 SRxD4 / SDA4 / TxD4 / ANEX1 / P96
2 CLK4 / ANEX0 / P95
3 SS4 / RTS4 / CTS4 / TB4IN / DA1 / P94
4 SS3 / RTS3 / CTS3 / TB3IN / DA0 / P93
5 IEOUT / ISTxD2 / OUTC20 / SRxD3 / SDA3 / TxD3 / TB2IN / P92
6 IEIN / ISRxD2 / STxD3 / SCL3 / RxD3 / TB1IN / P91
7 CLK3 / TB0IN / P90
8 P146
9 P145
10 P144
11 OUTC17 / INPC17 / P143
12 OUTC16 / INPC16 / P142
13 OUTC15 / P141
14 OUTC14 / P140
15 BYTE
16 CNVss
17 VCONT / XCIN / P87
18 XCOUT / P86
19 RESET
20 XOUT
21 Vss
22 XIN
23 Vcc
24 NMI / P85
25 INT2 / P84
26 CANIN / INT1 / P83
27 ISRxD3 / OUTC32 / CANOUT / INT0 / P82
28 ISTxD3 / OUTC30 / U / TA4IN / P81
29 BE0IN / ISRxD0 / INPC02 / U / TA4OUT / P80
30 CANIN / ISCLK0 / OUTC01 / INPC01 / TA3IN / P77
31 CANOUT / BE0OUT / ISTxD0 / OUTC00 / INPC00 / TA3OUT / P76
32 BE1IN / ISRxD1 / OUTC12 / INPC12 / W / TA2IN / P75
33 ISCLK1 / OUTC11 / INPC11 / W / TA2OUT / P74
34 BE1OUT / ISTxD1 / OUTC10 / SS2 / RTS2 / CTS2 / V / TA1IN / P73
35 CLK2 / V / TA1OUT / P72
36 (Note 3) IEIN / ISRxD2 / OUTC22 / STxD2 / SCL2 / RxD2 / TA0IN / TB5IN / P71
144P6Q-A

Notes :

1. P6$_4$ / CTS1 / RTS1 / SS1 / OUTC2$_1$ / ISCLK2
2. P7$_0$ / TA0$_{OUT}$ / TxD2 / SDA2 / SRxD2 / OUTC2$_0$ / ISTxD2 / IE$_{OUT}$
3. P7$_0$ and P7$_1$ are N-channel open drain output.

Figure 1.1.3. Pin Assignment for 144-Pin Package

Table 1.1.4. Pin Characteristics for 144-Pin Package

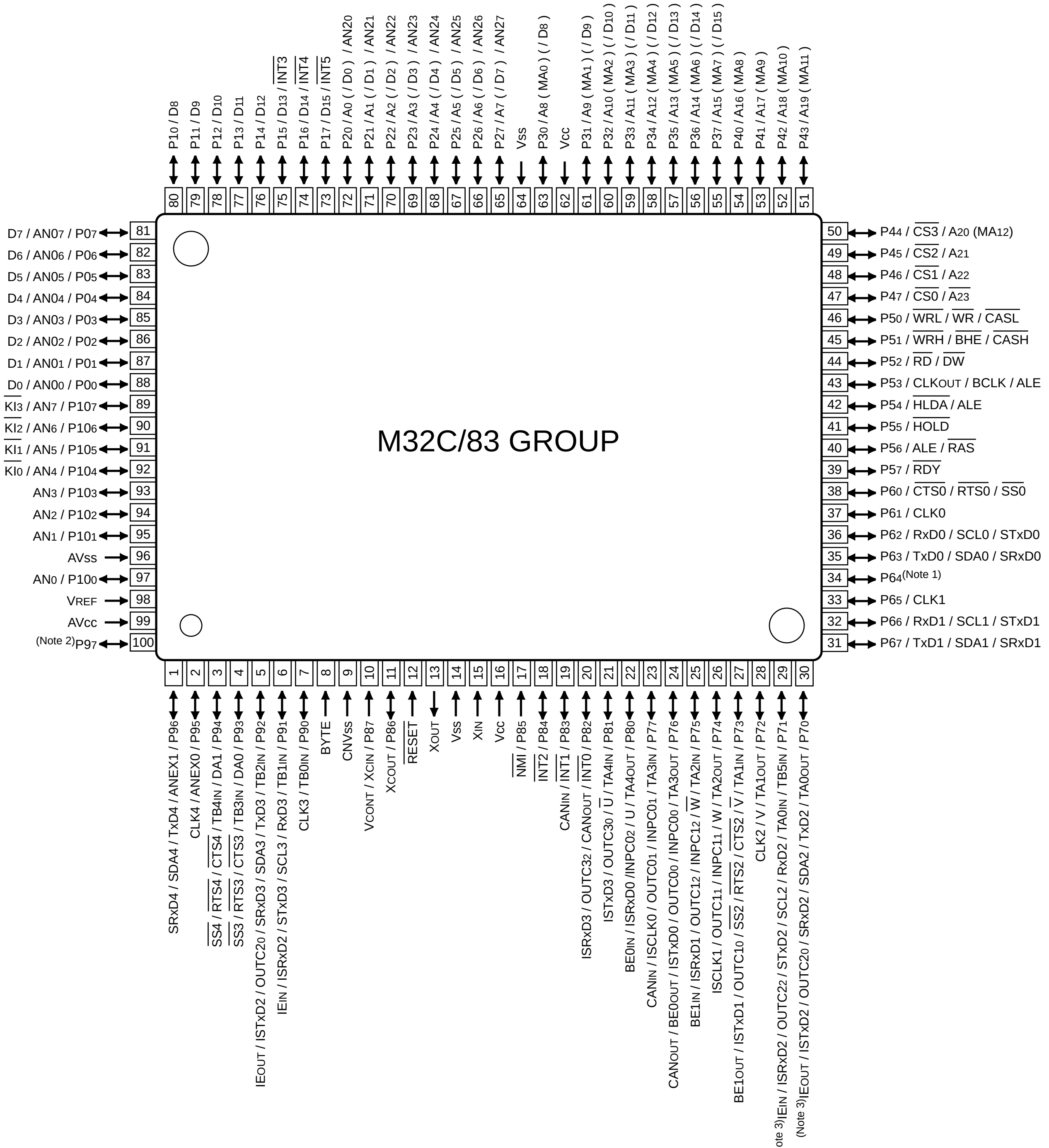
Pin No	Control pin	Port	Interrupt pin	Timer pin	UART/CAN pin	Intelligent I/O pin	Analog pin	Bus control pin
1		$P9_6$			TxD4/SDA4/SRxD4		ANEX1	
2		$P9_5$			CLK4		ANEX0	
3		$P9_4$		$TB4_{IN}$	$\overline{CTS4}/\overline{RTS4}/\overline{SS4}$		DA1	
4		$P9_3$		$TB3_{IN}$	$\overline{CTS3}/\overline{RTS3}/\overline{SS3}$		DA0	
5		$P9_2$		$TB2_{IN}$	TxD3/SDA3/SRxD3	$OUTC2_0/IE_{OUT}/ISTxD2$		
6		$P9_1$		$TB1_{IN}$	RxD3/SCL3/STxD3	$IE_{IN}/ISRxD2$		
7		$P9_0$		$TB0_{IN}$	CLK3			
8		$P14_6$						
9		$P14_5$						
10		$P14_4$						
11		$P14_3$				$INPC1_7/OUTC1_7$		
12		$P14_2$				$INPC1_6/OUTC1_6$		
13		$P14_1$				$OUTC1_5$		
14		$P14_0$				$OUTC1_4$		
15	BYTE							
16	CNV_{SS}							
17	X_{CIN}/V_{CONT}	$P8_7$						
18	X_{COUT}	$P8_6$						
19	$\overline{RESET}$							
20	X_{OUT}							
21	V_{SS}							
22	X_{IN}							
23	V_{CC}							
24		$P8_5$	$\overline{NMI}$					
25		$P8_4$	$\overline{INT2}$					
26		$P8_3$	$\overline{INT1}$		CAN_{IN}			
27		$P8_2$	$\overline{INT0}$		CAN_{OUT}	$OUTC3_2/ISRxD3$		
28		$P8_1$		$TA4_{IN}/\overline{U}$		$OUTC3_0/ISTxD3$		
29		$P8_0$		$TA4_{OUT}/U$		$INPC0_2/ISRxD0/BE0_{IN}$		
30		$P7_7$		$TA3_{IN}$	CAN_{IN}	$INPC0_1/OUTC0_1/ISCLK0$		
31		$P7_6$		$TA3_{OUT}$	CAN_{OUT}	$INPC0_0/OUTC0_0/ISTxD0/BE0_{OUT}$		
32		$P7_5$		$TA2_{IN}/\overline{W}$		$INPC1_2/OUTC1_2/ISRxD1/BE1_{IN}$		
33		$P7_4$		$TA2_{OUT}/W$		$INPC1_1/OUTC1_1/ISCLK1$		
34		$P7_3$		$TA1_{IN}/\overline{V}$	$\overline{CTS2}/\overline{RTS2}/\overline{SS2}$	$OUTC1_0/ISTxD1/BE1_{OUT}$		
35		$P7_2$		$TA1_{OUT}/V$	CLK2			
36		$P7_1$		$TB5_{IN}/TA0_{IN}$	RxD2/SCL2/STxD2	$OUTC2_2/ISRxD2/IE_{IN}$		
37		$P7_0$		$TA0_{OUT}$	TxD2/SDA2/SRxD2	$OUTC2_0/ISTxD2/IE_{OUT}$		
38		$P6_7$			TxD1/SDA1/SRxD1			
39	V_{CC}							
40		$P6_6$			RxD1/SCL1/STxD1			
41	V_{SS}							
42		$P6_5$			CLK1			
43		$P6_4$			$\overline{CTS1}/\overline{RTS1}/\overline{SS1}$	$OUTC2_1/ISCLK2$		
44		$P6_3$			TxD0/SDA0/SRxD0			
45		$P6_2$			RxD0/SCL0/STxD0			
46		$P6_1$			CLK0			
47		$P6_0$			$\overline{CTS0}/\overline{RTS0}/\overline{SS0}$			
48		$P13_7$				$OUTC2_7$		

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		P13$_6$				OUTC2$_1$/ISCLK2		
50		P13$_5$				OUTC2$_2$/ISRxD2/IE$_{IN}$		
51		P13$_4$				OUTC2$_0$/ISTxD2/IE$_{OUT}$		
52		P5$_7$						$\overline{RDY}$
53		P5$_6$						ALE/$\overline{RAS}$
54		P5$_5$						$\overline{HOLD}$
55		P5$_4$						$\overline{HLDA}$/ALE
56		P13$_3$				OUTC2$_3$		
57	V$_{SS}$							
58		P13$_2$				OUTC2$_6$		
59	V$_{CC}$							
60		P13$_1$				OUTC2$_5$		
61		P13$_0$				OUTC2$_4$		
62		P5$_3$						CLK$_{OUT}$/BCLK/ALE
63		P5$_2$						$\overline{RD}$/$\overline{DW}$
64		P5$_1$						$\overline{WRH}$/$\overline{BHE}$/$\overline{CASH}$
65		P5$_0$						$\overline{WRL}$/$\overline{WR}$/$\overline{CASL}$
66		P12$_7$				OUTC3$_7$		
67		P12$_6$				OUTC3$_6$		
68		P12$_5$				OUTC3$_5$		
69		P4$_7$						$\overline{CS0}$/$\overline{A_{23}}$
70		P4$_6$						$\overline{CS1}$/A$_{22}$
71		P4$_5$						$\overline{CS2}$/A$_{21}$
72		P4$_4$						$\overline{CS3}$/A$_{20}$(MA$_{12}$)
73		P4$_3$						A$_{19}$(MA$_{11}$)
74	V$_{CC}$							
75		P4$_2$						A$_{18}$(MA$_{10}$)
76	V$_{SS}$							
77		P4$_1$						A$_{17}$(MA$_9$)
78		P4$_0$						A$_{16}$(MA$_8$)
79		P3$_7$						A$_{15}$(MA$_7$)(/D$_{15}$)
80		P3$_6$						A$_{14}$(MA$_6$)(/D$_{14}$)
81		P3$_5$						A$_{13}$(MA$_5$)(/D$_{13}$)
82		P3$_4$						A$_{12}$(MA$_4$)(/D$_{12}$)
83		P3$_3$						A$_{11}$(MA$_3$)(/D$_{11}$)
84		P3$_2$						A$_{10}$(MA$_2$)(/D$_{10}$)
85		P3$_1$						A$_9$(MA$_1$)(/D$_9$)
86		P12$_4$				OUTC3$_4$		
87		P12$_3$				OUTC3$_3$		
88		P12$_2$				OUTC3$_2$/ISRxD3		
89		P12$_1$				OUTC3$_1$/ISCLK3		
90		P12$_0$				OUTC3$_0$/ISTxD3		
91	V$_{CC}$							
92		P3$_0$						A$_8$(MA$_0$)(/D$_8$)
93	V$_{SS}$							
94		P2$_7$					AN2$_7$	A$_7$(/D$_7$)
95		P2$_6$					AN2$_6$	A$_6$(/D$_6$)
96		P2$_5$					AN2$_5$	A$_5$(/D$_5$)

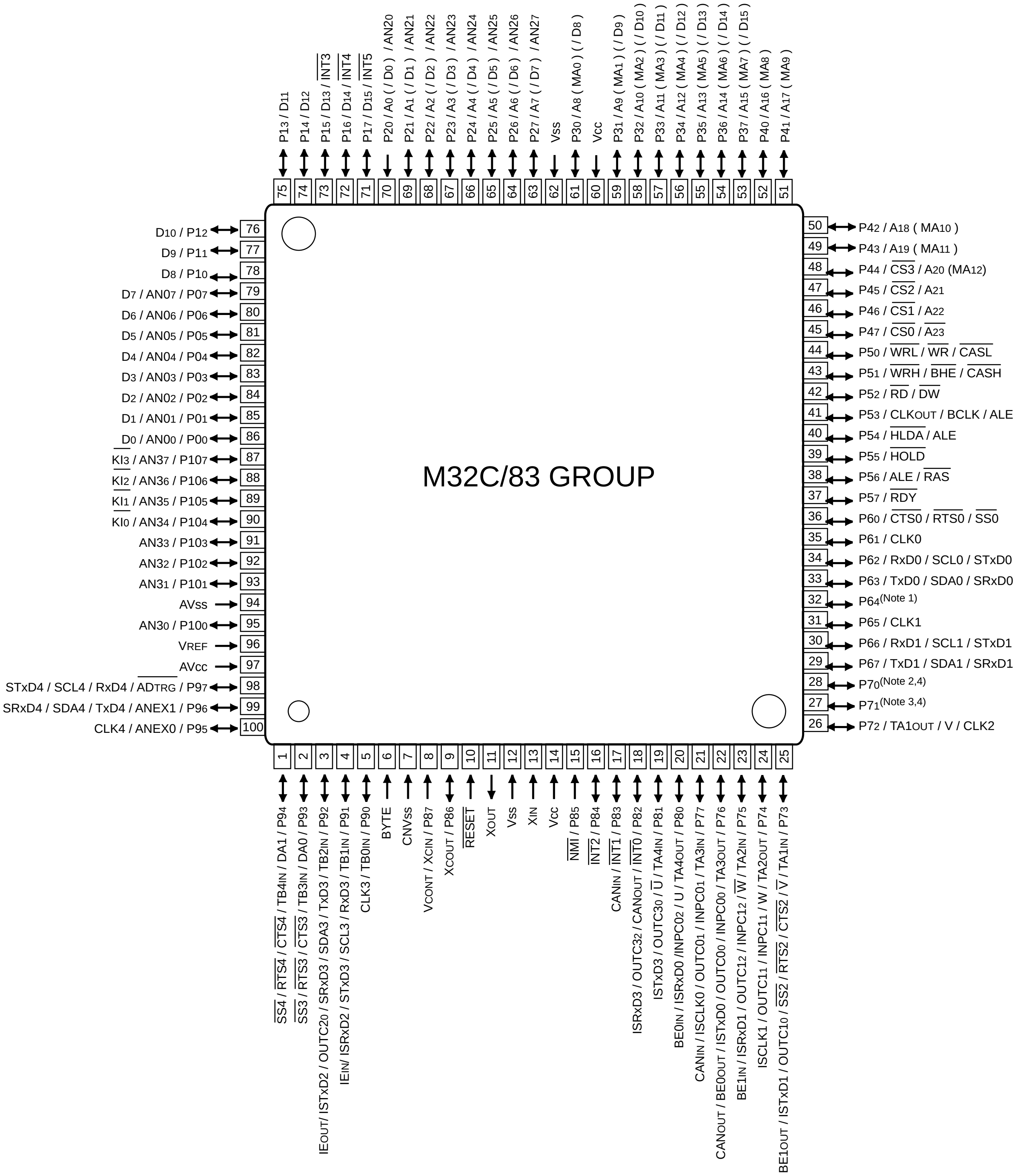
Table 1.1.4. Pin Characteristics for 144-Pin Package (Continued)

Pin No	Control pin	Port	Interrupt pin	Timer pin	UART/CAN pin	Intelligent I/O pin	Analog pin	Bus control pin
97		$P2_4$					$AN2_4$	A_4(/D_4)
98		$P2_3$					$AN2_3$	A_3(/D_3)
99		$P2_2$					$AN2_2$	A_2(/D_2)
100		$P2_1$					$AN2_1$	A_1(/D_1)
101		$P2_0$					$AN2_0$	A_0(/D_0)
102		$P1_7$	$\overline{INT5}$					D_{15}
103		$P1_6$	$\overline{INT4}$					D_{14}
104		$P1_5$	$\overline{INT3}$					D_{13}
105		$P1_4$						D_{12}
106		$P1_3$						D_{11}
107		$P1_2$						D_{10}
108		$P1_1$						D_9
109		$P1_0$						D_8
110		$P0_7$					$AN0_7$	D_7
111		$P0_6$					$AN0_6$	D_6
112		$P0_5$					$AN0_5$	D_5
113		$P0_4$					$AN0_4$	D_4
114		$P11_4$						
115		$P11_3$				$OUTC1_3$		
116		$P11_2$				$INPC1_2$/$OUTC1_2$/ISRxD1/$BE1_{IN}$		
117		$P11_1$				$INPC1_1$/$OUTC1_1$/ISCLK1		
118		$P11_0$				$OUTC1_0$/ISTxD1/$BE1_{OUT}$		
119		$P0_3$					$AN0_3$	D_3
120		$P0_2$					$AN0_2$	D_2
121		$P0_1$					$AN0_1$	D_1
122		$P0_0$					$AN0_0$	D_0
123		$P15_7$				$INPC0_7$	$AN15_7$	
124		$P15_6$				$INPC0_6$	$AN15_6$	
125		$P15_5$				$INPC0_5$/$OUTC0_5$	$AN15_5$	
126		$P15_4$				$INPC0_4$/$OUTC0_4$	$AN15_4$	
127		$P15_3$				$INPC0_3$	$AN15_3$	
128		$P15_2$				$INPC0_2$/ISRxD0/$BE0_{IN}$	$AN15_2$	
129		$P15_1$				$INPC0_1$/$OUTC0_1$/ISCLK0	$AN15_1$	
130	V_{SS}							
131		$P15_0$				$INPC0_0$/$OUTC0_0$/ISTxD0/$BE0_{OUT}$	$AN15_0$	
132	V_{CC}							
133		$P10_7$	$\overline{KI_3}$				AN_7	
134		$P10_6$	$\overline{KI_2}$				AN_6	
135		$P10_5$	$\overline{KI_1}$				AN_5	
136		$P10_4$	$\overline{KI_0}$				AN_4	
137		$P10_3$					AN_3	
138		$P10_2$					AN_2	
139		$P10_1$					AN_1	
140	AV_{SS}							
141		$P10_0$					AN_0	
142							V_{REF}	
143	AV_{CC}							
144		$P9_7$			RxD4/SCL4/STxD4		$\overline{AD_{TRG}}$	


M32C/83 GROUP
100P6S-A
80 P10 / D8
79 P11 / D9
78 P12 / D10
77 P13 / D11
76 P14 / D12
75 P15 / D13 / INT3
74 P16 / D14 / INT4
73 P17 / D15 / INT5
72 P20 / A0 (/ D0) / AN20
71 P21 / A1 (/ D1) / AN21
70 P22 / A2 (/ D2) / AN22
69 P23 / A3 (/ D3) / AN23
68 P24 / A4 (/ D4) / AN24
67 P25 / A5 (/ D5) / AN25
66 P26 / A6 (/ D6) / AN26
65 P27 / A7 (/ D7) / AN27
64 Vss
63 P30 / A8 (MA0) (/ D8)
62 Vcc
61 P31 / A9 (MA1) (/ D9)
60 P32 / A10 (MA2) (/ D10)
59 P33 / A11 (MA3) (/ D11)
58 P34 / A12 (MA4) (/ D12)
57 P35 / A13 (MA5) (/ D13)
56 P36 / A14 (MA6) (/ D14)
55 P37 / A15 (MA7) (/ D15)
54 P40 / A16 (MA8)
53 P41 / A17 (MA9)
52 P42 / A18 (MA10)
51 P43 / A19 (MA11)
50 P44 / CS3 / A20 (MA12)
49 P45 / CS2 / A21
48 P46 / CS1 / A22
47 P47 / CS0 / A23
46 P50 / WRL / WR / CASL
45 P51 / WRH / BHE / CASH
44 P52 / RD / DW
43 P53 / CLKOUT / BCLK / ALE
42 P54 / HLDA / ALE
41 P55 / HOLD
40 P56 / ALE / RAS
39 P57 / RDY
38 P60 / CTS0 / RTS0 / SS0
37 P61 / CLK0
36 P62 / RxD0 / SCL0 / STxD0
35 P63 / TxD0 / SDA0 / SRxD0
34 P64(Note 1)
33 P65 / CLK1
32 P66 / RxD1 / SCL1 / STxD1
31 P67 / TxD1 / SDA1 / SRxD1
81 D7 / AN07 / P07
82 D6 / AN06 / P06
83 D5 / AN05 / P05
84 D4 / AN04 / P04
85 D3 / AN03 / P03
86 D2 / AN02 / P02
87 D1 / AN01 / P01
88 D0 / AN00 / P00
89 KI3 / AN7 / P107
90 KI2 / AN6 / P106
91 KI1 / AN5 / P105
92 KI0 / AN4 / P104
93 AN3 / P103
94 AN2 / P102
95 AN1 / P101
96 AVss
97 AN0 / P100
98 VREF
99 AVcc
100 (Note 2)P97
1 SRxD4 / SDA4 / TxD4 / ANEX1 / P96
2 CLK4 / ANEX0 / P95
3 SS4 / RTS4 / CTS4 / TB4IN / DA1 / P94
4 SS3 / RTS3 / CTS3 / TB3IN / DA0 / P93
5 IEOUT / ISTxD2 / OUTC20 / SRxD3 / SDA3 / TxD3 / TB2IN / P92
6 IEIN / ISRxD2 / STxD3 / SCL3 / RxD3 / TB1IN / P91
7 CLK3 / TB0IN / P90
8 BYTE
9 CNVss
10 VCONT / XCIN / P87
11 XCOUT / P86
12 RESET
13 XOUT
14 Vss
15 XIN
16 Vcc
17 NMI / P85
18 INT2 / P84
19 CANIN / INT1 / P83
20 ISRxD3 / OUTC32 / CANOUT / INT0 / P82
21 ISTxD3 / OUTC30 / U / TA4IN / P81
22 BE0IN / ISRxD0 /INPC02 / U / TA4OUT / P80
23 CANIN / ISCLK0 / OUTC01 / INPC01 / TA3IN / P77
24 CANOUT / BE0OUT / ISTxD0 / OUTC00 / INPC00 / TA3OUT / P76
25 BE1IN / ISRxD1 / OUTC12 / INPC12 / W / TA2IN / P75
26 ISCLK1 / OUTC11 / INPC11 / W / TA2OUT / P74
27 BE1OUT / ISTxD1 / OUTC10 / SS2 / RTS2 / CTS2 / V / TA1IN / P73
28 CLK2 / V / TA1OUT / P72
29 (Note 3)IEIN / ISRxD2 / OUTC22 / STxD2 / SCL2 / RxD2 / TA0IN / TB5IN / P71
30 (Note 3)IEOUT / ISTxD2 / OUTC20 / SRxD2 / SDA2 / TxD2 / TA0OUT / P70

Notes :
1. $P6_4$ / $\overline{CTS1}$ / $\overline{RTS1}$ / $\overline{SS1}$ / $OUTC2_1$ / ISCLK2
2. $P9_7$ / $\overline{AD_{TRG}}$ / RxD4 / STxD4 / SCL4
3. $P7_0$ and $P7_1$ are N-channel open drain output.

Figure 1.1.4. Pin assignment for 100-Pin Package


M32C/83 GROUP
75 P13 / D11
74 P14 / D12
73 P15 / D13 / INT3
72 P16 / D14 / INT4
71 P17 / D15 / INT5
70 P20 / A0 (/ D0) / AN20
69 P21 / A1 (/ D1) / AN21
68 P22 / A2 (/ D2) / AN22
67 P23 / A3 (/ D3) / AN23
66 P24 / A4 (/ D4) / AN24
65 P25 / A5 (/ D5) / AN25
64 P26 / A6 (/ D6) / AN26
63 P27 / A7 (/ D7) / AN27
62 Vss
61 P30 / A8 (MA0) (/ D8)
60 Vcc
59 P31 / A9 (MA1) (/ D9)
58 P32 / A10 (MA2) (/ D10)
57 P33 / A11 (MA3) (/ D11)
56 P34 / A12 (MA4) (/ D12)
55 P35 / A13 (MA5) (/ D13)
54 P36 / A14 (MA6) (/ D14)
53 P37 / A15 (MA7) (/ D15)
52 P40 / A16 (MA8)
51 P41 / A17 (MA9)
D10 / P12 76
D9 / P11 77
D8 / P10 78
D7 / AN07 / P07 79
D6 / AN06 / P06 80
D5 / AN05 / P05 81
D4 / AN04 / P04 82
D3 / AN03 / P03 83
D2 / AN02 / P02 84
D1 / AN01 / P01 85
D0 / AN00 / P00 86
KI3 / AN37 / P107 87
KI2 / AN36 / P106 88
KI1 / AN35 / P105 89
KI0 / AN34 / P104 90
AN33 / P103 91
AN32 / P102 92
AN31 / P101 93
AVss 94
AN30 / P100 95
VREF 96
AVcc 97
STxD4 / SCL4 / RxD4 / ADTRG / P97 98
SRxD4 / SDA4 / TxD4 / ANEX1 / P96 99
CLK4 / ANEX0 / P95 100
50 P42 / A18 (MA10)
49 P43 / A19 (MA11)
48 P44 / CS3 / A20 (MA12)
47 P45 / CS2 / A21
46 P46 / CS1 / A22
45 P47 / CS0 / A23
44 P50 / WRL / WR / CASL
43 P51 / WRH / BHE / CASH
42 P52 / RD / DW
41 P53 / CLKOUT / BCLK / ALE
40 P54 / HLDA / ALE
39 P55 / HOLD
38 P56 / ALE / RAS
37 P57 / RDY
36 P60 / CTS0 / RTS0 / SS0
35 P61 / CLK0
34 P62 / RxD0 / SCL0 / STxD0
33 P63 / TxD0 / SDA0 / SRxD0
32 P64(Note 1)
31 P65 / CLK1
30 P66 / RxD1 / SCL1 / STxD1
29 P67 / TxD1 / SDA1 / SRxD1
28 P70(Note 2,4)
27 P71(Note 3,4)
26 P72 / TA1OUT / V / CLK2
1 SS4 / RTS4 / CTS4 / TB4IN / DA1 / P94
2 SS3 / RTS3 / CTS3 / TB3IN / DA0 / P93
3 IEOUT/ ISTxD2 / OUTC20 / SRxD3 / SDA3 / TxD3 / TB2IN / P92
4 IEIN/ ISRxD2 / STxD3 / SCL3 / RxD3 / TB1IN / P91
5 CLK3 / TB0IN / P90
6 BYTE
7 CNVss
8 VCONT / XCIN / P87
9 XCOUT / P86
10 RESET
11 XOUT
12 Vss
13 XIN
14 Vcc
15 NMI / P85
16 INT2 / P84
17 CANIN / INT1 / P83
18 ISRxD3 / OUTC32 / CANOUT / INT0 / P82
19 ISTxD3 / OUTC30 / U / TA4IN / P81
20 BE0IN / ISRxD0 /INPC02 / U / TA4OUT / P80
21 CANIN / ISCLK0 / OUTC01 / INPC01 / TA3IN / P77
22 CANOUT / BE0OUT / ISTxD0 / OUTC00 / INPC00 / TA3OUT / P76
23 BE1IN / ISRxD1 / OUTC12 / INPC12 / W / TA2IN / P75
24 ISCLK1 / OUTC11 / INPC11 / W / TA2OUT / P74
25 BE1OUT / ISTxD1 / OUTC10 / SS2 / RTS2 / CTS2 / V / TA1IN / P73

Notes :

1. P64 / CTS1 / RTS1 / SS1 / OUTC21 / ISCLK2
2. P70 / TA0OUT / TxD2 / SDA2 / SRxD2 / OUTC20 / ISTxD2 / IEOUT
3. P71 / TA0IN / TB5IN / RxD2 / SCL2 / STxD2 / OUTC22 / ISRxD2 / IEIN
4. P70 and P71 are N-channel open drain output.

100P6Q-A

Figure 1.1.5. Pin Assignment for 100-Pin Package

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$_6$			TxD4/SDA4/SRxD4		ANEX1	
2	100		P9$_5$			CLK4		ANEX0	
3	1		P9$_4$		TB4$_{IN}$	$\overline{CTS4}$/$\overline{RTS4}$/$\overline{SS4}$		DA1	
4	2		P9$_3$		TB3$_{IN}$	$\overline{CTS3}$/$\overline{RTS3}$/$\overline{SS3}$		DA0	
5	3		P9$_2$		TB2$_{IN}$	TxD3/SDA3/SRxD3	OUTC2$_0$/IE$_{OUT}$/ISTxD2		
6	4		P9$_1$		TB1$_{IN}$	RxD3/SCL3/STxD3	IE$_{IN}$/ISRxD2		
7	5		P9$_0$		TB0$_{IN}$	CLK3			
8	6	BYTE							
9	7	CNV$_{SS}$							
10	8	X$_{CIN}$/V$_{CONT}$	P8$_7$						
11	9	X$_{COUT}$	P8$_6$						
12	10	$\overline{RESET}$							
13	11	X$_{OUT}$							
14	12	V$_{SS}$							
15	13	X$_{IN}$							
16	14	V$_{CC}$							
17	15		P8$_5$	$\overline{NMI}$					
18	16		P8$_4$	$\overline{INT2}$					
19	17		P8$_3$	$\overline{INT1}$		CAN$_{IN}$			
20	18		P8$_2$	$\overline{INT0}$		CAN$_{OUT}$	OUTC3$_2$/ISRxD3		
21	19		P8$_1$		TA4$_{IN}$/$\overline{U}$		OUTC3$_0$/ISTxD3		
22	20		P8$_0$		TA4$_{OUT}$/U		INPC0$_2$/ISRxD0/BE0$_{IN}$		
23	21		P7$_7$		TA3$_{IN}$	CAN$_{IN}$	INPC0$_1$/OUTC0$_1$/ISCLK0		
24	22		P7$_6$		TA3$_{OUT}$	CAN$_{OUT}$	INPC0$_0$/OUTC0$_0$/ISTxD0/BE0$_{OUT}$		
25	23		P7$_5$		TA2$_{IN}$/$\overline{W}$		INPC1$_2$/OUTC1$_2$/ISRxD1/BE1$_{IN}$		
26	24		P7$_4$		TA2$_{OUT}$/W		INPC1$_1$/OUTC1$_1$/ISCLK1		
27	25		P7$_3$		TA1$_{IN}$/$\overline{V}$	$\overline{CTS2}$/$\overline{RTS2}$/$\overline{SS2}$	OUTC1$_0$/ISTxD1/BE1$_{OUT}$		
28	26		P7$_2$		TA1$_{OUT}$/V	CLK2			
29	27		P7$_1$		TB5$_{IN}$/TA0$_{IN}$	RxD2/SCL2/STxD2	OUTC2$_2$/ISRxD2/IE$_{IN}$		
30	28		P7$_0$		TA0$_{OUT}$	TxD2/SDA2/SRxD2	OUTC2$_0$/ISTxD2/IE$_{OUT}$		
31	29		P6$_7$			TxD1/SDA1/SRxD1			
32	30		P6$_6$			RxD1/SCL1/STxD1			
33	31		P6$_5$			CLK1			
34	32		P6$_4$			$\overline{CTS1}$/$\overline{RTS1}$/$\overline{SS1}$	OUTC2$_1$/ISCLK2		
35	33		P6$_3$			TxD0/SDA0/SRxD0			
36	34		P6$_2$			RxD0/SCL0/STxD0			
37	35		P6$_1$			CLK0			
38	36		P6$_0$			$\overline{CTS0}$/$\overline{RTS0}$/$\overline{SS0}$			
39	37		P5$_7$						$\overline{RDY}$
40	38		P5$_6$						ALE/$\overline{RAS}$
41	39		P5$_5$						$\overline{HOLD}$
42	40		P5$_4$						$\overline{HLDA}$/ALE
43	41		P5$_3$						CLK$_{OUT}$/BCLK/ALE
44	42		P5$_2$						$\overline{RD}$/$\overline{DW}$
45	43		P5$_1$						$\overline{WRH}$/$\overline{BHE}$/$\overline{CASH}$
46	44		P5$_0$						$\overline{WRL}$/$\overline{WR}$/$\overline{CASL}$
47	45		P4$_7$						$\overline{CS0}$/A$_{23}$
48	46		P4$_6$						$\overline{CS1}$/A$_{22}$
49	47		P4$_5$						$\overline{CS2}$/A$_{21}$
50	48		P4$_4$						$\overline{CS3}$/A$_{20}$(MA$_{12}$)

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$_3$						A$_{19}$(MA$_{11}$)
52	50		P4$_2$						A$_{18}$(MA$_{10}$)
53	51		P4$_1$						A$_{17}$(MA$_9$)
54	52		P4$_0$						A$_{16}$(MA$_8$)
55	53		P3$_7$						A$_{15}$(MA$_7$)(/D$_{15}$)
56	54		P3$_6$						A$_{14}$(MA$_6$)(/D$_{14}$)
57	55		P3$_5$						A$_{13}$(MA$_5$)(/D$_{13}$)
58	56		P3$_4$						A$_{12}$(MA$_4$)(/D$_{12}$)
59	57		P3$_3$						A$_{11}$(MA$_3$)(/D$_{11}$)
60	58		P3$_2$						A$_{10}$(MA$_2$)(/D$_{10}$)
61	59		P3$_1$						A$_9$(MA$_1$)(/D$_9$)
62	60	V$_{CC}$							
63	61		P3$_0$						A$_8$(MA$_0$)(/D$_8$)
64	62	V$_{SS}$							
65	63		P2$_7$					AN2$_7$	A$_7$(/D$_7$)
66	64		P2$_6$					AN2$_6$	A$_6$(/D$_6$)
67	65		P2$_5$					AN2$_5$	A$_5$(/D$_5$)
68	66		P2$_4$					AN2$_4$	A$_4$(/D$_4$)
69	67		P2$_3$					AN2$_3$	A$_3$(/D$_3$)
70	68		P2$_2$					AN2$_2$	A$_2$(/D$_2$)
71	69		P2$_1$					AN2$_1$	A$_1$(/D$_1$)
72	70		P2$_0$					AN2$_0$	A$_0$(/D$_0$)
73	71		P1$_7$	$\overline{\text{INT5}}$					D$_{15}$
74	72		P1$_6$	$\overline{\text{INT4}}$					D$_{14}$
75	73		P1$_5$	$\overline{\text{INT3}}$					D$_{13}$
76	74		P1$_4$						D$_{12}$
77	75		P1$_3$						D$_{11}$
78	76		P1$_2$						D$_{10}$
79	77		P1$_1$						D$_9$
80	78		P1$_0$						D$_8$
81	79		P0$_7$					AN0$_7$	D$_7$
82	80		P0$_6$					AN0$_6$	D$_6$
83	81		P0$_5$					AN0$_5$	D$_5$
84	82		P0$_4$					AN0$_4$	D$_4$
85	83		P0$_3$					AN0$_3$	D$_3$
86	84		P0$_2$					AN0$_2$	D$_2$
87	85		P0$_1$					AN0$_1$	D$_1$
88	86		P0$_0$					AN0$_0$	D$_0$
89	87		P10$_7$	$\overline{\text{KI}_3}$				AN$_7$	
90	88		P10$_6$	$\overline{\text{KI}_2}$				AN$_6$	
91	89		P10$_5$	$\overline{\text{KI}_1}$				AN$_5$	
92	90		P10$_4$	$\overline{\text{KI}_0}$				AN$_4$	
93	91		P10$_3$					AN$_3$	
94	92		P10$_2$					AN$_2$	
95	93		P10$_1$					AN$_1$	
96	94	AV$_{SS}$							
97	95		P10$_0$					AN$_0$	
98	96							V$_{REF}$	
99	97	AV$_{CC}$							
100	98		P9$_7$			RxD4/SCL4/STxD4		$\overline{\text{AD}_{\text{TRG}}}$	

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

Symbol	Function	I/O type	Description
V_{CC} V_{SS}	Power supply input	I I	Connect V_{CC} pin to 3.0 to 5.5 V. Connect V_{SS} pin to 0 V.
CNV_{SS}	CNV_{SS}	I	Switches processor mode. Connect this pin to V_{SS} to start up in single-chip the (memory expansion mode). Connect this pin to V_{CC} to start up in microprocessor mode.
$\overline{RESET}$	Reset input	I	The microcomputer remains in a reset state when setting the pin to "L".
X_{IN} X_{OUT}	Clock input Clock output	I O	I/O pins for the main clock generation circuit. Connect a ceramic resonator or crystal oscillator between X_{IN} pin and X_{OUT} pin. To use an externally generated clock, input a clock from X_{IN} pin to leave X_{OUT} 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 settig this pin to "H". Set it to either one. Connet this pin to V_{SS} when an external bus is not used.
AV_{CC}	Analog power supply input	I	Inputs the power supply for A-D converter and D-A converter. Connect this pin to V_{CC}.
AV_{SS}	Analog power supply input	I	Inputs the power supply for A-D converter and D-A converter. Connect this pin to V_{SS}.
V_{REF}	Reference voltage input	I	Inputs reference voltage for A-D converter.
$P0_0$ to $P0_7$	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.
D_0 to D_7	Data bus	I/O	Input and output data (D_0 to D_7) when setting these pins as data bus.
$AN0_0$ to $AN0_7$	Analog input pin	I	Analog input pins for A-D converter.
$P1_0$ to $P1_7$	I/O port P1	I/O	8-bit I/O ports equivalent to P0
$\overline{INT3}$ to $\overline{INT5}$	$\overline{INT}$ interrupt input pin	I	Input pins for $\overline{INT}$ interrupt
D_8 to D_{15}	Data bus	I/O	Input and output data (D_8 to D_{15}) when setting these pins as data bus.
$P2_0$ to $P2_7$	I/O port P2	I/O	8-bit I/O ports equivalent to P0
A_0 to A_7	Address bus	O	Output 8 low-order address bits (A_0 to A_7).
A_0/D_0 to A_7/D_7	Address bus/data bus	I/O	Input and output data (D_0 to D_7) and output 8 low-order address bits (A_0 to A_7) by time-sharing when setting these pins as multiplex bus.
$AN2_0$ to $AN2_7$	Analog input pin	I	Analog input pins for A-D converter.
$P3_0$ to $P3_7$	I/O port P3	I/O	I/O ports equivalent to P0
A_8 to A_{15}	Address bus	O	Output 8 middle-order address bits (A_8 to A_{15}).
A_8/D_8 to A_{15}/D_{15}	Address bus/data bus	I/O	Input and output data (D_8 to D_{15}) and output 8 middle-order address bits (A_8 to A_8) by time-sharing when setting an external 16-bit data bus as multiplex bus.
MA_0 to MA_7	Address bus	O	Output row addresses and column addresses by time-sharing when accessing the DRAM space.
$P4_0$ to $P4_7$	I/O port P4	I/O	8-bit I/O ports equivalent to P0
A_{16} to A_{22} $\overline{A_{23}}$	Address bus	O O	Output 8 high-order address bits (A_{16} to A_{22}, $\overline{A_{23}}$). The uppermost bit ($\overline{A_{23}}$) inversed is also output.
$\overline{CS0}$ to $\overline{CS3}$	Chip-select	O	Output $\overline{CS0}$ to $\overline{CS3}$ signals. $\overline{CS0}$ to $\overline{CS3}$ are chip-select signals to specifly access space.
MA_8 to MA_{12}	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

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

Symbol	Function	I/O type	Description
$P5_0$ to $P5_7$	I/O port P5	I/O	8-bit I/O ports equivalent to P0
CLK$_{OUT}$	Clock output	O	Outputs a X_{IN} divided by 8 or divided by 32 or a clock having the same frequency as X_{CIN} from $P5_3$.
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 shignal is set to "L" when reading data in extenal 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 addess. CASH is set to "L" when accessing an odd address. RAS signal latches row address.
$P6_0$ to $P6_7$	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 ($P6_0$ to $P6_3$) and UART1 ($P6_4$ to $P6_7$)
ISCLK $OUTC2_1$	Intelligent I/O pin	I/O O	ISCLK2 inputs and outputs a clock for the intelligent I/O communication function. $OUTC2_1$ outputs a clock for the waveform generation function.

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

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

Symbol	Function	I/O type	Description
P70 to P77	I/O port P7	I/O	8-bit I/O ports equivalent to P0 (except P70 and P71 for N-channel open drain outputs.)
TA0OUT to TA3OUT TA0IN to TA3IN	Timer A pin	I/O I	I/O ports for the timer A0 to timer A3.
TB5IN	Timer B pin	I	Input pin for the timer B5
V, $\overline{V}$	Three-phase motor control output pin	O	V-phase output pin
W, $\overline{W}$			W-phase output pin
$\overline{CTS2}$ $\overline{RTS2}$ $\overline{SS2}$ CLK2 RxD2 SCL2 STxD2 TxD2 SDA2 SRxD2	UART pin	I O I I/O I I/O O O O I	I/O pins for UART2
INPC00, INPC01, INPC11, INPC12 OUTC00, OUTC01, OUTC10 to OUTC12, OUTC20 to OUTC22 ISCLK0, ISCLK1 ISTxD0 to ISTxD2 ISRxD1, ISRxD2 IEOUT IEIN BE0OUT BE1OUT BE1IN	Intellligent I/O pin	I O I/O O I O I O O I	INPC00, INPC01, INPC11 and INPC12 are input pins for the time measurement function. OUTC00, OUTC01 and OUTC10 to OUTC12, OUTC20, OUTC22 are output pins for the waveform generation function. ISCLK0 and ISCLK1 input and output a clock for the intelligent I/O communication function. IEIN, ISRxD1, ISRxD2 and BE1IN input received data for the intelligent I/O communication function. IEOUT, ISTxD0 to ISTxD2, BE0OUT and BE1OUT output transmit data for the intelligent I/O communication function.
CANOUT CANIN	CAN pin	O I	I/O pins for the CAN communication function
P80 to P84, P86, P87	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 frequeny synthesizer. Connect P86 to Vss to make a PLL frequency stable.
TA4OUT TA4IN	Timer A pin	I/O I	I/O pins for the timer A4
U, $\overline{U}$	Three phase motor control output pin	O	U-phase output pins
$\overline{INT0}$ to $\overline{INT2}$	$\overline{INT}$ interrupt input pin	I	Input pins for $\overline{INT}$ interrupt
INPC02 ISRxD0 BE0IN OUTC30 ISTxD3 OUTC32 ISRxD3	Intelligent I/O pin	I I I O O O I	INPC02 is an input pin for the time measurement function. OUTC30 and OUTC32 are output pins for the waveform generation function. ISRxD0 and BE0IN input received data 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.
CANOUT CANIN	CAN pin	O I	I/O pins for CAN communication function
P85/$\overline{NMI}$	$\overline{NMI}$ interrupt input pin	I	Input pin for the $\overline{NMI}$ interrupt. Pin status can be read in the P8_5 bit in the P8 register.

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

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

Symbol	Function	I/O type	Description
$P9_0$ to $P9_7$	I/O port P9	I/O	8-bit I/O ports equivalent to P0. PRCR register prevents PD9 and PS3 registers from rewriting.
$TB0_{IN}$ to $TB4_{IN}$	Timer B pin	I	Input pins for timer B0 to B4
$\overline{CTS3}$, $\overline{CTS4}$ $\overline{RTS3}$, $\overline{RTS4}$ $\overline{SS3}$, $\overline{SS4}$ 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 ($P9_0$ to $P9_3$) and UART 4 ($P9_4$ to $P9_7$)
DA0, DA1	D-A output pin	O	Input pins for D-A converter
ANEX0 ANEX1 $\overline{AD_{TRG}}$	A-D related pin	I/O I I	ANEX0 and ANEX1 are expanded analog I/O pins for A-D converter. $\overline{AD_{TRG}}$ is an A-D trigger input pin.
$OUTC2_0$ ISTxD2 IE_{OUT} IE_{IN} ISRxD2	Intelligent I/O pin	O O O I I	$OUTC2_0$ is an output pin for the waveform generation fuction. ISTxD2 outputs transmit data for the intelligent I/O communication function. IE_{OUT} outputs transmit data for the intelligento I/O communication function or IE mode. IE_{IN} inputs receive data for the intelligento I/O communication function or IE mode. ISRxD2 inputs receive data for the intelligento I/O communication function.
$P10_0$ to $P10_7$	I/O port P10	I/O	8-bit I/O ports equivalent to P0
$\overline{KI_0}$ to $\overline{KI_3}$	Key input interrupt pin	I	Input pins for key input interrupt
AN_0 to AN_7	Analog input pin	I	Analog I/O pins for A-D converter

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

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

Symbol	Function	I/O type	Description
P11$_0$ to P11$_4$	I/O port P11	I/O	5-bit I/O ports equivalent to P0
INPC1$_1$, INPC1$_2$ OUTC1$_0$ to OUTC1$_3$ ISCLK1 ISRxD1 BE1$_{IN}$ ISTxD1 BE1$_{OUT}$	Intelligent I/O pin	I O I/O I I O O	INPC1$_1$ and INPC1$_2$ are input pins for the time measurement function. OUTC1$_0$ to OUTC1$_3$ are output pins for the waveform generation function. ISCLK1 inputs and outputs a clock for the intelligent I/O communication function. ISRxD1 and BE1$_{IN}$ input receive data for the intelligent I/O communication function. ISTxD1 and BE1$_{OUT}$ output receive data for the intelligent I/O communication function.
P12$_0$ to P12$_7$	I/O port P12	I/O	8-bit I/O ports equivalent to P0
OUTC3$_0$ to OUTC3$_7$ ISCLK3 ISTxD3 ISRxD3	Intelligent I/O pin	O I/O O I	OUTC3$_0$ to OUTC3$_7$ 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.
P13$_0$ to P13$_7$	I/O port P13	I/O	8-bit I/O ports equivalent to P0
OUTC2$_0$ to OUTC2$_7$ ISCLK2 ISRxD2 IE$_{IN}$ ISTxD2 IE$_{OUT}$	Intelligent I/O pin	O I/O I I O O	OUTC2$_0$ to OUTC2$_7$ are output pins for the waveform generation function. ISCLK2 inputs and outputs a clock for the intelligent I/O communication function. ISTxD2 and IE$_{OUT}$ output transmit data for the intelligent I/O communication function. ISRxD1 and IE$_{IN}$ input receive data for the intelligent I/O communication function.
P14$_0$ to P14$_6$	I/O port P14	I/O	7-bit I/O ports equivalent to P0
INPC1$_6$ and INPC1$_7$ OUTC1$_4$ to OUTC1$_7$	Intelligent I/O pin	I O	INPC1$_6$ to INPC1$_7$ are input pins for the time measurement function. OUTC1$_4$ to OUTC1$_7$ are output pins for the waveform generation function.
P15$_0$ to P15$_7$	I/O port P15	I/O	8-bit I/O ports equivalent to P0
INPC0$_0$ to INPC0$_7$ OUTC0$_0$, OUTC0$_1$, OUTC0$_4$, OUTC0$_5$ ISCLK0 ISRxD0 BE0$_{IN}$ ISTxD0 BE0$_{OUT}$	Intelligent I/O pin	I O I/O I I O O	INPC0$_0$ to INPC0$_7$ are input pins for the time measurement function. OUTC0$_0$, OUTC0$_1$, OUTC0$_4$ and OUTC0$_5$ are output pins for the waveform generation function. ISCLK0 inputs and outputs a clock for the intelligent I/O communication function. ISRxD0 and BE$_{IN}$ input receive data for the intelligent I/O communication function. ISTxD0 and BE$_{OUT}$ output transmit data for the intelligent I/O communication function.
AN15$_0$ to AN15$_7$	Analog input port	I	Analog input pin for A-D converter

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

Memory

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

Total address space are 16M bytes from addresses 000000_{16} to $FFFFFF_{16}$.

Internal ROM is allocated in lower addresses beginning with address $FFFFFF_{16}$. For example, a 64K-byte internal ROM is allocated in addresses from $FF0000_{16}$ to $FFFFFF_{16}$.

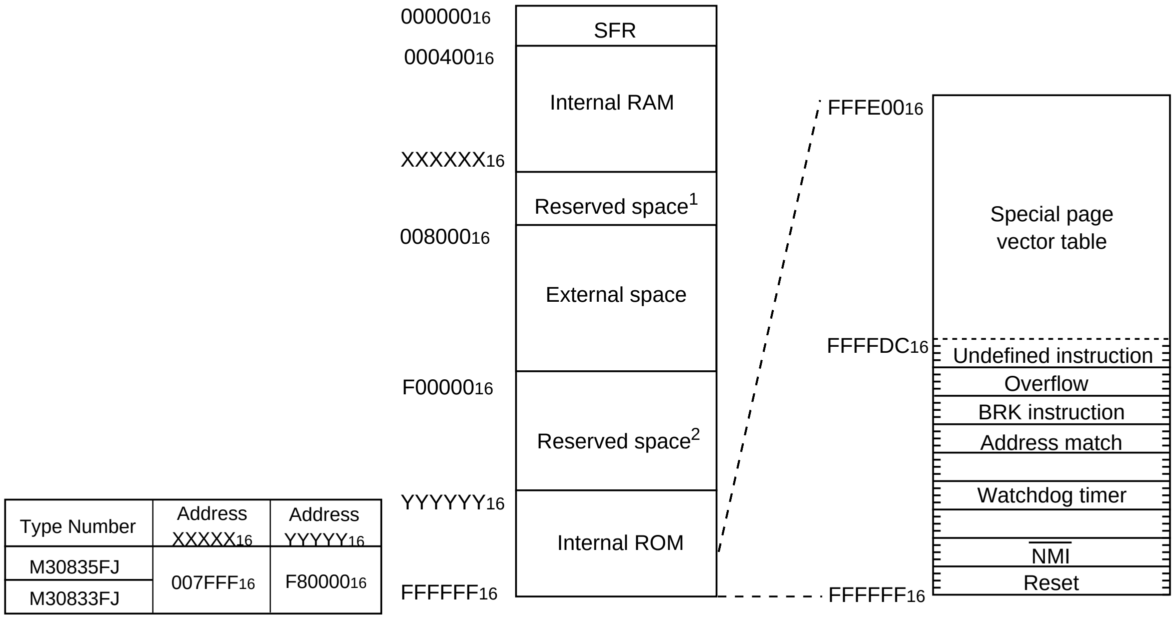
Fixed interrupt vectors are allocated in addresses from $FFFFDC_{16}$ to $FFFFFF_{16}$. The start address of each interrupt routine is stored there addresses. Refer to the section "Interrupts" for details.

Internal RAM is allocated in higher addresses beginning with address 000400_{16}. For example, a 10K-byte internal RAM are allocated in addresses from 000400_{16} to $002BFF_{16}$. Internal RAM stores data as well as a stack used to call subroutines and interrupts.

SFR is allocated in addresses from 000000_{16} to $0003FF_{16}$. The peripheral function contorl 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 cannnot used by users.

Special page vectors are allocated in addresses $FFFE00_{16}$ to $FFFFDB_{16}$. This vector is used for the JMPS instruction and JSRS instraction. Refer to the Mitubishi Electric publication "Software Mannual" for the details.

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



Type Number	Address $XXXXX_{16}$	Address $YYYYY_{16}$
M30835FJ	$007FFF_{16}$	$F80000_{16}$
M30833FJ		

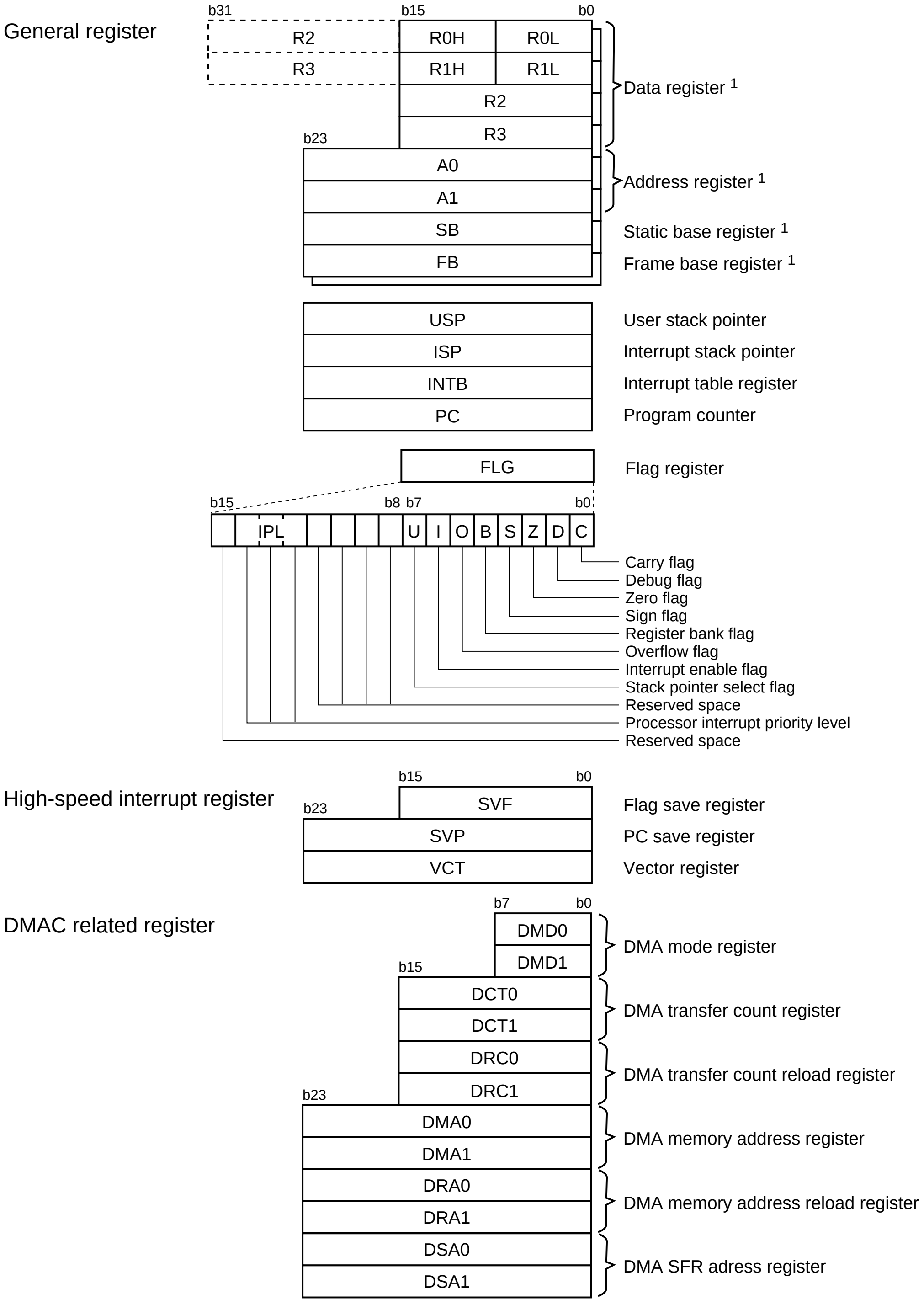
Notes:
1. In memory expansion and microprocessor modes.
2. In memory expansion mode.

Figure 1.2.1. Memory Map

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.



Notes :
1. These registers comprise a register bank. Two sets of register banks are provided.

Figure 1.3.1. CPU Register

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)

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 negative 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 recommanded performance requirement. (See Table 1.4.1.)
Main clock ocsillation 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.

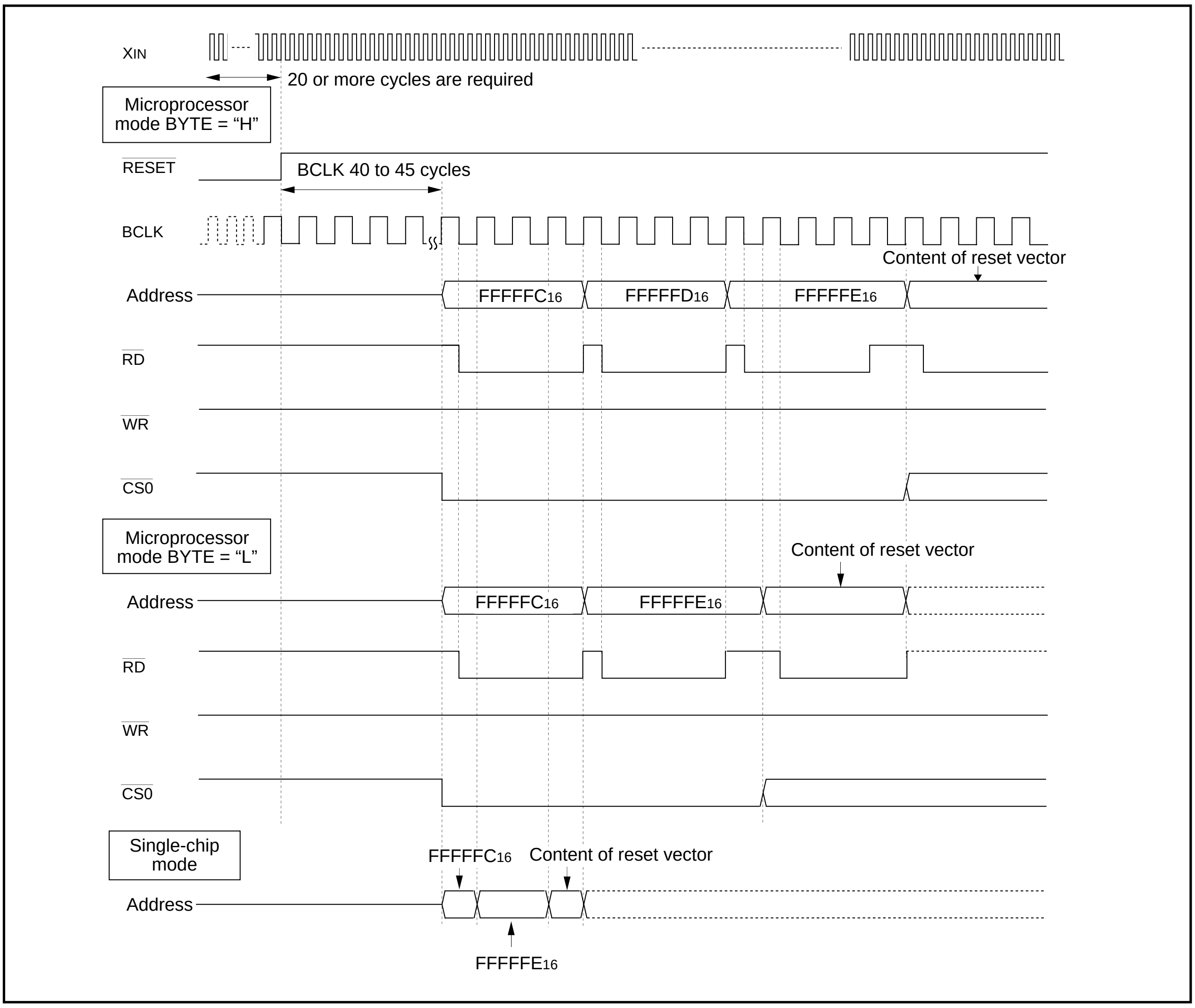


Figure 1.4.1. Reset Sequence

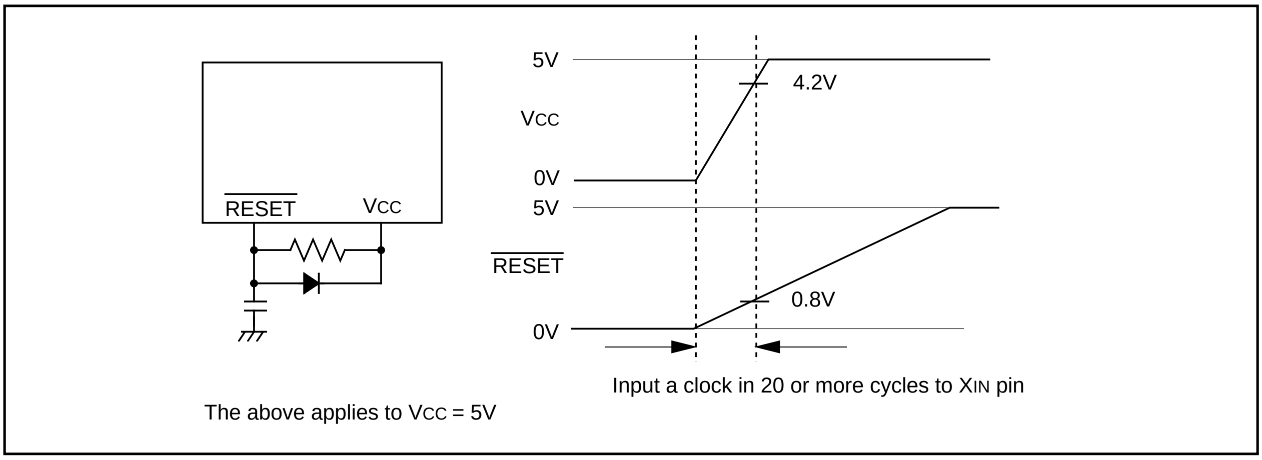

RESET
VCC
5V
VCC
4.2V
0V
5V
RESET
0.8V
0V
Input a clock in 20 or more cycles to XIN pin
The above applies to VCC = 5V

Figure 1.4.2. Reset Circuit

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)	
$P5_0$	Input port (high-impedance)	$\overline{WR}$ output (output "H")	
$P5_1$	Input port (high-impedance)	$\overline{BHE}$ output (indeterminate)	
$P5_2$	Input port (high-impedance)	$\overline{RD}$ output (output "H")	
$P5_3$	Input port (high-impedance)	BCLK output	
$P5_4$	Input port (high-impedance)	$\overline{HLDA}$ output (output value depends on an input to $\overline{HOLD}$ pin)	
$P5_5$	Input port (high-impedance)	$\overline{HOLD}$ input (high-impedance)	
$P5_6$	Input port (high-impedance)	$\overline{RAS}$ output	
$P5_7$	Input port (high-impedance)	$\overline{RDY}$ input (high-impedance)	
P6 to P15 [1]	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.

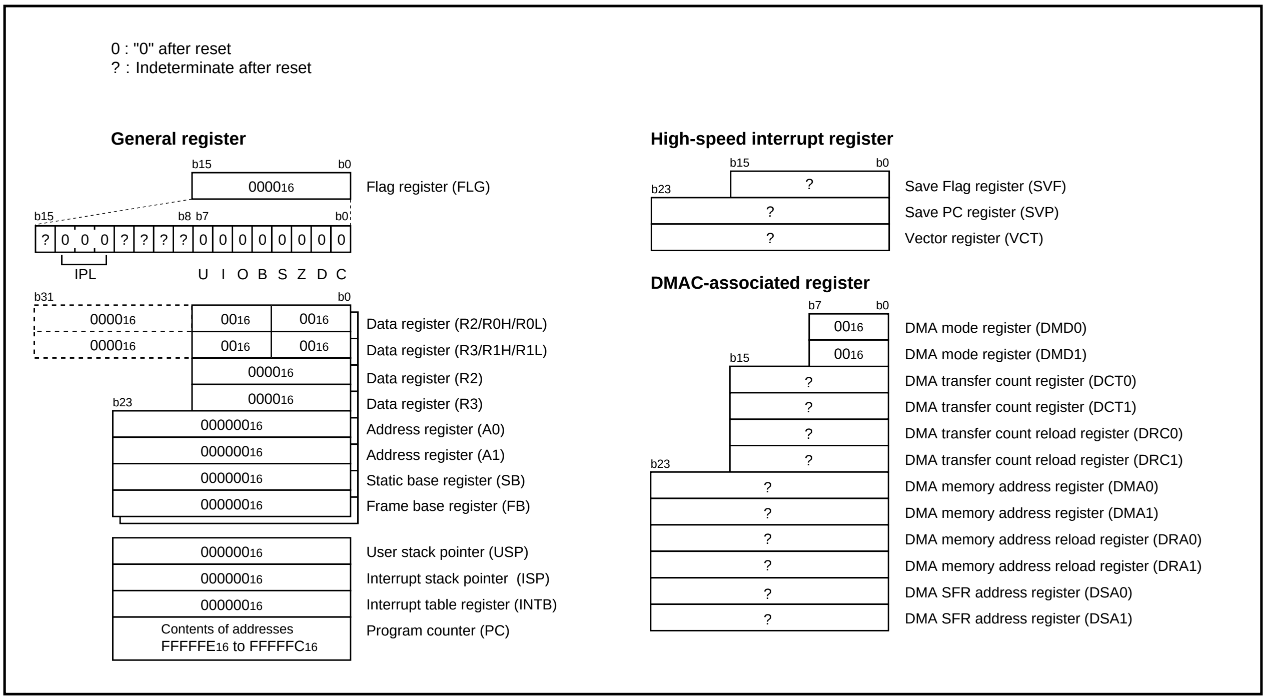

0 : "0" after reset
? : Indeterminate after reset
General register
b15 b0
0000_{16} Flag register (FLG)
b15 b8 b7 b0
? 0 0 0 ? ? ? ? 0 0 0 0 0 0 0 0
IPL U I O B S Z D C
b31 b0
0000_{16} 00_{16} 00_{16} Data register (R2/R0H/R0L)
0000_{16} 00_{16} 00_{16} Data register (R3/R1H/R1L)
0000_{16} Data register (R2)
b23 0000_{16} Data register (R3)
000000_{16} Address register (A0)
000000_{16} Address register (A1)
000000_{16} Static base register (SB)
000000_{16} Frame base register (FB)
000000_{16} User stack pointer (USP)
000000_{16} Interrupt stack pointer (ISP)
000000_{16} Interrupt table register (INTB)
Contents of addresses $FFFFFE_{16}$ to $FFFFFC_{16}$ Program counter (PC)
High-speed interrupt register
b15 b0
b23 ? Save Flag register (SVF)
? Save PC register (SVP)
? Vector register (VCT)
DMAC-associated register
b7 b0
00_{16} DMA mode register (DMD0)
b15 00_{16} DMA mode register (DMD1)
? DMA transfer count register (DCT0)
? DMA transfer count register (DCT1)
? DMA transfer count reload register (DRC0)
b23 ? DMA transfer count reload register (DRC1)
? DMA memory address register (DMA0)
? DMA memory address register (DMA1)
? DMA memory address reload register (DRA0)
? DMA memory address reload register (DRA1)
? DMA SFR address register (DSA0)
? DMA SFR address register (DSA1)

Figure 1.4.3. CPU Register after Reset

SFR

Address	Register	Symbol	Value after RESET	
0000_{16}				
0001_{16}				
0002_{16}				
0003_{16}				
0004_{16}	Processor mode register 0	PM0	$1000\ 0000_2$(CNVss pin ="L") $0000\ 0011_2$(CNVss pin ="H")	
0005_{16}	Processor mode register 1	PM1	$0X00\ 0000_2$	
0006_{16}	System clock control register 0	CM0	$0000\ X000_2$	
0007_{16}	System clock control register 1	CM1	$0010\ 0000_2$	
0008_{16}	Wait control register	WCR	$1111\ 1111_2$	
0009_{16}	Address match interrupt enable register	AIER	$XXXX\ 0000_2$	
$000A_{16}$	Protect register	PRCR	$XXXX\ 0000_2$	
$000B_{16}$	External data bus width control register	DS	$XXXX\ 1000_2$(BYTE pin ="L") $XXXX\ 0000_2$(BYTE pin ="H")	
$000C_{16}$	Main clock division register	MCD	$XXX0\ 1000_2$	
$000D_{16}$	Oscillation stop detect register	CM2	$0000\ 0000_2$	
$000E_{16}$	Watchdog timer start register	WDTS	$??_{16}$	
$000F_{16}$	Watchdog timer control register	WDC	$000?\ ????_2$	
0010_{16} 0011_{16} 0012_{16}	Address match interrupt register 0	RMAD0	000000_{16}	
0013_{16}				
0014_{16} 0015_{16} 0016_{16}	Address match interrupt register 1	RMAD1	000000_{16}	
0017_{16}	VDC control register for PLL	PLV	$XXXX\ XX01_2$	
0018_{16} 0019_{16} $001A_{16}$	Address match interrupt register 2	RMAD2	000000_{16}	
$001B_{16}$	VDC control register 0	VDC0	$0000\ 0000_2$	
$001C_{16}$ $001D_{16}$ $001E_{16}$	Address match interrupt register 3	RMAD3	000000_{16}	
$001F_{16}$	VDC control register 1	VDC1	$0000\ 0000_2$	*
0020_{16} 0021_{16} 0022_{16}	Emulator interrupt vector table register	EIAD	$F00000_{16}$	*
0023_{16}	Emulator interrupt detect register	EITD	$XXXX\ XX00_2$	*
0024_{16}	Emulator protect register	EPRR	$XXXX\ XXX0_2$	*
0025_{16}	Emulator protect control register	EMU	$XXXX\ X000_2$	*
0026_{16}				
0027_{16}				
0028_{16}				
0029_{16}				
$002A_{16}$				
$002B_{16}$				
$002C_{16}$				
$002D_{16}$				
$002E_{16}$				
$002F_{16}$				

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.

Address	Register	Symbol	Value after RESET	
0030_{16}	ROM space set register	ROA	XXXX X000$_2$	*
0031_{16}	Debug monitor space set register	DBA	1111 0000$_2$	*
0032_{16}	Expanded space set register 0	EXA0	0000 0000$_2$	*
0033_{16}	Expanded space set register 1	EXA1	0000 0000$_2$	*
0034_{16}	Expanded space set register 2	EXA2	0000 0000$_2$	*
0035_{16}	Expanded space set register 3	EXA3	0000 0000$_2$	*
0036_{16}				
0037_{16}				
0038_{16}				
0039_{16}				
$003A_{16}$				
$003B_{16}$				
$003C_{16}$				
$003D_{16}$				
$003E_{16}$				
$003F_{16}$				
0040_{16}	DRAM control register	DRAMCONT	?XXX ????$_2$	
0041_{16}	DRAM refresh interval set register	REFCNT	??$_{16}$	
0042_{16}				
0043_{16}				
0044_{16}				
0045_{16}				
0046_{16}				
0047_{16}				
0048_{16}				
0049_{16}				
$004A_{16}$				
$004B_{16}$				
$004C_{16}$				
$004D_{16}$				
$004E_{16}$				
$004F_{16}$				
0050_{16}				
0051_{16}				
0052_{16}				
0053_{16}				
0054_{16}				
0055_{16}	Flash memory control register 2	FMR2	XXXX X0X0$_2$	*
0056_{16}	Flash memory control register 1	FMR1	XXXX XXX0$_2$	*
0057_{16}	Flash memory control register 0	FMR0	XX00 0001$_2$	
0058_{16}				
0059_{16}				
$005A_{16}$				
$005B_{16}$				
$005C_{16}$				
$005D_{16}$				
$005E_{16}$				
$005F_{16}$				

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.

Address	Register	Symbol	Value after RESET
0060$_{16}$			
0061$_{16}$			
0062$_{16}$			
0063$_{16}$			
0064$_{16}$			
0065$_{16}$			
0066$_{16}$			
0067$_{16}$			
0068$_{16}$	DMA0 interrupt control register	DM0IC	XXXX ?000$_2$
0069$_{16}$	Timer B5 interrupt control register	TB5IC	XXXX ?000$_2$
006A$_{16}$	DMA2 interrupt control register	DM2IC	XXXX ?000$_2$
006B$_{16}$	UART2 receive /ACK interrupt control register	S2RIC	XXXX ?000$_2$
006C$_{16}$	Timer A0 interrupt control register	TA0IC	XXXX ?000$_2$
006D$_{16}$	UART3 receive /ACK interrupt control register	S3RIC	XXXX ?000$_2$
006E$_{16}$	Timer A2 interrupt control register	TA2IC	XXXX ?000$_2$
006F$_{16}$	UART4 receive /ACK interrupt control register	S4RIC	XXXX ?000$_2$
0070$_{16}$	Timer A4 interrupt control register	TA4IC	XXXX ?000$_2$
0071$_{16}$	UART0/UART3 bus conflict detect interrupt control register	BCN0IC/BCN3IC	XXXX ?000$_2$
0072$_{16}$	UART0 receive/ACK interrupt control register	S0RIC	XXXX ?000$_2$
0073$_{16}$	A-D0 conversion interrupt control register	AD0IC	XXXX ?000$_2$
0074$_{16}$	UART1 receive/ACK interrupt control register	S1RIC	XXXX ?000$_2$
0075$_{16}$	Intelligent I/O interrupt control register 0	IIO0IC	XXXX ?000$_2$
0076$_{16}$	Timer B1 interrupt control register	TB1IC	XXXX ?000$_2$
0077$_{16}$	Intelligent I/O interrupt control register 2	IIO2IC	XXXX ?000$_2$
0078$_{16}$	Timer B3 interrupt control register	TB3IC	XXXX ?000$_2$
0079$_{16}$	Intelligent I/O interrupt control register 4	IIO4IC	XXXX ?000$_2$
007A$_{16}$	INT5 interrupt control register	INT5IC	XX00 ?000$_2$
007B$_{16}$	Intelligent I/O interrupt control register 6	IIO6IC	XXXX ?000$_2$
007C$_{16}$	INT3 interrupt control register	INT3IC	XX00 ?000$_2$
007D$_{16}$	Intelligent I/O interrupt control register 8	IIO8IC	XXXX ?000$_2$
007E$_{16}$	INT1 interrupt control register	INT1IC	XX00 ?000$_2$
007F$_{16}$	Intelligent I/O interrupt control register 10/ CAN interrupt 1 control register	IIO10IC CAN1IC	XXXX ?000$_2$
0080$_{16}$			
0081$_{16}$	Intelligent I/O interrupt control register 11/ CAN interrupt 2 control register	IIO11IC CAN2IC	XXXX ?000$_2$
0082$_{16}$			
0083$_{16}$			
0084$_{16}$			
0085$_{16}$			
0086$_{16}$	A-D1 conversion interrupt control register	AD1IC	XXXX ?000$_2$
0087$_{16}$			
0088$_{16}$	DMA1 interrupt control register	DM1IC	XXXX ?000$_2$
0089$_{16}$	UART2 transmit /NACK interrupt control register	S2TIC	XXXX ?000$_2$
008A$_{16}$	DMA3 interrupt control register	DM3IC	XXXX ?000$_2$
008B$_{16}$	UART3 transmit /NACK interrupt control register	S3TIC	XXXX ?000$_2$
008C$_{16}$	Timer A1 interrupt control register	TA1IC	XXXX ?000$_2$
008D$_{16}$	UART4 transmit /NACK interrupt control register	S4TIC	XXXX ?000$_2$
008E$_{16}$	Timer A3 interrupt control register	TA3IC	XXXX ?000$_2$
008F$_{16}$	UART2 bus conflict detect interrupt control register	BCN2IC	XXXX ?000$_2$

X : Nothing is assigned ? : Indetermination

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

Address	Register	Symbol	Value after RESET
0090$_{16}$	UART0 transmit /NACK interrupt control register	S0TIC	XXXX ?000$_2$
0091$_{16}$	UART1/UART4 bus conflict detect interrupt control register	BCN1IC/BCN4IC	XXXX ?000$_2$
0092$_{16}$	UART1 transmit/NACK interrupt control register	S1TIC	XXXX ?000$_2$
0093$_{16}$	Key input interrupt control register	KUPIC	XXXX ?000$_2$
0094$_{16}$	Timer B0 interrupt control register	TB0IC	XXXX ?000$_2$
0095$_{16}$	Intelligent I/O interrupt control register 1	IIO1IC	XXXX ?000$_2$
0096$_{16}$	Timer B2 interrupt control register	TB2IC	XXXX ?000$_2$
0097$_{16}$	Intelligent I/O interrupt control register 3	IIO3IC	XXXX ?000$_2$
0098$_{16}$	Timer B4 interrupt control register	TB4IC	XXXX ?000$_2$
0099$_{16}$	Intelligent I/O interrupt control register 5	IIO5IC	XXXX ?000$_2$
009A$_{16}$	INT4 interrupt control register	INT4IC	XX00 ?000$_2$
009B$_{16}$	Intelligent I/O interrupt control register 7	IIO7IC	XXXX ?000$_2$
009C$_{16}$	INT2 interrupt control register	INT2IC	XX00 ?000$_2$
009D$_{16}$	Intelligent I/O interrupt control register 9/ CAN interrupt 0 control register	IIO9IC CAN0IC	XXXX ?000$_2$
009E$_{16}$	INT0 interrupt control register	INT0IC	XX00 ?000$_2$
009F$_{16}$	Exit priority control register	RLVL	XX0X 0000$_2$
00A0$_{16}$	Interrupt request register 0	IIO0IR	0000 000X$_2$
00A1$_{16}$	Interrupt request register 1	IIO1IR	0000 000X$_2$
00A2$_{16}$	Interrupt request register 2	IIO2IR	0000 000X$_2$
00A3$_{16}$	Interrupt request register 3	IIO3IR	0000 000X$_2$
00A4$_{16}$	Interrupt request register 4	IIO4IR	0000 000X$_2$
00A5$_{16}$	Interrupt request register 5	IIO5IR	0000 000X$_2$
00A6$_{16}$	Interrupt request register 6	IIO6IR	0000 000X$_2$
00A7$_{16}$	Interrupt request register 7	IIO7IR	0000 000X$_2$
00A8$_{16}$	Interrupt request register 8	IIO8IR	0000 000X$_2$
00A9$_{16}$	Interrupt request register 9	IIO9IR	0000 000X$_2$
00AA$_{16}$	Interrupt request register 10	IIO10IR	0000 000X$_2$
00AB$_{16}$	Interrupt request register 11	IIO11IR	0000 000X$_2$
00AC$_{16}$			
00AD$_{16}$			
00AE$_{16}$			
00AF$_{16}$			
00B0$_{16}$	Interrupt enable register 0	IIO0IE	0000 0000$_2$
00B1$_{16}$	Interrupt enable register 1	IIO1IE	0000 0000$_2$
00B2$_{16}$	Interrupt enable register 2	IIO2IE	0000 0000$_2$
00B3$_{16}$	Interrupt enable register 3	IIO3IE	0000 0000$_2$
00B4$_{16}$	Interrupt enable register 4	IIO4IE	0000 0000$_2$
00B5$_{16}$	Interrupt enable register 5	IIO5IE	0000 0000$_2$
00B6$_{16}$	Interrupt enable register 6	IIO6IE	0000 0000$_2$
00B7$_{16}$	Interrupt enable register 7	IIO7IE	0000 0000$_2$
00B8$_{16}$	Interrupt enable register 8	IIO8IE	0000 0000$_2$
00B9$_{16}$	Interrupt enable register 9	IIO9IE	0000 0000$_2$
00BA$_{16}$	Interrupt enable register 10	IIO10IE	0000 0000$_2$
00BB$_{16}$	Interrupt enable register 11	IIO11IE	0000 0000$_2$
00BC$_{16}$			
00BD$_{16}$			
00BE$_{16}$			
00BF$_{16}$			

X : Nothing is assigned ? : Indetermination

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

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

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

Address	Register	Symbol	Value after RESET
00F0$_{16}$	Group 0 data compare register 0	G0CMP0	??$_{16}$
00F1$_{16}$	Group 0 data compare register 1	G0CMP1	??$_{16}$
00F2$_{16}$	Group 0 data compare register 2	G0CMP2	??$_{16}$
00F3$_{16}$	Group 0 data compare register 3	G0CMP3	??$_{16}$
00F4$_{16}$	Group 0 data mask register 0	G0MSK0	??$_{16}$
00F5$_{16}$	Group 0 data mask register 1	G0MSK1	??$_{16}$
00F6$_{16}$			
00F7$_{16}$			
00F8$_{16}$	Group 0 receive CRC code register	G0RCRC	??$_{16}$
00F9$_{16}$			??$_{16}$
00FA$_{16}$	Group 0 transmit CRC code register	G0TCRC	0000 0000$_2$
00FB$_{16}$			0000 0000$_2$
00FC$_{16}$	Group 0 SI/O extended mode register	G0EMR	0000 0000$_2$
00FD$_{16}$	Group 0 SI/O extended receive control register	G0ERC	0000 0000$_2$
00FE$_{16}$	Group 0 SI/O special communication interrupt detect register	G0IRF	0000 00XX$_2$
00FF$_{16}$	Group 0 SI/O extended transmit control register	G0ETC	0000 0XXX$_2$
0100$_{16}$	Group 1 time measurement/waveform generation register 0	G1TM0/G1PO0	??$_{16}$
0101$_{16}$			??$_{16}$
0102$_{16}$	Group 1 time measurement/waveform generation register 1	G1TM1/G1PO1	??$_{16}$
0103$_{16}$			??$_{16}$
0104$_{16}$	Group 1 time measurement/waveform generation register 2	G1TM2/G1PO2	??$_{16}$
0105$_{16}$			??$_{16}$
0106$_{16}$	Group 1 time measurement/waveform generation register 3	G1TM3/G1PO3	??$_{16}$
0107$_{16}$			??$_{16}$
0108$_{16}$	Group 1 time measurement/waveform generation register 4	G1TM4/G1PO4	??$_{16}$
0109$_{16}$			??$_{16}$
010A$_{16}$	Group 1 time measurement/waveform generation register 5	G1TM5/G1PO5	??$_{16}$
010B$_{16}$			??$_{16}$
010C$_{16}$	Group 1 time measurement/waveform generation register 6	G1TM6/G1PO6	??$_{16}$
010D$_{16}$			??$_{16}$
010E$_{16}$	Group 1 time measurement/waveform generation register 7	G1TM7/G1PO7	??$_{16}$
010F$_{16}$			??$_{16}$
0110$_{16}$	Group 1 waveform generation control register 0	G1POCR0	0X00 X000$_2$
0111$_{16}$	Group 1 waveform generation control register 1	G1POCR1	0X00 X000$_2$
0112$_{16}$	Group 1 waveform generation control register 2	G1POCR2	0X00 X000$_2$
0113$_{16}$	Group 1 waveform generation control register 3	G1POCR3	0X00 X000$_2$
0114$_{16}$	Group 1 waveform generation control register 4	G1POCR4	0X00 X000$_2$
0115$_{16}$	Group 1 waveform generation control register 5	G1POCR5	0X00 X000$_2$
0116$_{16}$	Group 1 waveform generation control register 6	G1POCR6	0X00 X000$_2$
0117$_{16}$	Group 1 waveform generation control register 7	G1POCR7	0X00 X000$_2$
0118$_{16}$	Group 1 time measurement control register 0	G1TMCR0	0000 0000$_2$
0119$_{16}$	Group 1 time measurement control register 1	G1TMCR1	0000 0000$_2$
011A$_{16}$	Group 1 time measurement control register 2	G1TMCR2	0000 0000$_2$
011B$_{16}$	Group 1 time measurement control register 3	G1TMCR3	0000 0000$_2$
011C$_{16}$	Group 1 time measurement control register 4	G1TMCR4	0000 0000$_2$
011D$_{16}$	Group 1 time measurement control register 5	G1TMCR5	0000 0000$_2$
011E$_{16}$	Group 1 time measurement control register 6	G1TMCR6	0000 0000$_2$
011F$_{16}$	Group 1 time measurement control register 7	G1TMCR7	0000 0000$_2$

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

Address	Register	Symbol	Value after RESET
0120$_{16}$ 0121$_{16}$	Group 1 base timer register	G1BT	??$_{16}$??$_{16}$
0122$_{16}$	Group 1 base timer control register 0	G1BCR0	0000 0000$_2$
0123$_{16}$	Group 1 base timer control register 1	G1BCR1	0000 0000$_2$
0124$_{16}$	Group 1 time measurement prescaler register 6	G1TPR6	0000 0000$_2$
0125$_{16}$	Group 1 time measurement prescaler register 7	G1TPR7	0000 0000$_2$
0126$_{16}$	Group 1 function enable register	G1FE	0000 0000$_2$
0127$_{16}$	Group 1 function select register	G1FS	0000 0000$_2$
0128$_{16}$ 0129$_{16}$	Group 1 SI/O receive buffer register	G1RB	???? ????$_2$ XX00 XXXX$_2$
012A$_{16}$	Group 1 transmit buffer/receive data register	G1TB/G1DR	??$_{16}$
012B$_{16}$			
012C$_{16}$	Group 1 receive input register	G1RI	??$_{16}$
012D$_{16}$	Group 1 SI/O communication mode register	G1MR	0000 0000$_2$
012E$_{16}$	Group 1 transmit output register	G1TO	??$_{16}$
012F$_{16}$	Group 1 SI/O communication control register	G1CR	0000 X000$_2$
0130$_{16}$	Group 1 data compare register 0	G1CMP0	??$_{16}$
0131$_{16}$	Group 1 data compare register 1	G1CMP1	??$_{16}$
0132$_{16}$	Group 1 data compare register 2	G1CMP2	??$_{16}$
0133$_{16}$	Group 1 data compare register 3	G1CMP3	??$_{16}$
0134$_{16}$	Group 1 data mask register 0	G1MSK0	??$_{16}$
0135$_{16}$	Group 1 data mask register 1	G1MSK1	??$_{16}$
0136$_{16}$			
0137$_{16}$			
0138$_{16}$ 0139$_{16}$	Group 1 receive CRC code register	G1RCRC	??$_{16}$??$_{16}$
013A$_{16}$ 013B$_{16}$	Group 1 transmit CRC code register	G1TCRC	0000 0000$_2$ 0000 0000$_2$
013C$_{16}$	Group 1 SI/O extended mode register	G1EMR	0000 0000$_2$
013D$_{16}$	Group 1 SI/O extended receive control register	G1ERC	0000 0000$_2$
013E$_{16}$	Group 1 SI/O special communication interrupt detect register	G1IRF	0000 00XX$_2$
013F$_{16}$	Group 1 SI/O extended transmit control register	G1ETC	0000 0XXX$_2$
0140$_{16}$ 0141$_{16}$	Group 2 waveform generation register 0	G2PO0	??$_{16}$??$_{16}$
0142$_{16}$ 0143$_{16}$	Group 2 waveform generation register 1	G2PO1	??$_{16}$??$_{16}$
0144$_{16}$ 0145$_{16}$	Group 2 waveform generation register 2	G2PO2	??$_{16}$??$_{16}$
0146$_{16}$ 0147$_{16}$	Group 2 waveform generation register 3	G2PO3	??$_{16}$??$_{16}$
0148$_{16}$ 0149$_{16}$	Group 2 waveform generation register 4	G2PO4	??$_{16}$??$_{16}$
014A$_{16}$ 014B$_{16}$	Group 2 waveform generation register 5	G2PO5	??$_{16}$??$_{16}$
014C$_{16}$ 014D$_{16}$	Group 2 waveform generation register 6	G2PO6	??$_{16}$??$_{16}$
014E$_{16}$ 014F$_{16}$	Group 2 waveform generation register 7	G2PO7	??$_{16}$??$_{16}$

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

Address	Register	Symbol	Value after RESET
0150$_{16}$	Group 2 waveform generation control register 0	G2POCR0	0000 0000$_2$
0151$_{16}$	Group 2 waveform generation control register 1	G2POCR1	0000 0000$_2$
0152$_{16}$	Group 2 waveform generation control register 2	G2POCR2	0000 0000$_2$
0153$_{16}$	Group 2 waveform generation control register 3	G2POCR3	0000 0000$_2$
0154$_{16}$	Group 2 waveform generation control register 4	G2POCR4	0000 0000$_2$
0155$_{16}$	Group 2 waveform generation control register 5	G2POCR5	0000 0000$_2$
0156$_{16}$	Group 2 waveform generation control register 6	G2POCR6	0000 0000$_2$
0157$_{16}$	Group 2 waveform generation control register 7	G2POCR7	0000 0000$_2$
0158$_{16}$			
0159$_{16}$			
015A$_{16}$			
015B$_{16}$			
015C$_{16}$			
015D$_{16}$			
015E$_{16}$			
015F$_{16}$			
0160$_{16}$ 0161$_{16}$	Group 2 base timer register	G2BT	??$_{16}$??$_{16}$
0162$_{16}$	Group 2 base timer control register 0	G2BCR0	0000 0000$_2$
0163$_{16}$	Group 2 base timer control register 1	G2BCR1	0000 0000$_2$
0164$_{16}$	Base timer start register	BTSR	XXXX 0000$_2$
0165$_{16}$			
0166$_{16}$	Group 2 function enable register	G2FE	0000 0000$_2$
0167$_{16}$	Group 2 RTP output buffer register	G2RTP	0000 0000$_2$
0168$_{16}$			
0169$_{16}$			
016A$_{16}$	Group 2 SI/O communication mode register	G2MR	00XX X000$_2$
016B$_{16}$	Group 2 SI/O communication control register	G2CR	0000 X000$_2$
016C$_{16}$ 016D$_{16}$	Group 2 SI/O transmit buffer register	G2TB	???? ????$_2$???X X???$_2$
016E$_{16}$ 016F$_{16}$	Group 2 SI/O receive buffer register	G2RB	???? ????$_2$ XXX? XXXX$_2$
0170$_{16}$ 0171$_{16}$	Group 2 IE Bus address register	IEAR	???? ????$_2$ XXXX ????$_2$
0172$_{16}$	Group 2 IE Bus control register	IECR	00XX X000$_2$
0173$_{16}$	Group 2 IE Bus transmit interrupt cause detect register	IETIF	XXX0 0000$_2$
0174$_{16}$	Group 2 IE Bus receive interrupt cause detect register	IERIF	XXX0 0000$_2$
0175$_{16}$			
0176$_{16}$			
0177$_{16}$			
0178$_{16}$	Input function select register	IPS	0000 0000$_2$
0179$_{16}$			
017A$_{16}$	Group 3 SI/O communication mode register	G3MR	00XX 0000$_2$
017B$_{16}$	Group 3 SI/O communication control register	G3CR	0000 X000$_2$
017C$_{16}$ 017D$_{16}$	Group 3 SI/O transmit buffer register	G3TB	??$_{16}$??$_{16}$
017E$_{16}$ 017F$_{16}$	Group 3 SI/O receive buffer register	G3RB	??$_{16}$??$_{16}$

X : Nothing is assigned ? : Indetermination

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

Address	Register	Symbol	Value after RESET	
0180_{16} 0181_{16}	Group 3 waveform generation register 0	G3PO0	$??_{16}$ $??_{16}$	
0182_{16} 0183_{16}	Group 3 waveform generation register 1	G3PO1	$??_{16}$ $??_{16}$	
0184_{16} 0185_{16}	Group 3 waveform generation register 2	G3PO2	$??_{16}$ $??_{16}$	
0186_{16} 0187_{16}	Group 3 waveform generation register 3	G3PO3	$??_{16}$ $??_{16}$	
0188_{16} 0189_{16}	Group 3 waveform generation register 4	G3PO4	$??_{16}$ $??_{16}$	
$018A_{16}$ $018B_{16}$	Group 3 waveform generation register 5	G3PO5	$??_{16}$ $??_{16}$	
$018C_{16}$ $018D_{16}$	Group 3 waveform generation register 6	G3PO6	$??_{16}$ $??_{16}$	
$018E_{16}$ $018F_{16}$	Group 3 waveform generation register 7	G3PO7	$??_{16}$ $??_{16}$	
0190_{16}	Group 3 waveform generation control register 0	G3POCR0	$0000\ 0000_2$	
0191_{16}	Group 3 waveform generation control register 1	G3POCR1	$0000\ 0000_2$	
0192_{16}	Group 3 waveform generation control register 2	G3POCR2	$0000\ 0000_2$	
0193_{16}	Group 3 waveform generation control register 3	G3POCR3	$0000\ 0000_2$	
0194_{16}	Group 3 waveform generation control register 4	G3POCR4	$0000\ 0000_2$	
0195_{16}	Group 3 waveform generation control register 5	G3POCR5	$0000\ 0000_2$	
0196_{16}	Group 3 waveform generation control register 6	G3POCR6	$0000\ 0000_2$	
0197_{16}	Group 3 waveform generation control register 7	G3POCR7	$0000\ 0000_2$	
0198_{16} 0199_{16}	Group 3 waveform generation mask register 4	G3MK4	$??_{16}$ $??_{16}$	
$019A_{16}$ $019B_{16}$	Group 3 waveform generation mask register 5	G3MK5	$??_{16}$ $??_{16}$	
$019C_{16}$ $019D_{16}$	Group 3 waveform generation mask register 6	G3MK6	$??_{16}$ $??_{16}$	
$019E_{16}$ $019F_{16}$	Group 3 waveform generation mask register 7	G3MK7	$??_{16}$ $??_{16}$	
$01A0_{16}$ $01A1_{16}$	Group 3 base timer register	G3BT	$??_{16}$ $??_{16}$	
$01A2_{16}$	Group 3 base timer control register 0	G3BCR0	$0000\ 0000_2$	
$01A3_{16}$	Group 3 base timer control register 1	G3BCR1	$0000\ 0000_2$	
$01A4_{16}$				
$01A5_{16}$				
$01A6_{16}$	Group 3 function enable register	G3FE	$0000\ 0000_2$	
$01A7_{16}$	Group 3 RTP output buffer register	G3RTP	$0000\ 0000_2$	
$01A8_{16}$				
$01A9_{16}$				
$01AA_{16}$				
$01AB_{16}$				
$01AC_{16}$	Group 3 HDLC communication control register	HDLC	$0000\ 0000_2$	*
$01AD_{16}$	Group 3 SI/O communication flag register	G3FLG	$XXXX\ XXX0_2$	
$01AE_{16}$ $01AF_{16}$	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.

Address	Register	Symbol	Value after RESET	
01B0$_{16}$ 01B1$_{16}$	Group 3 HDLC address compare register 0	HADR0	0000 0000$_2$ 0000 0000$_2$	*
01B2$_{16}$ 01B3$_{16}$	Group 3 HDLC address mask register 0	HMSK0	0000 0000$_2$ 0000 0000$_2$	*
01B4$_{16}$ 01B5$_{16}$	Group 3 HDLC address compare register 1	HADR1	0000 0000$_2$ 0000 0000$_2$	*
01B6$_{16}$ 01B7$_{16}$	Group 3 HDLC address mask register 1	HMSK1	0000 0000$_2$ 0000 0000$_2$	*
01B8$_{16}$ 01B9$_{16}$	Group 3 HDLC address compare register 2	HADR2	0000 0000$_2$ 0000 0000$_2$	*
01BA$_{16}$ 01BB$_{16}$	Group 3 HDLC address mask register 2	HMSK2	0000 0000$_2$ 0000 0000$_2$	*
01BC$_{16}$ 01BD$_{16}$	Group 3 HDLC address compare register 3	HADR3	0000 0000$_2$ 0000 0000$_2$	*
01BE$_{16}$ 01BF$_{16}$	Group 3 HDLC address mask register 3	HMSK3	0000 0000$_2$ 0000 0000$_2$	*
01C0$_{16}$ 01C1$_{16}$	A-D1 register 0	AD10	???? ????$_2$ XXXX XX??$_2$	
01C2$_{16}$ 01C3$_{16}$	A-D1 register 1	AD11	???? ????$_2$ XXXX XX??$_2$	
01C4$_{16}$ 01C5$_{16}$	A-D1 register 2	AD12	???? ????$_2$ XXXX XX??$_2$	
01C6$_{16}$ 01C7$_{16}$	A-D1 register 3	AD13	???? ????$_2$ XXXX XX??$_2$	
01C8$_{16}$ 01C9$_{16}$	A-D1 register 4	AD14	???? ????$_2$ XXXX XX??$_2$	
01CA$_{16}$ 01CB$_{16}$	A-D1 register 5	AD15	???? ????$_2$ XXXX XX??$_2$	
01CC$_{16}$ 01CD$_{16}$	A-D1 register 6	AD16	???? ????$_2$ XXXX XX??$_2$	
01CE$_{16}$ 01CF$_{16}$	A-D1 register 7	AD17	???? ????$_2$ XXXX XX??$_2$	
01D0$_{16}$				
01D1$_{16}$				
01D2$_{16}$				
01D3$_{16}$				
01D4$_{16}$	A-D1 control register 2	AD1CON2	X00X X000$_2$	
01D5$_{16}$				
01D6$_{16}$	A-D1 control register 0	AD1CON0	0000 0000$_2$	
01D7$_{16}$	A-D1 control register 1	AD1CON1	XX00 0000$_2$	
01D8$_{16}$				
01D9$_{16}$				
01DA$_{16}$				
01DB$_{16}$				
01DC$_{16}$				
01DD$_{16}$				
01DE$_{16}$				
01DF$_{16}$				

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.

Address	Register	Symbol	Value after RESET
01E0$_{16}$	CAN0 message slot buffer 0 standard ID0	C0SLOT0_0	XXX? ????$_2$
01E1$_{16}$	CAN0 message slot buffer 0 standard ID1	C0SLOT0_1	XX?? ????$_2$
01E2$_{16}$	CAN0 message slot buffer 0 extended ID0	C0SLOT0_2	XXXX ????$_2$
01E3$_{16}$	CAN0 message slot buffer 0 extended ID1	C0SLOT0_3	??$_{16}$
01E4$_{16}$	CAN0 message slot buffer 0 extended ID2	C0SLOT0_4	XX?? ????$_2$
01E5$_{16}$	CAN0 message slot buffer 0 data length code	C0SLOT0_5	XXXX ????$_2$
01E6$_{16}$	CAN0 message slot buffer 0 data 0	C0SLOT0_6	??$_{16}$
01E7$_{16}$	CAN0 message slot buffer 0 data 1	C0SLOT0_7	??$_{16}$
01E8$_{16}$	CAN0 message slot buffer 0 data 2	C0SLOT0_8	??$_{16}$
01E9$_{16}$	CAN0 message slot buffer 0 data 3	C0SLOT0_9	??$_{16}$
01EA$_{16}$	CAN0 message slot buffer 0 data 4	C0SLOT0_10	??$_{16}$
01EB$_{16}$	CAN0 message slot buffer 0 data 5	C0SLOT0_11	??$_{16}$
01EC$_{16}$	CAN0 message slot buffer 0 data 6	C0SLOT0_12	??$_{16}$
01ED$_{16}$	CAN0 message slot buffer 0 data 7	C0SLOT0_13	??$_{16}$
01EE$_{16}$	CAN0 message slot buffer 0 time stamp high-order	C0SLOT0_14	??$_{16}$
01EF$_{16}$	CAN0 message slot buffer 0 time stamp low-order	C0SLOT0_15	??$_{16}$
01F0$_{16}$	CAN0 message slot buffer 1 standard ID0	C0SLOT1_0	XXX? ????$_2$
01F1$_{16}$	CAN0 message slot buffer 1 standard ID1	C0SLOT1_1	XX?? ????$_2$
01F2$_{16}$	CAN0 message slot buffer 1 extended ID0	C0SLOT1_2	XXXX ????$_2$
01F3$_{16}$	CAN0 message slot buffer 1 extended ID1	C0SLOT1_3	??$_{16}$
01F4$_{16}$	CAN0 message slot buffer 1 extended ID2	C0SLOT1_4	XX?? ????$_2$
01F5$_{16}$	CAN0 message slot buffer 1 data length code	C0SLOT1_5	XXXX ????$_2$
01F6$_{16}$	CAN0 message slot buffer 1 data 0	C0SLOT1_6	??$_{16}$
01F7$_{16}$	CAN0 message slot buffer 1 data 1	C0SLOT1_7	??$_{16}$
01F8$_{16}$	CAN0 message slot buffer 1 data 2	C0SLOT1_8	??$_{16}$
01F9$_{16}$	CAN0 message slot buffer 1 data 3	C0SLOT1_9	??$_{16}$
01FA$_{16}$	CAN0 message slot buffer 1 data 4	C0SLOT1_10	??$_{16}$
01FB$_{16}$	CAN0 message slot buffer 1 data 5	C0SLOT1_11	??$_{16}$
01FC$_{16}$	CAN0 message slot buffer 1 data 6	C0SLOT1_12	??$_{16}$
01FD$_{16}$	CAN0 message slot buffer 1 data 7	C0SLOT1_13	??$_{16}$
01FE$_{16}$	CAN0 message slot buffer 1 time stamp high-order	C0SLOT1_14	??$_{16}$
01FF$_{16}$	CAN0 message slot buffer 1 time stamp low-order	C0SLOT1_15	??$_{16}$
0200$_{16}$	CAN0 control register 0	C0CTLR0	XX01 0X01$_2$[1]
0201$_{16}$			XXXX 0000$_2$[1]
0202$_{16}$	CAN0 status register	C0STR	0000 0000$_2$[1]
0203$_{16}$			X000 0X01$_2$[1]
0204$_{16}$	CAN0 extended ID register	C0IDR	0000 0000$_2$[1]
0205$_{16}$			0000 0000$_2$[1]
0206$_{16}$	CAN0 configuration register	C0CONR	0000 XXXX$_2$[1]
0207$_{16}$			0000 0000$_2$[1]
0208$_{16}$	CAN0 time stamp register	C0TSR	0000 0000$_2$[1]
0209$_{16}$			0000 0000$_2$[1]
020A$_{16}$	CAN0 transmit error count register	C0TEC	0000 0000$_2$[1]
020B$_{16}$	CAN0 receive error count register	C0REC	0000 0000$_2$[1]
020C$_{16}$	CAN0 slot interrupt status register	C0SISTR	0000 0000$_2$[1]
020D$_{16}$			0000 0000$_2$[1]
020E$_{16}$			
020F$_{16}$			

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_{16} 0211_{16}	CAN0 slot interrupt mask register	C0SIMKR	0000 0000$_2$ [2] 0000 0000$_2$ [2]	
0212_{16}				
0213_{16}				
0214_{16}	CAN0 error interrupt mask register	C0EIMKR	XXXX X000$_2$ [2]	
0215_{16}	CAN0 error interrupt status register	C0EISTR	XXXX X000$_2$ [2]	
0216_{16}				
0217_{16}	CAN0 baud rate prescaler	C0BRP	0000 0001$_2$ [2]	
0218_{16}				
0219_{16}				
$021A_{16}$				
$021B_{16}$				
$021C_{16}$				
$021D_{16}$				
$021E_{16}$				
$021F_{16}$				
0220_{16}				
0221_{16}				
0222_{16}				
0223_{16}				
0224_{16}				
0225_{16}				
0226_{16}				
0227_{16}				
0228_{16}	CAN0 global mask register standard ID0	C0GMR0	XXX0 0000$_2$ [2]	
0229_{16}	CAN0 global mask register standard ID1	C0GMR1	XX00 0000$_2$ [2]	
$022A_{16}$	CAN0 global mask register extended ID0	C0GMR2	XXXX 0000$_2$ [2]	
$022B_{16}$	CAN0 global mask register extended ID1	C0GMR3	0000 0000$_2$ [2]	
$022C_{16}$	CAN0 global mask register extended ID2	C0GMR4	XX00 0000$_2$ [2]	
$022D_{16}$				
$022E_{16}$				
$022F_{16}$				(Note 1)
0230_{16}	CAN0 message slot 0 control register / CAN0 local mask register A standard ID0	C0MCTL0/ C0LMAR0	0000 0000$_2$ [2] XXX0 0000$_2$ [2]	
0231_{16}	CAN0 message slot 1 control register / CAN0 local mask register A standard ID1	C0MCTL1/ C0LMAR1	0000 0000$_2$ [2] XX00 0000$_2$ [2]	
0232_{16}	CAN0 message slot 2 control register / CAN0 local mask register A extended ID0	C0MCTL2/ C0LMAR2	0000 0000$_2$ [2] XXXX 0000$_2$ [2]	
0233_{16}	CAN0 message slot 3 control register / CAN0 local mask register A extended ID1	C0MCTL3/ C0LMAR3	0000 0000$_2$ [2] 0000 0000$_2$ [2]	
0234_{16}	CAN0 message slot 4 control register / CAN0 local mask register A extended ID2	C0MCTL4/ C0LMAR4	0000 0000$_2$ [2] XX00 0000$_2$ [2]	
0235_{16}	CAN0 message slot 5 control register	C0MCTL5	0000 0000$_2$ [2]	
0236_{16}	CAN0 message slot 6 control register	C0MCTL6	0000 0000$_2$ [2]	
0237_{16}	CAN0 message slot 7 control register	C0MCTL7	0000 0000$_2$ [2]	
0238_{16}	CAN0 message slot 8 control register / CAN0 local mask register B standard ID0	C0MCTL8/ C0LMBR0	0000 0000$_2$ [2] XXX0 0000$_2$ [2]	

X : Nothing is assigned ? : Indetermination

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

Notes :

1. Addresses 0230_{16} to $023F_{16}$ are switched its function in the BankSel bit of C0CTLR1 register.
2. 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	
0239$_{16}$	CAN0 message slot 9 control register / CAN0 local mask register B standard ID1	C0MCTL9/ C0LMBR1	0000 0000$_2$ [2] XX00 0000$_2$ [2]	(Note 1)
023A$_{16}$	CAN0 message slot 10 control register / CAN0 local mask register B extended ID0	C0MCTL10/ C0LMBR2	0000 0000$_2$ [2] XXXX 0000$_2$ [2]	
023B$_{16}$	CAN0 message slot 11 control register / CAN0 local mask register B extended ID1	C0MCTL11/ C0LMBR3	0000 0000$_2$ [2] 0000 0000$_2$ [2]	
023C$_{16}$	CAN0 message slot 12 control register / CAN0 local mask register B extended ID2	C0MCTL12/ C0LMBR4	0000 0000$_2$ [2] XX00 0000$_2$ [2]	
023D$_{16}$	CAN0 message slot 13 control register	C0MCTL13	0000 0000$_2$ [2]	
023E$_{16}$	CAN0 message slot 14 control register	C0MCTL14	0000 0000$_2$ [2]	
023F$_{16}$	CAN0 message slot 15 control register	C0MCTL15	0000 0000$_2$ [2]	
0240$_{16}$	CAN0 slot buffer select register	C0SBS	0000 0000$_2$ [2]	
0241$_{16}$	CAN0 control register 1	C0CTLR1	XX00 00XX$_2$ [2]	
0242$_{16}$	CAN0 sleep control register	C0SLPR	XXXX XXX0$_2$	
0243$_{16}$				
0244$_{16}$ 0245$_{16}$	CAN0 acceptance filter support register	C0AFS	0000 0000$_2$ [2] 0000 0001$_2$ [2]	

X : Nothing is assigned ? : Indetermination

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

Notes :

1. Addresses 0230$_{16}$ to 023F$_{16}$ are switched its function in the BANKSEL bit in the C0CTLR1 register.
2. 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
02C0$_{16}$ 02C1$_{16}$	X0 register Y0 register	X0R,Y0R	??$_{16}$??$_{16}$
02C2$_{16}$ 02C3$_{16}$	X1 register Y1 register	X1R,Y1R	??$_{16}$??$_{16}$
02C4$_{16}$ 02C5$_{16}$	X2 register Y2 register	X2R,Y2R	??$_{16}$??$_{16}$
02C6$_{16}$ 02C7$_{16}$	X3 register Y3 register	X3R,Y3R	??$_{16}$??$_{16}$
02C8$_{16}$ 02C9$_{16}$	X4 register Y4 register	X4R,Y4R	??$_{16}$??$_{16}$
02CA$_{16}$ 02CB$_{16}$	X5 register Y5 register	X5R,Y5R	??$_{16}$??$_{16}$
02CC$_{16}$ 02CD$_{16}$	X6 register Y6 register	X6R,Y6R	??$_{16}$??$_{16}$
02CE$_{16}$ 02CF$_{16}$	X7 register Y7 register	X7R,Y7R	??$_{16}$??$_{16}$
02D0$_{16}$ 02D1$_{16}$	X8 register Y8 register	X8R,Y8R	??$_{16}$??$_{16}$
02D2$_{16}$ 02D3$_{16}$	X9 register Y9 register	X9R,Y9R	??$_{16}$??$_{16}$
02D4$_{16}$ 02D5$_{16}$	X10 register Y10 register	X10R,Y10R	??$_{16}$??$_{16}$
02D6$_{16}$ 02D7$_{16}$	X11 register Y11 register	X11R,Y11R	??$_{16}$??$_{16}$
02D8$_{16}$ 02D9$_{16}$	X12 register Y12 register	X12R,Y12R	??$_{16}$??$_{16}$
02DA$_{16}$ 02DB$_{16}$	X13 register Y13 register	X13R,Y13R	??$_{16}$??$_{16}$
02DC$_{16}$ 02DD$_{16}$	X14 register Y14 register	X14R,Y14R	??$_{16}$??$_{16}$
02DE$_{16}$ 02DF$_{16}$	X15 register Y15 register	X15R,Y15R	??$_{16}$??$_{16}$
02E0$_{16}$	XY control register	XYC	XXXX XX00$_2$
02E1$_{16}$			
02E2$_{16}$			
02E3$_{16}$			
02E4$_{16}$	UART1 special mode register 4	U1SMR4	0000 0000$_2$
02E5$_{16}$	UART1 special mode register 3	U1SMR3	0000 0000$_2$
02E6$_{16}$	UART1 special mode register 2	U1SMR2	0000 0000$_2$
02E7$_{16}$	UART1 special mode register	U1SMR	0000 0000$_2$
02E8$_{16}$	UART1 transmit/receive mode register	U1MR	0000 0000$_2$
02E9$_{16}$	UART1 baud rate register	U1BRG	??$_{16}$
02EA$_{16}$ 02EB$_{16}$	UART1 transmit buffer register	U1TB	???? ????$_2$ XXXX XXX?$_2$
02EC$_{16}$	UART1 transmit/receive control register 0	U1C0	0000 1000$_2$
02ED$_{16}$	UART1 transmit/receive control register 1	U1C1	0000 0010$_2$
02EE$_{16}$ 02EF$_{16}$	UART1 receive buffer register	U1RB	???? ????$_2$???? ?XX?$_2$

X : Nothing is assigned ? : Indetermination

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

Address	Register	Symbol	Value after RESET
$02F0_{16}$			
$02F1_{16}$			
$02F2_{16}$			
$02F3_{16}$			
$02F4_{16}$	UART4 special mode register 4	U4SMR4	$0000\ 0000_2$
$02F5_{16}$	UART4 special mode register 3	U4SMR3	$0000\ 0000_2$
$02F6_{16}$	UART4 special mode register 2	U4SMR2	$0000\ 0000_2$
$02F7_{16}$	UART4 special mode register	U4SMR	$0000\ 0000_2$
$02F8_{16}$	UART4 transmit/receive mode register	U4MR	$0000\ 0000_2$
$02F9_{16}$	UART4 baud rate register	U4BRG	$??_{16}$
$02FA_{16}$ $02FB_{16}$	UART4 transmit buffer register	U4TB	$????\ ????_2$ $XXXX\ XXX?_2$
$02FC_{16}$	UART4 transmit/receive control register 0	U4C0	$0000\ 1000_2$
$02FD_{16}$	UART4 transmit/receive control register 1	U4C1	$0000\ 0010_2$
$02FE_{16}$ $02FF_{16}$	UART4 receive buffer register	U4RB	$????\ ????_2$ $????\ ?XX?_2$
0300_{16}	Timer B3,B4,B5 count start flag	TBSR	$000X\ XXXX_2$
0301_{16}			
0302_{16} 0303_{16}	Timer A1-1 register	TA11	$??_{16}$ $??_{16}$
0304_{16} 0305_{16}	Timer A2-1 register	TA21	$??_{16}$ $??_{16}$
0306_{16} 0307_{16}	Timer A4-1 register	TA41	$??_{16}$ $??_{16}$
0308_{16}	Three-phase PWM control register 0	INVC0	$0000\ 0000_2$
0309_{16}	Three-phase PWM control register 1	INVC1	$0000\ 0000_2$
$030A_{16}$	Three-phase output buffer register 0	IDB0	$XX11\ 1111_2$
$030B_{16}$	Three-phase output buffer register 1	IDB1	$XX11\ 1111_2$
$030C_{16}$	Dead time timer	DTT	$??_{16}$
$030D_{16}$	Timer B2 interrupt generation frequency set counter	ICTB2	$XXXX\ ????_2$
$030E_{16}$			
$030F_{16}$			
0310_{16} 0311_{16}	Timer B3 register	TB3	$??_{16}$ $??_{16}$
0312_{16} 0313_{16}	Timer B4 register	TB4	$??_{16}$ $??_{16}$
0314_{16} 0315_{16}	Timer B5 register	TB5	$??_{16}$ $??_{16}$
0316_{16}			
0317_{16}			
0318_{16}			
0319_{16}			
$031A_{16}$			
$031B_{16}$	Timer B3 mode register	TB3MR	$00??\ 0000_2$
$031C_{16}$	Timer B4 mode register	TB4MR	$00?X\ 0000_2$
$031D_{16}$	Timer B5 mode register	TB5MR	$00?X\ 0000_2$
$031E_{16}$			
$031F_{16}$	External interrupt cause select register	IFSR	$0000\ 0000_2$

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

Address	Register	Symbol	Value after RESET
0320_{16}			
0321_{16}			
0322_{16}			
0323_{16}			
0324_{16}	UART3 special mode register 4	U3SMR4	0000 0000_2
0325_{16}	UART3 special mode register 3	U3SMR3	0000 0000_2
0326_{16}	UART3 special mode register 2	U3SMR2	0000 0000_2
0327_{16}	UART3 special mode register	U3SMR	0000 0000_2
0328_{16}	UART3 transmit/receive mode register	U3MR	0000 0000_2
0329_{16}	UART3 baud rate register	U3BRG	$??_{16}$
$032A_{16}$	UART3 transmit buffer register	U3TB	???? $????_2$
$032B_{16}$			XXXX $XXX?_2$
$032C_{16}$	UART3 transmit/receive control register 0	U3C0	0000 1000_2
$032D_{16}$	UART3 transmit/receive control register 1	U3C1	0000 0010_2
$032E_{16}$	UART3 receive buffer register	U3RB	???? $????_2$
$032F_{16}$			???? $?XX?_2$
0330_{16}			
0331_{16}			
0332_{16}			
0333_{16}			
0334_{16}	UART2 special mode register 4	U2SMR4	0000 0000_2
0335_{16}	UART2 special mode register 3	U2SMR3	0000 0000_2
0336_{16}	UART2 special mode register 2	U2SMR2	0000 0000_2
0337_{16}	UART2 special mode register	U2SMR	0000 0000_2
0338_{16}	UART2 transmit/receive mode register	U2MR	0000 0000_2
0339_{16}	UART2 baud rate register	U2BRG	$??_{16}$
$033A_{16}$	UART2 transmit buffer register	U2TB	???? $????_2$
$033B_{16}$			XXXX $XXX?_2$
$033C_{16}$	UART2 transmit/receive control register 0	U2C0	0000 1000_2
$033D_{16}$	UART2 transmit/receive control register 1	U2C1	0000 0010_2
$033E_{16}$	UART2 receive buffer register	U2RB	???? $????_2$
$033F_{16}$			???? $?XX?_2$
0340_{16}	Count start flag	TABSR	0000 0000_2
0341_{16}	Clock prescaler reset flag	CPSRF	0XXX $XXXX_2$
0342_{16}	One-shot start flag	ONSF	0000 0000_2
0343_{16}	Trigger select register	TRGSR	0000 0000_2
0344_{16}	Up-down flag	UDF	0000 0000_2
0345_{16}			
0346_{16}	Timer A0 register	TA0	$??_{16}$
0347_{16}			$??_{16}$
0348_{16}	Timer A1 register	TA1	$??_{16}$
0349_{16}			$??_{16}$
$034A_{16}$	Timer A2 register	TA2	$??_{16}$
$034B_{16}$			$??_{16}$
$034C_{16}$	Timer A3 register	TA3	$??_{16}$
$034D_{16}$			$??_{16}$
$034E_{16}$	Timer A4 register	TA4	$??_{16}$
$034F_{16}$			$??_{16}$

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

Address	Register	Symbol	Value after RESET
0350$_{16}$ 0351$_{16}$	Timer B0 register	TB0	??$_{16}$??$_{16}$
0352$_{16}$ 0353$_{16}$	Timer B1 register	TB1	??$_{16}$??$_{16}$
0354$_{16}$ 0355$_{16}$	Timer B2 register	TB2	??$_{16}$??$_{16}$
0356$_{16}$	Timer A0 mode register	TA0MR	0000 0X00$_2$
0357$_{16}$	Timer A1 mode register	TA1MR	0000 0X00$_2$
0358$_{16}$	Timer A2 mode register	TA2MR	0000 0X00$_2$
0359$_{16}$	Timer A3 mode register	TA3MR	0000 0X00$_2$
035A$_{16}$	Timer A4 mode register	TA4MR	0000 0X00$_2$
035B$_{16}$	Timer B0 mode register	TB0MR	00?? 0000$_2$
035C$_{16}$	Timer B1 mode register	TB1MR	00?X 0000$_2$
035D$_{16}$	Timer B2 mode register	TB2MR	00?X 0000$_2$
035E$_{16}$	Timer B2 special mode register	TB2SC	XXXX XXX0$_2$
035F$_{16}$	Count source prescaler register	TCSPR	0XXX 0000$_2$
0360$_{16}$			
0361$_{16}$			
0362$_{16}$			
0363$_{16}$			
0364$_{16}$	UART0 special mode register 4	U0SMR4	0000 0000$_2$
0365$_{16}$	UART0 special mode register 3	U0SMR3	0000 0000$_2$
0366$_{16}$	UART0 special mode register 2	U0SMR2	0000 0000$_2$
0367$_{16}$	UART0 special mode register	U0SMR	0000 0000$_2$
0368$_{16}$	UART0 transmit/receive mode register	U0MR	0000 0000$_2$
0369$_{16}$	UART0 baud rate register	U0BRG	??$_{16}$
036A$_{16}$ 036B$_{16}$	UART0 transmit buffer register	U0TB	???? ????$_2$ XXXX XXX?$_2$
036C$_{16}$	UART0 transmit/receive control register 0	U0C0	0000 1000$_2$
036D$_{16}$	UART0 transmit/receive control register 1	U0C1	0000 0010$_2$
036E$_{16}$ 036F$_{16}$	UART0 receive buffer register	U0RB	???? ????$_2$???? ?XX?$_2$
0370$_{16}$			
0371$_{16}$			
0372$_{16}$			
0373$_{16}$			
0374$_{16}$			
0375$_{16}$			
0376$_{16}$	PLL control register 0	PLC0	0011 X100$_2$
0377$_{16}$	PLL control register 1	PLC1	XXXX 0000$_2$
0378$_{16}$	DMA0 cause select register	DM0SL	0X00 0000$_2$
0379$_{16}$	DMA1 cause select register	DM1SL	0X00 0000$_2$
037A$_{16}$	DMA2 cause select register	DM2SL	0X00 0000$_2$
037B$_{16}$	DMA3 cause select register	DM3SL	0X00 0000$_2$
037C$_{16}$ 037D$_{16}$	CRC data register	CRCD	??$_{16}$??$_{16}$
037E$_{16}$	CRC input register	CRCIN	??$_{16}$
037F$_{16}$			

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

Address	Register	Symbol	Value after RESET
0380_{16}	A-D0 register 0	AD00	???? ????$_2$
0381_{16}			XXXX XX??$_2$
0382_{16}	A-D0 register 1	AD01	???? ????$_2$
0383_{16}			XXXX XX??$_2$
0384_{16}	A-D0 register 2	AD02	???? ????$_2$
0385_{16}			XXXX XX??$_2$
0386_{16}	A-D0 register 3	AD03	???? ????$_2$
0387_{16}			XXXX XX??$_2$
0388_{16}	A-D0 register 4	AD04	???? ????$_2$
0389_{16}			XXXX XX??$_2$
$038A_{16}$	A-D0 register 5	AD05	???? ????$_2$
$038B_{16}$			XXXX XX??$_2$
$038C_{16}$	A-D0 register 6	AD06	???? ????$_2$
$038D_{16}$			XXXX XX??$_2$
$038E_{16}$	A-D0 register 7	AD07	???? ????$_2$
$038F_{16}$			XXXX XX??$_2$
0390_{16}			
0391_{16}			
0392_{16}			
0393_{16}			
0394_{16}	A-D0 control register 2	AD0CON2	X000 0000$_2$
0395_{16}			
0396_{16}	A-D0 control register 0	AD0CON0	0000 0000$_2$
0397_{16}	A-D0 control register 1	AD0CON1	0000 0000$_2$
0398_{16}	D-A register 0	DA0	??$_{16}$
0399_{16}			
$039A_{16}$	D-A register 1	DA1	??$_{16}$
$039B_{16}$			
$039C_{16}$	D-A control register	DACON	XXXX XX00$_2$
$039D_{16}$			
$039E_{16}$			
$039F_{16}$			

X : Nothing is assigned ? : Indetermination

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

<144-pin package>

Address	Register	Symbol	Value after RESET
$03A0_{16}$	Function select register A8	PS8	$X000\ 0000_2$
$03A1_{16}$	Function select register A9	PS9	$0000\ 0000_2$
$03A2_{16}$			
$03A3_{16}$			
$03A4_{16}$			
$03A5_{16}$			
$03A6_{16}$			
$03A7_{16}$			
$03A8_{16}$			
$03A9_{16}$			
$03AA_{16}$			
$03AB_{16}$			
$03AC_{16}$			
$03AD_{16}$			
$03AE_{16}$			
$03AF_{16}$	Function select register C	PSC	$00X0\ 0000_2$
$03B0_{16}$	Function select register A0	PS0	$0000\ 0000_2$
$03B1_{16}$	Function select register A1	PS1	$0000\ 0000_2$
$03B2_{16}$	Function select register B0	PSL0	$0000\ 0000_2$
$03B3_{16}$	Function select register B1	PSL1	$0000\ 0000_2$
$03B4_{16}$	Function select register A2	PS2	$00X0\ 0000_2$
$03B5_{16}$	Function select register A3	PS3	$0000\ 0000_2$
$03B6_{16}$	Function select register B2	PSL2	$00X0\ 0000_2$
$03B7_{16}$	Function select register B3	PSL3	$0000\ 0000_2$
$03B8_{16}$			
$03B9_{16}$	Function select register A5	PS5	$XXX0\ 0000_2$
$03BA_{16}$			
$03BB_{16}$			
$03BC_{16}$	Function select register A6	PS6	$0000\ 0000_2$
$03BD_{16}$	Function select register A7	PS7	$0000\ 0000_2$
$03BE_{16}$			
$03BF_{16}$			
$03C0_{16}$	Port P6 register	P6	$??_{16}$
$03C1_{16}$	Port P7 register	P7	$??_{16}$
$03C2_{16}$	Port P6 direction register	PD6	$0000\ 0000_2$
$03C3_{16}$	Port P7 direction register	PD7	$0000\ 0000_2$
$03C4_{16}$	Port P8 register	P8	$??_{16}$
$03C5_{16}$	Port P9 register	P9	$??_{16}$
$03C6_{16}$	Port P8 direction register	PD8	$00X0\ 0000_2$
$03C7_{16}$	Port P9 direction register	PD9	$0000\ 0000_2$
$03C8_{16}$	Port P10 register	P10	$??_{16}$
$03C9_{16}$	Port P11 register	P11	$XXX?\ ????_2$
$03CA_{16}$	Port P10 direction register	PD10	$0000\ 0000_2$
$03CB_{16}$	Port P11 direction register	PD11	$XXX0\ 0000_2$
$03CC_{16}$	Port P12 register	P12	$??_{16}$
$03CD_{16}$	Port P13 register	P13	$??_{16}$
$03CE_{16}$	Port P12 direction register	PD12	$0000\ 0000_2$
$03CF_{16}$	Port P13 direction register	PD13	$0000\ 0000_2$

X : Nothing is assigned ? : Indetermination

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

<144-pin package>

Address	Register	Symbol	Value after RESET
03D0$_{16}$	Port P14 register	P14	X??? ????$_2$
03D1$_{16}$	Port P15 register	P15	??$_{16}$
03D2$_{16}$	Port P14 direction register	PD14	X000 0000$_2$
03D3$_{16}$	Port P15 direction register	PD15	0000 0000$_2$
03D4$_{16}$			
03D5$_{16}$			
03D6$_{16}$			
03D7$_{16}$			
03D8$_{16}$			
03D9$_{16}$			
03DA$_{16}$	Pull-up control register 2	PUR2	0000 0000$_2$
03DB$_{16}$	Pull-up control register 3	PUR3	0000 0000$_2$
03DC$_{16}$	Pull-up control register 4	PUR4	XXXX 0000$_2$
03DD$_{16}$			
03DE$_{16}$			
03DF$_{16}$			
03E0$_{16}$	Port P0 register	P0	??$_{16}$
03E1$_{16}$	Port P1 register	P1	??$_{16}$
03E2$_{16}$	Port P0 direction register	PD0	0000 0000$_2$
03E3$_{16}$	Port P1 direction register	PD1	0000 0000$_2$
03E4$_{16}$	Port P2 register	P2	??$_{16}$
03E5$_{16}$	Port P3 register	P3	??$_{16}$
03E6$_{16}$	Port P2 direction register	PD2	0000 0000$_2$
03E7$_{16}$	Port P3 direction register	PD3	0000 0000$_2$
03E8$_{16}$	Port P4 register	P4	??$_{16}$
03E9$_{16}$	Port P5 register	P5	??$_{16}$
03EA$_{16}$	Port P4 direction register	PD4	0000 0000$_2$
03EB$_{16}$	Port P5 direction register	PD5	0000 0000$_2$
03EC$_{16}$			
03ED$_{16}$			
03EE$_{16}$			
03EF$_{16}$			
03F0$_{16}$	Pull-up control register 0	PUR0	0000 0000$_2$
03F1$_{16}$	Pull-up control register 1	PUR1	XXXX 0000$_2$
03F2$_{16}$			
03F3$_{16}$			
03F4$_{16}$			
03F5$_{16}$			
03F6$_{16}$			
03F7$_{16}$			
03F8$_{16}$			
03F9$_{16}$			
03FA$_{16}$			
03FB$_{16}$			
03FC$_{16}$			
03FD$_{16}$			
03FE$_{16}$			
03FF$_{16}$	Port control register	PCR	XXXX XXX0$_2$

X : Nothing is assigned ? : Indetermination

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

<100-pin package>

Address	Register	Symbol	Value after RESET	
$03A0_{16}$	☐	☐	☐	(Note 2)
$03A1_{16}$	☐	☐	☐	
$03A2_{16}$				
$03A3_{16}$				
$03A4_{16}$				
$03A5_{16}$				
$03A6_{16}$				
$03A7_{16}$				
$03A8_{16}$				
$03A9_{16}$				
$03AA_{16}$				
$03AB_{16}$				
$03AC_{16}$				
$03AD_{16}$				
$03AE_{16}$				
$03AF_{16}$	Function select register C	PSC	$0X00\ 0000_2$	
$03B0_{16}$	Function select register A0	PS0	$0000\ 0000_2$	
$03B1_{16}$	Function select register A1	PS1	$0000\ 0000_2$	
$03B2_{16}$	Function select register B0	PSL0	$0000\ 0000_2$	
$03B3_{16}$	Function select register B1	PSL1	$0000\ 0000_2$	
$03B4_{16}$	Function select register A2	PS2	$00X0\ 0000_2$	
$03B5_{16}$	Function select register A3	PS3	$0000\ 0000_2$	
$03B6_{16}$	Function select register B2	PSL2	$00X0\ 0000_2$	
$03B7_{16}$	Function select register B3	PSL3	$0000\ 0000_2$	
$03B8_{16}$				
$03B9_{16}$	☐	☐	☐	(Note 2)
$03BA_{16}$				
$03BB_{16}$				
$03BC_{16}$	☐	☐	☐	(Note 2)
$03BD_{16}$	☐	☐	☐	
$03BE_{16}$				
$03BF_{16}$				
$03C0_{16}$	Port P6 register	P6	$??_{16}$	
$03C1_{16}$	Port P7 register	P7	$??_{16}$	
$03C2_{16}$	Port P6 direction register	PD6	$0000\ 0000_2$	
$03C3_{16}$	Port P7 direction register	PD7	$0000\ 0000_2$	
$03C4_{16}$	Port P8 register	P8	$??_{16}$	
$03C5_{16}$	Port P9 register	P9	$??_{16}$	
$03C6_{16}$	Port P8 direction register	PD8	$00X0\ 0000_2$	
$03C7_{16}$	Port P9 direction register	PD9	$0000\ 0000_2$	
$03C8_{16}$	Port P10 register	P10	$??_{16}$	
$03C9_{16}$	☐	☐	☐	(Note 2)
$03CA_{16}$	Port P10 direction register	PD10	$0000\ 0000_2$	
$03CB_{16}$	▨	▨	▨	(Note 1)
$03CC_{16}$	☐	☐	☐	(Note 2)
$03CD_{16}$	☐	☐	☐	
$03CE_{16}$	▨	▨	▨	(Note 1)
$03CF_{16}$	▨	▨	▨	

X : Nothing is assigned ? : Indetermination

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

Notes :

1. ▨ Address space $03CB_{16}$, $03CE_{16}$ and $03CF_{16}$ should be set to "FF_{16}" in the 100-pin package.
2. ☐ No address space $03A0_{16}$, $03A1_{16}$, $03B9_{16}$, $03BD_{16}$, $03C9_{16}$, $03CC_{16}$ and $03CD_{16}$ is provided in the 100-pin package.

<100-pin package>

Address	Register	Symbol	Value after RESET	
03D0$_{16}$	□	□	□	(Note 3)
03D1$_{16}$	□	□	□	
03D2$_{16}$	▨	▨	▨	(Note 1)
03D3$_{16}$	▨	▨	▨	
03D4$_{16}$				
03D5$_{16}$				
03D6$_{16}$				
03D7$_{16}$				
03D8$_{16}$				
03D9$_{16}$				
03DA$_{16}$	Pull-up control register 2	PUR2	0000 0000$_2$	
03DB$_{16}$	Pull-up control register 3	PUR3	0000 0000$_2$	
03DC$_{16}$	▨	▨	▨	(Note 2)
03DD$_{16}$				
03DE$_{16}$				
03DF$_{16}$				
03E0$_{16}$	Port P0 register	P0	??$_{16}$	
03E1$_{16}$	Port P1 register	P1	??$_{16}$	
03E2$_{16}$	Port P0 direction register	PD0	0000 0000$_2$	
03E3$_{16}$	Port P1 direction register	PD1	0000 0000$_2$	
03E4$_{16}$	Port P2 register	P2	??$_{16}$	
03E5$_{16}$	Port P3 register	P3	??$_{16}$	
03E6$_{16}$	Port P2 direction register	PD2	0000 0000$_2$	
03E7$_{16}$	Port P3 direction register	PD3	0000 0000$_2$	
03E8$_{16}$	Port P4 register	P4	??$_{16}$	
03E9$_{16}$	Port P5 register	P5	??$_{16}$	
03EA$_{16}$	Port P4 direction register	PD4	0000 0000$_2$	
03EB$_{16}$	Port P5 direction register	PD5	0000 0000$_2$	
03EC$_{16}$				
03ED$_{16}$				
03EE$_{16}$				
03EF$_{16}$				
03F0$_{16}$	Pull-up control register 0	PUR0	0000 0000$_2$	
03F1$_{16}$	Pull-up control register 1	PUR1	XXXX 0000$_2$	
03F2$_{16}$				
03F3$_{16}$				
03F4$_{16}$				
03F5$_{16}$				
03F6$_{16}$				
03F7$_{16}$				
03F8$_{16}$				
03F9$_{16}$				
03FA$_{16}$				
03FB$_{16}$				
03FC$_{16}$				
03FD$_{16}$				
03FE$_{16}$				
03FF$_{16}$	Port control register	PCR	XXXX XXX0$_2$	

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

Notes :

1. ▨ Address space 03D2$_{16}$ and 03D3$_{16}$ should be set to "FF$_{16}$" in the 100-pin package.
2. ▨ Address space 03DC$_{16}$ should be set to "00$_{16}$" in the 100-pin package.
3. □ No address space 03D0$_{16}$ and 03D1$_{16}$ 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 CNV$_{SS}$ pin and the PM01 to PM00 bits in the PM0 register are combined to set a processor mode. Avoid setting "01$_2$" 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 CNV$_{SS}$ 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.

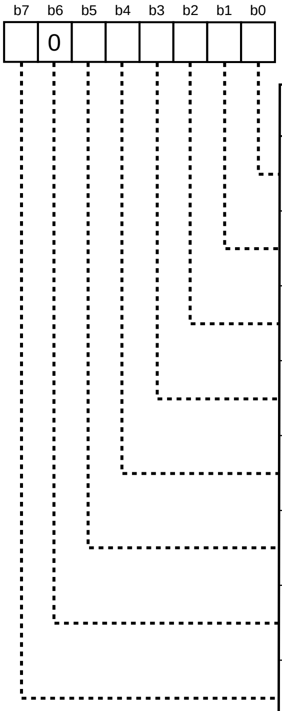
1. Applying V$_{SS}$ to CNV$_{SS}$ Pin

The microcomputer enters single-chip mode after reset. The PM01 to PM00 bits should be set to "01$_2$" (memory expansion mode) to switch to memory expansion mode after an operation start.

2. Applying V$_{CC}$ to CNV$_{SS}$ Pin

The microcomputer enters microprocessor mode after reset.

Processor mode register 0[1]



Symbol
PM0

Address
0004_{16}

When reset
$1000\ 0000_2$ (CNVss = "L")
$0000\ 0011_2$ (CNVss = "H")

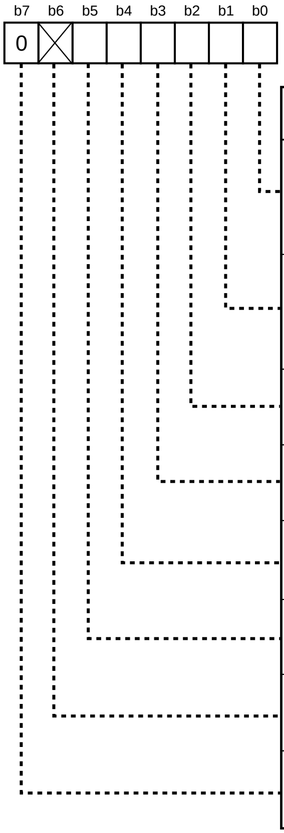
Bit symbol	Bit name	Function	RW
PM00	Processor mode bit[2, 3]	b1 b0 0 0: Single-chip mode 0 1: Memory expansion mode 1 0: Avoid this setting 1 1: Microprocessor mode	RW
PM01			RW
PM02	R/W mode select bit[4]	0: $\overline{RD}$ / $\overline{BHE}$ / $\overline{WR}$ 1: $\overline{RD}$ / $\overline{WRH}$ / $\overline{WRL}$	RW
PM03	Software reset bit	The microcomputer is reset when this bit is set to "1". When read, its content is indeterminate.	RW
PM04	Multiplex bus space select bit[5]	b5 b4 0 0 : Multiplex bus is not used 0 1 : Allocated to $\overline{CS2}$ space 0 1 : Allocated to $\overline{CS1}$ space 1 1 : Allocated to entire $\overline{CS}$ space[6]	RW
PM05			RW
——	Reserved bit	Should set to "0"	RW
PM07	BCLK output disable bit[7]	0 : BCLK is output[8] 1 : BCLK is not output The CM01 and CM00 bits in the CM0 register determine functions.	RW

Notes :

1. The PM0 registers should be set after the PRC1 bit in the PRCR register is set to "1".
2. Processor mode is not exited even if setting the PM03 bit to "1" .
3. Avoid setting the PM01 to PM00 bits and other bits simultaneously when setting the PM01 to PM00 bits to "01_2" or "11_2". Another bits should be set first to rewrite before setting the PM01 to PM00 bits.
4. When using 16-bit data bus in DRAMC controler, this bit should be set to "1".
5. This bit is available in microprocessor and memory expansion modes.
   The PM05 to PM04 bits should be set to "00_2" in mode 0.
   Avoid setting the PM05 to PM04 bits to "01_2" in mode 2.
6. The PM05 to PM04 bits cannot be set to "11_2" in microprocessor mode since a separate bus is performed after reset.
   When setting the PM05 to PM04 bits to "11_2" in memory expansion mode, space to be accessed is 64K bytes per chip-select. $\overline{CS0}$ to $\overline{CS2}$ are selected in mode 1, $\overline{CS0}$ and $\overline{CS1}$ in mode 2 and $\overline{CS0}$ to $\overline{CS3}$ in mode 3.
7. No BCLK is output in single-chip mode even if the PM07 bit is set to "0". When halting a clock output in microprocessor or memory expansion mode, the PM07 bit should be set to "1" and the CM01 to CM00 bits in the CM0 register be set to "00_2" (I/O port $P5_3$). "L" is output from $P5_3$.
8. When setting the PM07 bit to "0" (BCLK output), the CM01 and CM00 bits should be set to "00_2".

Figure 1.6.1. PM0 Register

Processor mode register 1[1]



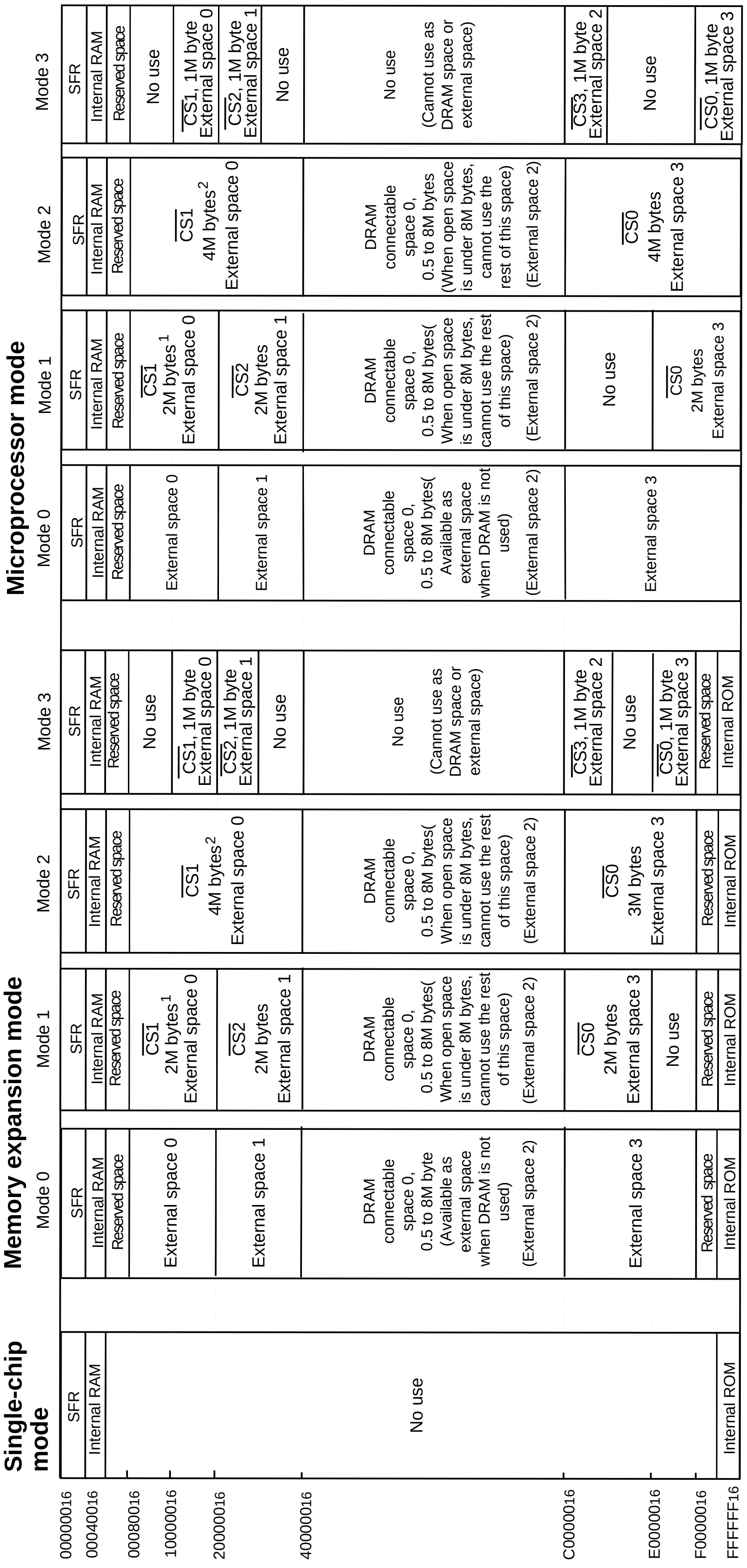
Symbol	Address	When reset
PM1	0005_{16}	$0X00\ 0000_2$

Bit symbol	Bit name	Function	RW
PM10	External memory space mode bit[2]	b1 b0 0 0 : Mode 0 (A_{20} to $\overline{A_{23}}$ for $P4_4$ to $P4_7$) 0 1 : Mode 1 (A_{20} for $P4_4$, $\overline{CS2}$ to $\overline{CS0}$ for $P4_5$ to $P4_7$) 1 0 : Mode 2 (A_{20}, A_{21} for $P4_4$, $P4_5$, $\overline{CS1}$, $\overline{CS0}$ for $P4_6$, $P4_7$) 1 1 : Mode 3[3] ($\overline{CS3}$ to $\overline{CS0}$ for $P4_4$ to $P4_7$)	RW
PM11			RW
PM12	Internal memory wait bit	0 : No wait 1 : Wait	RW
PM13	SFR space wait bit 0	0 : One wait 1 : Two waits[4]	RW
PM14	ALE pin select bit[2]	b5 b4 0 0 : No ALE 0 1 : $P5_3$/BCLK[5] 1 0 : $P5_6$/$\overline{RAS}$ 1 1 : $P5_4$/$\overline{HLDA}$	RW
PM15			RW
——	Nothing is assigned. When read, its content is indeterminate.		–
——	Reserved bit	Should set to "0"	RW

Notes :

1. The register should be set after the PRC1 bit in the PRCR register is set to "1".
2. This bit is available in memory expansion mode or in microprocessor mode.
3. When setting the PM11 and PM10 bits to "11_2" (mode 3), DRAMC cannot be used.
4. When accessing CAN-associated registers (addresses $01E0_{16}$ to 0245_{16}), the PM13 bit is set to "1" (Two waits).
5. When setting the PM15 and PM14 bits to "01_2" ($P5_3$/BCLK select), the CM01 and CM00 bits in the CM0

Figure 1.6.2. PM1 Register


Single-chip mode
Memory expansion mode
Microprocessor mode
Mode 0
Mode 1
Mode 2
Mode 3
SFR
Internal RAM
Reserved space
No use
External space 0
External space 1
External space 2
External space 3
$\overline{CS1}$ 2M bytes[1]
$\overline{CS2}$ 2M bytes
$\overline{CS1}$ 4M bytes[2]
$\overline{CS1}$, 1M byte
$\overline{CS2}$, 1M byte
$\overline{CS3}$, 1M byte
$\overline{CS0}$, 1M byte
$\overline{CS0}$ 2M bytes
$\overline{CS0}$ 3M bytes
$\overline{CS0}$ 4M bytes
DRAM connectable space 0, 0.5 to 8M byte (Available as external space when DRAM is not used)
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)
Internal ROM
000000_{16}
000400_{16}
008000_{16}
010000_{16}
020000_{16}
040000_{16}
$C00000_{16}$
$E00000_{16}$
$F00000_{16}$
$FFFFFF_{16}$

Each $\overline{CS0}$ to $\overline{CS3}$ can be set for 0 to 3 wait(s) in the WCR register.

Notes :
1. 200000_{16}–008000_{16}=2016K bytes. 32K bytes less than 2M bytes.
2. 400000_{16}–008000_{16}=4064K bytes. 32K bytes less than 4M bytes.

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. A_0 to A_{22}, $\overline{A_{23}}$, D_0 to D_{15}, MA_0 to MA_{12}, CS_0 to CS_3, $\overline{WR}/\overline{WR}/\overline{CASL}$, $\overline{WRH}/\overline{BHE}/\overline{CASH}$, $\overline{RD}/\overline{DW}$, BCLK/ALE, $\overline{HLDA}$/ALE, $\overline{HOLD}$, ALE/$\overline{RAS}$, $\overline{RDY}$ are included as bus control pins.

Bus Settings

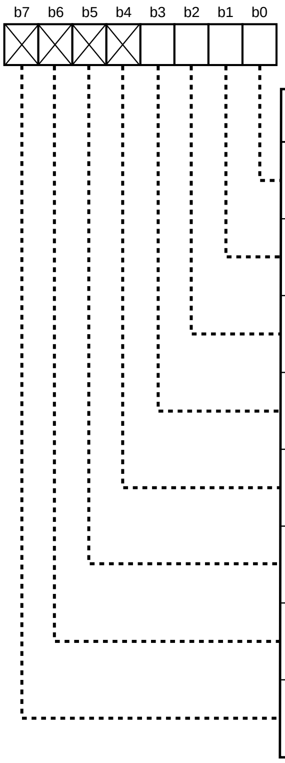
The BYTE pin, the DS register, the PM05 to PM04 bits in the PM0 register and the PM11 to PM10 bits in the PM1 register determine bus settings.

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

External data bus width control register



Symbol: DS
Address: $000B_{16}$
When reset: $XXXX\ 1000_2$ (BYTE pin = "L")
$XXXX\ 0000_2$ (BYTE pin = "H")

Bit symbol	Bit name	Function	RW
DS0	External space 0 data bus width select bit	0 : 8 bits wide 1 : 16 bits wide	RW
DS1	External space 1 data bus width select bit	0 : 8 bits wide 1 : 16 bits wide	RW
DS2	External space 2 data bus width select bit	0 : 8 bits wide 1 : 16 bits wide	RW
DS3	External space 3 data bus width select bit [1]	0 : 8 bits wide 1 : 16 bits wide	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
——			—
——			—
——			—

Notes :
1. After reset, this bit is set to "1" when the BYTE pin is set to "L". This bit is set to "0" when the BYTE pin is set to "H".

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 DSi bits in the DS register (i=0 to 3) are set to "0" (8-bit data bus), port P_0 becomes a data bus and port P_1 becomes a programmable I/O port. When setting one of the DSi bits to "1" (16-bit data bus), ports P0 and P1 become the data bus. When setting the DSi bits to "0", port P1 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_0 to D_7 are multiplexed with A_0 to A_7 in 8-bit space selected by the DSi bit. D_0 to D_{15} are multiplexed with A_0 to A_{15} in16-bit space selected by the DSi 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 "11_2" (allocated to entire $\overline{CS}$ space), only 16 bits from A_0 to A_{15} are output as an address.
The PM05 to PM04 bits cannot set to "11_2" 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	
PM05 to PM04 bits in PM0 register		"01_2", "10_2" ($\overline{CS1}$ or $\overline{CS2}$ as multiplex bus. Another as separate bus)		"00_2" (Separate bus)		"11_2" [1] (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
$P0_0$ to $P0_7$	I/O port	Data bus D_0 to D_7	Data bus D_0 to D_7	Data bus D_0 to D_7	Data bus D_0 to D_7	I/O port	I/O port
$P1_0$ to $P1_7$	I/O port	I/O port	Data bus D_8 to D_{15}	I/O port	Data bus D_8 to D_{15}	I/O port	I/O port
$P2_0$ to $P2_7$	I/O port	Address bus data bus [2] A_0/D_0 to A_7/D_7	Address bus data bus [2] A_0/D_0 to A_7/D_7	Address bus A_0/D_0 to A_7/D_7	Address bus A_0/D_0 to A_7/D_7	Address bus data bus A_0/D_0 to A_7/D_7	Address bus data bus A_0/D_0 to A_7/D_7
$P3_0$ to $P3_7$	I/O port	Address bus A_8 to A_{15}	Address bus/ data bus [2] A_8/D_8 to A_{15}/D_{15}	Address bus A_8 to A_{15}	Address bus A_8 to A_{15}	Address bus A_8 to A_{15}	Address bus/ data bus A_8/D_8 to A_{15}/D_{15}
$P4_0$ to $P4_3$	I/O port	Address bus A_{16} to A_{19}	Address bus A_{16} to A_{19}	Address bus A_{16} to A_{19}	Address bus A_{16} to A_{19}	I/O port	I/O port
$P4_4$ to $P4_6$	I/O port	$\overline{CS}$ (chip-select) or address bus (A_{23}) (Refet to the paragraph "Bus control" for details) [5]					
$P4_7$	I/O port	$\overline{CS}$ (chip-select) or address bus (A_{23}) (Refer to the paragraph "Bus control" for details) [5]					
$P5_0$ to $P5_3$	I/O port	$\overline{RD}$, $\overline{WRL}$, $\overline{WRH}$ and BCLK output or $\overline{RD}$, $\overline{BHE}$, $\overline{WR}$ and BCLK output (Refer to the paragraph "Bus control" for details) [3,4]					
$P5_4$	I/O port	$\overline{HDLA}$ [3]	$\overline{HDLA}$ [3]	$\overline{HDLA}$ [3]	$\overline{HDLA}$ [3]	$\overline{HDLA}$ [3]	$\overline{HDLA}$ [3]
$P5_5$	I/O port	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$	$\overline{HOLD}$
$P5_6$	I/O port	$\overline{RAS}$ [3]	$\overline{RAS}$ [3]	$\overline{RAS}$ [3]	$\overline{RAS}$ [3]	$\overline{RAS}$ [3]	$\overline{RAS}$ [3]
$P5_7$	I/O port	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$	$\overline{RDY}$

Notes :

1. Avoid setting the PM05 to PM04 bits to "11_2" (all $\overline{CS}$ space as multiplex bus) in microprocessor mode because running a separate bus after reset.
   When selecting "11_2" in memory expansion mode, address bus accesses with 64K bytes per chip-select.
2. Address bus is selected in seprarete bus configuration.
3. 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.
4. When selecting the DRAMC and accessing the DRAM space, $\overline{CASL}$, $\overline{CASH}$, $\overline{DW}$ and BCLK output occurs.
5. 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 A_0 to A_{22} and $\overline{A_{23}}$. $\overline{A_{23}}$ 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 D_0 to D_7 or 16-bit data bus as D_0 to D_{15} 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 A_8 to A_{20}.

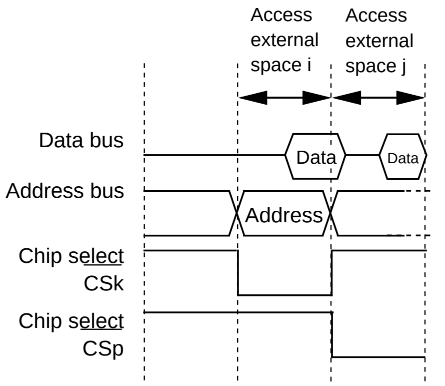
(2) Chip-Select Signal

The chip-select signal is shared with A_0 to A_{22} and $\overline{A_{23}}$. 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 $\overline{A_{23}}$ as chip-select signal. "L" is output while CSi (i=0 to 3) accesses the corresponding external space. "H" is output when CSi accesses internal space and another external memory space. Figure 1.7.2 shows an example of the address bus and chip-select signal outputs.

Example 1:

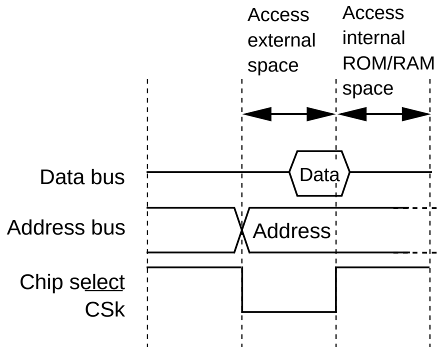
When accessing external space j specified by another chip-select signal in the next cycle after having accessed external space i, both address bus and chip-select signal change.



i = 0 to 3
j = 0 to 3, excluding i
k = 0 to 3
p= 0 to 3, excluding k
(See Figure 1.6.3 about i, j and p, k)

Example 2:

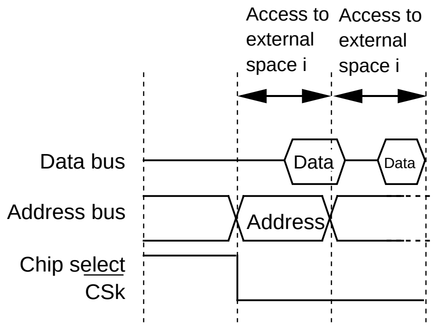
When accessing internal ROM/RAM space in the next cycle after having accessed external space, the chip-select signal changes but the address bus does not.



k = 0 to 3

Example 3:

When accessing the space i specified by the same chip-select signal in the next cycle after having accessed external space i, address bus changes but the chip-select signal does not.

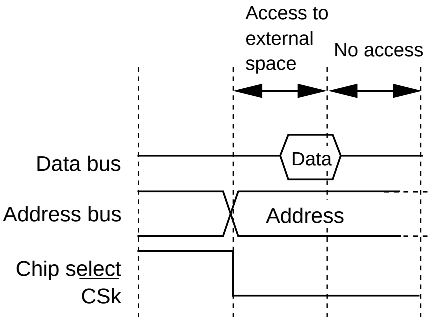


i = 0 to 3
k = 0 to 3

(See Figure 1.6.3 about i and k)

Example 4:

When CPU does not access any space in the next cycle after having accessed external space (no pre-fetch of an instruction is generated), niether address bus nor the chip select signal change .



k = 0 to 3

Notes :

1. The above applies to address bus and chip-select signal in two consecutive cycles. By combining these examples, chip-select signal may output two or more cycles.

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[1]	Not used	Write 1-byte data
	L	H[1]	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 $01E0_{16}$ to 0245_{16}), the PM13 bit should be set to "1".

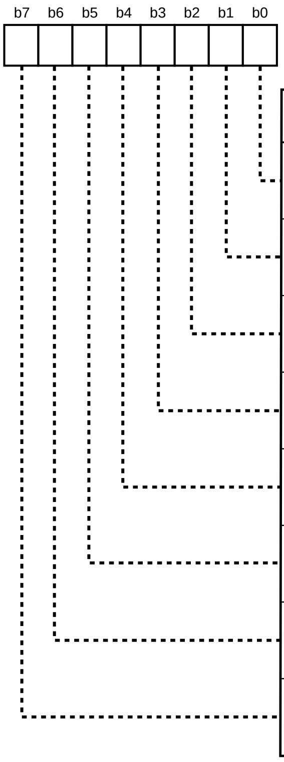
Bus cycle for an external space is basically one BCLK cycle. The WCR register inserts wait(s), equivalent to one to three BCLK cycles, into an external space. Bus cycles are two BCLK cycles if selecting one wait. Bus cycles are four BCLK cycles if selecting three waits.

If applicable to the followings, bus cycles vary from the ones selected by the WCR register. Figure1.7.5 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.

Wait control register [1,2]



Symbol	Address	When reset
WCR	0008_{16}	$1111\ 1111_2$

Bit symbol	Bit name	Function	RW
WCR0	External space 0 wait bit	b1 b0 0 0: No wait 0 1: 1 wait 1 0: 2 waits 1 1: 3 waits	RW
WCR1			RW
WCR2	External space 1 wait bit	b3 b2 0 0: No wait 0 1: 1 wait 1 0: 2 waits 1 1: 3 waits	RW
WCR3			RW
WCR4	External space 2 wait bit	b5 b4 0 0: No wait 0 1: 1 wait 1 0: 2 waits 1 1: 3 waits	RW
WCR5			RW
WCR6	External space 3 wait bit	b7 b6 0 0: No wait 0 1: 1 wait 1 0: 2 waits 1 1: 3 waits	RW
WCR7			RW

Notes :

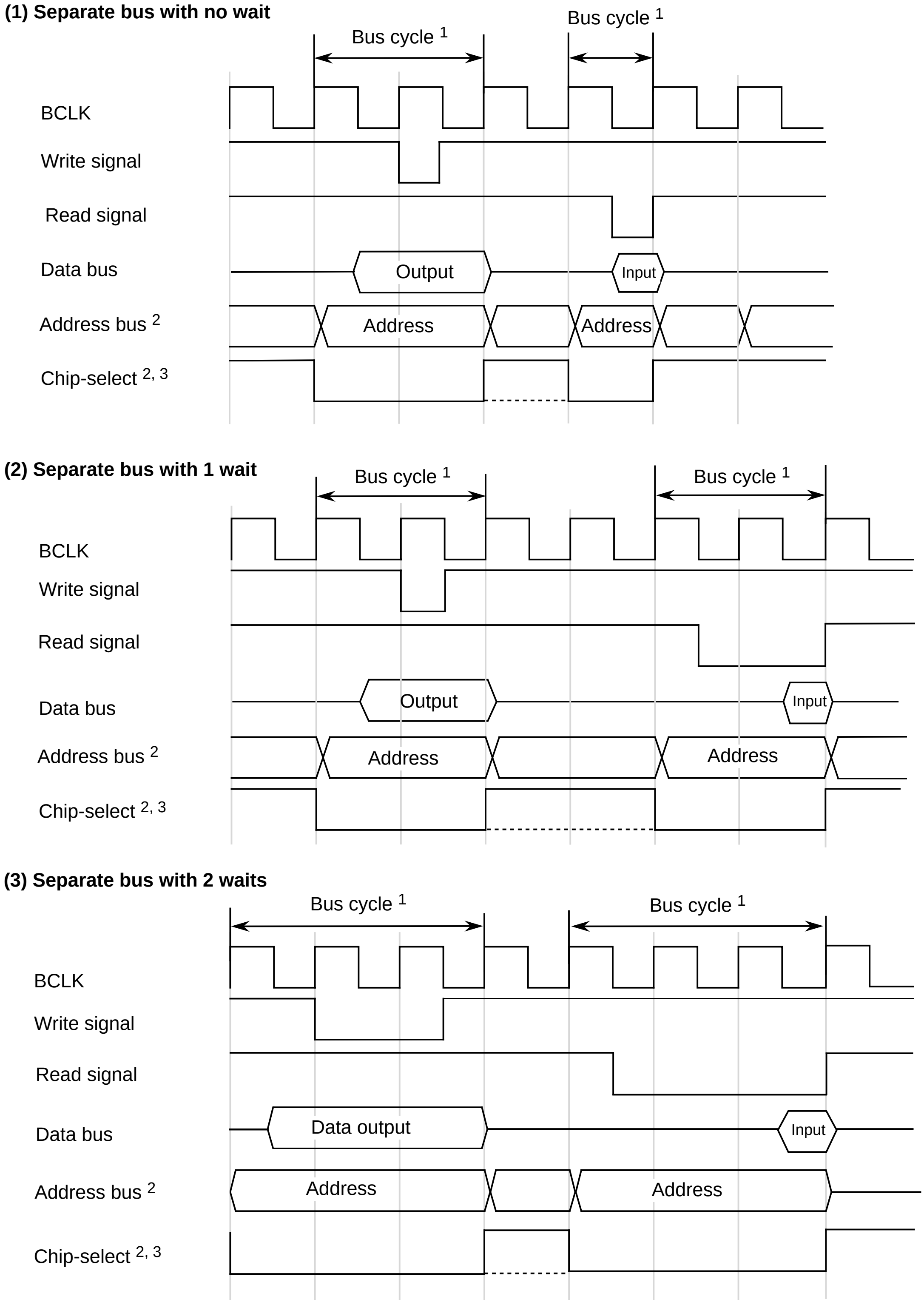
1. With a multiplex bus, the WCR register must be two waits even if setting it to "00_2" (no wait) or "01_2" (one wait).
2. With a separate bus, the read bus cycle is one BCLK cycle and the write bus cycle is two BCLK cycles (one wait) when setting the WCR register to "00_2".

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	—	—	00_2	Read :1 BCLK cycle Write : 2 BCLK cycles
				01_2	2 BCLK cycles
				10_2	3 BCLK cycles
				11_2	4 BCLK cycles
	Multiplex bus	—	—	00_2	3 BCLK cycle
				01_2	3 BCLK cycles
				10_2	3 BCLK cycles
				11_2	4 BCLK cycles

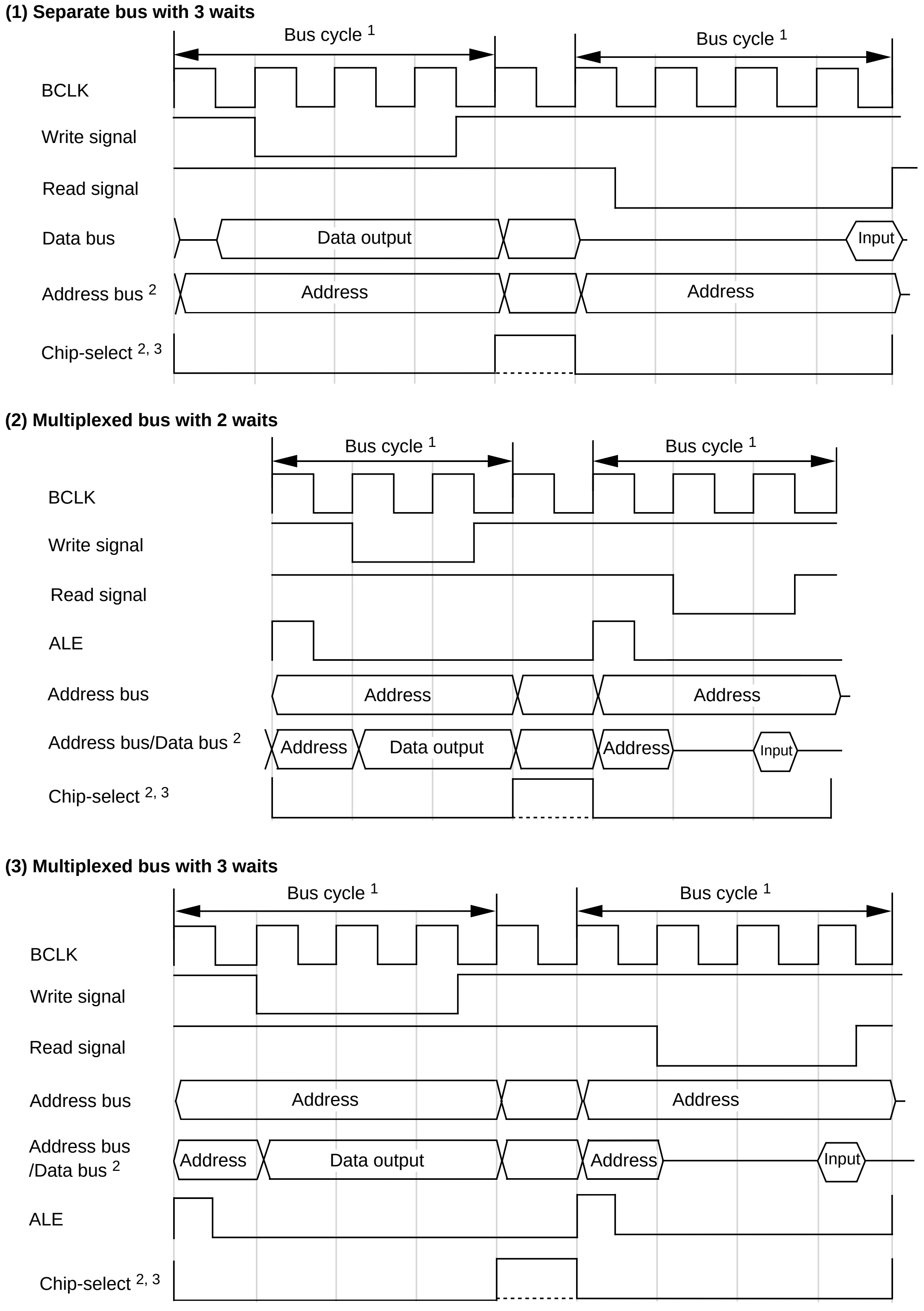
i = 0, 2, 4, 6   j = i + 1



Notes :

1. This timing illustrates bus cycle length. Read cycle and write cycle may occur consecutively.
2. Address bus and chip-select signal may be output longer depending on CPU status such as an instruction queue buffer.
3. When accessing the same external space (same $\overline{CS}$ space) consecutively, the chip-select may be output consecutively.

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

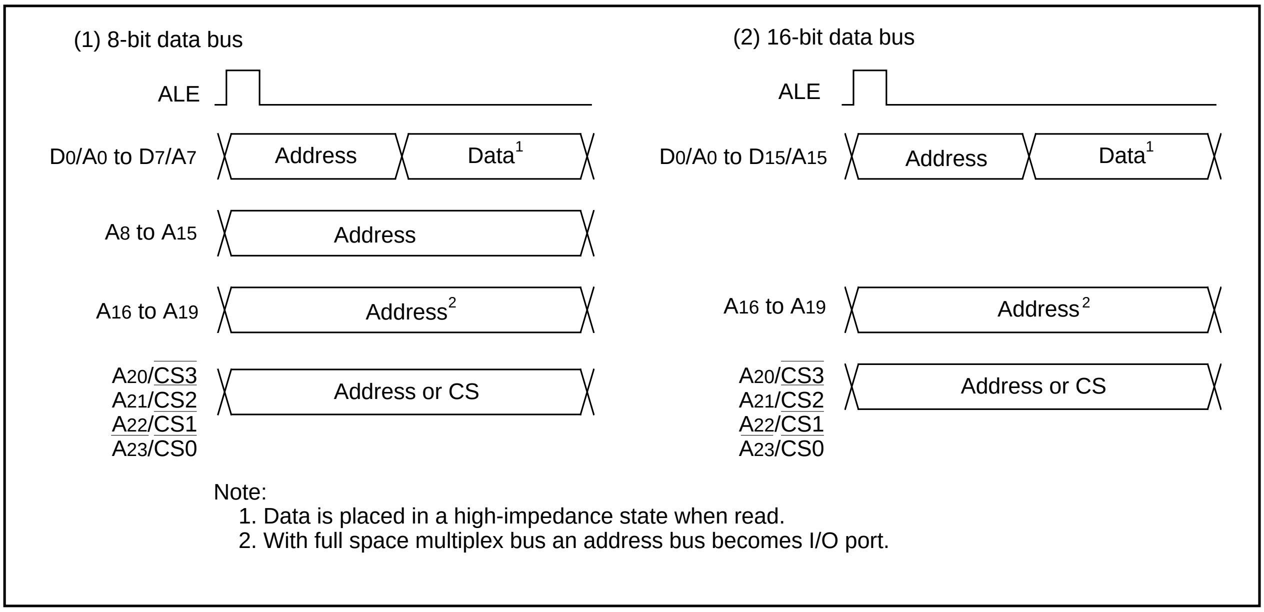


Notes :
1. This timing illustrates bus cycle length. Read cycle and write cycle may be occur consecutively.
2. Address bus and chip-select signal may be output longer depending on CPU status such as an instruction queue buffer.
3. When accessing the same external space (same $\overline{CS}$ space) consecutively, the chip select may be output consecutively.

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.



Note:
1. Data is placed in a high-impedance state when read.
2. With full space multiplex bus an address bus becomes I/O port.

Figure 1.7.6. ALE Signal and Address/data Bus

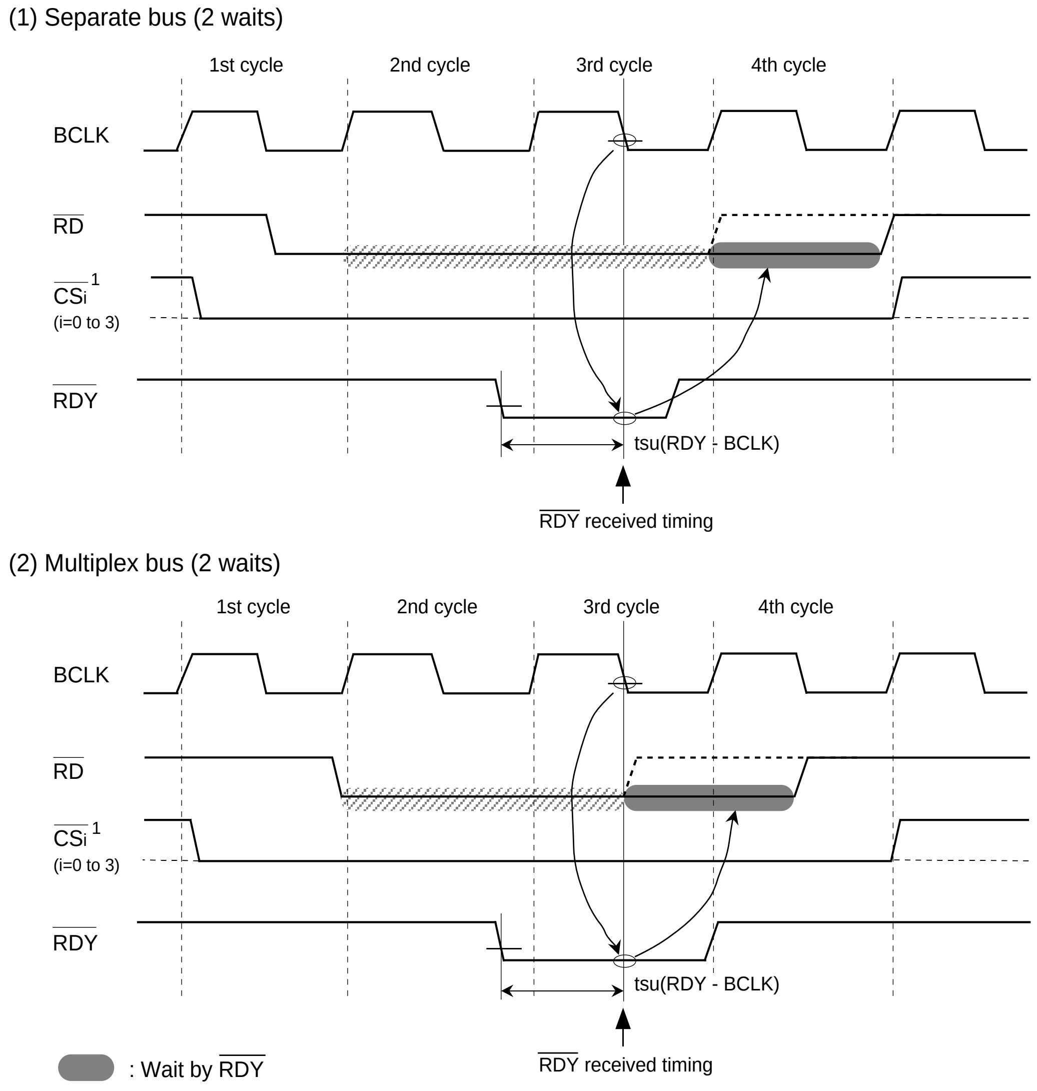
(6) $\overline{\text{RDY}}$ Signal

The $\overline{\text{RDY}}$ signal facilitates an access to external devices which need a longer access time. When setting the $\overline{\text{RDY}}$ 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 $\overline{\text{RDY}}$ 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 $\overline{\text{RDY}}$ signal inserts a wait into the bus cycle. Figure 1.7.7 shows an example of the $\overline{\text{RD}}$ signal that is output longer by the $\overline{\text{RDY}}$ signal.

Table 1.7.6. Microcomputer Status in a Wait State[1]

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

Note:
1. The $\overline{\text{RDY}}$ signal cannot be received immediately before a software wait.



▬ : Wait by $\overline{\text{RDY}}$

▨ : Wait by program

tsu(RDY-BCLK): $\overline{\text{RDY}}$ input setup time
$\overline{\text{RDY}}$ receive timing for i wait(s): j + 1st cycles (j = 1 to 3)

Note:
1. Chip-select ($\overline{\text{CSi}}$) may be output ,longer depending on CPU status such as an instruction queue buffer.

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

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

The $\overline{\text{HOLD}}$ signal transfers a bus privileges from the CPU to external circuits. When setting the $\overline{\text{HOLD}}$ 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 $\overline{\text{HOLD}}$ pin is set to "L". The $\overline{\text{HLDA}}$ pin outputs "L". Table 1.7.7 shows a microcomputer state in a hold state.

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

$\overline{\text{HOLD}}$ > 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 $\overline{HOLD}$ 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 "00$_2$", the CPU clock signal is output from P5$_3$ 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

Clock Generation Circuit

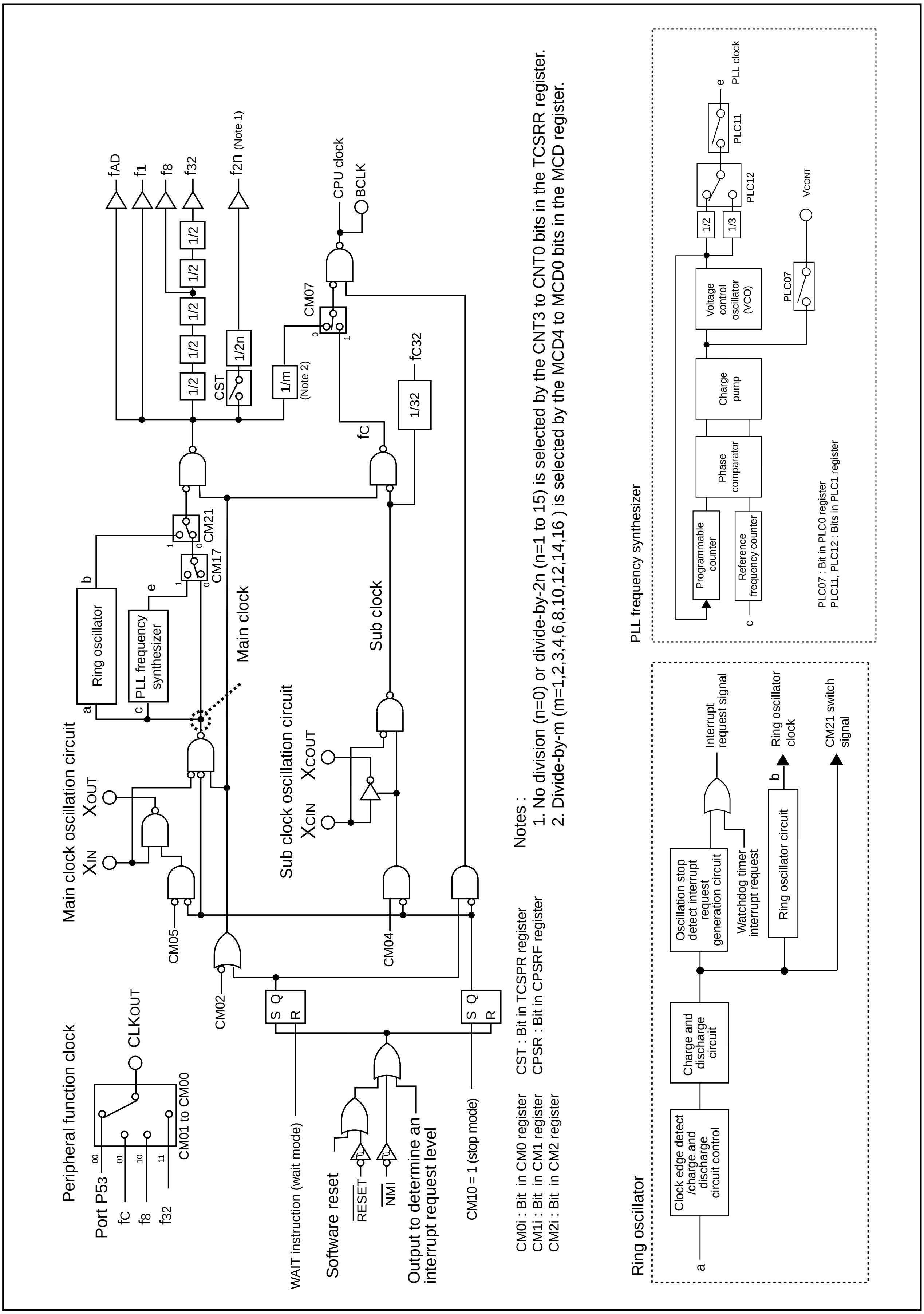
Four circuits are incorporated to generate the system clock signal :

- 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	• CPU clock source • Peripheral function clock source	• CPU clock source • Timer A and B clock source	• CPU clock source • Peripheral function clock source	• CPU clock source • Peripheral function clock source
Clock frequency	Up to 30 MHz	32.768 kHz	Approximatly 1 MHz	20 MHz, 30 MHz
Connectable or additional circuit oscillator	• Ceramic resonator • Crystal oscillator	• Crystal oscillator	———	• Low pass filter
Pins for oscillator or for additional circuit	X_{IN}, X_{OUT}	X_{CIN}, X_{COUT}	———	V_{COUT} (connect to Low pass filter) $P8_6$ (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.


Main clock oscillation circuit
XIN
XOUT
CM05
Ring oscillator
PLL frequency synthesizer
a
b
c
e
0
1
CM17
CM21
Main clock
Sub clock oscillation circuit
XCIN
XCOUT
CM04
Sub clock
fC
1/32
fC32
1/2
1/2
1/2
1/2
1/2
fAD
f1
f8
f32
CST
1/2n
f2n (Note 1)
1/m
(Note 2)
CM07
CPU clock
BCLK
Peripheral function clock
Port P53
fC
f8
f32
00
01
10
11
CM01 to CM00
CLKOUT
CM02
S Q
R
S Q
R
WAIT instruction (wait mode)
Software reset
RESET
NMI
Output to determine an interrupt request level
CM10 = 1 (stop mode)
PLL frequency synthesizer
Programmable counter
Reference frequency counter
c
Phase comparator
Charge pump
Voltage control oscillator (VCO)
PLC07
VCONT
1/2
1/3
PLC12
PLC11
e
PLL clock
Ring oscillator
a
Clock edge detect /charge and discharge circuit control
Charge and discharge circuit
Oscillation stop detect interrupt request generation circuit
Watchdog timer interrupt request
Interrupt request signal
Ring oscillator circuit
b
Ring oscillator clock
CM21 switch signal

Notes :
1. No division (n=0) or divide-by-2n (n=1 to 15) is selected by the CNT3 to CNT0 bits in the TCSRR register.
2. Divide-by-m (m=1,2,3,4,6,8,10,12,14,16) is selected by the MCD4 to MCD0 bits in the MCD register.

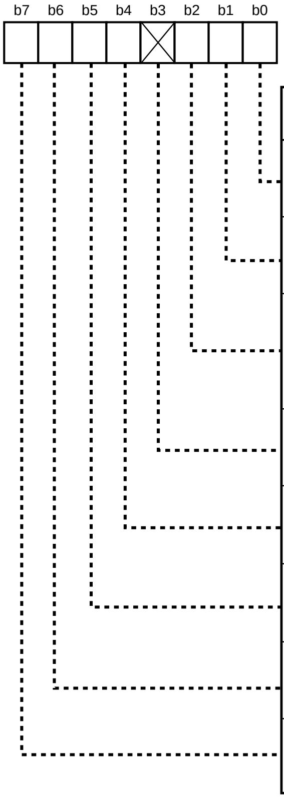
CM0i : Bit in CM0 register
CM1i : Bit in CM1 register
CM2i : Bit in CM2 register

CST : Bit in TCSPR register
CPSR : Bit in CPSRF register

PLC07 : Bit in PLC0 register
PLC11, PLC12 : Bits in PLC1 register

Figure 1.8.1. Clock Generation Circuit

System clock control register 0[1]



Symbol	Address	When reset
CM0	0006_{16}	$0000\ X000_2$

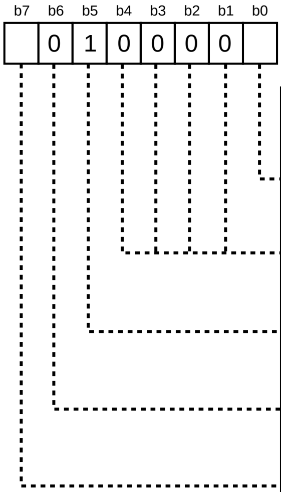
Bit symbol	Bit name	Function	RW
CM00	Clock output function select bit[2]	b1 b0 0 0 : I/O port $P5_3$ 0 1 : f_C output 1 0 : f_8 output 1 1 : f_{32} output	RW
CM01			RW
CM02	In WAIT peripheral function clock stop bit	0 : Peripheral clock does not stop in wait mode 1 : Peripheral clock stops in wait mode[3]	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
CM04	Port X_C switch bit	0 : I/O port function 1 : X_{CIN}-X_{COUT} oscilation function[4]	RW
CM05	Main clock (X_{IN}-X_{OUT}) stop bit[5]	0 : Main clock oscillation 1 : Main clock stop[6]	RW
CM06	Watchdog timer function select bit	0 : Watchdog timer interrupt 1 : Reset[7]	RW
CM07	System clock select bit[8]	0 : Select X_{IN} - X_{OUT} 1 : Select X_{CIN} - X_{COUT}	RW

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 "00_2". When the PM15 to PM14 bits in the PM1 register is set to "01_2" (ALE output to $P5_3$), the CM01 to CM00 bits should be set to "00_2". 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 "00_2" to output "L" from port $P5_3$ (port $P5_3$ does not function as an I/O port).
3. f_{C32} 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" (X_{CIN}-X_{COUT} oscillation), the PD8_7 to PD8_6 bits should be set to "00_2" (with port $P8_7$ and $P8_6$ 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), X_{OUT} is set to "H". Also, an internal feedback resistance remains ON. X_{IN} is pulled up to X_{OUT} ("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 "01000_2" (divide-by-8 mode). In ring oscillation mode, the MCD register is not divided by eight even if X_{IN}-X_{out} 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 1.8.2. CM0 register

System clock control register 1[1]



Symbol	Address	When reset
CM1	0007_{16}	$0010\ 0000_2$

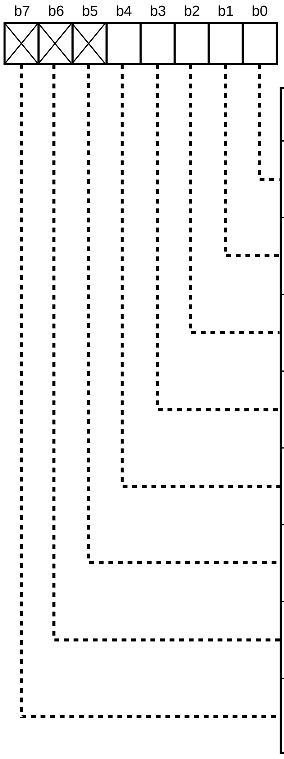
Bit symbol	Bit name	Function	RW
CM10	All clock stop control bit[2]	0 : Clock ocsillating 1 : All clocks stop (stop mode)[3]	RW
——	Reserved bit	Should set to "0"	RW
——	Reserved bit	Should set to "1"	RW
——	Reserved bit	Should set to "0"	RW
CM17	CPU clock switch bit 2	0 : Main clock[4] 1 : PLL clock[4]	RW

Notes :

1. The PRC0 bit in the PRCR register is set to "1" (write enable) before rewriting CM1 register.
2. When setting the CM10 bit to "1" to set XOUT to "H", an internal feedback resistance is disabled. XIN, XCIN and XCOUT are placed on high-impedance.
3. When setting the CM10 bit to "1", the MCD4 to MCD0 bits in the MCD register are set to "01000_2" (divide-by-8 mode). When setting the CM20 bit to "1" (oscillation stop detect function enabled) or the CM21 bit to "1" (ring oscillator selected), avoid setting the CM10 bit to "1".
4. CM17 bit is enabled only when the CM21 bit in the CM2 register is set to "0". Follow a procedure shown Figure in 1.8.13 to set the CM17 bit to "1".

Figure 1.8.3. CM1 Register

Main clock division register[1]



Symbol	Address	When reset
MCD	$000C_{16}$	$XXX01000_2$

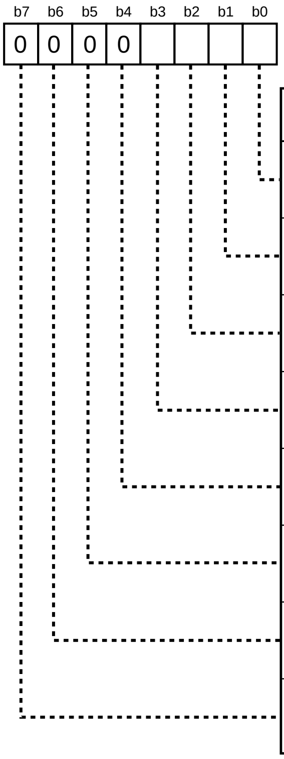
Bit symbol	Bit name	Function	RW
MCD0	Main clock division select bit[2,4]	b4 b3 b2 b1 b0 1 0 0 1 0 : Divide-by-1(no devision) mode 0 0 0 1 0 : Divide-by-2 mode 0 0 0 1 1 : Divide-by-3 mode 0 0 1 0 0 : Divide-by-4 mode 0 0 1 1 0 : Divide-by-6 mode 0 1 0 0 0 : Divide-by-8 mode 0 1 0 1 0 : Divide-by-10 mode 0 1 1 0 0 : Divide-by-12 mode 0 1 1 1 0 : Divide-by-14 mode 0 0 0 0 0 : Divide-by-16 mode (Note 3)	RW
MCD1			RW
MCD2			RW
MCD3			RW
MCD4			RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
——			—
——			—

Notes :

1. The PRC0 bit in the PRCR register is set to "1" (write enable) before rewriting the MCD register.
2. While entering to stop mode or low power consumption mode, the MCD4 to MCD0 bits are set to "01000_2"(divide-by-8 mode).
   Divide-by-8 mode cannot be entered even if the CM05 bit in the CM0 register is set to "1"(XIN-XOUT stopped) in ring oscillator mode.
3. Avoid setting a bit combination except the above.
4. CAN-associated registers should be accessed with setting the MCD4 to MCD0 bits to "10010_2" (no division mode).

Figure 1.8.4. MCD Register

Oscillation stop detect register[1]



Symbol CM2
Address 000D$_{16}$
When reset 0000 0000$_2$

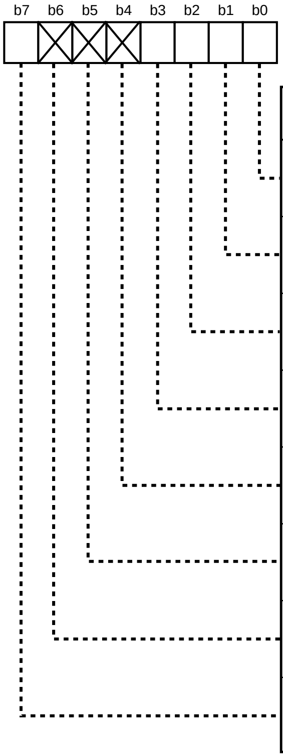
Bit symbol	Bit name	Function	RW
CM20	Oscillation stop detect enable bit	0: Oscillation stop detect function disabled 1: Oscillation stop detect function enabled	RW
CM21	CPU clock switch bit[2,3]	0: Main clock[6] 1: Ring oscillator clock	RW
CM22	Oscillation stop detect flag[4]	0: Main clock not stopped 1: Detect main clock stop	RW
CM23	XIN clock monitor flag[5]	0: Main clock oscillating 1: Main clock stop	RO
——	Reserved bit	Should set to "0"	RW
——			RW
——			RW
——			RW

Notes :

1. The PRC0 bit in the PRCR register is set to "1" (write enable) before rewriting the CM2 register.
2. When the main clock oscillation stop is detected while setting the CM20 bit to "1" (oscillation stop detect function enabled), the CM21 bit is set to "1" (ring oscillator clock). Although the main clock starts oscillating, the CM21 bit is not set to "0" (main clock). When the main clock is used as a CPU clock source after restarting the main clock oscillation, the CM21 bit should be set to "0" by program.
3. When setting the CM20 bit to "1" and the CM22 bit to "1" (main clock stop detected), avoid setting the CM21 bit to "0".
4. When detecting the main clock stop, the CM22 bit is set to "1" (detect main clock stop). The CM22 bit can be set to "0" (main clock not stopped) only by program.
   When setting the CM22 bit to "0" by program during a main clock oscillation stop, the CM22 bit cannot be set to "1" before detecting the next main clock stop.
5. A main clock status should be determine by reading the CM23 bit with an oscillation stop interrupt processing program several times.
6. When setting the CM21 bit to "0", the CPU clock is selected by the CM17 bit in the CM1 register.

Figure 1.8.5. CM2 Register

Count source prescaler register



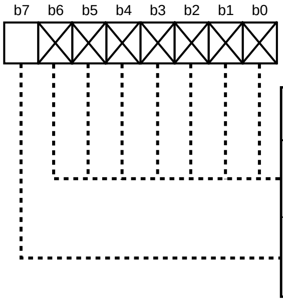
Symbol	Address	When reset
TCSPR	$035F_{16}$	$0XXX\ 0000_2$

Bit symbol	Bit name	Function	RW
CNT0	Division rate select bit[1]	As a setting value is n, f_{2n} is divided main clock, ring oscillator clock or PLL clock by 2n. When n is set to "0", it is no division.	RW
CNT1			RW
CNT2			RW
CNT3			RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
——			—
——			—
CST	Operation enable bit	0: Divider stops 1: Divider starts	RW

Notes :

1. The CST bit should be set to "0" before rewriting the CNT3 to CNT0 bits.

Clock prescaler reset flag

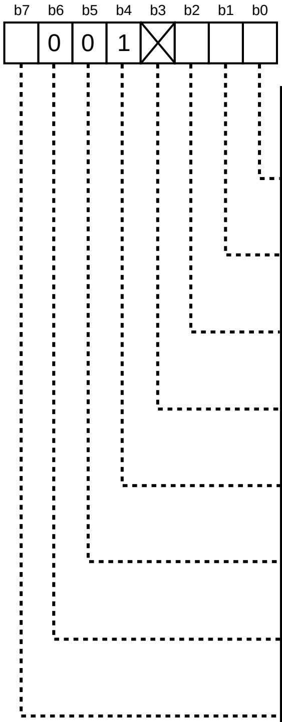


Symbol	Address	When reset
CPSRF	0341_{16}	$0XXX\ XXXX_2$

Bit symbol	Bit name	Function	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
CPSR	Clock prescaler reset flag	When setting this bit to "1", fc divided by 32 is initialized. When read, its content is "0".	RW

Figure 1.8.6. TCSPR and CPSRF Registers

PLL control register 0[1]



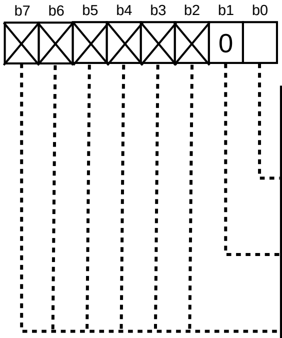
Symbol	Address	When reset
PLC0	0376_{16}	$0011\ X100_2$

Bit symbol	Bit name	Function	RW
PLC00	Programmable counter select bit[2]	See Table 1.8.2	RW
PLC01			RW
PLC02			RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
——	Reserved bit[2]	Should set to "1"	RW
——	Reserved bit[2]	Should set to "0"	RW
——	Reserved bit	Should set to "0"	RW
PLC07	Operation enable bit[3,4]	0: PLL Off 1: PLL On	RW

Notes :

1. The PRC0 bit in the PRCR register is set to "1" (write enable) before rewriting the PLC0 register.
2. A counter value should be set when setting the PLC07 bit to "0" (PLL off). Once a value is written, it cannot be changed.
3. With PLL function, the PD8_7 bit in the PD8 register is set to "0" (input) and the CM04 bit in the CM0 register is set to "0" (I/O port). The PD8_6 bit in the PD8 register is set to "0" (input) to connect $P8_6$ to Vss.
4. The CM17 bit is set to "0" (main clock as CPU clock source), the PLC07 bit be set to "0" (PLL off) and PLV00 bit be set to "0" (cut off power to PLL) before entering wait or stop mode.

VDC control register for PLL[1]



Symbol	Address	When reset
PLV	0017_{16}	$XXXX\ XX01_2$

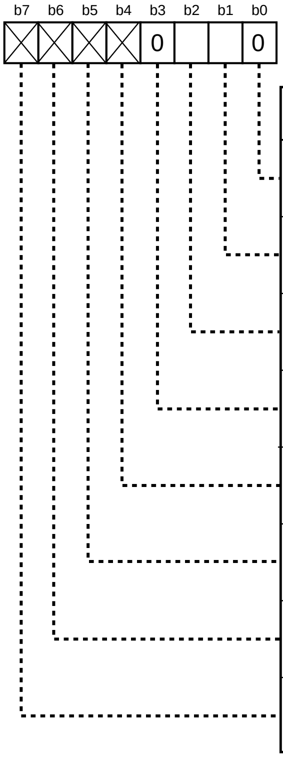
Bit symbol	Bit name	Function	RW
PLV00	PLL VDC enable bit[2]	0 : Cut off power to PLL 1 : Power to PLL	RW
——	Reserved bit	Should set to "0"	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—

Notes :

1. The PRC3 bit in the PRCR register is set to "1" (write enable) before rewriting the PLV register.
2. The CM17 bit is set to "0" (main clock as CPU clock source), the PLC07 bit be set to "0" (PLL off) and PLV00 bit be set to "0" (Cut off power to PLL) before entering wait or stop mode.

Figure 1.8.7. PLC0 and PLV Registers

PLL control register 1[1,2]



Symbol	Address	When reset
PLC1	0377_{16}	$XXXX\ 0000_2$

Bit symbol	Bit name	Function	RW
——	Reserved bit	Should set to "0"	RW
PLC11	PLL clock division enable bit[3]	0 : Division disabled 1 : Division enabled	RW
PLC12	PLL clock division switch bit	0 : Divide-by-2 1 : Divide-by-3	RW
——	Reserved bit	Should set to "0"	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
——			—
——			—
——			—

Notes :

1. The PRC0 bit in the PRCR register is set to "1" (write enable) before rewriting the PLC1 register.
2. The CM17 bit in the CM1 register is set to "0" (main clock) before rewriting the PLC1 register.
3. When the CM21 bit in the CM2 register is set to "0" (main clock as CPU clock source), the PLC11 bit should be set to "1" before setting the CM17 bit to "1" (PLL clock as CPU clock source). The PLL clock divided by 2 or divided by 3 becomes a clock source of the CPU clock and peripheral function clock.

Figure 1.8.8. PLC1 Register

1. Main Clock

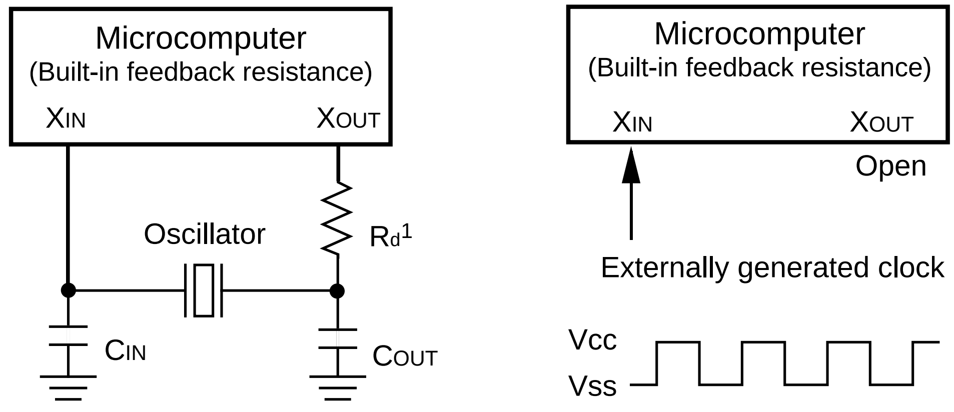
Main clock oscillation circuit generates the main clock. The main clock becomes a clock source of the CPU clock and peripheral function clock.

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.


Microcomputer
(Built-in feedback resistance)
XIN
XOUT
Oscillator
Rd1
CIN
COUT
Microcomputer
(Built-in feedback resistance)
XIN
XOUT
Open
Externally generated clock
Vcc
Vss

Notes :

1. A damping resistance should be inserted if required. The resistance varies depending on an oscillator setting. Use values recommended by each oscillator manufacturer.
   A feedback resistance should be inserted between XIN and XOUT if an oscillation manufacturer recommands to add the resistance externally.

Figure 1.8.9. Main Clock Circuit Connection

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 fc as the sub clock can be output from the CLKOUT pin.

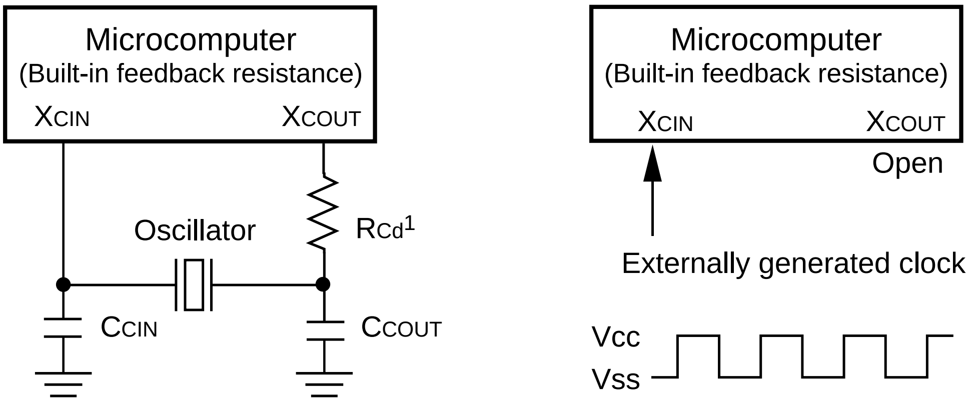
The sub clock oscillation circuit is configured by connecting an oscillator between the XCIN and XCOUT pins. The circuit has a built-in feedback resistance. The feedback resistance is separated from an oscillation circuit in stop mode to reduce power consumption. An externally generated clock can be 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.


Microcomputer
(Built-in feedback resistance)
XCIN
XCOUT
Oscillator
RCd1
CCIN
CCOUT
Microcomputer
(Built-in feedback resistance)
XCIN
XCOUT
Open
Externally generated clock
Vcc
Vss

Notes :
1. A damping resistance should be inserted if required. The resistance varies depending on the oscillator setting. Use values recommended by each oscillator manufacturer.
A feedback resistance should be inserted between XIN and XOUT if an oscillation manufacturer recommands to add the resistance externally.

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".

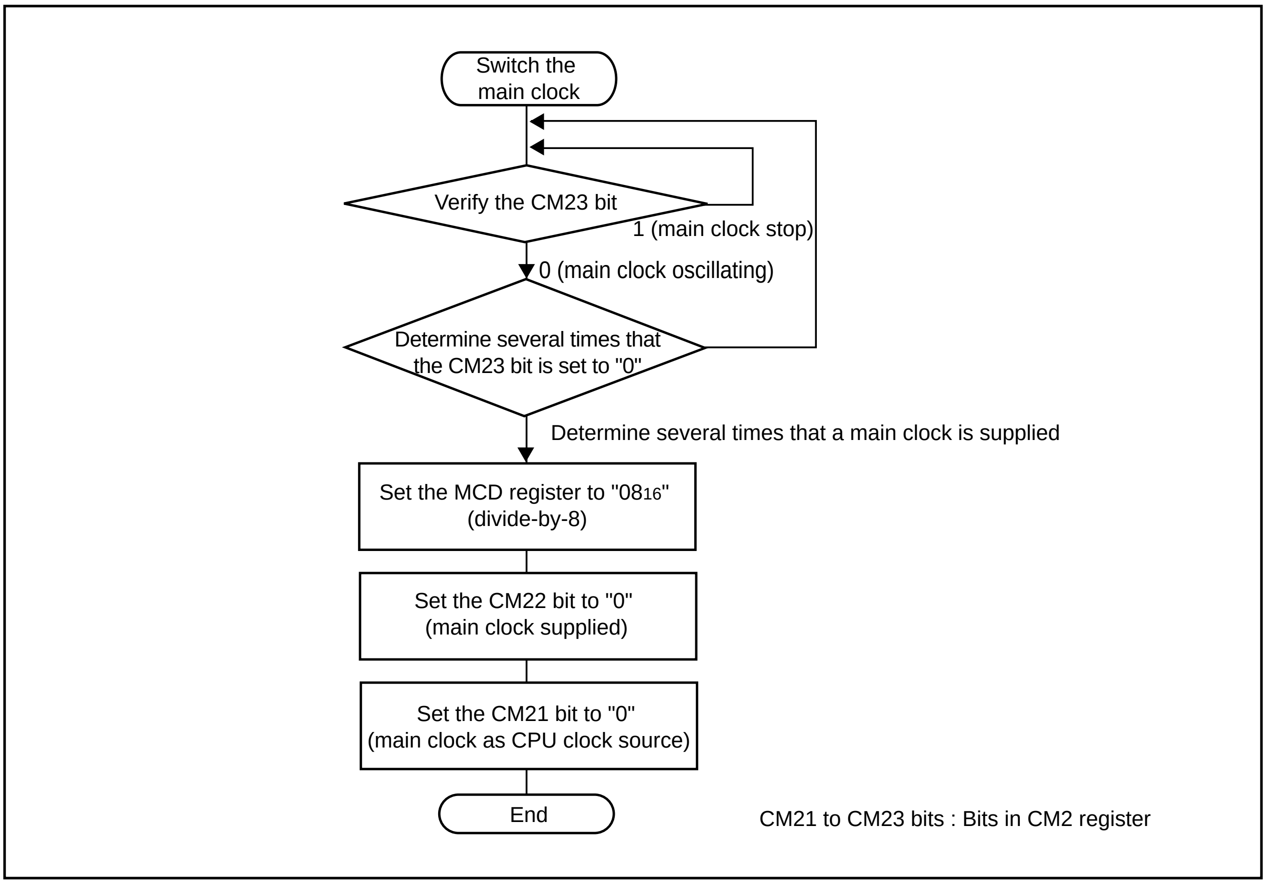
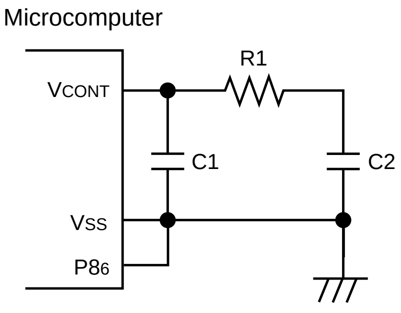

Switch the main clock
Verify the CM23 bit
1 (main clock stop)
0 (main clock oscillating)
Determine several times that the CM23 bit is set to "0"
Determine several times that a main clock is supplied
Set the MCD register to "0816" (divide-by-8)
Set the CM22 bit to "0" (main clock supplied)
Set the CM21 bit to "0" (main clock as CPU clock source)
End
CM21 to CM23 bits : Bits in CM2 register

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 XCOUNT pins. When using the PLL frequency synthesizer, the sub clock cannot be used.



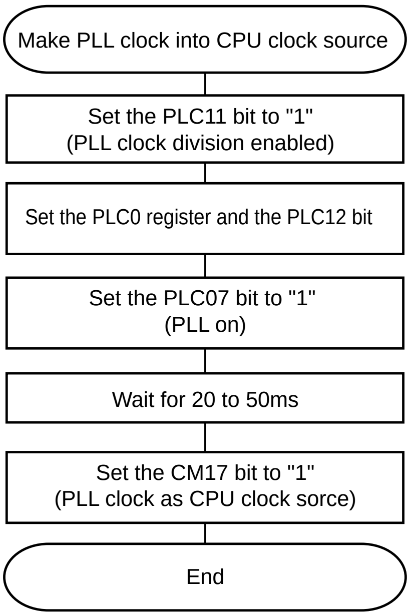
C1=220pF, C2=0.1 µF, R1=1 kΩ [2]

Notes :
1. VSS should be connected to GND via C1 and C2 with shorter wiring. Ground pattern should be formed around the circuit.
2. Optimal value of a low pass filter circuit element, connected to the VCONT pin, depends on the enviroments (e.g.noise). The most appropriate value for a system should be used after evaluating with the optimal value and with several optimal values.
C1 and C2 should be used with best temperature characteristics.

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


Make PLL clock into CPU clock source
Set the PLC11 bit to "1" (PLL clock division enabled)
Set the PLC0 register and the PLC12 bit
Set the PLC07 bit to "1" (PLL on)
Wait for 20 to 50ms
Set the CM17 bit to "1" (PLL clock as CPU clock sorce)
End

PLC11 bit : Bit in PLC1 register
PLC00 to PLC07 bits : Bits in PLC0 register
CM17 bit : Bit in CM1 register

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

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 "08_{16}" (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_1, f_8, f_{32} and f_{2n}

f_1, f_8, f_{32} and f_{2n} are the main clock[1] 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 stops. The clocks also stops in low-power consumption mode.

f_1, f_8 and f_{2n} 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_{2n} division. f_1 is also used for an operation clock of the intelligent I/O.

The CLKOUT pin outputs f_8 and f_{32} . Refer to the paragraph "Clock Output Function" for details.

2. f_{AD}

f_{AD} is an operation clock of the A-D convertor and has the same frequency as the main clock[1] 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_{AD} stops. f_{AD} 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).

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 [1]	CM0 register [2]		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 [1]		PM0 register [1]	CM0 register [2]		CLKOUT pin function
PM15	PM14	PM07	CM01	CM00	
00_2, 10_2, 11_2,		0	0 [3]	0 [3]	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 [3]	0 [3]	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 "01_2" (P53/BCLK), the CM01 to CM00 bits should be set to "00_2" (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.

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[1] becomes the CPU clock and a clock source for the peripheral function clock. When the sub clock runs, f_{C32} 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, f_{C32} 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. f_{C32} 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 f_{C32} can be used as the peripheral function clock. In low-power consumption mode, the MCD register should be set to "08_{16}" (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, f_{C32} 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, f_{C32} 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[2] 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=08_{16}).

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.

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), f_1, f_8, f_{32}, f_{2n} and f_{AD} stop in wait mode. Power consumption can be reduced. f_{C32} 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, $\overline{\mathrm{NMI}}$ interrupt or peripheral function interrupt.

When using a hardware reset or $\overline{\mathrm{NMI}}$ interrupt to exit wait mode, the ILVL2 to ILVL0 bits for the peripheral function interrupt should be set to "000_2" (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, $\overline{\mathrm{CS0}}$ to $\overline{\mathrm{CS3}}$, $\overline{\mathrm{BHE}}$		Maintains status before entering wait mode	
$\overline{\mathrm{RD}}$, $\overline{\mathrm{WR}}$, $\overline{\mathrm{WRL}}$, $\overline{\mathrm{WRH}}$, $\overline{\mathrm{DW}}$, $\overline{\mathrm{CASL}}$, $\overline{\mathrm{CASH}}$		"H" [1]	
$\overline{\mathrm{RAS}}$		"H" [1]	
$\overline{\mathrm{HLDA}}$,BCLK		"H"	
ALE		"L"	
Port		Maintains status before entering wait mode	
CLK_{OUT}	When f_C is selected	Clock output	
	When f_8, f_{32} 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, $\overline{\mathrm{CAS}}$ and $\overline{\mathrm{RAS}}$ are set to "L".

Table 1.8.7. Interrupts to Exit Wait Mode

Interrupt	When CM02=0	When CM02=1
$\overline{\text{NMI}}$ 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 f_{C32}
$\overline{\text{INT}}$ 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 "000_2" (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 "08_{16}" (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 "000_2" (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 "000_2" (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.

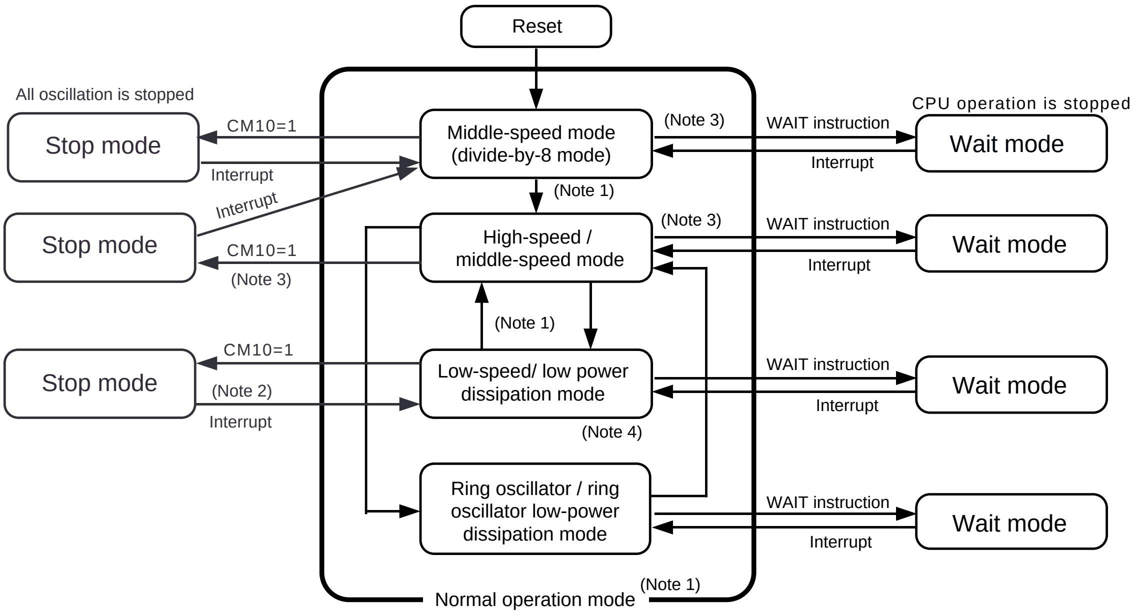
- When the sub clock is set as the CPU clock before entering stop mode : Sub clock
- When the main clock is set as the CPU clock before entering stop mode : Main clock divided by eight

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]	
$\overline{HLDA}$, BCLK		"H"	
ALE		"H"	
Port		Maintains status before entering stop mode	
CLKOUT	When fC selected	"H"	
	When f8, f32 selected	Maintains status before entering stop mode	
XIN		High-impedance	
XOUT		"H"	
XCIN, XCOUT		High-impedance	

Notes :

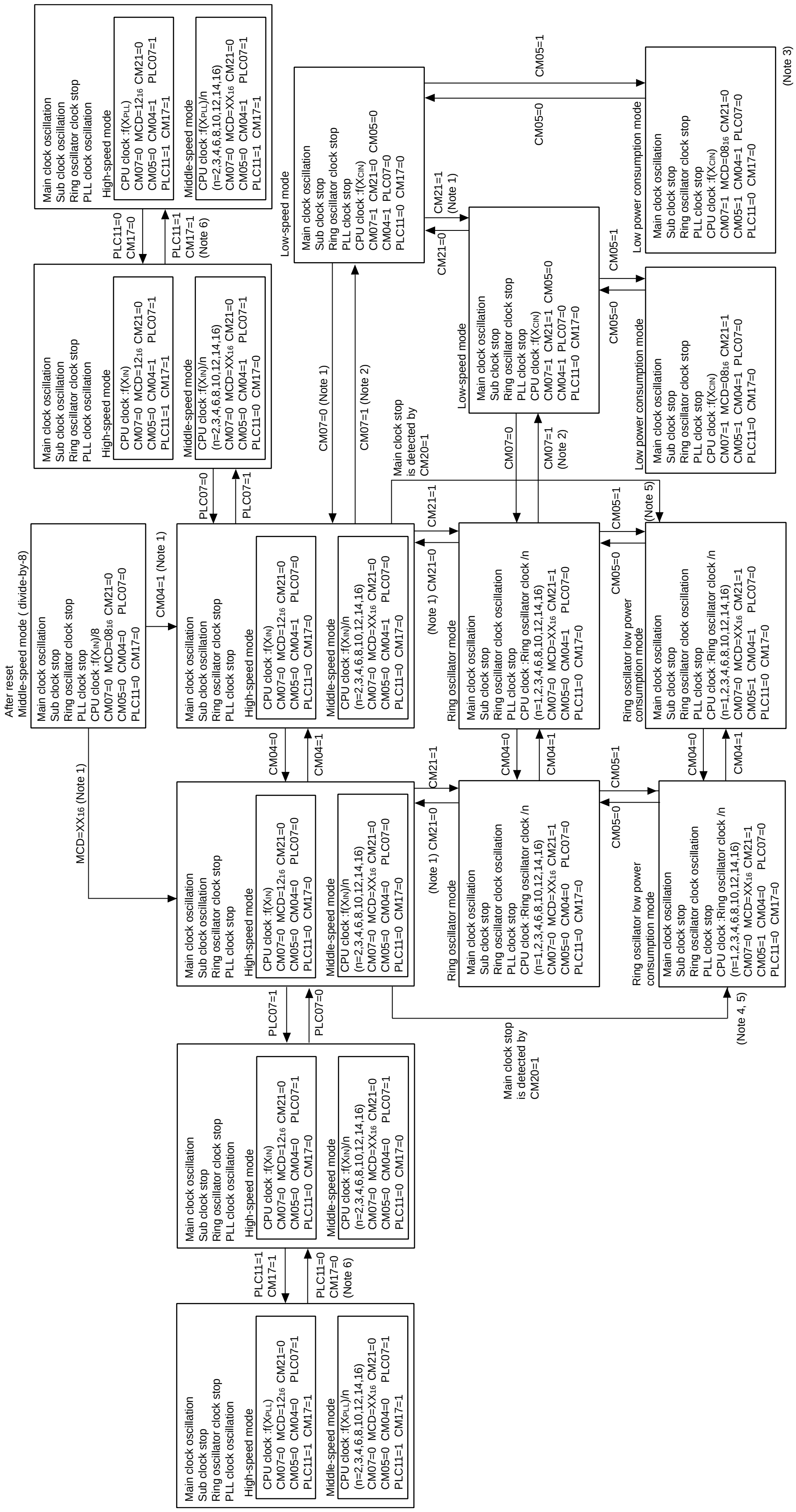
1. In self-refresh operation with the DRAMC, $\overline{CAS}$ and $\overline{RAS}$ are set to "L".



Notes :

1. See Figure 1.8.15.
2. The MCD4 to MCD0 bits in the MCD register is set to "01000$_2$"(divide-by-8 mode).
3. 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) before setting the PLC07 bit to "0" (PLL off). The PLV00 bit should be set to "0" (no power to PLL) before entering wait mode or stop mode.
4. When using the PLL frequency synthesizer, low-speed and low-power consumption mode cannot be entered.

Figure 1.8.14. Status Transition in Wait Mode and Stop Mode



→ : Change according to a direction of an arrow.

MCD=XX_{16}: Set the desired division in the MCD register.

Notes :

1. Clocks should be switched after main clock oscillation is fully stable.
2. Clocks should be switched after sub clock oscillation is fully stable.
3. The MCD register is set to "08_{16}" automatically.
4. With the PLL frequency synthesizer, low-speed mode and low power consumption mode cannot be entered.

Figure 1.8.15. Status Transition

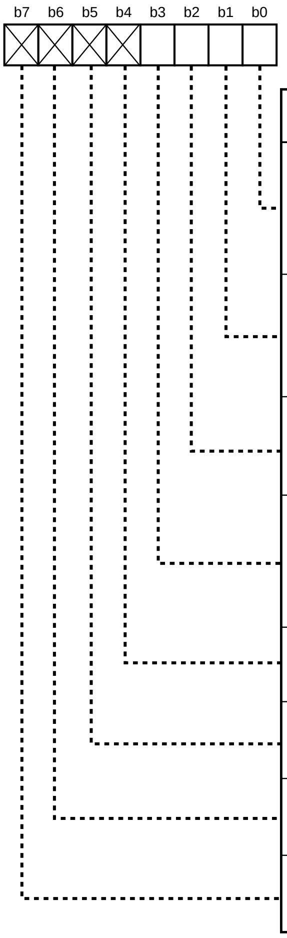
Protection

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 PR3 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.

Protect register



Symbol	Address	When reset
PRCR	$000A_{16}$	$XXXX\ 0000_2$

Bit symbol	Bit name	Function	RW
PRC0	Protect bit 0	Enables writing to CM0, CM1, CM2, MCD, PLC0, PLC1 registers 0 : Write disable 1 : Write enable	RW
PRC1	Protect bit 1	Enables writing to PM0, PM1, INVC0, INVC1 registers 0 : Write disable 1 : Write enable	RW
PRC2	Protect bit 2[1]	Enables writing to PD9, PS3 registers 0 : Write disable 1 : Write enable	RW
PRC3	Protect bit 3	Enables writing to PLV, VDC0, VDC1 registers 0 : Write disable 1 : Write enable	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–
——			–
——			–

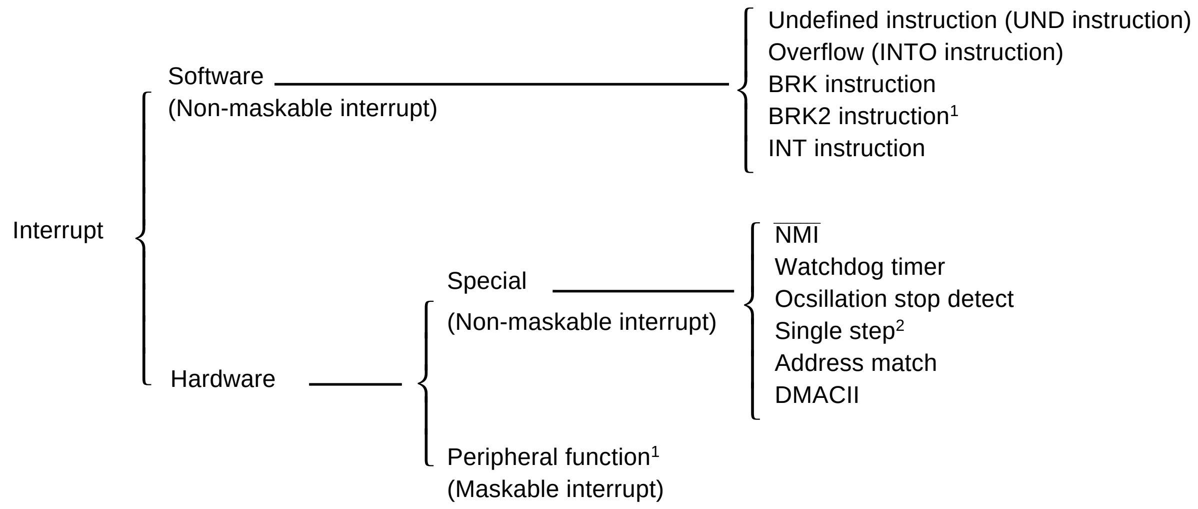
Notes :
1. The PRC2 bit is set to "0" by writing into any addresses after setting the PRC2 bit to "1".
The PRC0, PRC1 and PRC3 bits are not automatically set to "0". They should be set to"0" by program.

Figure 1.8.16. PRCR Register

Interrupts

Types of Interrupts

Figure 1.9.1 shows types of interrupts.



Notes :

1. Peripheral function interrupts are generated by the peripheral functions built into the microcomputer system.
2. Avoid using this interrupt because this is a dedicated interrupt for development suport tools only.

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

(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.

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

(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 "111_2" (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.

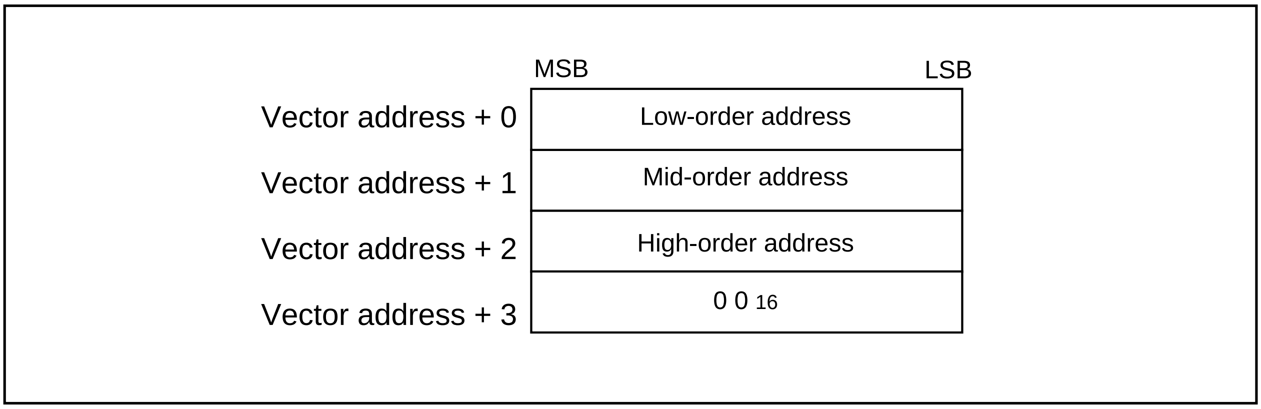


Figure 1.9.2. Interrupt Vector

(1) Fixed Vector Tables

Fixed vector tables are allocated in addresses from FFFFDC$_{16}$ to FFFFFF$_{16}$. 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$_{16}$ to FFFFDF$_{16}$		M32C/80 series software manual
Overflow	FFFFE0$_{16}$ to FFFFE3$_{16}$		
BRK instruction	FFFFE4$_{16}$ to FFFFE7$_{16}$	If a content of FFFFE7$_{16}$ is FF$_{16}$, a program is executed from an address stored into software interrupt number 0 in variable vector table	
Address matching	FFFFE8$_{16}$ to FFFFEB$_{16}$		
-	FFFFEC$_{16}$ to FFFFEF$_{16}$	Reserved space	
Watchdog timer	FFFFF0$_{16}$ to FFFFF3$_{16}$	These address are shared by the watchdog timer and oscillation stop detect interrupt	Watchdog timer
-	FFFFF4$_{16}$ to FFFFF7$_{16}$	Reserved space	
$\overline{\text{NMI}}$	FFFFF8$_{16}$ to FFFFFB$_{16}$		
Reset	FFFFFC$_{16}$ to FFFFFF$_{16}$		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.

Table 1.9.2. Relocatable Vector Tables

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

Table 1.9.2. Relocatable Vector Tables (Continued)

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

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.

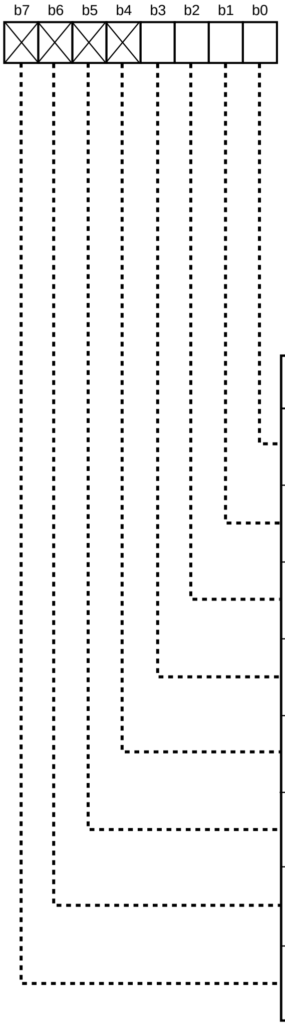
Table 1.9.4. Interrupt Priority Levels Field Encording

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.

Interrupt control register



Symbol	Address	When reset
TA0IC to TA4IC	$006C_{16}$, $008C_{16}$, $006E_{16}$, $008E_{16}$, 0070_{16}	XXXX $X000_2$
TB0IC to TA5IC	0094_{16}, 0076_{16}, 0096_{16}, 0078_{16}, 0098_{16}, 0069_{16}	XXXX $X000_2$
S0TIC to S4TIC	0090_{16}, 0092_{16}, 0089_{16}, $008B_{16}$, $008D_{16}$	XXXX $X000_2$
S0RIC to S4RIC	0072_{16}, 0074_{16}, $006B_{16}$, $006D_{16}$, $006F_{16}$	XXXX $X000_2$
BCN0IC to BCN4IC	0071_{16}, 0091_{16}, $008F_{16}$, 0071_{16}[1], 0091_{16}[2]	XXXX $X000_2$
DM0IC to DM3IC	0068_{16}, 0088_{16}, $006A_{16}$, $008A_{16}$	XXXX $X000_2$
AD0IC, AD1IC	0073_{16}, 0086_{16}	XXXX $X000_2$
KUPIC	0093_{16}	XXXX $X000_2$
IIO0IC to IIO5IC	0075_{16}, 0095_{16}, 0077_{16}, 0097_{16}, 0079_{16}, 0099_{16}	XXXX $X000_2$
IIO6IC to IIO11IC	$007B_{16}$, $009B_{16}$, $007D_{16}$, $009D_{16}$, $007F_{16}$, 0081_{16}	XXXX $X000_2$
CAN0IC0 to CAN2IC	$009D_{16}$, $007F_{16}$, 0081_{16}[3]	XXXX $X000_2$

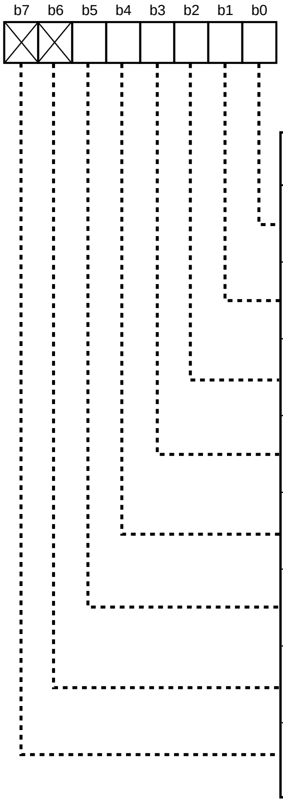
Bit symbol	Bit name	Function	RW
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt request bit	0 : No interrupt requested 1 : Interrupt requested[4]	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–
——			–
——			–

Notes :

1. The BCN0IC register shares an address with the BCN3IC register.
2. The BCN1IC register shares an address with the BCN4IC register.
3. The IIO9IC register shares an address with the CAN0IC register.
   The IIO10IC register shares an address with the CAN1IC register.
   The IIO11IC register shares an address with the CAN2IC register.
4. The IR bit can be set to "0" only (avoid setting to "1").

Figure 1.9.3. Interrupt Control Register (1)

Interrupt control register



Symbol	Address	When reset
INT0IC to INT2IC	$009E_{16}$, $007E_{16}$, $009C_{16}$	XX00 $X000_2$
INT3IC to INT5IC[1]	$007C_{16}$, $009A_{16}$, $007A_{16}$	XX00 $X000_2$

Bit symbol	Bit name	Function	RW
ILVL0	Interrupt priority level select bit	b2 b1 b0 0 0 0 : Level 0 (interrupt disabled) 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
ILVL1			RW
ILVL2			RW
IR	Interrupt request bit	0 : No interrupt requested 1 : Interrupt requested[2]	RW
POL	Polarity switch bit	0 : Selects falling edge or "L" level 1 : Selects rising edge or "H" level[3]	RW
LVS	Level sense/edge sense switch bit	0 : Edge sense 1 : Level sense[4]	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
——			—

Notes

1. When using a 16-bit data bus in microprocessor or memory expansion mode, each pin for the INT3 to INT5 bits is used as data bus. The ILVL2 to ILVL0 bits in the INT3IC, INT4IC and INT5IC registers should be set to "000_2".
2. The IR bit can be set to "0" only (avoid setting to "1").
3. When a bit corresponding to the IFSR register is set to "1" (both edge), the POL bit should set to "0" (falling edge).
4. With a level sense, a bit corresponding to the IFSR register should be set to "0" (one edge).

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.

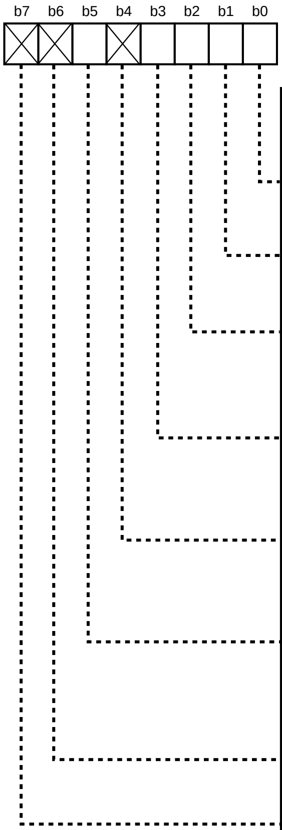
When an interrupt request is generated, its interrupt priority level is compared to IPL. This interrupt is enabled only when its interrupt priority level is greater than IPL. When setting the ILVL2 to ILVL0 bits to "00_2" (level 0), its interrupt is disabled.

- **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").

Exit priority register



Symbol	Address	When reset
RLVL	009F$_{16}$	XX0X 0000$_2$

Bit symbol	Bit name	Function	RW
RLVL0	Interrupt priority set bits to exit stop/wait mode[1]	b2 b1 b0 0 0 0 : Level 0 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
RLVL1			RW
RLVL2			RW
FSIT	High-speed interrupt set bit[2]	0: Interrupt priority level 7 is used for normal interrupt 1: Interrupt priority level 7 is used for high-speed interrupt	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
DMA II	DMAC II select bit[3]	0: Interrupt priority level 7 is used for interrupt 1: Interrupt priority level 7 is used for DMAC II transfer	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
——			—

Notes :

1. Stop or wait mode is exited when a requested interrupt priority level is higher than a level set in the RLVL2 to RLVL0 bits. The RLVL2 to RLVL0 bits should be set the same value as IPL in the FLG register.
2. When setting the FSIT bit to "1" (high-speed interrupt), interrupt level 7 as the highest interrupt priority level becomes high-speed interrupt. Only one interrupt should be set to level 7 in interrupt priority level to set the DMA II bit to "0" (normal interrupt).
3. Avoid setting the DMA II bit to "0" (normal interrupt) again once setting it to "1" (DMAC II transfer). When setting the DMA II bit to "1", the FSIT bit should be set to "0" (normal interrupt). DMAC II cannot be used with high-speed interrupt simultaneously. The I flag and IPL do not affect DMAC II transfer.

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 detemination.

The interrupt sequence is handled as follows:

(1) The CPU obtains interrupt information (interrupt number and interrupt request level) by reading address 000000_{16} (address 000002_{16} 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[1] 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)

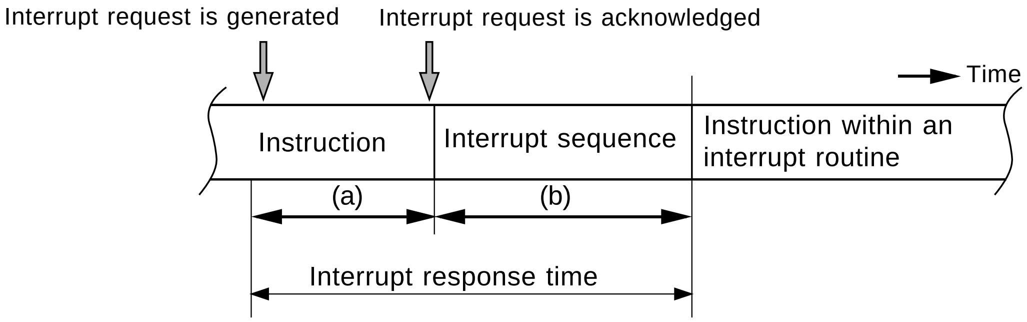
(4) A temporary register within the CPU is saved to the stack space and to the SVF register for the high-speed interrupt.
(5) Content of PC is saved to the stack space and to the SVP register for the high-speed interrupt.
(6) An interrupt priority level of an acknowledged interrupt in IPL is set.
(7) 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.

(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).


Interrupt request is generated
Interrupt request is acknowledged
Time
Instruction
Interrupt sequence
Instruction within an interrupt routine
(a)
(b)
Interrupt response time

(a) Period between an interrupt request generation and an instruction being completed.

(b) Period required to handle an interrupt sequence.

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).

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[1]	16 cycles	16 cycles
INT instruction	Even address	12 cycles	14 cycles
	Odd address [1]	14 cycles	14 cycles
$\overline{\text{NMI}}$ Watchdog timer Undefined instruction Address match	Even address [2]	13 cycles	15 cycles
Overflow	Even address [2]	14 cycles	16 cycles
BRK instruction (variable vector table)	Even address	17 cycles	19 cycles
	Odd address [1]	19 cycles	19 cycles
BRK instruction (fixed vector table)	Even address [2]	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.

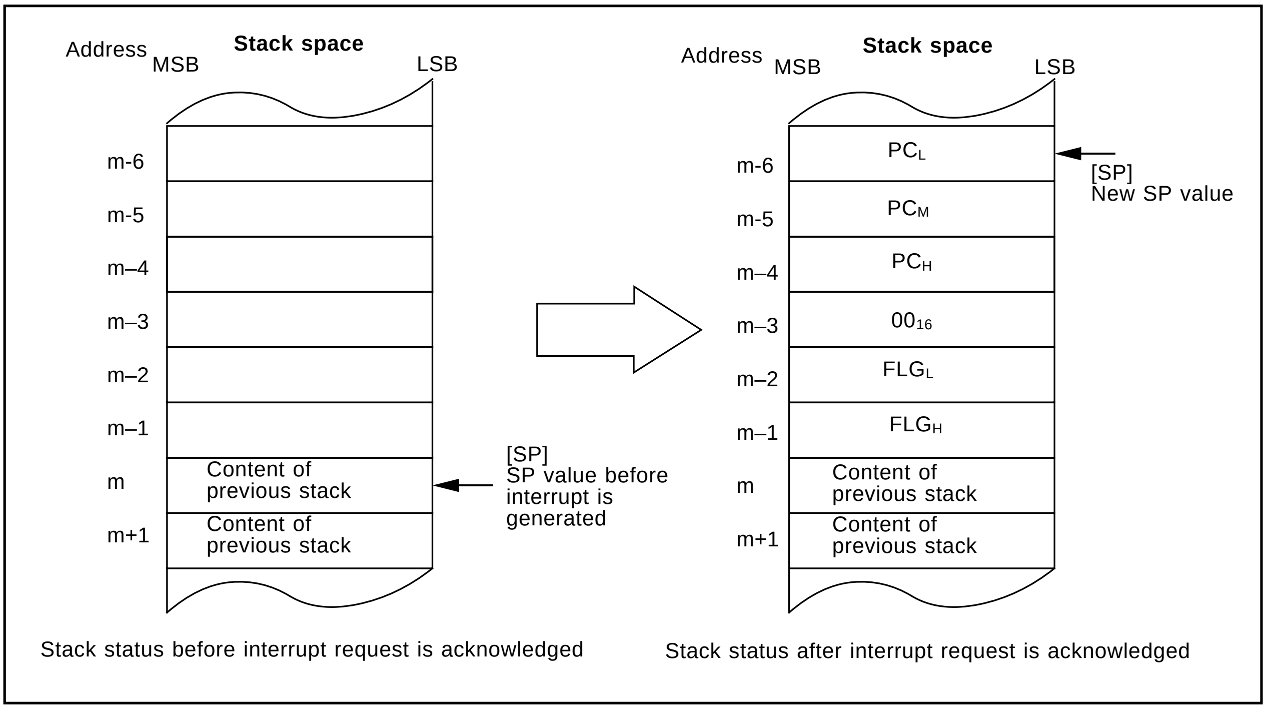

Address
MSB
Stack space
LSB
m-6
m-5
m–4
m–3
m–2
m–1
m
m+1
Content of previous stack
Content of previous stack
[SP]
SP value before interrupt is generated
Stack status before interrupt request is acknowledged
Address
MSB
Stack space
LSB
m-6
m-5
m–4
m–3
m–2
m–1
m
m+1
PCL
PCM
PCH
0016
FLGL
FLGH
Content of previous stack
Content of previous stack
[SP]
New SP value
Stack status after interrupt request is acknowledged

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.

Reset > $\overline{\text{NMI}}$ > Watchdog > Peripheral function > Address match

Figure 1.9.8. Interrupt Priority

(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.

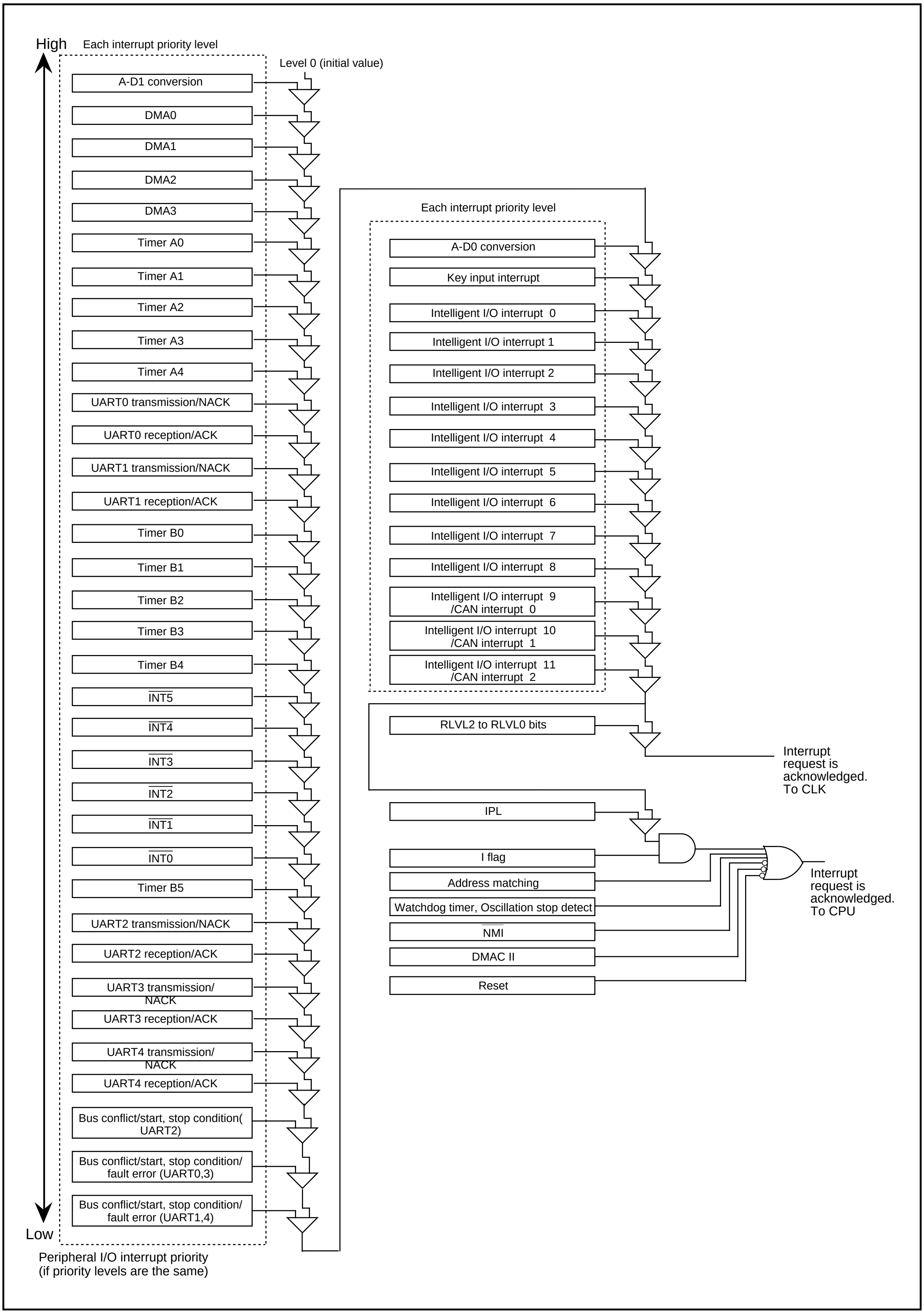

High
Each interrupt priority level
Level 0 (initial value)
A-D1 conversion
DMA0
DMA1
DMA2
DMA3
Timer A0
Timer A1
Timer A2
Timer A3
Timer A4
UART0 transmission/NACK
UART0 reception/ACK
UART1 transmission/NACK
UART1 reception/ACK
Timer B0
Timer B1
Timer B2
Timer B3
Timer B4
$\overline{\text{INT5}}$
$\overline{\text{INT4}}$
$\overline{\text{INT3}}$
$\overline{\text{INT2}}$
$\overline{\text{INT1}}$
$\overline{\text{INT0}}$
Timer B5
UART2 transmission/NACK
UART2 reception/ACK
UART3 transmission/ NACK
UART3 reception/ACK
UART4 transmission/ NACK
UART4 reception/ACK
Bus conflict/start, stop condition(UART2)
Bus conflict/start, stop condition/ fault error (UART0,3)
Bus conflict/start, stop condition/ fault error (UART1,4)
Low
Peripheral I/O interrupt priority
(if priority levels are the same)
Each interrupt priority level
A-D0 conversion
Key input interrupt
Intelligent I/O interrupt 0
Intelligent I/O interrupt 1
Intelligent I/O interrupt 2
Intelligent I/O interrupt 3
Intelligent I/O interrupt 4
Intelligent I/O interrupt 5
Intelligent I/O interrupt 6
Intelligent I/O interrupt 7
Intelligent I/O interrupt 8
Intelligent I/O interrupt 9 /CAN interrupt 0
Intelligent I/O interrupt 10 /CAN interrupt 1
Intelligent I/O interrupt 11 /CAN interrupt 2
RLVL2 to RLVL0 bits
Interrupt request is acknowledged. To CLK
IPL
I flag
Address matching
Watchdog timer, Oscillation stop detect
$\overline{\text{NMI}}$
DMAC II
Reset
Interrupt request is acknowledged. To CPU

Figure 1.9.9. Interrupt Priority Select Circuit

$\overline{\text{INT}}$ Interrupt

The $\overline{\text{INT}}$i interrupts (i = 0 to 5) are generated by an external input. The LVS bit in the INTiIC 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 INTiIC register can determine polarity. With an edge sense, when the IFSRi 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 IFSRi bit to "1", the POL bit in a corresponding register should be set to "0" (falling edge).

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

Figure 1.9.10 shows the IFSR register.

External interrupt request cause select register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
IFSR	031F$_{16}$	0000 0000$_2$

Bit symbol	Bit name	Function	RW
IFSR0	INT0 interrupt polarity select bit[1]	0 : One edge 1 : Both edges	RW
IFSR1	INT1 interrupt polarity select bit[1]	0 : One edge 1 : Both edges	RW
IFSR2	INT2 interrupt polarity select bit[1]	0 : One edge 1 : Both edges	RW
IFSR3	INT3 interrupt polarity select bit[1]	0 : One edge 1 : Both edges	RW
IFSR4	INT4 interrupt polarity select bit[1]	0 : One edge 1 : Both edges	RW
IFSR5	INT5 interrupt polarity select bit[1]	0 : One edge 1 : Both edges	RW
IFSR6	UART0, 3 interrupt cause select bit	0 : UART3 bus conflict, start condition detect, stop condition detect, fault error detect 1 : UART0 bus conflict, start condition detect, stop condition detect, fault error detect	RW
IFSR7	UART1, 4 interrupt cause select bit	0 : UART4 bus conflict, start condition detect, stop condition detect, fault error detect 1 : UART1 bus conflict, start condition detect, stop condition detect, fault error detect	RW

Notes :

1.When selecting a level sense, this bit should be set to "0".
When selecting both edges, the POL bit in the INTiIC register (i = 0 to 5) should be set to "0" (falling edge).

Figure 1.9.10. IFSR Register

$\overline{\text{NMI}}$ 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[1] 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 "1", 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.

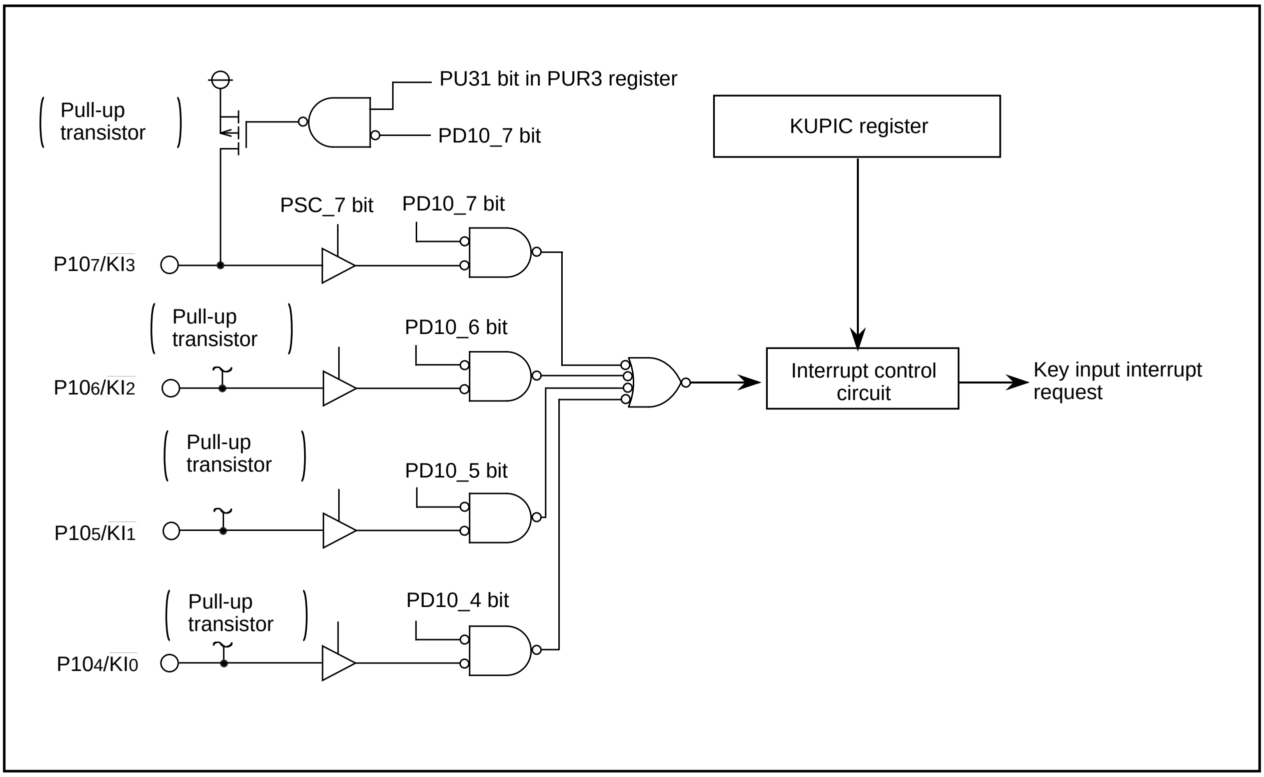

Pull-up transistor
PU31 bit in PUR3 register
PD10_7 bit
KUPIC register
PSC_7 bit
PD10_7 bit
P107/$\overline{\text{KI3}}$
Pull-up transistor
PD10_6 bit
P106/$\overline{\text{KI2}}$
Interrupt control circuit
Key input interrupt request
Pull-up transistor
PD10_5 bit
P105/$\overline{\text{KI1}}$
Pull-up transistor
PD10_4 bit
P104/$\overline{\text{KI0}}$

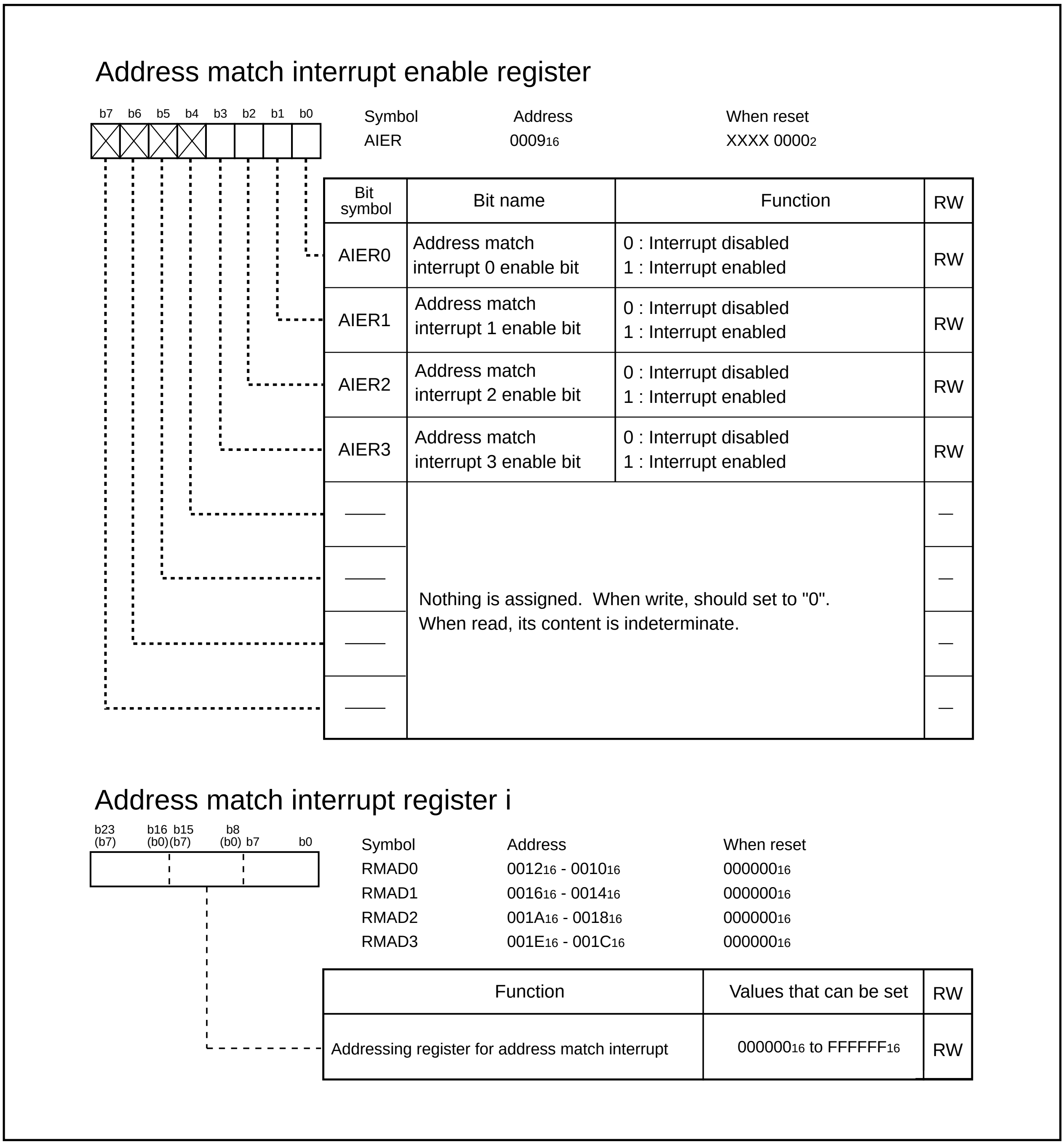
Figure 1.9.11. Key Input Interrupt

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.



Address match interrupt enable register

Symbol	Address	When reset
AIER	0009_{16}	XXXX 0000_2

Bit symbol	Bit name	Function	RW
AIER0	Address match interrupt 0 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
AIER1	Address match interrupt 1 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
AIER2	Address match interrupt 2 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
AIER3	Address match interrupt 3 enable bit	0 : Interrupt disabled 1 : Interrupt enabled	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
——			—
——			—
——			—

Address match interrupt register i

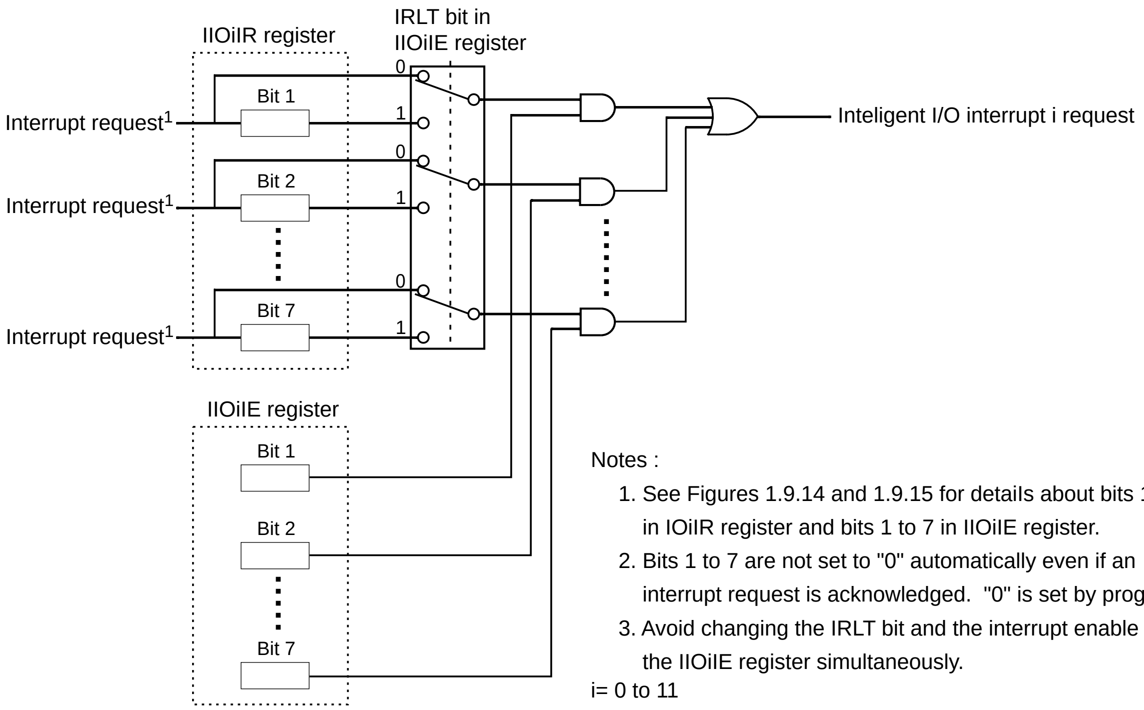
Symbol	Address	When reset
RMAD0	0012_{16} - 0010_{16}	000000_{16}
RMAD1	0016_{16} - 0014_{16}	000000_{16}
RMAD2	$001A_{16}$ - 0018_{16}	000000_{16}
RMAD3	$001E_{16}$ - $001C_{16}$	000000_{16}

Function	Values that can be set	RW
Addressing register for address match interrupt	000000_{16} to $FFFFFF_{16}$	RW

Figure 1.9.12. AIER Register and RMAD0 to RMAD3 Registers

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.



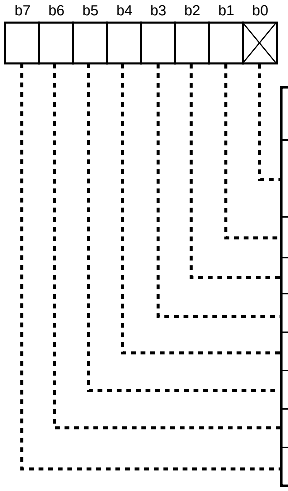
Notes :
1. See Figures 1.9.14 and 1.9.15 for details about bits 1 to 7 in IOiIR register and bits 1 to 7 in IIOiIE register.
2. Bits 1 to 7 are not set to "0" automatically even if an interrupt request is acknowledged. "0" is set by program.
3. Avoid changing the IRLT bit and the interrupt enable bit in the IIOiIE register simultaneously.

i= 0 to 11

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.

Interrupt request register



Symbol: IIO0IR to IIO11IR
Address: See below
When reset: 0000 000X$_2$

Bit symbol	Function	RW
	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.	–
(Note 1)	0 : No interrupt request 1 : Interrupt request (Note 2)	RW
(Note 1)		RW
(Note 1)		RW
(Note 1)		RW
(Note 1)		RW
(Note 1)		RW
(Note 1)		RW

Notes :
1. See the table below about the bit symbols.
2. Only "0" can be set (nothing is changed even if "1" is set).

Bit symbols for the interrupt request register

Symbol	Address	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
IIO0IR	00A0$_{16}$	-	-	SIO0RR	G0RIR	-	PO13R	TM02R	-
IIO1IR	00A1$_{16}$	-	-	SIO0TR	G0TOR	-	PO14R	TM00R/PO00R	-
IIO2IR	00A2$_{16}$	-	-	SIO1RR	G1RIR	-	TM12R/PO12R	-	-
IIO3IR	00A3$_{16}$	-	-	SIO1TR	G1TOR	PO27R	PO10R	TM03R	-
IIO4IR	00A4$_{16}$	SRT0R	SRT1R	-	BT1R	PO32R	TM17R/PO17R	TM04R/PO04R	-
IIO5IR	00A5$_{16}$	-	-	-	SIO2RR	PO33R	PO21R	TM05R/PO05R	-
IIO6IR	00A6$_{16}$	-	-	-	SIO2TR	PO34R	PO20R	TM06R	-
IIO7IR	00A7$_{16}$	IE0R	-	-	BT0R	PO35R	PO22R	TM07R	-
IIO8IR	00A8$_{16}$	IE1R	IE2R	-	BT2R	PO36R	PO23R	TM11R/PO11R	-
IIO9IR	00A9$_{16}$	CAN0R	-	-	SIO3RR	PO31R	PO24R	PO15R	-
IIO10IR	00AA$_{16}$	CAN1R	-	-	SIO3TR	PO30R	PO25R	TM16R/PO16R	-
IIO11IR	00AB$_{16}$	CAN2R	-	-	BT3R	PO37R	PO26R	TM01R/PO01R	-

BTiR	: Intelligent I/O group i base timer interrupt request (i=0 to 3)
TMijR	: Intelligent I/O group i time measurement j interrupt request (j=0 to 7)
POijR	: Intelligent I/O group i waveform generation function j interrupt request
SIOiRR/SIOiTR	: Intelligent I/O group i communication function interrupt request (RR:receive, TR:transmit)
GiRIR/GiTOR	: Intelligent I/O group i HDLC data processing function interrupt request (RR:input to receive, TOR:input to transmit)
SRTiR	: Intelligent I/O group i special communication function interrupt request (i=0,1)
IER	: Intelligent I/O group 2 IE Bus communication function interrupt request
HDLCR	: Intelligent I/O group 3 HDLC communication function interrupt request
CANkR	: CAN communication function interrupt request (k = 0 to 2)
-	: Nothing is assigned in this bit. Should set to "0".

Figure 1.9.14. IIO0IR to IIO11IR Registers

Interrupt enable register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol: IIO0IE to IIO11IE
Address: See below
When reset: $0000\ 0000_2$

Bit	Bit symbol	Bit name	Function	RW
b0	IRLT	Interrupt request select bit	0 : Interrupt request is used for DMAC, DMAC II 1 : Interrupt request is used for interrupt	RW
b1	(Note 1)	———	0 : Interrupt disabled by bit 1 in IIOiIR register 1 : Interrupt enabled by bit 1 in IIOiIR register	RW
b2	(Note 1)	———	0 : Interrupt disabled by bit 2 in IIOiIR register 1 : Interrupt enabled by bit 2 in IIOiIR register	RW
b3	(Note 1)	———	0 : Interrupt disabled by bit 3 in IIOiIR register 1 : Interrupt enabled by bit 3 in IIOiIR register	RW
b4	(Note 1)	———	0 : Interrupt disabled by bit 4 in IIOiIR register 1 : Interrupt enabled by bit 4 in IIOiIR register	RW
b5	(Note 1)	———	0 : Interrupt disabled by bit 5 in IIOiIR register 1 : Interrupt enabled by bit 5 in IIOiIR register	RW
b6	(Note 1)	———	0 : Interrupt disabled by bit 6 in IIOiIR register 1 : Interrupt enabled by bit 6 in IIOiIR register	RW
b7	(Note 1)	———	0 : Interrupt disabled by bit 7 in IIOiIR register 1 : Interrupt enabled by bit 7 in IIOiIR register	RW

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	$00B0_{16}$	-	-	SIO0RE	G0RIE	-	PO13E	TM02E	IRLT
IIO1IE	$00B1_{16}$	-	-	SIO0TE	G0TOE	-	PO14E	TM00E/PO00E	IRLT
IIO2IE	$00B2_{16}$	-	-	SIO1RE	G1RIE	-	TM12E/PO12E	-	IRLT
IIO3IE	$00B3_{16}$	-	-	SIO1TE	G1TOE	PO27E	PO10E	TM03E	IRLT
IIO4IE	$00B4_{16}$	SRT0E	SRT1E	-	BT1E	PO32E	TM17E/PO17E	TM04E/PO04E	IRLT
IIO5IE	$00B5_{16}$	-	-	-	SIO2RE	PO33E	PO21E	TM05E/PO05E	IRLT
IIO6IE	$00B6_{16}$	-	-	-	SIO2TE	PO34E	PO20E	TM06E	IRLT
IIO7IE	$00B7_{16}$	IE0E	-	-	BT0E	PO35E	PO22E	TM07E	IRLT
IIO8IE	$00B8_{16}$	IE1E	IE2E	-	BT2E	PO36E	PO23E	TM11E/PO11E	IRLT
IIO9IE	$00B9_{16}$	CAN0E	-	-	SIO3RE	PO31E	PO24E	PO15E	IRLT
IIO10IE	$00BA_{16}$	CAN1E	-	-	SIO3TE	PO30E	PO25E	TM16E/PO16E	IRLT
IIO11IE	$00BB_{16}$	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 "000000_{16}" 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 V_{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. 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.

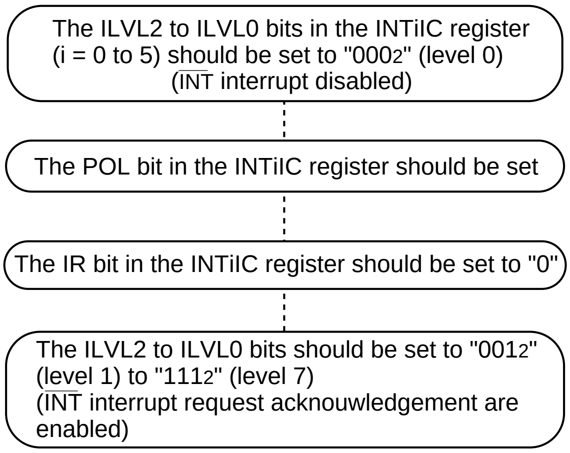
(3) $\overline{\text{INT}}$ Interrupt

- Edge sense

  At least 250ns should be input as "L" or "H" width to the $\overline{\text{INTi}}$ 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 $\overline{\text{INTi}}$ pins. (At least 234ns when X_{IN} = 30MHz and no division.)
- When a polarity of the $\overline{\text{INTi}}$ 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.



The ILVL2 to ILVL0 bits in the INTiIC register (i = 0 to 5) should be set to "000_2" (level 0) ($\overline{\text{INT}}$ interrupt disabled)

The POL bit in the INTiIC register should be set

The IR bit in the INTiIC register should be set to "0"

The ILVL2 to ILVL0 bits should be set to "001_2" (level 1) to "111_2" (level 7) ($\overline{\text{INT}}$ interrupt request acknouwledgement are enabled)

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

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 IIOiIR Register (i = 0 to 15)

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

AND, BCLR

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

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

When the sub clock is selected as the CPU clock

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

For example, with a 30MHz CPU clock and a prescalar 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 starts 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.

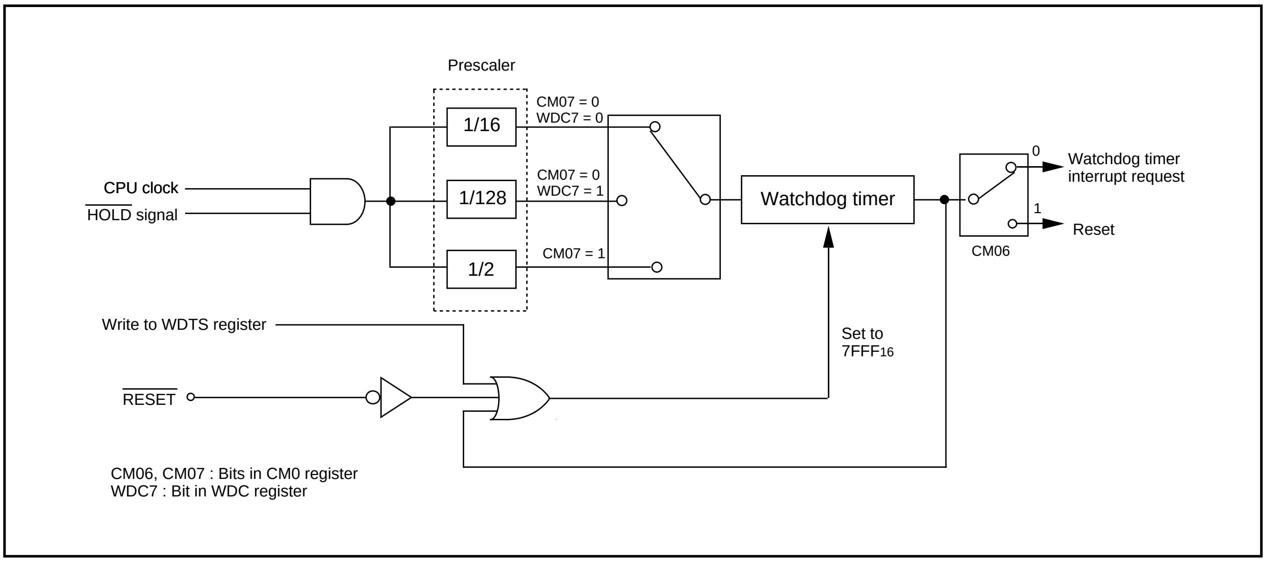
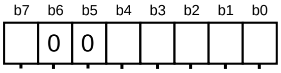


Figure 1.10.1. Watchdog Timer Block Diagram

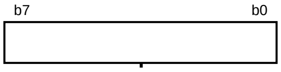
Watchdog timer control register



Symbol	Address	When reset
WDC	$000F_{16}$	$000X\ XXXX_2$

Bit symbol	Bit name	Function	RW
——	High-order bit of watchdog timer		RO
——			RO
——			RO
——			RO
——			RO
——	Reserved bit	Should set to "0"	RW
——			RW
WDC7	Prescaler select bit	0 : Divide-by-16 1 : Divide-by-128	RW

Watchdog timer start register

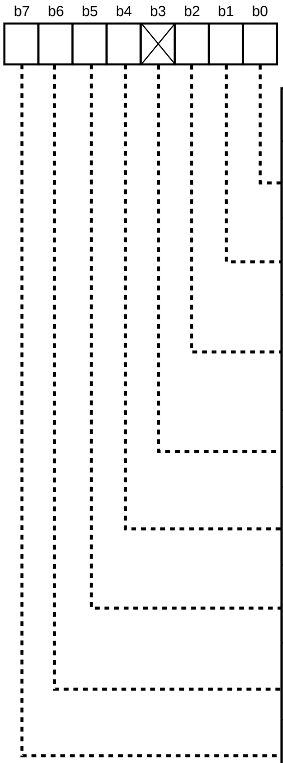


Symbol	Address	When reset
WDTS	$000E_{16}$	Indeterminate

Function	RW
The watchdog timer is initialized to start counting by a write instruction to the WDTS register. Default value of the watchdog timer is always set to "$7FFF_{16}$" regardless of a value written.	WO

Figure 1.10.2. WDC Register and WDTS Register

System clock control register 0[1]



Symbol	Address	When reset
CM0	0006_{16}	$0000\ X000_2$

Bit symbol	Bit name	Function	RW
CM00	Clock output function select bit[2]	b1 b0 0 0 : I/O port $P5_3$ 0 1 : f_C output 1 0 : f_8 output 1 1 : f_{32} output	RW
CM01			RW
CM02	In WAIT peripheral function clock stop bit	0 : Peripheral clock does not stop in wait mode 1 : Peripheral clock stops in wait mode[3]	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
CM04	Port X_C switch bit	0 : I/O port function 1 : X_{CIN}-X_{COUT} oscilation function[4]	RW
CM05	Main clock (X_{IN}-X_{OUT}) stop bit[5]	0 : Main clock oscillation 1 : Main clock stop[6]	RW
CM06	Watchdog timer function select bit	0 : Watchdog timer interrupt 1 : Reset[7]	RW
CM07	System clock select bit[8]	0 : Select X_{IN} - X_{OUT} 1 : Select X_{CIN} - X_{COUT}	RW

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 "00_2". When the PM15 to PM14 bits in the PM1 register is set to "01_2" (ALE output to $P5_3$), the CM01 to CM00 bits should be set to "00_2". 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 "00_2" to output "L" from port $P5_3$ (port $P5_3$ does not function as an I/O port).
3. f_{C32} 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" (X_{CIN}-X_{COUT} oscillation), the PD8_7 to PD8_6 bits should be set to "00_2" (with port $P8_7$ and $P8_6$ 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), X_{OUT} is set to "H". Also, an internal feedback resistance remains ON. X_{IN} is pulled up to X_{OUT} ("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 "01000_2" (divide-by-8 mode). In ring oscillation mode, the MCD register is not divided by eight even if X_{IN}-X_{out} 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.

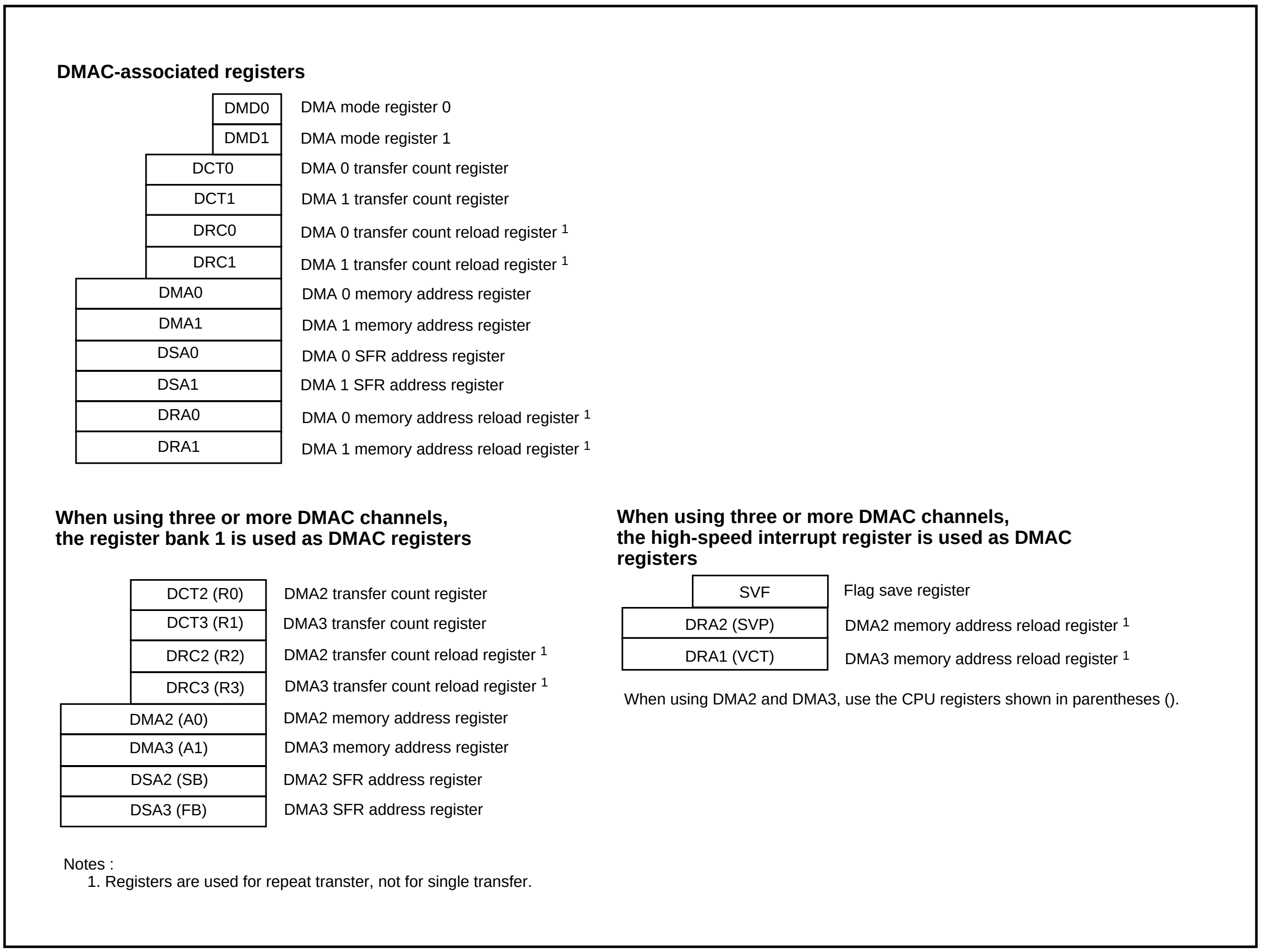
Figure 1.10.3. CM0 Register

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.



DMAC-associated registers

DMD0	DMA mode register 0
DMD1	DMA mode register 1
DCT0	DMA 0 transfer count register
DCT1	DMA 1 transfer count register
DRC0	DMA 0 transfer count reload register [1]
DRC1	DMA 1 transfer count reload register [1]
DMA0	DMA 0 memory address register
DMA1	DMA 1 memory address register
DSA0	DMA 0 SFR address register
DSA1	DMA 1 SFR address register
DRA0	DMA 0 memory address reload register [1]
DRA1	DMA 1 memory address reload register [1]

When using three or more DMAC channels, the register bank 1 is used as DMAC registers

DCT2 (R0)	DMA2 transfer count register
DCT3 (R1)	DMA3 transfer count register
DRC2 (R2)	DMA2 transfer count reload register [1]
DRC3 (R3)	DMA3 transfer count reload register [1]
DMA2 (A0)	DMA2 memory address register
DMA3 (A1)	DMA3 memory address register
DSA2 (SB)	DMA2 SFR address register
DSA3 (FB)	DMA3 SFR address register

When using three or more DMAC channels, the high-speed interrupt register is used as DMAC registers

SVF	Flag save register
DRA2 (SVP)	DMA2 memory address reload register [1]
DRA1 (VCT)	DMA3 memory address reload register [1]

When using DMA2 and DMA3, use the CPU registers shown in parentheses ().

Notes :
1. Registers are used for repeat transter, not for single transfer.

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.

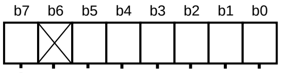
Table 1.11.1. DMAC Specifications

Item		Specification
Channels		4 channels (cycle-steal method)
Transfer memory space		• From any address in a 16M-byte space to a fixed address in a 16M-bytes space • From a fixed address in a 16M-byte space to any address in a 16M-bytes space
Maximum bytes transferred		128K-byte (with 16-bit transfer) or 64K-byte (with 8-bit transfer)
DMA request factors[1]		Falling edge or both edge of inputs to $\overline{\text{INT0}}$ to $\overline{\text{INT3}}$ 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_{16}"
	Repeat transfer	When DCTi register is set to "0000_{16}", 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_{16}" from "0001_{16}"
DMA startup	Single transfer	Transfer starts when the DCTi register is set to "0001_{16}" or more and the DMA is requested after the MDi1 to MD0 bits in the DMDj register (j = 0 to 1) are set to "01_2" (single transfer)
	Repeat transfer	Transfer starts when DCTi register is set to "0001_{16}" and more and DMA is requested after "11_2" (repeat transfer) is written to the MDi1 to MD0 bits
DMA shutdown	Single transfer	When the MDi1 to MD0 bits is set to "00_2" (DMA inhibited) and DCTi register is set to "0000_{16}" (number of times for DMA transfer 0) by DMA transfer or write
	Repeat transfer	The MDi1 to MD0 bits is set to "00_2" and the DCTi register is set to "0000_{16}" with the DRCi register set to "0000_{16}"
Reload timing to DCTi or DMAi register		When the DCTi register is set to "0000_{16}" from "0001_{16}" in repeat transfer mode
DMA transfer cycles		Minimum 3 cycles

Notes :

1.No DMA transfer affects an interrupt.

DMAi request factor select register (i=0 to 3)



Symbol	Address	When reset
DM0SL to DM3SL	0378_{16}, 0379_{16}, $037A_{16}$, $037B_{16}$	0X00 0000_2

Bit symbol	Bit name	Function	RW
DSEL0	DMA request cause select bit[1]	See Table 1.11.2 about DMiSL register (i = 0 to 3) function	RW
DSEL1			RW
DSEL2			RW
DSEL3			RW
DSEL4			RW
DSR	Software DMA request bit[2]	When a software trigger is selected, a DMA request is generated by setting this bit to "1" (When read, its content is always "0")	RW
——	Nothing is assigned. When write, should set "0". When read, its content is indeterminate.		–
DRQ	DMA request bit[2, 3]	0 : Not requested 1 : Requested	RW

Notes :

1. The MDi1 to MDi0 bits in the DMA0 or DMD1 register should be set to "00_2" (DMA inhibit) before changing the DSEL4 to DSEL0 bits. Also, the DRQ bit should be set to "1" simultaneously when changing the DSEL4 to DSEL0 bits.
   e.g MOV.B #083h, DMiSL ; Set timer A0
2. When setting the DSR bit to "1", the DRQ should be set to "1" simultaneously.
   e.g OR.B #0A0h, DMiSL

Figure 1.11.2. DM0SL to DM3SL Registers

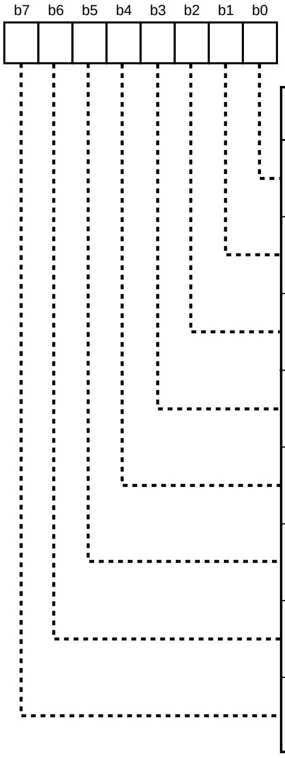
Table 1.11.2. DMiSL Register (i = 0 to 3) Function

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[1]	(Note 2)
0 0 0 1 0	Both edges of INT0	Both edges of INT1	Both edges of INT2	Both edges of INT3[1]	(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[3]				
1 0 0 0 0	UART1 transmit interrupt request				
1 0 0 0 1	UART1 receive or ACK interrupt request[3]				
1 0 0 1 0	UART2 transmit interrupt request				
1 0 0 1 1	UART2 receive or ACK interrupt request[3]				
1 0 1 0 0	UART3 transmit interrupt request				
1 0 1 0 1	UART3 receive or ACK interrupt request[3]				
1 0 1 1 0	UART4 transmit interrupt request				
1 0 1 1 1	UART4 receive or ACK interrupt request[3]				
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[4]	
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[5]	
1 1 0 1 1	Intelligent I/O interrupt 2 request	Intelligent I/O interrupt 9 request[4]	Intelligent I/O interrupt 4 request	Intelligent I/O interrupt 11 request[6]	
1 1 1 0 0	Intelligent I/O interrupt 3 request	Intelligent I/O interrupt 10 request[5]	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[6]	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 $\overline{\text{INT3}}$ pin is data bus in the memory expansion mode or microprocessor mode, DMA3 request cannot be generated by an $\overline{\text{INT3}}$ pin interrupt.
2. The falling edge and both edge of input to the $\overline{\text{INTj}}$ pin (j = 0 to 3) cause a DMA request. An $\overline{\text{INT}}$ interrupt (the POL bit in the INTiIC 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 UARTj 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.

DMA mode register 0[1]



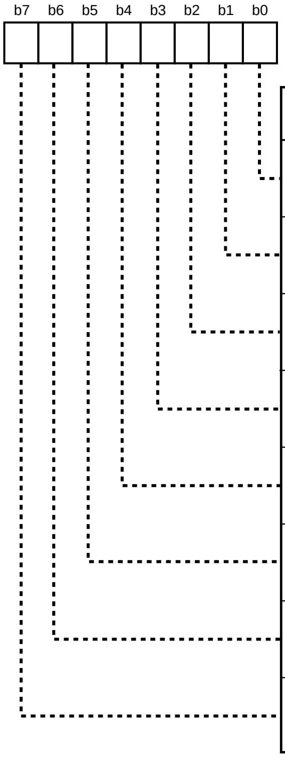
Symbol	Address	When reset
DMD0	CPU internal register	$0000\ 0000_2$

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

Notes :

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

DMA mode register 1[1]



Symbol	Address	When reset
DMD1	CPU internal register	$0000\ 0000_2$

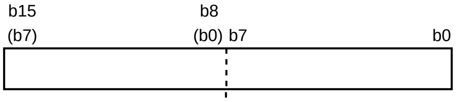
Bit symbol	Bit name	Function	RW
MD20	Channel 2 transfer mode select bit	b1 b0 0 0 : DMA inhibited 0 1 : Single transfer 1 0 : Avoid this setting 1 1 : Repeat transfer	RW
MD21			RW
BW2	Channel 2 transfer unit select bit	0 : 8 bits 1 : 16 bits	RW
RW2	Channel 2 transfer direction select bit	0 : Fixed address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

Notes :

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

Figure 1.11.3. DMD0 Register, DMD1 Register

DMAi transfer count register (i = 0 to 3)



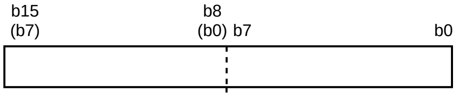
Symbol	Address	When reset
DCT0[2]	(CPU internal register)	XXXX$_{16}$
DCT1[2]	(CPU internal register)	XXXX$_{16}$
DCT2 (bank 1;R0)[3]	(CPU internal register)	0000$_{16}$
DCT3 (bank 1;R1)[4]	(CPU internal register)	0000$_{16}$

Function	Setting range	RW
Set the number of transfers	0000$_{16}$ to FFFF$_{16}$ [1]	RW

Notes :

1. When setting this register to "0000$_{16}$", no data transfer occurs regardless of a DMA request.
2. The LDC instruction should be used to set this register.
3. When setting the DCT2 register, the B flag in the FLG register should be set to "1" (register bank 1) to set the R0 register. The MOV instruction should be used to set this register.
4. When setting the DCT3 register, the B flag should be set to "1" to set the R1 register. The MOV instruction should be used to set this register.

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



Symbol	Address	When reset
DRC0[1]	(CPU internal register)	XXXX$_{16}$
DRC1[1]	(CPU internal register)	XXXX$_{16}$
DRC2 (bank 1;R2)[2]	(CPU internal register)	0000$_{16}$
DRC3 (bank 1;R3)[3]	(CPU internal register)	0000$_{16}$

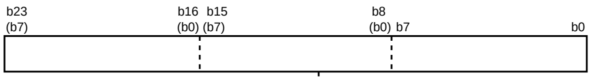
Function	Setting range	RW
Set the number of transfers	0000$_{16}$ to FFFF$_{16}$	RW

Notes :

1. The LDC instruction should be used to set this register.
2. When setting the DRC2 register, the B flag in the FLG register should be set to "1" (register bank 1) to set the R2 register. The MOV instruction should be used to set this register.
3. When setting the DRC3 register, the B flag should be set to "1" to set the R3 register. The MOV should be used to set this register.

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

DMAi memory address register (i = 0 to 3)

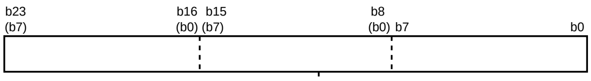


Symbol	Address	When reset
DMA0[2]	(CPU internal register)	$XXXXXX_{16}$
DMA1[2]	(CPU internal register)	$XXXXXX_{16}$
DMA2 (bank 1;A0)[3]	(CPU internal register)	000000_{16}
DMA3 (bank 1;A1)[4]	(CPU internal register)	000000_{16}

Function	Setting range	RW
Set a source or destination memory address[1]	000000_{16} to $FFFFFF_{16}$ (16M-byte space)	RW

Notes :
1. When the RWk bit (k = 0 to 3) in the DMDj register (j = 0,1) is set to "0" (fixed address to memory), an address for this register becomes a destination address.
   When the RWk bit is set to "1" (memory to fixed address), an address for this register becomes a source address.
2. The LDC instruction should be used to set this register.
3. When setting the DMA2 register, the B flag in the FLG register should be set to "1" (register bank 1) to set the A0 register. The MOV instruction should be used to set this register.
4. When setting the DMA3 register, the B flag should be set to "1" to set the A1 register. The MOV instruction should be used to set this register.

DMAi SFR address register (i = 0 to 3)

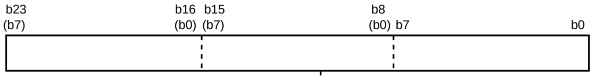


Symbol	Address	When reset
DSA0[2]	(CPU internal register)	$XXXXXX_{16}$
DSA1[2]	(CPU internal register)	$XXXXXX_{16}$
DSA2 (bank 1;SB)[3]	(CPU internal register)	000000_{16}
DSA3 (bank 1;FB)[4]	(CPU internal register)	000000_{16}

Function	Setting range	RW
Set a source or destination fixed address[1]	000000_{16} to $FFFFFF_{16}$ (16M-byte space)	RW

Notes :
1. When the RWk (k=0 to 3) bit in the DMDj (j=0 to 3) register is set to "0" (fixed address to memory), an address for this register becomes a source fixed address.
   When the RWk bit is set to "1" (memory to fixed address), an address for this register becomes a destination address.
2. The LDC instruction should be used to set this register.
3. When setting the DSA2 register, the B flag in the FLG register should be set to "1" (register bank 1) to set the SB register. The MOV instruction should be used to set this register.
4. When setting the DSA3 register, the B flag should be set to "1" to set the FB register. The MOV instruction should be used to set this register.

DMAi memory address reload register (i = 0 to 3)[1]



Symbol	Address	When reset
DRA0	(CPU internal register)	$XXXXXX_{16}$
DRA1	(CPU internal register)	$XXXXXX_{16}$
DRA2 (bank 1;SVP)[2]	(CPU internal register)	$XXXXXX_{16}$
DRA3 (bank 1;VCT)[3]	(CPU internal register)	$XXXXXX_{16}$

Function	Setting range	RW
Set a source or destination memory address	000000_{16} to $FFFFFF_{16}$ (16M-byte space)	RW

Notes :
1. The LDC instruction should be used to set this register.
2. When setting for DRA2, the SVP register should be set.
3. When setting for DRA3, the VCT register should be set.

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

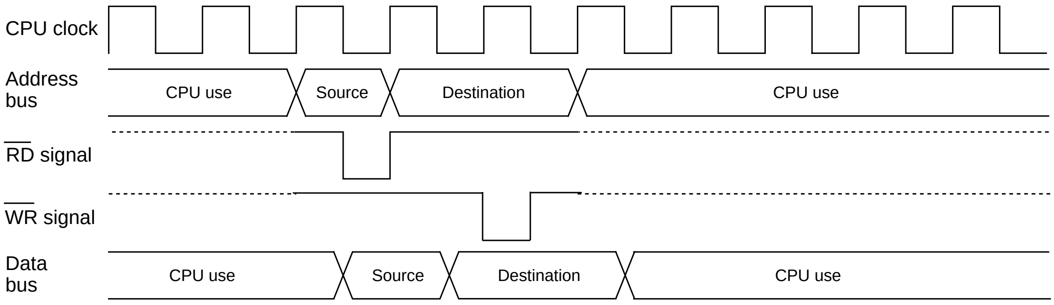
When a SFR space or a memory space with a software wait is accessed, the number of cycles is incremented by software wait(s).

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 illustllted 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.

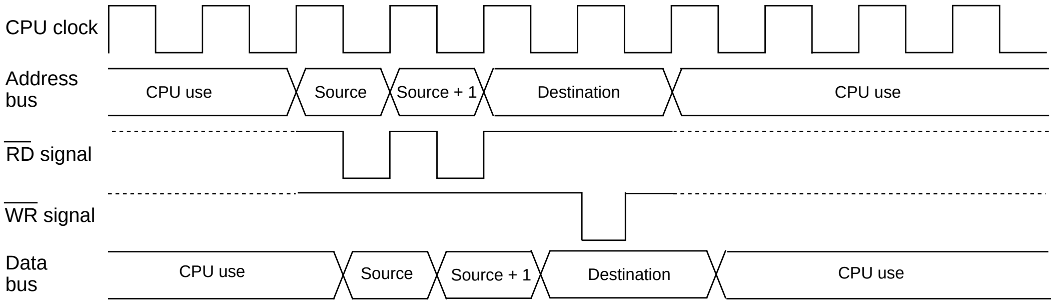
4. Effect of $\overline{\text{RDY}}$ Signal

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.

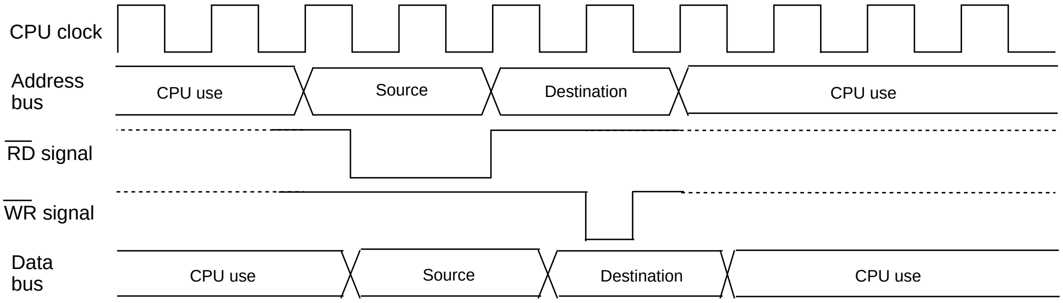
(1) When 8-bit data is transferred or when 16-bit data is transferred with a 16-bit data bus and even source address



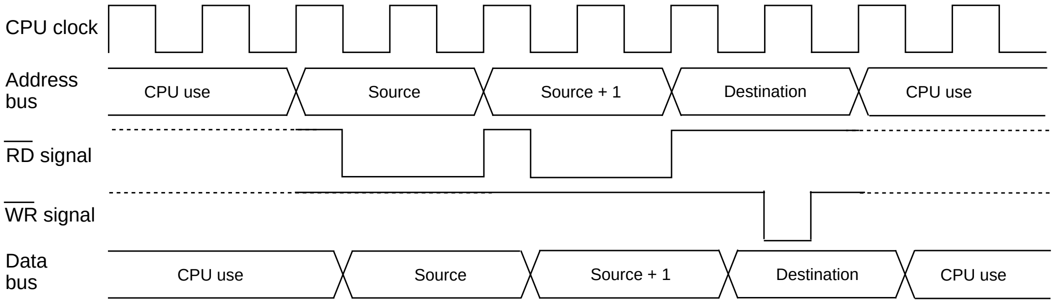
(2) When 16-bit data is transferred with odd source address or when 16-bit data is transferred with 8-bit bus to access a source address



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



(4) When one wait is inserted into a source read under the conditions in (2)



Notes :

1. The above applies to 2 cycles (1 bus cycle) for a destination write cycle.
   A destination address has the same timing changes as a source address has, under each condition.

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.

Transfer cycles per transfer unit = Number of read cycle(s) x j + Number of write cycle(s) x 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 illustllates. 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.

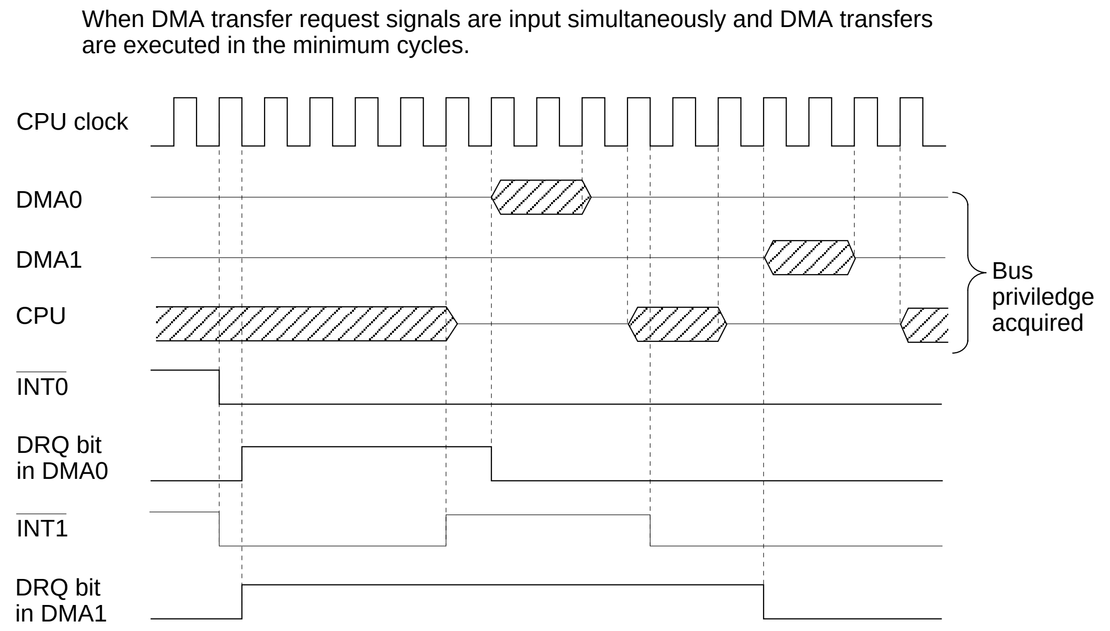

When DMA transfer request signals are input simultaneously and DMA transfers are executed in the minimum cycles.
CPU clock
DMA0
DMA1
CPU
INT0
DRQ bit in DMA0
INT1
DRQ bit in DMA1
Bus priviledge acquired

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 "00_2" (DMA disabled). Then the MDi1 to MDi0 bits should be set to "01_2" (single transfer) or "11_2" (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[1], 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 "00_2" or the DCTi register is set to "0000_{16}" (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 "01_2" or "11_2".

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 $\overline{\text{INT}}$ interrupt for DMA request factor), avoid setting the DCTi register to "1".

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 "111_2" (level 7)
Transfer data	• Memory -> memory (memory-to-memory transfer) • Immediate data -> memory (immediate data transfer) • Memory (or immediate data) + memory -> memory (arithmetic transfer)
Transfer block	8 or 16 bits
Transfer space	64-Kbyte space at addresses 0000_{16} to $0FFFF_{16}$[1]
Transfer direction	Fixed or forward address Can be selected for either a source address or destination address.
Transfer mode	• Single transfer • Burst transfer
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 $0FFFF_{16}$, it is transferred to $0FFFF_{16}$ and 10000_{16}. When to a source address as $0FFFF_{16}$, data is transferred as well.

DMAC II Settings

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

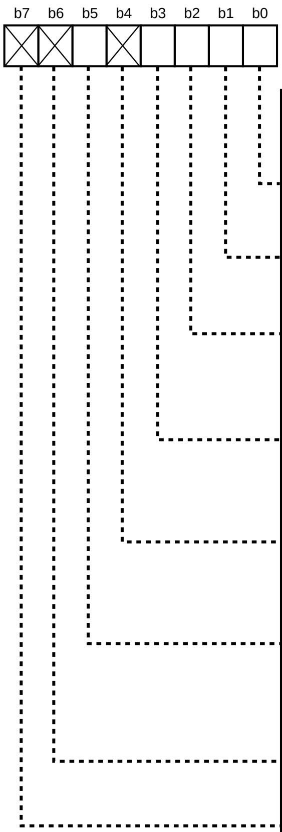
- RLVL register
- DMAC II Index
- Interrupt control register for the peripheral function causing DMAC II request
- Relocatable vector table for the peripheral function causing DMAC II request
- With the intelligent I/O or CAN interrupt, set the IRLT bit of IIOiIE register (i = 0 to 11).
  Refer to the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

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 "111$_2$" (level 7).

Figure 1.12.1 shows the RLVL register.

Exit priority register



Symbol	Address	When reset
RLVL	009F$_{16}$	XX0X 0000$_2$

Bit symbol	Bit name	Function	RW
RLVL0	Interrupt priority set bits to exit stop/wait mode[1]	b2 b1 b0 0 0 0 : Level 0 0 0 1 : Level 1 0 1 0 : Level 2 0 1 1 : Level 3 1 0 0 : Level 4 1 0 1 : Level 5 1 1 0 : Level 6 1 1 1 : Level 7	RW
RLVL1			RW
RLVL2			RW
FSIT	High-speed interrupt set bit[2]	0: Interrupt priority level 7 is used for normal interrupt 1: Interrupt priority level 7 is used for high-speed interrupt	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
DMA II	DMAC II select bit[3]	0: Interrupt priority level 7 is used for interrupt 1: Interrupt priority level 7 is used for DMAC II transfer	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–

Notes :

1. Stop or wait mode is exited when a requested interrupt priority level is higher than a level set in the RLVL2 to RLVL0 bits. The RLVL2 to RLVL0 bits should be set the same value as IPL in the FLG register.
2. When setting the FSIT bit to "1" (high-speed interrupt), interrupt level 7 as the highest interrupt priority level becomes high-speed interrupt. Only one interrupt should be set to level 7 in interrupt priority level to set the DMA II bit to "0" (normal interrupt).
3. Avoid setting the DMA II bit to "0" (normal interrupt) again once setting it to "1" (DMAC II transfer). When setting the DMA II bit to "1", the FSIT bit should be set to "0" (normal interrupt). DMAC II cannot be used with high-speed interrupt simultaneously. The I flag and IPL do not affect DMAC II transfer.

Figure 1.12.1. RLVL Register

(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.

Memory-to-memory transfer, Immediate transfer, Arithmetic transfer

16 bits

DMAC II Index starting address (BASE)	Transfer mode	(MOD)
BASE + 2	Transfer counter	(COUNT)
BASE + 4	Transfer source address (or imm data)	(SADR)
BASE + 6	Operation address[1]	(OADR)
BASE + 8	Transfer destination address	(DADR)
BASE + 10	Chained transfer address[2]	(CADR0)
BASE + 12	Chained transfer address[2]	(CADR1)
BASE + 14	End-of-transfer interrupt address[3]	(IADR0)
BASE + 16	End-of-transfer interrupt address[3]	(IADR1)

Multiple transfer

16 bits

BASE	Transfer mode	(MOD)
BASE + 2	Transfer counter	(COUNT)
BASE + 4	Transfer source address	(SADR1)
BASE + 6	Transfer destination address	(DADR1)
BASE + 8	Transfer source address	(SADR2)
BASE + 10	Transfer destination address	(DADR2)
	≀	
BASE + 28	Transfer source address	(SADR7)
BASE + 30	Transfer destination address	(DADR7)

Notes :
1. This data is not required when not using the arithmetic transfer function.
2. This data is not required when not using the chained transfer function.
3. This data is not required when not using an end-of-transfer interrupt.

DMAC II index should be located on the RAM. Necessary data is set front-aligned. For example, if not using the arithmetic transfer function, a destination address should be set to BASE+6. (See Table 1.12.2.)
A starting address of DMAC II index should be set in an interrupt vector for a peripheral function interrupt causing a DMAC II request.

Figure 1.12.2. DMAC II Index

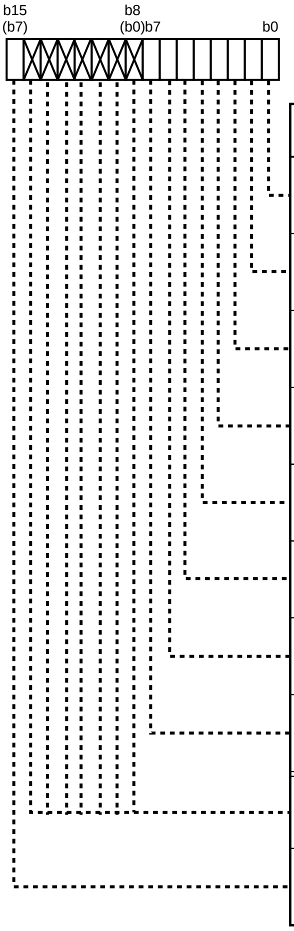
The detail for DMAC II index is below. These parameters should be set in the specified order listed on Table 1.12.2, according to DMAC II transfer mode.

- 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.

Table 1.12.2. DMAC II Index Configuration in Transfer Mode

Transmit data	Memory-to-memory transfer /immediate data transfer				Arithmetic transfer				Multiple transfer
Chained transfer	Not use	Use	Not use	Use	Not use	Use	Not use	Use	Cannot use
Interrupt at end of transfer	Not use	Not use	Use	Use	Not use	Not use	Use	Use	Cannot use
DMAC II index	MOD COUNT SADR DADR 8 bytes	MOD COUNT SADR DADR CADR0 CADR1 12 bytes	MOD COUNT SADR DADR IADR0 IADR1 12 bytes	MOD COUNT SADR DADR CADR0 CADR1 IADR0 IADR1 16 bytes	MOD COUNT SADR OADR DADR 10 bytes	MOD COUNT SADR OADR DADR CADR0 CADR1 14 bytes	MOD COUNT SADR OADR DADR IADR0 IADR1 14 bytes	MOD COUNT SADR OADR DADR CADR0 CADR1 IADR0 IADR1 18 bytes	MOD COUNT SADR1 DADR1 SADRi DADRi i=1 to 7 Max 32 bytes (when i=7)

Transfer mode(MOD)[1]



Bit symbol	Bit name	Function (MULT=0)	Function (MULT=1)	RW
SIZE	Transfer unit select bit	0: 8 bits 1: 16 bits		RW
IMM	Transfer data select bit	0: Immediate 1: Memory	Should set to "1"	RW
UPDS	Transfer source direction select bit	0: Fixed address 1: Forward address		RW
UPDD	Transfer destination direction select bit	0: Fixed address 1: Forward address		RW
OPER/ CNT0[2]	Arithmetic transfer function select bit	0: Not used 1: Used	b6 b5 b4 0 0 0: Avoid this setting 0 0 1: Once 0 1 0: Twice : : 1 1 0: 6 times 1 1 1: 7 times	RW
BRST/ CNT1[2]	Burst transfer select bit	0: Single transfer 1: Burst transfer		RW
INTE/ CNT2[2]	End of transfer interrupt select bit	0: No interrupt used 1: Interrupt used		RW
CHAIN	Chained transfer select bit	0: No chained-transfer used 1: Chained-transfer used	Should set to "0"	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.			—
MULT	Multiple transfer select bit	0: No multiple transfer	1: Multiple transfer	RW

Notes :
1. The MOD register should be located on the RAM.
2. When setting the MULT bit to "0" (no muliple transfer), bits 4 to 6 shift to the OPER,BRST, INTE bits. When setting the MULT bit to "1" (muliple transfer), bits 4 to 6 shift to the CNT0 to CNT2 bits.

Figure 1.12.3. MOD

(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 "111_2" (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 IIOiIE Register (i=0 to 11)

When using the intelligent I/O interrupt or CAN interrupt to activate DMAC II, the IRLT bit in the IIOiIE 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 "111_2" (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 "111_2" (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_{16} to $0FFFF_{16}$) 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 or16-bit data are added together and the result is transferred to any memory location in a 64K-byte space.

When to transferring a 16-bit data to a destination address as $0FFFF_{16}$, it is transferred to $0FFFF_{16}$ and 10000_{16}. 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 $0FFFF_{16}$ as a result of address incrementation, a source or destination address is back to address 00000_{16} to increment. A source or destination address should be maintained below.

(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_{16}".

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_2" (once) to "111_2" (7 times). Avoid setting the CNT2 to CNT0 bits to "000_2".

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.

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.

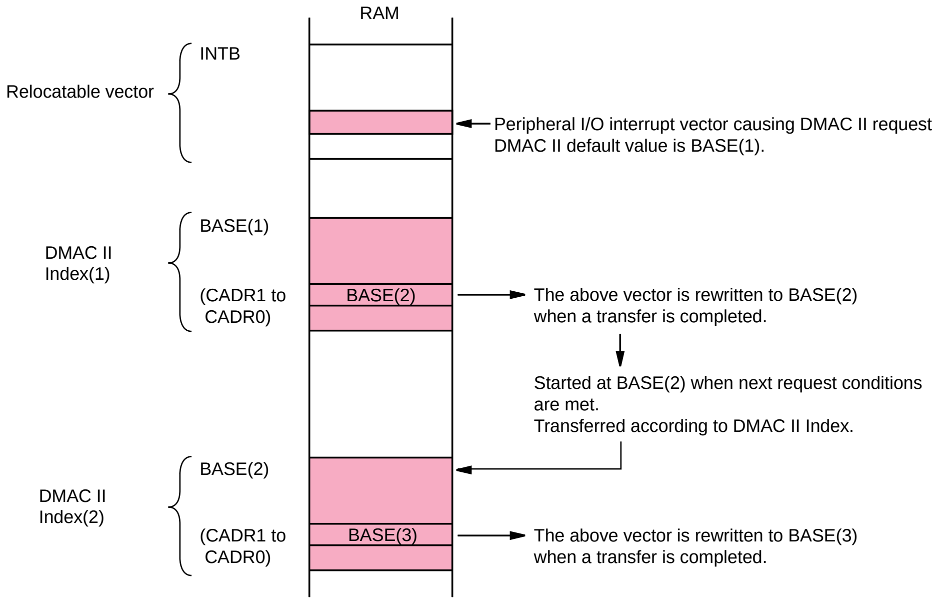

RAM
INTB
Relocatable vector
Peripheral I/O interrupt vector causing DMAC II request
DMAC II default value is BASE(1).
BASE(1)
DMAC II
Index(1)
(CADR1 to
CADR0)
BASE(2)
The above vector is rewritten to BASE(2)
when a transfer is completed.
Started at BASE(2) when next request conditions
are met.
Transferred according to DMAC II Index.
BASE(2)
DMAC II
Index(2)
(CADR1 to
CADR0)
BASE(3)
The above vector is rewritten to BASE(3)
when a transfer is completed.

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."

Execution Time

DMAC II execution cycle is calculated by the equation below.

Other than multiple transfers, t = 6 + (26 + a + b + c + d) X m + (4 + e) X n (cycles)
Multiple transfers, t = 21 + (11 + b + c) X 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 (arithmetic 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 = 0 (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.

When using an end-of-transfer interrupt (transfer counter = 2) after performing a memory to memory transfer, source address relocatable, destination address fixed, single transfer and double transfer with a chained transfer function unselected

a=-1 b=1 c=1 d=0 e=0 m=1

First DMAC II transfer t=6+27x1+4x1=37 cycles
Second DMAC II transfer t=6+27x1+4x0=33 cycles

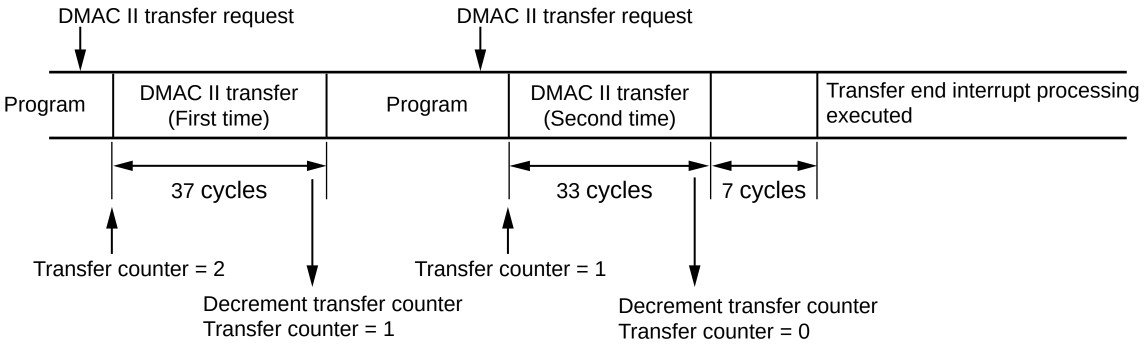
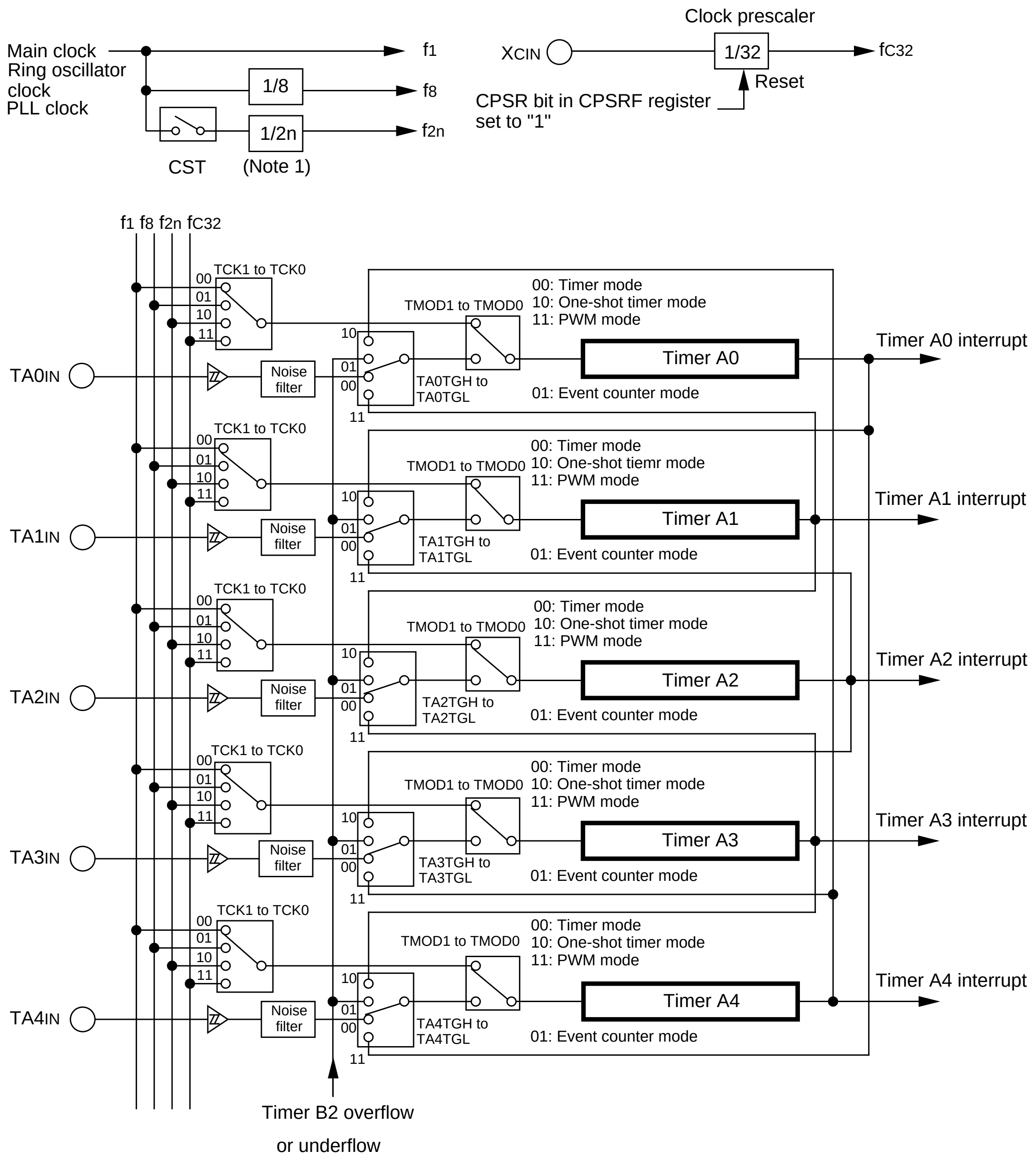


Figure 1.12.5. Transfer Cycle

When an interrupt request causing DMAC II request conditions and interrupt request with higher priority (e.g. $\overline{\text{NMI}}$ 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

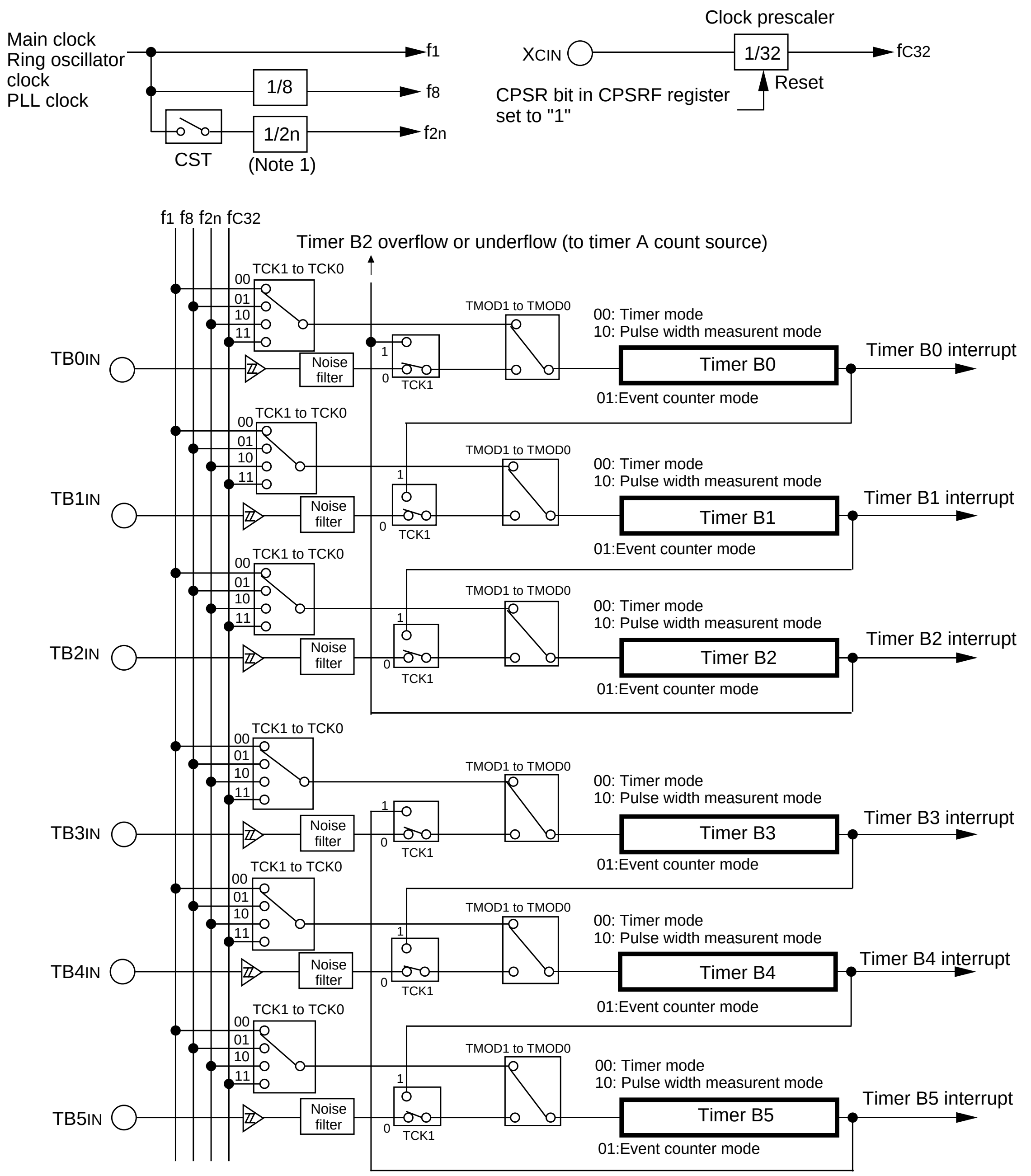
The microcomputer has eleven 16-bit timers. Five timers A and six timers B have different functions. Each timer functions independently. Count source of each timer is a clock for the timer operation like counting and reloading, etc. Figures 1.13.1 and 1.13.2 show block diagrams of the timer A and timer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



CST: Bit in TCSPR register
TCK1 to TCK0, TMOD1 to TMOD0 : Bits in TAiMR register
TA0GH to TA0GL: Bits in ONSF register
TAiGH to TAiGL: Bits in TRGSR register (i=1 to 4)

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

Figure 1.13.1. Timer A Configuration



CST : Bit in TCSPR register
TCK1 to TCK0, TMOD1 to TMOD0 : Bits in TBiMR register (i=0 to 5)

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

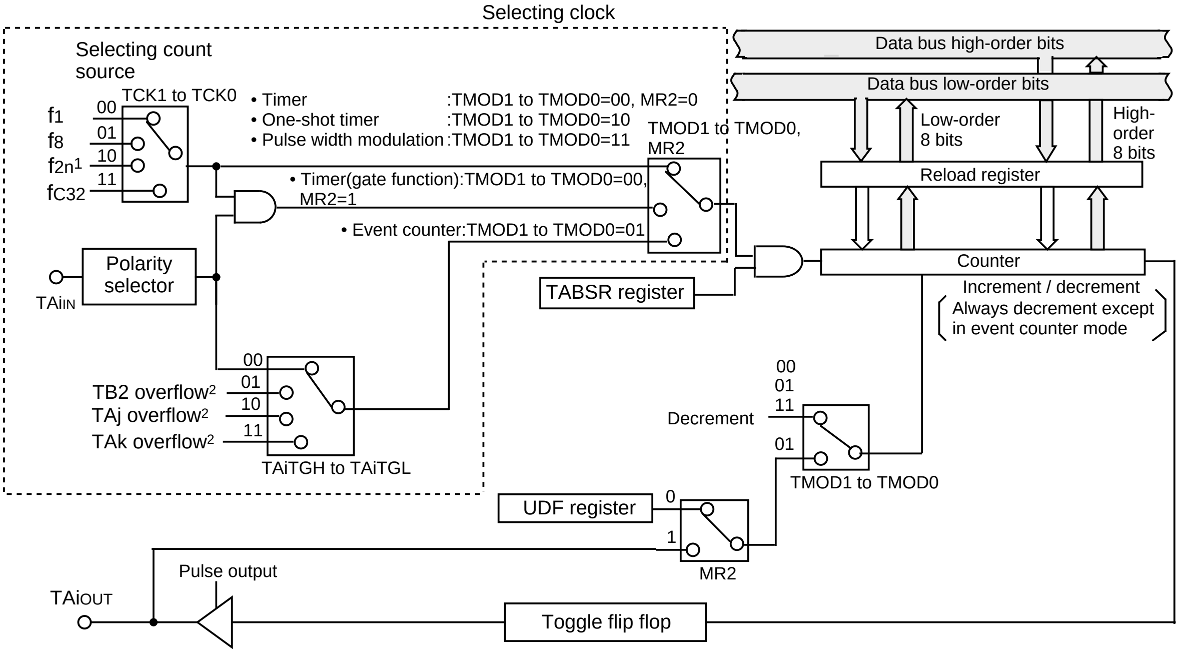
Figure 1.13.2. Timer B Configuration

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 TAiMR 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 "0000$_{16}$".
- Pulse width modulation mode: The timer consecutively outputs a given pulse width.



i=0 to 4
j=i-1, except j=4 if i=0
k=i+1, except k=0 if i=4

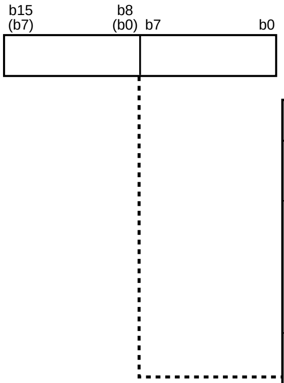
Notes :
1. The CNT3 to CNT0 bits in the TCSPR register determines either "No division (n=0)" or "divide-by-2n (n=1 to 15)" .
2. Overflow or underflow

TCK1 to TCK0, TMOD1 to TMOC0, MR2 to MR1 : Bits in TAiMR register
TAiTGH to TAiTGL : Bit in ONSF register if i=0, Bits in TRGSR register if i=1 to 4

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

Figure 1.14.1. Timer A Block Diagram

Timer Ai register (i=0 to 4)[1]



Symbol	Address	When reset
TA0 to TA2	0347_{16}-0346_{16}, 0349_{16}-0348_{16}, $034B_{16}$-$034A_{16}$	Indeterminate
TA3, TA4	$034D_{16}$-$034C_{16}$, $034F_{16}$-$034E_{16}$	Indeterminate

Mode	Function	Value that ca be set	RW
Timer mode	A count source divided by n+1, given n as a setting value	0000_{16} to $FFFF_{16}$	RW
Event counter mode	A count source divided by $FFFF_{16}$ - n+1 when a count source is incremented and by n+1 when a count source is decremented given n as the setting value[2]	0000_{16} to $FFFF_{16}$	RW
One-shot timer mode	A count source divided by n, given n as a setting value, then stopped[4]	0000_{16} to $FFFF_{16}$[3]	WO
Pulse width modulation mode (16-bit PWM)	Given fj as a frequency of count source and n as a setting value of TAi register, PWM cycle: (2 -1) / fj "H" width of PWM pulse: n / fj (Note 5)	0000_{16} to $FFFF_{16}$[3]	WO
Pulse width modulation mode (8-bit PWM)	Given fj as a frequency of count source, n as high-order address setting value of TAi register and m as low-order address setting value of TAi register, PWM cycle: $(2^8 -1)x(m+1) / fj$ "H" width of PWM pulse: (m+1)n / fj (Note 5)	00_{16} to FE_{16}[3] (High-order address) 00_{16} to FE_{16}[3] (Low-order address)	WO

Notes :
1. For read and write, 16-bit data should be used as a transfer unit.
2. The TAi register counts how many pulses are input from an external input pulse or how many times another timer overflows and underflows.
3. The MOV instruction should be used to set the TAi register.
4. When setting the TAi register to "000016", the counter does not operate and a timer Ai interrupt request is not generated.
5. When setting the TAi register to "000016", a pulse width modulator does not operate and an output level of the TAiOUT pin remains in "L". the TAi interrupt request is also not generated. This also occurs in an 8-bit pulse width modulator mode when 8 high-order bits in the TAi register are set to "0016".

Figure 1.14.2. TA0 to TA4 Registers

Timer Ai mode register (i=0 to 4)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TA0MR0 to TA4MR	0356$_{16}$, 0357$_{16}$, 0358$_{16}$, 0359$_{16}$, 035A$_{16}$	0000 0X00$_2$

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : One-shot timer mode 1 1 : Pulse width modulation (PWM) mode	RW
TMOD1			RW
MR0	Nothing is assigned. When write, should set to "0".		—
MR1	—————	Function varies depending on an operation mode	RW
MR2			RW
MR3			RW
TCK0	Count source select bit	Function varies depending on an operation mode	RW
TCK1			RW

Count start flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TABSR	0340$_{16}$	0000 0000$_2$

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

Figure 1.14.3. TA0MR to TA4MR Registers and TABSR Register

Up/down flag[1]

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
UDF	0344_{16}	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
TA0UD	Timer A0 up/down flag	0 : Decrement 1 : Increment (Note 2)	RW
TA1UD	Timer A1 up/down flag	0 : Decrement 1 : Increment (Note 2)	RW
TA2UD	Timer A2 up/down flag	0 : Decrement 1 : Increment (Note 2)	RW
TA3UD	Timer A3 up/down flag	0 : Decrement 1 : Increment (Note 2)	RW
TA4UD	Timer A4 up/down flag	0 : Decrement 1 : Increment (Note 2)	RW
TA2P	Timer A2 two-phase pulse signal processing function select bit	0 : two-phase pulse signal processing function disabled 1 : two-phase pulse signal processing function enabled (Note 3)	WO
TA3P	Timer A3 two-phase pulse signal processing function select bit	0 : two-phase pulse signal processing function disabled 1 : two-phase pulse signal processing function enabled (Note 3)	WO
TA4P	Timer A4 two-phase pulse signal processing function select bit	0 : two-phase pulse signal processing function disabled 1 : two-phase pulse signal processing function enabled (Note 3)	WO

Notes :
1. The MOV instruction should be used to set this register.
2. This bit is enabled when setting the MR2 bit in the TAiMR register to "0" (contens of the UDF register cause increment/decrement switching) in the event counter mode.
3. This bit should be set to "0" when not using a two-phase pulse signal processing function.

One-shot start flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
ONSF	0342_{16}	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
TA0OS	Timer A0 one-shot start flag	0 : idle 1 : Timer start (Note 1)	RW
TA1OS	Timer A1 one-shot start flag	0 : idle 1 : Timer start (Note 1)	RW
TA2OS	Timer A2 one-shot start flag	0 : idle 1 : Timer start (Note 1)	RW
TA3OS	Timer A3 one-shot start flag	0 : idle 1 : Timer start (Note 1)	RW
TA4OS	Timer A4 one-shot start flag	0 : idle 1 : Timer start (Note 1)	RW
TAZIE	Z-phase input enable bit	0 : Z-phase input disabled 1 : Z-phase input enabled	RW
TA0TGL	Timer A0 event/trigger select bit	b1 b0 0 0 : Selects input on $TA0_{IN}$ 0 1 : Selects TB2 overflow[2] 1 0 : Selects TA4 overflow[2] 1 1 : Selects TA1 overflow[2]	RW
TA0TGH			RW

Notes :
1. When read, a value is "0".
2. Overflow or underflow.

Figure 1.14.4. UDF Register and ONSF Register

Trigger select register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TRGSR	0343_{16}	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
TA1TGL	Timer A1 event/trigger select bit	b1 b0 0 0 : Selects input on $TA1_{IN}$ 0 1 : Selects TB2 overflow[1] 1 0 : Selects TA0 overflow[1] 1 1 : Selects TA2 overflow[1]	RW
TA1TGH			RW
TA2TGL	Timer A2 event/trigger select bit	b1 b0 0 0 : Selects input on $TA2_{IN}$ 0 1 : Selects TB2 overflow[1] 1 0 : Selects TA1 overflow[1] 1 1 : Selects TA3 overflow[1]	RW
TA2TGH			RW
TA3TGL	Timer A3 event/trigger select bit	b1 b0 0 0 : Selects input on $TA3_{IN}$ 0 1 : Selects TB2 overflow[1] 1 0 : Selects TA2 overflow[1] 1 1 : Selects TA4 overflow[1]	RW
TA3TGH			RW
TA4TGL	Timer A4 event/trigger select bit	b1 b0 0 0 : Selects input on $TA4_{IN}$ 0 1 : Selects TB2 overflow[1] 1 0 : Selects TA3 overflow[1] 1 1 : Selects TA0 overflow[1]	RW
TA4TGH			RW

Notes :
1. Overflow or underflow.

Count source prescaler register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TCSPR	$035F_{16}$	$0XXX\ 0000_2$

Bit symbol	Bit name	Function	RW
CNT0	Divide ratio select bit	f_{2n} is equal to the one that is divided the main clock, ring pscillator and PLL clock by 2n, given n as the setting value. Not divided if n=0. (Note 1)	RW
CNT1			RW
CNT2			RW
CNT3			RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		RW
——			RW
——			RW
CST	Operation enable bit	0 : Stopping a divider 1 : Operating a divider	RW

Notes :
1. The CNT3 to CNT0 bits should be rewritten after setting the CST bit to "0" (stops a divider).

Figure 1.14.5. TRGSR Register and TCSPR Register

Table 1.14.1. Pin Settings for Output from TAi_{OUT} Pin

Pin	Bit and setting values		
	PS1, PS2 registers	PSL1, PSL2 registers	PSC register
$P7_0$/$TA0_{OUT}$[1]	PS1_0= 1	PSL1_0=1	-
$P7_2$/$TA1_{OUT}$	PS1_2= 1	PSL1_2=1	-
$P7_4$/$TA2_{OUT}$	PS1_4= 1	PSL1_4=0	PSC_4= 0
$P7_6$/$TA3_{OUT}$	PS1_6= 1	PSL1_6=1	-
$P8_0$/$TA4_{OUT}$	PS2_0= 1	PSL2_0=0	-

Notes :

1. N-channel open drain

Table 1.14.2. Pin Settings for Input from TAi_{IN} and TAi_{OUT} Pins

Pin	Bits and setting values	
	PS1, PS2 registers	PD7, PD8 registers
$P7_0$/$TA0_{OUT}$	PS1_0=0	PD7_0=0
$P7_1$/$TA0_{IN}$	PS1_1=0	PD7_1=0
$P7_2$/$TA1_{OUT}$	PS1_2=0	PD7_2=0
$P7_3$/$TA1_{IN}$	PS1_3=0	PD7_3=0
$P7_4$$TA2_{OUT}$	PS1_4=0	PD7_4=0
$P7_5$/$TA2_{IN}$	PS1_5=0	PD7_5=0
$P7_6$$TA3_{OUT}$	PS1_6=0	PD7_6=0
$P7_7$/$TA3_{IN}$	PS1_7=0	PD7_7=0
$P8_0$$TA4_{OUT}$	PS2_0=0	PD8_0=0
$P8_1$/$TA4_{IN}$	PS2_1=0	PD8_1=0

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 TAiMR register in timer mode.

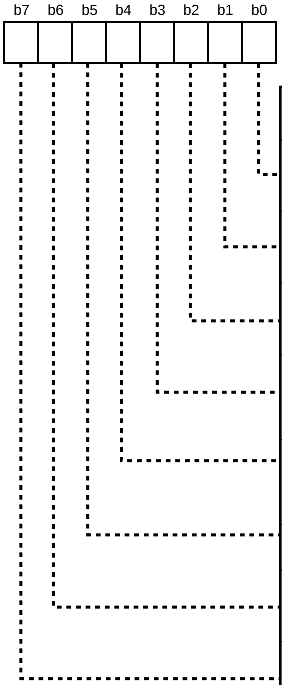
Table 1.14.3. Specifications in Timer Mode

Item	Specification
Count source	f_1, f_8, f_{2n}[1], f_{C32}
Count operation	• The timer decrements the counter • When the timer underflows, it reloads contents of the reload register into ones of the count register to continue counting.
Divide ratio	1/(n+1) n: setting value of TAi register (i=0 to 4) 0000_{16} to $FFFF_{16}$
Count start condition	The TAiS bit in the TABSR register is set to "1" (starts counting)
Count stop condition	The TAiS bit is set to "0" (stops counting)
Interrupt request generation timing	Timer underflows
TAi_{IN} pin function	Programmable I/O port or gate input
TAi_{OUT} pin function	Programmable I/O port or pulse output
Read from timer	The Ai register indicates a value of the counter
Write to timer	• 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 • 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)
Selectable function	• Gate function Input signal to the TAi_{IN} pin determines whether the timer starts or stops counting • Pulse output function A polarity of the TAi_{OUT} pin is inversed whenever the timer underflows

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)".

Timer Ai mode register (i=0 to 4) (Timer mode)



Symbol	Address	When reset
TA0MR0 to TA4MR	0356_{16}, 0357_{16}, 0358_{16}, 0359_{16}, $035A_{16}$	0000 $0X00_2$

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	RW
TMOD1			RW
MR0	Nothing is assigned. When write, should set to "0".		—
MR1	Gate function select bit	b1 b0 0 X : Gate function disabled[1] (TAiIN pin is a programmable I/O pin) 1 0 : Timer counts only when the TAiIN pin is set to "L" 1 1 : Timer counts only when the TAiIN pin is set to "H"	RW
MR2			RW
MR3	Shold be set to "0" in timer mode		RW
TCK0	Count source select bit	b1 b0 0 0 : f_1 0 1 : f_8 1 0 : f_{2n}[2] 1 1 : f_{C32}	RW
TCK1			RW

Notes :
1. X value can be set to either way "0" or "1".
2. The CNT3 to CNT0 bits in the TCSPR register determines either "no division (n=0)" or "divide-by-2n (n=1 to 15)".

Figure 1.14.6. TA0MR to TA4MR Registers

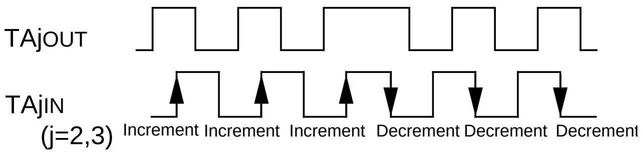
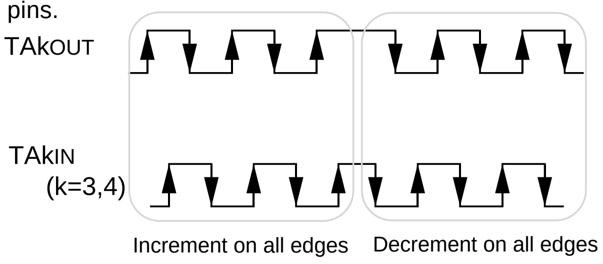
(2) Event Counter Mode

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 TAiMR 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	• External signal input to the TAiIN pin (i = 0 to4) (valid edge can be selected by program) • The timer B2 overflows or underflows, timer Aj overflows or underflows (j=i-1, except j=4 if i=0) and timer Ak overflows or underflows (k=i+1, except k=0 if i=4)
Count operation	• Determines whether the timer increments or decrements the counter • 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
Divide ratio	• 1/($FFFF_{16}$ - n + 1) for a counter increment • 1/(n + 1) for a counter decrement n : setting value of the TAi register 0000_{16} to $FFFF_{16}$
Count start condition	The TAiS bit is set to "1" (starts counting)
Count stop condition	The TAiS bit is set to "0" (stops counting)
Interrupt request generation timing	The timer overflows or underflows
TAiIN pin function	Programmable I/O port or count source input
TAiOUT pin function	Programmable I/O port, pulse output or input to a select the counter increment/decrement
Read from timer	The Ai register indicates a value of the counter
Write to timer	• 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. • 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).
Selectable function	• Free-running count function Contents of the reload register are not reloaded even if the The timer overflows or underflows • Pulse output function A polarity of the TAiOUT is inversed whenever the timer overflows or underflows

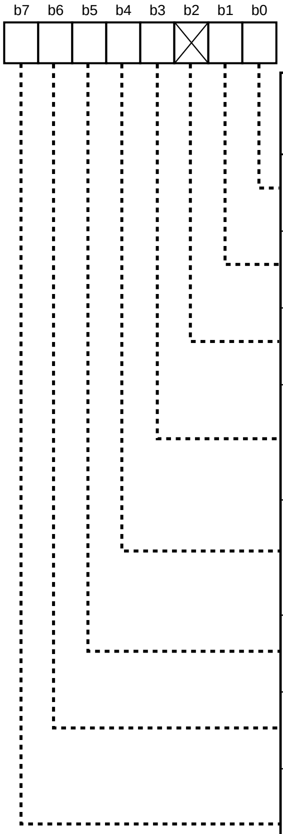
Table 1.14.5. Specifications in Event Counter Mode (when processing two-phase pulse signal on timer A2, A3 and A4)

Item	Specification
Count source	Two-phase pulse signal input to the TAiIN or TAiOUT pin (i = 2 to 4)
Count operation	• Determines whether the timer increments or decrements the counter • 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
Divide ratio	• 1/ ($FFFF_{16}$ - n + 1) for a counter increment • 1/ (n + 1) for a counter decrement n : setting value of the TAi register
Count start condition	The TAiS bit is set to "1" (starts counting)
Count stop condition	The TAiS bit is set to "0" (stops counting)
Interrupt request generation timing	The timer overflows or underflows
TAiIN pin function	Two-phase pulse input
TAiOUT pin function	Two-phase pulse input
Read from timer	The Ai register indicates a value of the counter
Write to timer	• 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 • While counting A value written to the TAi register is written to the reload register only (Transferred to counter at next reload time).
Selectable function[1]	• Normal processing operation (the timer A2 and timer A3) While input signal to the TAjOUT pin is set to "H" the timer increments the counter on the rising edge of the TAjIN pin or decrements the counter on the falling edge  TAjOUT TAjIN (j=2,3) Increment Increment Increment Decrement Decrement Decrement • Multiplied-by-4 processing operation (the timer A3 and timer A4) When an input signal to the TAkOUT pin is set to "H" with the rising edge of the TAkIN pin, the timer increments the counter on the rising and falling edges of the TAkout and TAkin pins. When an input signal to the TAkOUT pin is set to "H" with the falling edge of the TAkIN pin, the timer decrements the counter on the rising and falling edges of the TAkOUT and TAkIN pins.  TAkOUT TAkIN (k=3,4) Increment on all edges Decrement on all edges

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 Ai mode register (i=0 to 4) (Event counter mode)



Symbol	Address	When reset
TA0MR to TA4MR	0356$_{16}$, 0357$_{16}$, 0358$_{16}$, 0359$_{16}$, 035A$_{16}$	0000 0X00$_2$

Bit symbol	Bit name	Function (When not using two-phase pulse signal processing)	Function (When using two-phase pulse signal processing)	RW
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode1		RW
TMOD1				RW
MR0	Nothing is assigned. When write, should set to "0".			–
MR1	Count polarity select bit[2]	0 : Counts falling edges of external signal 1 : Counts rising edges of external signal	Should set to "0"	RW
MR2	Increment/decrement switching cause select bit	0 : Content of UDF 1 : Input signal to TAiOUT pin[3]	Should set to "1"	RW
MR3	Should set to "0" in event counter mode			RW
TCK0	Count operation type select bit	0 : Reloading 1 : Free running		RW
TCK1	Two-phase pulse signal processing operation select bit[4,5]	Should set to "0"	0 : Normal processing operation 1 : Multiplied-by-4 processing operation	RW

Notes :

1. The TAiTGH and TAiTGL bits in the ONSF or TRGSR register determine a counter source in the event counter mode.
2. The MR1 bit is enabled only when counting how many times external signals are input.
3. The timer decrements the counter when an input signal to the TAiOUT pin is set to "L" and the timer increments the counter when an input signal to the TAiOUT pin is set to "H".
4. The TCK bit is enabled for the TA3MR register.
5. For two-phase pulse signal processing, the TAjP bit in the UDF register (j=2 to 4) should be set to "1" (two-phase pulse signal processing function enabled). Also, the TAiTGH and TAiTGL bits should be set to "00$_2$" (input to the TAjIN pin).

Figure 1.14.7. TA0MR to TA4MR Registers

- **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.

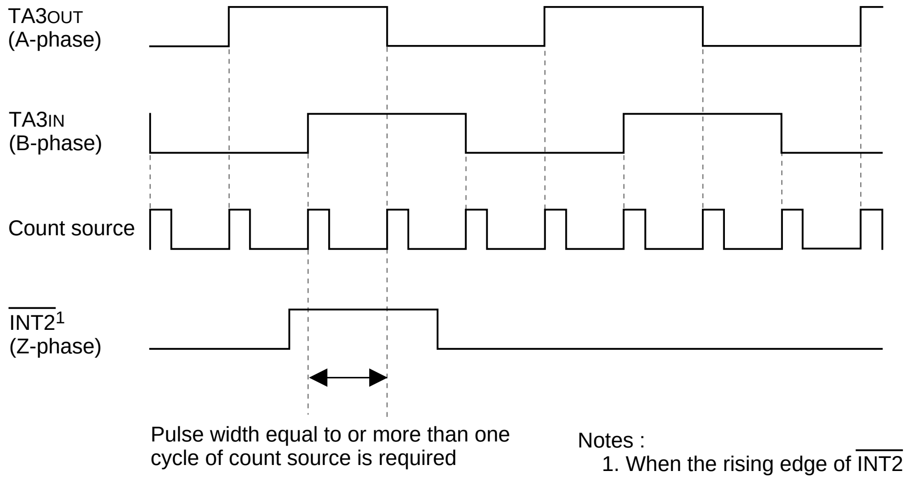
This function can be used in timer A3 event counter mode, two-phase pulse signal processing, free-run type and multiplied-by-4 processing. The Z-phase is input to the $\overline{\text{INT2}}$ pin.

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_{16}" 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.

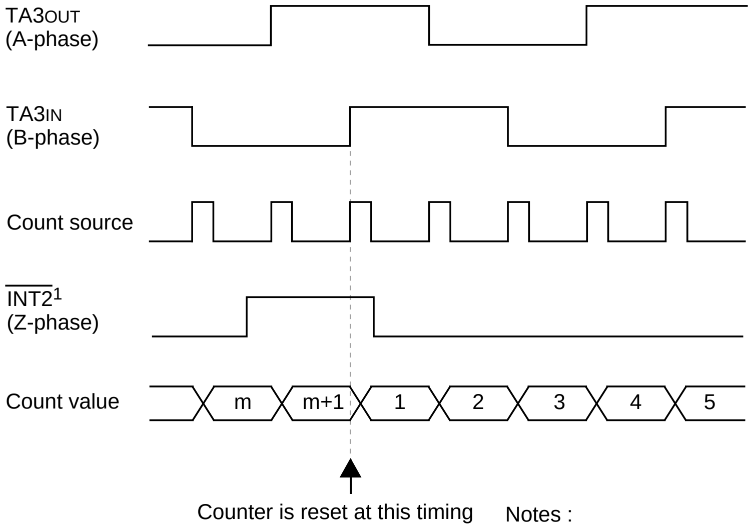
Z-phase input resets the counter in the next count source following Z-phase input. Figure 1.14.9 shows the counter reset timing.

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.


TA3OUT
(A-phase)
TA3IN
(B-phase)
Count source
INT2¹
(Z-phase)
Pulse width equal to or more than one cycle of count source is required

Notes :
1. When the rising edge of $\overline{\text{INT2}}$ is selected.

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



Notes :
1. When the rising edge of $\overline{\text{INT2}}$ is selected.

Figure 1.14.9. Counter Reset Timing

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 TAiMR register (i=0 to 4) in one-shot timer mode.

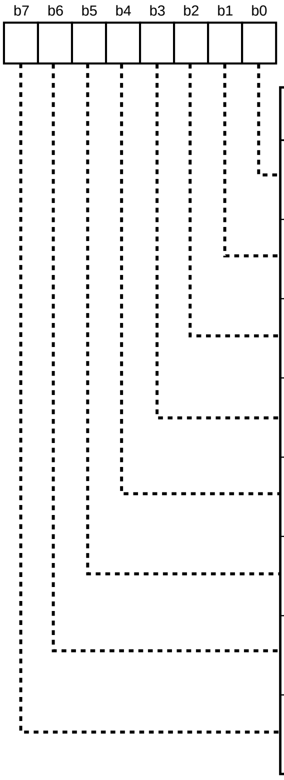
Table 1.14.6. Specifications in One-shot Timer Mode

Item	Specification
Count source	f_1, f_8, f_{2n}[1], f_{C32}
Count operation	• The timer decrements the counter • When the counter reaches "0000_{16}", the timer stops counting after reloading • If a trigger occurs while counting, the timer reloads to resume counting
Divide ratio	1/n n : setting value of the TAi register (i=0 to 4) 0000_{16} to $FFFF_{16}$, the counter do not run if n=0000_{16}
Count start condition	• The TAiS bit in the TABSR register is set to "1" (starts counting) and following triggers are generated • External trigger is input • The timer overflows and underflows • The TAiOS bit in the ONSF register is set to "1" (timer starts)
Count stop condition	• Reload after the counter has reached "0000_{16}" • The TAiOS bit is set to "0" (timer stops)
Interrupt request generation timing	Counter reaches "0000_{16}"
TAi_{IN} pin function	Programmable I/O port or trigger input
TAi_{OUT} pin function	Programmable I/O port or pulse output
Read from timer	Value is indeterminate by reading the TAi register
Write to timer	• 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. • 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.)

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)".

Timer Ai mode register (i=0 to 4) (One-shot timer mode)



Symbol	Address	When reset
TA0MR0 to TA4MR	0356_{16}, 0357_{16}, 0358_{16}, 0359_{16}, $035A_{16}$	$0000\ 0X00_2$

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 1 : One-shot timer mode	RW
TMOD1			RW
MR0	Nothing is assigned. When write, should set to "0".		—
MR1	External trigger select bit[1]	0 : Falling edge of input signal to TAi$_{IN}$ pin 1 : Rising edge of input signal to TAi$_{IN}$ pin	RW
MR2	Trigger select bit	0 : TAiOS bit is enabled 1 : Selected by TAiTGH and TAiTGL bits	RW
MR3	Should set to "0" in the ohe-shot timer mode		RW
TCK0	Count source select bit	b1 b0 0 0 : f_1 0 1 : f_8 1 0 : f_{2n}[2] 1 1 : f_{C32}	RW
TCK1			RW

Notes :

1. The MR1 bit is enabled only when the TAiTGH and TAiTGL bits in the TRGSR are set to "00_2". The MR1 bit can be set to either way "0" or "1" when setting to "01_2" (TB2 overflow and underflow), "10_2" (TA4 overflow and underflow) or "11_2" (TA1 overflow and underflow).
2. The CNT3 to CNT0 bits in the TCSPR register determines either "no division (n=0)" or "divide-by-2n (n=1 to 15)".

Figure 1.14.10. TA0MR to TA4MR Registers

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 TAiMR 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_1, f_8, f_{2n}[1], f_{C32}
Count operation	• The timer decrements the counter (operating as an 8-bit or a 16-bit pulse width modulator) • The timer reloads on the rising edge of PWM pulse and continues counting • The timer is not affected by a trigger that occurs during counting
16-bit PWM	• "H" width = n / fi n : setting value of TAi register 0000_{16} to $FFFF_{16}$ f_j : Count source frequency • Cycle = $(2^{16}-1)$ / fi fixed
8-bit PWM	• "H" width = n x (m+1) / f_j n : setting value of high-order address of the TAi register 00_{16} to FF_{16} • Cycles = (2^8-1) x (m+1) / f_j m : setting value of low-order address of the TAi register 00_{16} to FF_{16}
Count start condition	• External trigger is input • The timer overflows and underflows • The TAiS bit in the TABSR register is set to "1" (start counting)
Count stop condition	The TAiS bit is set to "0" (stop counting)
Interrupt request generation timing	Falling edge of PWM pulse
TAj_{IN} pin function	Programmable I/O port or trigger input
TAj_{OUT} pin function	Pulse output
Read from timer	The Ai register indicates a value of the counter
Write to timer	• When the counter stops A value written to the TAi register is also written to both reload register and counter. • 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.)

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)".

Timer Ai mode register (i=0 to 4) (Pulse width modulator mode)

Symbol	Address	When reset
TA0MR0 to TA4MR	0356_{16}, 0357_{16}, 0358_{16}, 0359_{16}, $035A_{16}$	$0000\ 0X00_2$

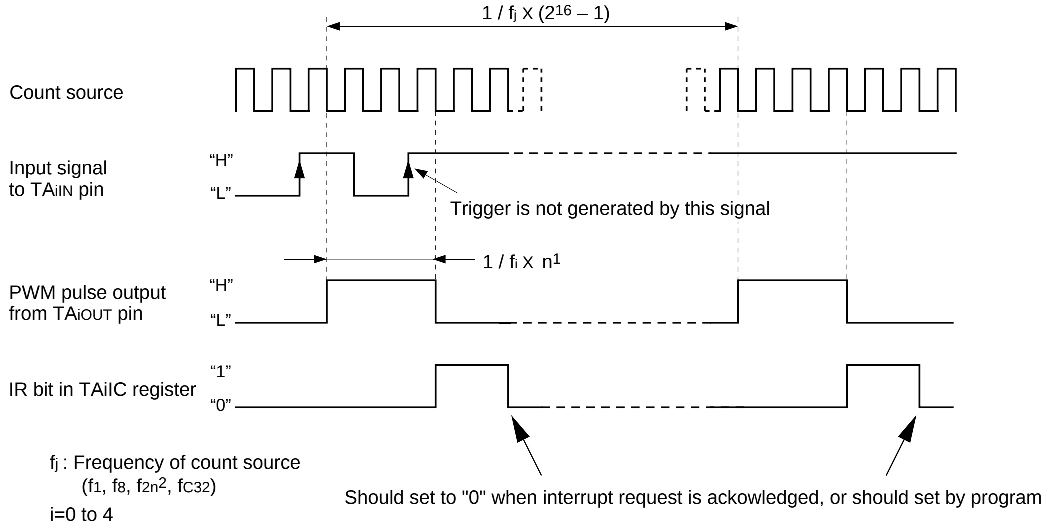
Bit	Bit symbol	Bit name	Function	RW
b0	TMOD0	Operation mode select bit	b1 b0 1 1 : Pulse width modulation (PWM) mode	RW
b1	TMOD1			RW
b2	MR0	Nothing is assigned. When write, should set to "0".		—
b3	MR1	External trigger select bit[1]	0 : Falling edge of input signal to TAi$_{IN}$ pin 1 : Rising edge of input signal to TAi$_{IN}$ pin	RW
b4	MR2	Trigger select bit	0 : TAiOS bit is enabled 1 : Selected by TAiTGH and TAiTGL bits	RW
b5	MR3	16/8-bit PWM mode select bit	0: Functions as a 16-bit pulse width modulator 1: Functions as an 8-bit pulse width modulator	RW
b6	TCK0	Count source select bit	b1 b0 0 0 : f_1 0 1 : f_8 1 0 : f_{2n}[2] 1 1 : f_{C32}	RW
b7	TCK1			RW

Notes :

1. The MR1 bit is enabled only when the TAiTGH and TAiTGL bits in the TRGSR are set to "00_2". The MR1 bit can be set to either way "0" or "1" when setting to "01_2" (TB2 overflow and underflow), "10_2" (TA4 overflow and underflow) or "11_2" (TA1 overflow and underflow).
2. The CNT3 to CNT0 bits in the TCSPR register determines either "no division (n=0)" or "divide-by-2n (n=1 to 15)".

Figure 1.14.11. TA0MR to TA4MR Registers

When reload register = 0003_{16} and external trigger (rising edge of input signal to the TAiIN pin) is selected


1 / fj X (2^{16} – 1)
Count source
Input signal to TAiIN pin
"H"
"L"
Trigger is not generated by this signal
1 / fi X n1
PWM pulse output from TAiOUT pin
"H"
"L"
IR bit in TAiIC register
"1"
"0"

Should set to "0" when interrupt request is ackowledged, or should set by program

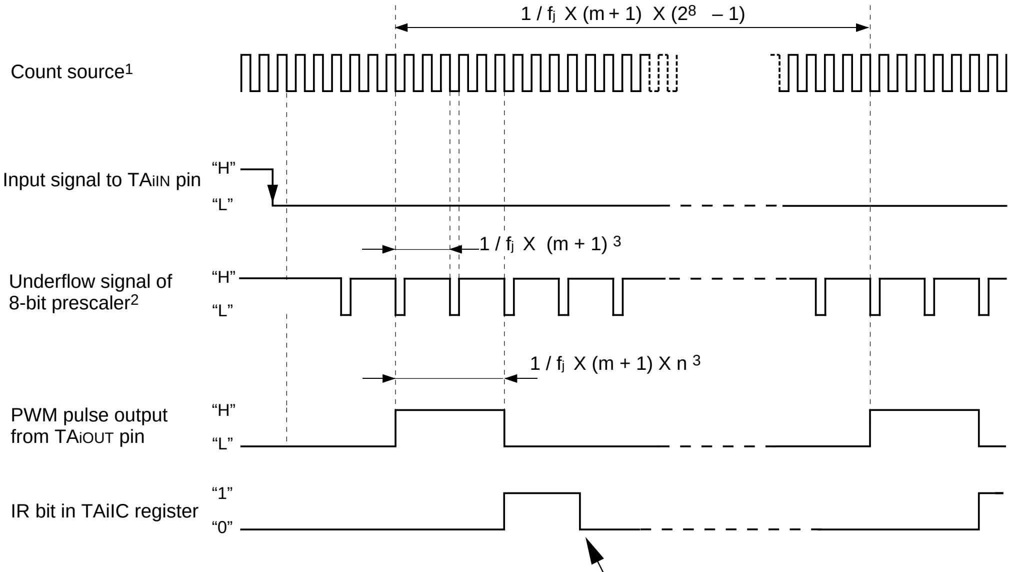
fj : Frequency of count source
(f_1, f_8, f_{2n}2, fC32)

i=0 to 4

Notes :
1. n = 0000_{16} to $FFFE_{16}$.
2. "No division (n=0)" or "divide-by-2n (n=1 to 15)" should be set, depending on the CNT3 to CNT0 bits in the TCSPR register.

Figure 1.14.12. 16-bit Pulse Width Modulator Operation

When reload high-order 8 bits of reload register = 02_{16},
reload low-order 8 bits of reload register = 02_{16} and
external trigger (falling edge of input signal to the TAiIN pin) is selected


1 / fj X (m + 1) X (2^8 – 1)
Count source1
Input signal to TAiIN pin
"H"
"L"
1 / fj X (m + 1) 3
Underflow signal of 8-bit prescaler2
"H"
"L"
1 / fj X (m + 1) X n 3
PWM pulse output from TAiOUT pin
"H"
"L"
IR bit in TAiIC register
"1"
"0"

fj : Frequency of count source
(f_1, f_8, f_{2n}2, fC32)

Should set to "0" when interrupt request is ackowledged, or should set by program

i=0 to 4

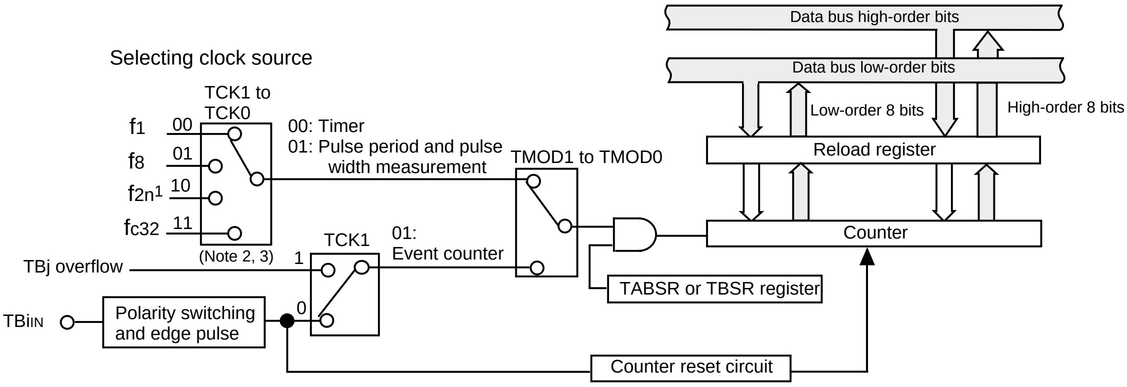
Notes :
1. 8-bit prescaler counts the count source.
2. 8-bit pulse width modulator counts underflow signal of the 8-bit prescaler.
3. m = 00_{16} to FF_{16}, n = 00_{16} to FE_{16}
4. The CNT3 to CNT0 bits in the TCSPR register determines either "no division (n=0)" or "divide-by-2n (n=1 to 15)" .

Figure 1.14.13. 8-bit Pulse Width Modulator Operation

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.



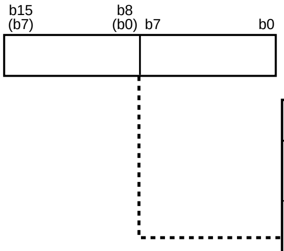
i=0 to 5
Notes :
1. The CNT3 to CNT0 bits in the TCSPR register determines either "no division (n=0)" or "divide-by-2n (n=1 to 15)".
2. Overflows or underflows.
3. j=i-1, except j=2 when i=0 j=5 when i=3

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

TCK1 to TCK0, TMOD1 to TMOD0: Bits in TAiMR register

Figure 1.15.1. Timer B Block Diagram

Timer Bi register (i=0 to 5)[1]



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

Mode	Function	Value that can be set	RW
Timer mode	A count source divided by n+1, given n as a setting value	0000_{16} to $FFFF_{16}$	RW
Event counter mode	A count source divided by n+1, given n as a setting value[2]	0000_{16} to $FFFF_{16}$	RW
Pulse period/pulse width measurement mode	A count source is incremented at a period between an valid edge and another valid edge of TBiIN pulse	0000_{16} to $FFFF_{16}$[3]	RO

Notes :
1. For read and write, 16-bit data should be used as a transfer unit.
2. The TBi register counts how many pulses are input from an external input pulse or how many times another timer overflows and underflows.

Figure 1.15.2. TB0 to TB5 Registers

Timer Bi mode register (i=0 to 5)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TB0MR0 to TB5MR	$035B_{16}$, $035C_{16}$, $035D_{16}$, $031B_{16}$, $031C_{16}$, $031D_{16}$	$0000\ 0X00_2$

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode 0 1 : Event counter mode 1 0 : Pulse period measurement mode, pulse width measurement mode 1 1 : Avoid this setting	RW
TMOD1			RW
MR0	————	Function varies depending on an operation mode[1,2]	RW
MR1			RW
MR2			RW
MR3			RW
TCK0	Count source select bit	Function varies depending on an operation mode	RW
TCK1			RW

Notes :
1. The MR2 bits in theTB0MR and TB3MR registers are enabled.
2. Nothing is assigned in the MR2 bit in the TB1MR, TB2MR, TB4MR and TB5MR registers. When write, should set to "0". When read, its content is indeterminate.

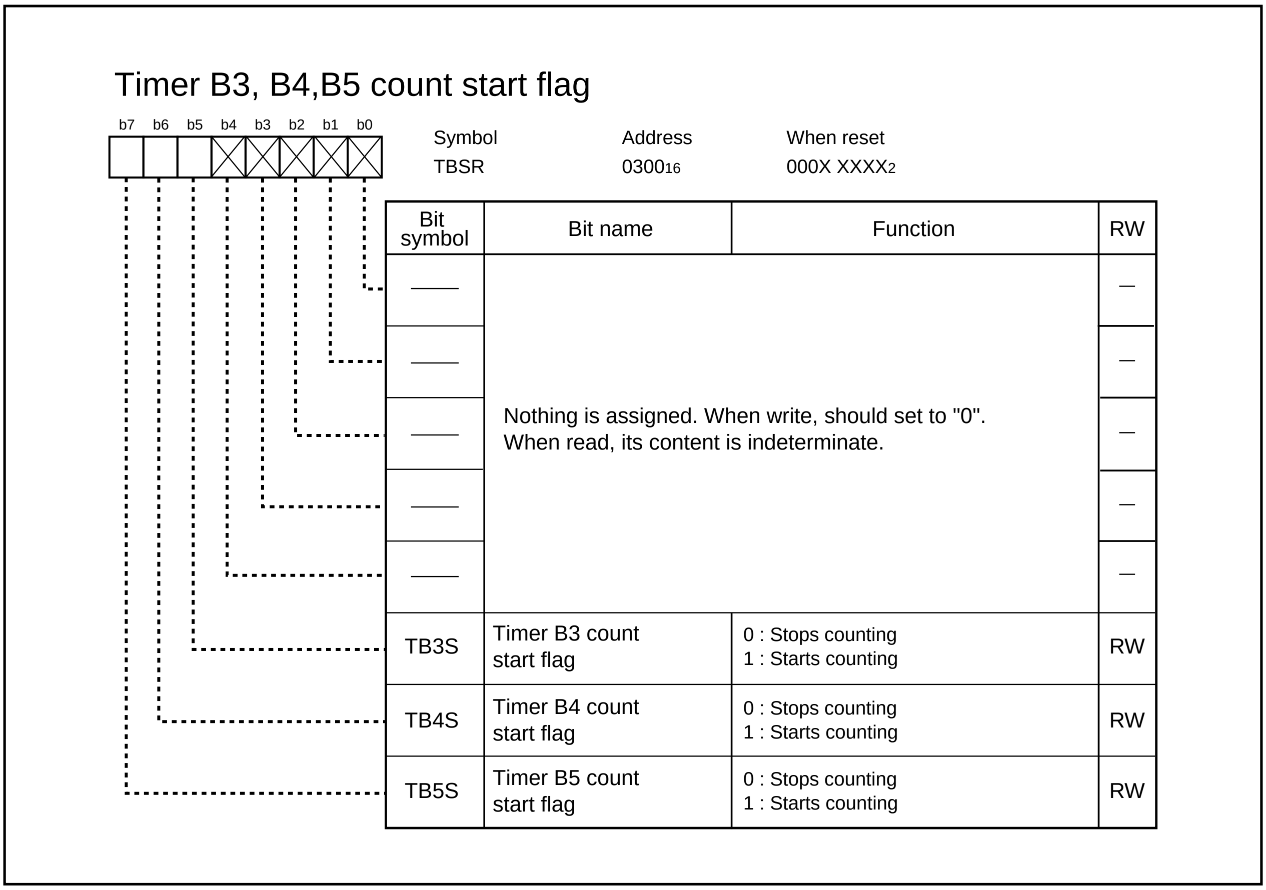
Count start flag

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
TABSR	0340_{16}	$0000\ 0000_2$

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

Figure 1.15.3. TB0MR to TB5MR Registers, TABSR Register



Timer B3, B4,B5 count start flag

Symbol	Address	When reset
TBSR	0300_{16}	$000X\ XXXX_2$

Bit symbol	Bit name	Function	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
——			—
——			—
——			—
——			—
TB3S	Timer B3 count start flag	0 : Stops counting 1 : Starts counting	RW
TB4S	Timer B4 count start flag	0 : Stops counting 1 : Starts counting	RW
TB5S	Timer B5 count start flag	0 : Stops counting 1 : Starts counting	RW

Figure 1.15.4. TBSR Register

Table 1.15.1. Settings for the TBi$_{IN}$ Pins (i=0 to 5)

Port name	Function	Bits and Setting values	
		PS1, PS3[1] registers	PD7, PD9[1] registers
$P9_0$	$TB0_{IN}$	PS3_0=0	PD9_0=0
$P9_1$	$TB1_{IN}$	PS3_1=0	PD9_1=0
$P9_2$	$TB2_{IN}$	PS3_2=0	PD9_2=0
$P9_3$	$TB3_{IN}$	PS3_3=0	PD9_3=0
$P9_4$	$TB4_{IN}$	PS3_4=0	PD9_4=0
$P7_1$	$TB5_{IN}$	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.

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.

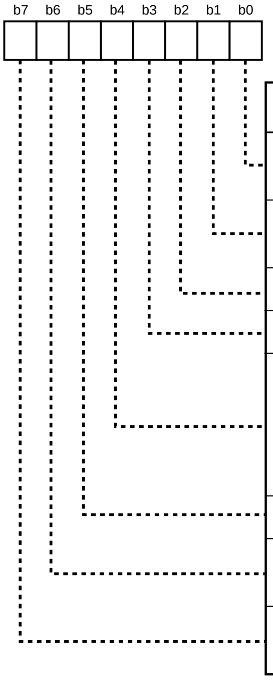
Table 1.15.2. Specifications in Timer Mode

Item	Specification
Count source	f_1, f_8, f_{2n}[1], f_{C32}
Count operation	• The timer decrements the counter • When the timer overflows of 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.
Divide ratio	1/(n+1) n : setting value of the TBi register (i=0 to 5) 0000_{16} to $FFFF_{16}$
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
TB_{iIN} pin function	Programmable I/O port
Read from timer	The Ai register indicates a value of the counter
Write to timer	• 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. • 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).

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)".

Timer Bi mode register (i=0 to 5) (Timer mode)



Symbol	Address	When reset
TB0MR to TB5MR	$035B_{16}$, $035C_{16}$, $035D_{16}$, $031B_{16}$, $031C_{16}$, $031D_{16}$	$00XX\ 0000_2$

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 0 : Timer mode	RW
TMOD1			RW
MR0	Disabled in timer mode. Can be set to "0" or "1".		RW
MR1			RW
MR2	TB0MR, TB3MR registers Should set to "0" in timer mode		RW
	TB1MR, TB2MR TB4MR, TB5MR registers Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
MR3	Shold be set to "0" in timer mode		RW
TCK0	Count source select bit	b1 b0 0 0 : f_1 0 1 : f_8 1 0 : f_{2n}[1] 1 1 : f_{C32}	RW
TCK1			RW

Notes :

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

Figure 1.15.5. TB0MR to TB5MR Registers

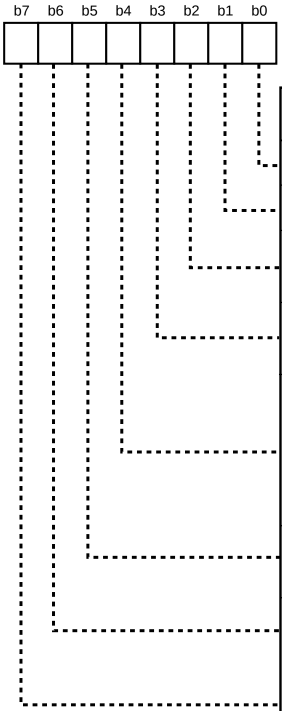
2. Event Counter Mode

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	• 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 an valid edge of a count source by program. • TBj overflows or underflows (j=i-1, except j=2 when i=0, j=5 when i=3)
Count operation	• The timer decrements the counter • When the timer underflows, it reloads contents the reload register into ones of the register to continue counting
Divide ratio	1/(n+1) n : setting value of the TBi register 0000_{16} to $FFFF_{16}$
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	• When the counter stops or before the first count source prior to stating of a coutner is input A value written to the TBi register is also written to both reload register and counter. • While counting A value written to the TBi register is written to the reload register only (Transferred to counter at next reload time).

Timer Bi mode register (i=0 to 5) (Event counter mode)



Symbol	Address	When reset
TB0MR to TB5MR	$035B_{16}$, $035C_{16}$, $035D_{16}$, $031B_{16}$, $031C_{16}$, $031D_{16}$	$00XX\ 0000_2$

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 0 1 : Event counter mode	RW
TMOD1			RW
MR0	Count polarity select bit[1]	b1 b0 0 0 : Counts falling edges of external signal 0 1 : Counts falling edges of external signal 1 0 : Counts falling and rising edges of external signal 1 1 : Avoid this setting	RW
MR1			RW
MR2	TB0MR, TB3MR registers Should set to "0" in Event counter mode		RW
	TB1MR, TB2MR TB4MR, TB5MR registers Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
MR3	Disabled in event coutner mode. When write, should set to "0". When read, its content is indeterminate.		RO
TCK0	Disabled in event coutner mode. Can be set to "0" or "1".		RW
TCK1	Count source select bit	0 : Input from TBiIN pin 1 : TBj overflow[2]	RW

Notes :
1. This bit is enabled when setting the TCK1 bit to "0" (select TBiIN pin input as event clock). The MR1 bit can be set to either way "0" or "1" when setting the TCK1 bit to "1" (select timer overflow or underflow).
2. j=i – 1, except j=2 when i=0 and j=5 when i=3.

Figure 1.15.6. TB0MR to TB5MR Registers

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.

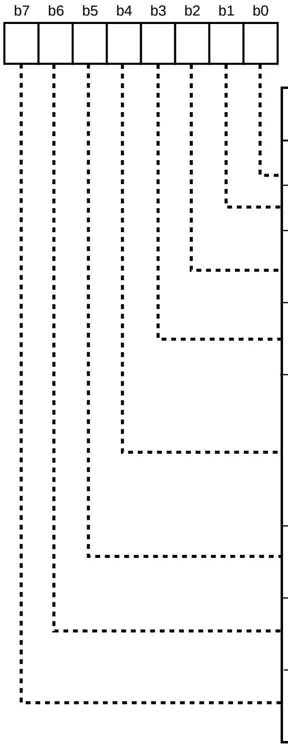
Table 1.15.4. Specifications in Pulse Period and Pulse Width Measurement Mode

Item	Specification
Count source	f_1, f_8, f_{2n}[3], f_{C32}
Count operation	• The timer increments the counter • Counter value is transferred to the reload register on a valid edge of measurement pulse The counter value is set to "0000_{16}" to continue counting
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	• When valid edge of measurement pulse is input[1] • 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.
TBi_{IN} pin function	Measurement pulse input
Read from timer	The Ai register indicates a value of the counter[2]
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 Bi mode register (i=0 to 5)
(Pulse period / pulse width measurement mode)



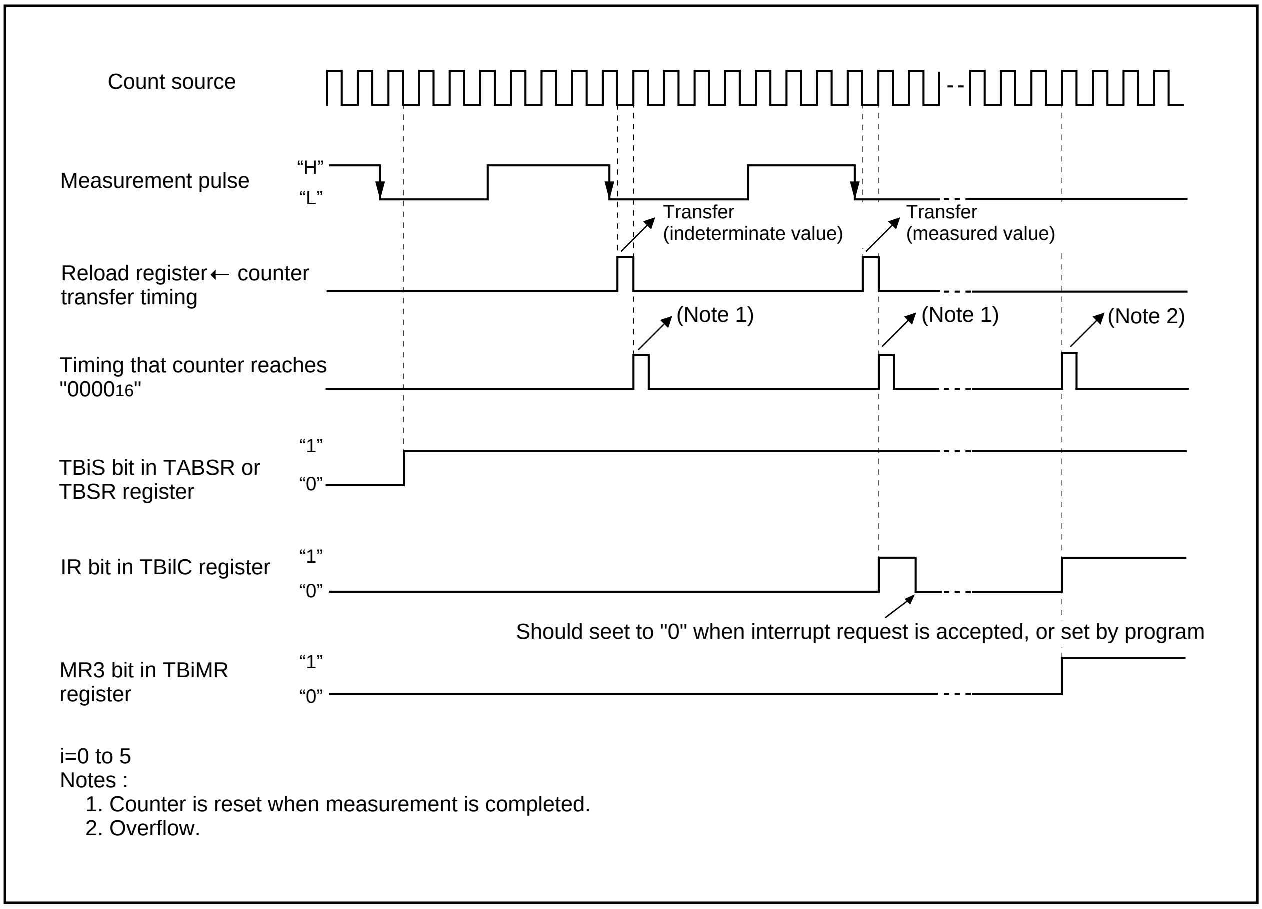
Symbol	Address	When reset
TB0MR to TB5MR	$035B_{16}$, $035C_{16}$, $035D_{16}$, $031B_{16}$, $031C_{16}$, $031D_{16}$	$00XX\ 0000_2$

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	b1 b0 1 0 : Pulse period measurement mode, pulse width measurement mode	RW
TMOD1			RW
MR0	Measurement mode select bit	b1 b0 0 0 : Pulse period measurement 1 0 1 : Pulse period measurement 2 1 0 : Pulse width measurement 1 1 : Avoid this setting[1]	RW
MR1			RW
MR2	TB0MR, TB3MR registers Should set to "0" in Event counter mode		RW
	TB1MR, TB2MR TB4MR, TB5MR registers Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
MR3	Timer Bi overflow flag[2]	0 : No overflow 1 : Overflow	RO
TCK0	Count source select bit	b1 b0 0 0 : f_1 0 1 : f_8 1 0 : f_{2n}[3] 1 1 : f_{C32}	RW
TCK1			RW

Notes :

1. The MR1 to MR0 bits measure as follows.
   Pulse period measurement 1 (MR1 to MR0 bits = 00_2) :
   Measurement between a falling edge and the next falling edge of measurement pulse.
   Pulse period measurement 2 (MR1 to MR0 bits = 01_2) :
   Measurement between a rising edge and the next rising edge of measurement pulse .
   Pulse width measurement (MR1 to MR0 bits = 10_2) :
   Measurement between a falling edge and the next rising edge of measurement pulse, between a rising edge and the falling edge of measurement pulse.
2. This bit is indeterminate when reset.
   When the timer overflows, the MR3 bit 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", the MR3 bit is set to "0" (no overflow) by setting the TBiMR register again. The MR3 bit cannot be set to "1" by program.
3. The CNT3 to CNT0 bits in the TCSPR register determines either "no division (n=0)" or "divide-by-2n (n=1 to 15)".

Figure 1.15.7. TB0MR to TB5MR Registers

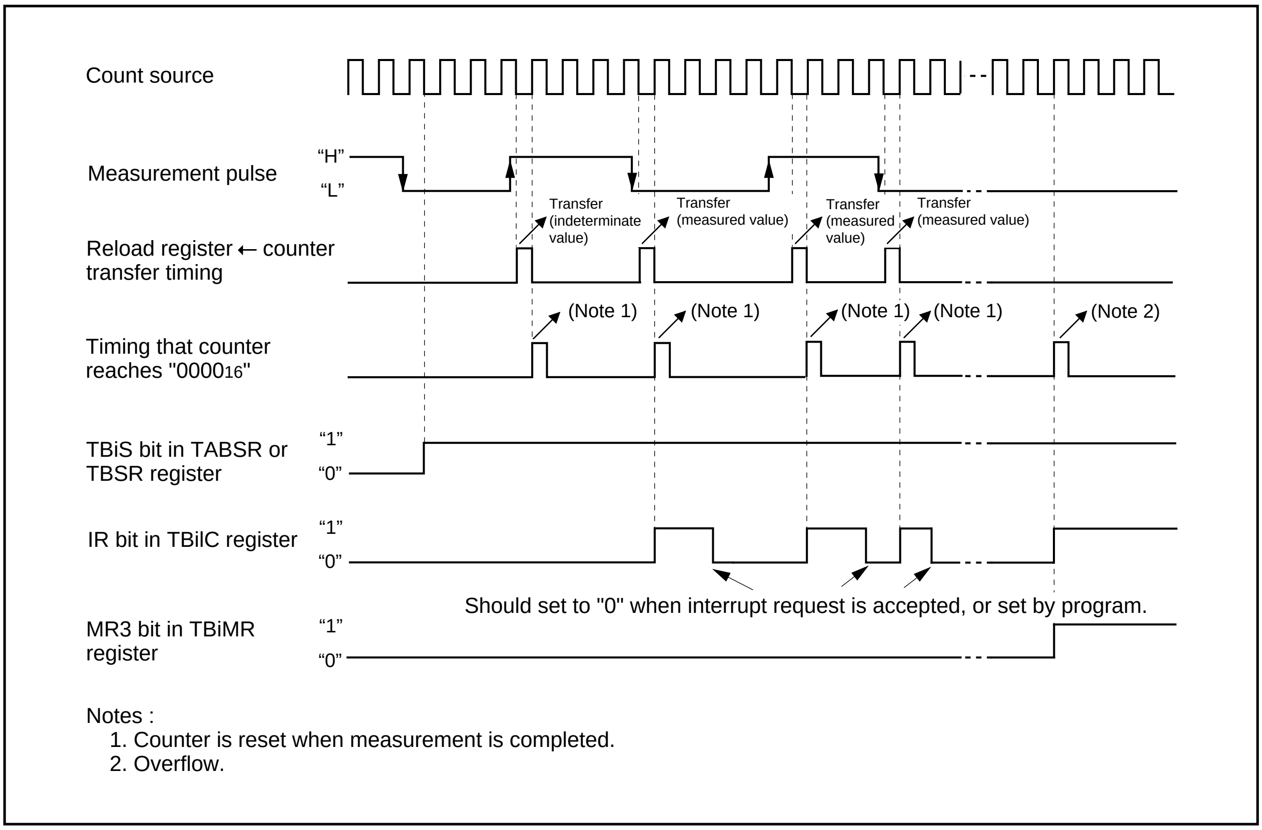

Count source
Measurement pulse
"H"
"L"
Transfer (indeterminate value)
Transfer (measured value)
Reload register ← counter transfer timing
(Note 1)
(Note 1)
(Note 2)
Timing that counter reaches "0000₁₆"
TBiS bit in TABSR or TBSR register
"1"
"0"
IR bit in TBiIC register
"1"
"0"
Should seet to "0" when interrupt request is accepted, or set by program
MR3 bit in TBiMR register
"1"
"0"

i=0 to 5

Notes :

1. Counter is reset when measurement is completed.
2. Overflow.

Figure 1.15.8. Pulse Period 1 Measurement


Count source
Measurement pulse
"H"
"L"
Transfer (indeterminate value)
Transfer (measured value)
Transfer (measured value)
Transfer (measured value)
Reload register ← counter transfer timing
(Note 1)
(Note 1)
(Note 1)
(Note 1)
(Note 2)
Timing that counter reaches "0000₁₆"
TBiS bit in TABSR or TBSR register
"1"
"0"
IR bit in TBiIC register
"1"
"0"
Should set to "0" when interrupt request is accepted, or set by program.
MR3 bit in TBiMR register
"1"
"0"

Notes :

1. Counter is reset when measurement is completed.
2. Overflow.

Figure 1.15.9. Pulse Width Measurement

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, $\overline{U}$, V, $\overline{V}$, W, $\overline{W}$)
Forced cutoff input[1]	Input "L" to the $\overline{\text{NMI}}$ pin
Timers to be used	Timer A4, A1, A2 (used in one-shot timer mode) Timer A4: U- and $\overline{U}$-phase waveform control Timer A1: V- and $\overline{V}$-phase waveform control Timer A2: W- and $\overline{W}$-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_{16} to $FFFF_{16}$ Count source: f_1, f_8, f_{2n}[2], f_{c32}
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_{16} to $FFFF_{16}$ Count source: f_1, f_8, f_{2n}, f_{c32}
Dead time	Count source x p, or no dead time p: setting value of the DTT register, 00_{16} to FF_{16} Count source: f_1, or f_1 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

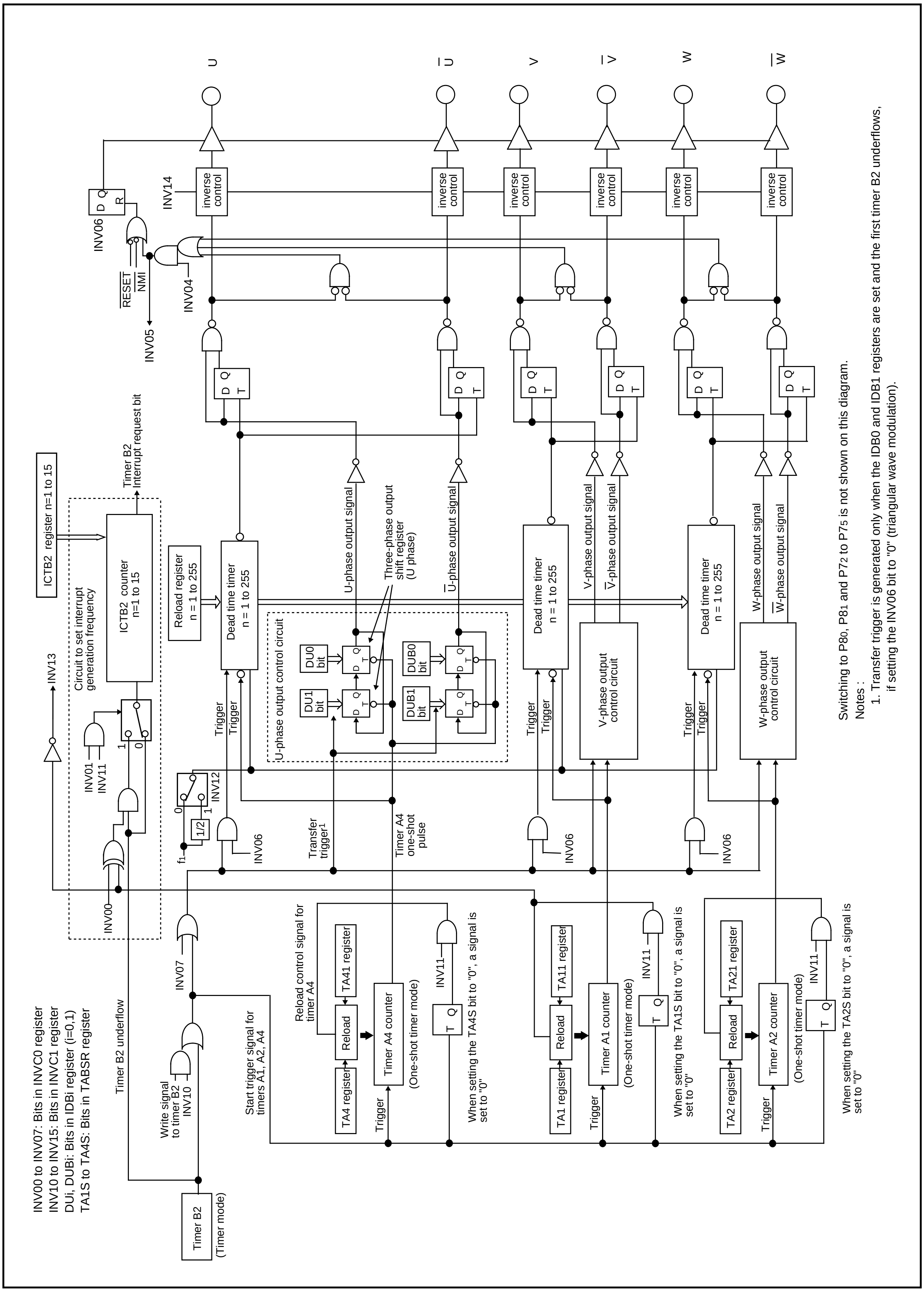
Notes :

1. Forced cutoff with $\overline{\text{NMI}}$ 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)".

Table 1.16.2. Pin Settings

Pin	Bits and Setting values		
	PS1, PS2 registers[1]	PSL1, PSL2 registers	PSC register
$P7_2$/V	PS1_2 bit =1	PSL1_2 bit =0	PSC_2 bit =1
$P7_3/\overline{V}$	PS1_3 bit =1	PSL1_3 bit =1	——
$P7_4$/W	PS1_4 bit =1	PSL1_4 bit =1	——
$P7_5/\overline{W}$	PS1_5 bit =1	PSL1_5 bit =0	——
$P8_0$/U	PS2_0 bit =1	PSL2_0 bit =1	——
$P8_1/\overline{U}$	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".


U
$\overline{U}$
V
$\overline{V}$
W
$\overline{W}$
inverse control
INV14
INV06
D Q
R
$\overline{RESET}$
$\overline{NMI}$
INV04
INV05
D Q
T
Timer B2 Interrupt request bit
ICTB2 register n=1 to 15
Circuit to set interrupt generation frequency
ICTB2 counter n=1 to 15
INV13
INV01
INV11
1
0
INV00
Reload register n = 1 to 255
Dead time timer n = 1 to 255
Trigger
Trigger
U-phase output signal
Three-phase output shift register (U phase)
$\overline{U}$-phase output signal
DU0 bit
DU1 bit
DUB0 bit
DUB1 bit
U-phase output control circuit
V-phase output signal
$\overline{V}$-phase output signal
V-phase output control circuit
W-phase output signal
$\overline{W}$-phase output signal
W-phase output control circuit
INV12
0
1
1/2
f1
INV06
Transfer trigger1
Timer A4 one-shot pulse
Timer B2 underflow
INV07
Write signal to timer B2
INV10
Timer B2
(Timer mode)
Start trigger signal for timers A1, A2, A4
Reload control signal for timer A4
TA41 register
Reload
TA4 register
Timer A4 counter
(One-shot timer mode)
INV11
T Q
When setting the TA4S bit to "0", a signal is set to "0"
TA11 register
TA1 register
Timer A1 counter
When setting the TA1S bit to "0", a signal is set to "0"
TA21 register
TA2 register
Timer A2 counter
When setting the TA2S bit to "0", a signal is set to "0"

INV00 to INV07: Bits in INVC0 register
INV10 to INV15: Bits in INVC1 register
DUi, DUBi: Bits in IDBi register (i=0,1)
TA1S to TA4S: Bits in TABSR register

Switching to P80, P81 and P72 to P75 is not shown on this diagram.

Notes :

1. Transfer trigger is generated only when the IDB0 and IDB1 registers are set and the first timer B2 underflows, if setting the INV06 bit to "0" (triangular wave modulation).

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

Three-phase PWM control register 0[1]

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
INVC0	0308_{16}	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
INV00	Interrupt enable output polarity select bit	0: ICTB2 counter is incremented by one when timer B2 underflows every add times 1: ICTB2 counter is incremented by one when timer B2 underflows every even times (Note 3)	RW
INV01	Interrupt enable output specification bit[2]	0: ICTB2 counter is incremented by one when timer B2 underflows 1: Selected by the INV00 bit (Note 3)	RW
INV02	Mode select bit[4,5]	0: Not use the three-phase control timer functions (High-impedance) 1: Three-phase control timer function[6]	RW
INV03	Output control bit	0: Three-phase control timer output disabled[6] 1: Three-phase control timer output enabled[7]	RW
INV04	Positive and negative-phases concurrent active disable function enable bit	0: Concurrent active enabled 1: Concurrent active disabled	RW
INV05	Positive and negative-phases concurrent active output detect flag	0: Not detected 1: Detected[8]	RW
INV06	Modulation mode select bit[9]	0: Triangular wave modulation mode 1: Sawtooth wave modulation mod[10]	RW
INV07	Software trigger select bit	Transfer trigger is generated when setting the INV07 bit to "1". Trigger to the dead time timer is also generated when setting the INV06 bit to "1". Its value is "0" when reading	RW

Notes :

1. This register should be set after the PRC1 bit in the PRCR register is set to "1" (write enable).
2. The INV01 bit should be set to "1" after setting a value in the ICTB2 register .
3. This bit is enabled only when setting the INV11 bit to "1" (three-phase mode 1). ICTB2 counter is incremented by one whenever timer B2 underflows, regardless of INV00 and INV01 bits, when setting to "0" (three-phase mode).
4. When setting the INV02 bit to "1", dead time timer, U-, V-and W-phase output control circuits and ICTB2 counter are operated.
5. Pins should be set after setting the INV02 bit to "1". See Table 1.16.2 about pin settings.
6. When setting the INV02 bit to "1" (three-phase control timer functions) and the INV03 bit to "0" (three-phase control timer output disabled), U, $\overline{U}$, V, $\overline{V}$, W and $\overline{W}$ pins, including on other output functions shared pins, to goes into a high-impedance state.
7. The INV03 bit is set to "0" in the following condition is :
   - Reset
   - When setting to "0" by program
   - When $\overline{NMI}$ pin is is set to "H" from "L" (Cannot be set the INV03 bit to "1" when setting $\overline{NMI}$ pin to "L".)
8. "1" cannot be set by program but "0" can.
9. The following table lists how the INV06 bit works.

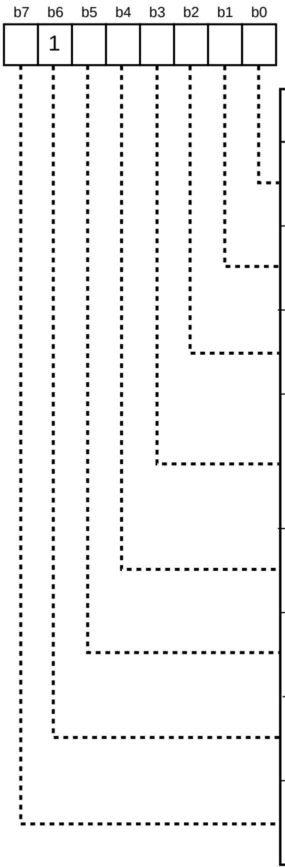
Item	INV06 = 0	INV06 = 1
mode	Triangular wave modulation mode	Sawtooth wave modulation mode
Timing to transfer from the IDB0 and IDB1 registers to three output shift register	Transfer once in sync with a transfer trigger after setting the IDB0 and IDB1 registers	Transfer every transfer trigger

Transfer trigger : Timer B2 underflow and write to the INV07 bit, or write to the TB2 register when INV10 = 1

10. When setting the INV06 bit to "1", the INV11 bit to "0" (three-phase mode 0), and the PWCON bit to "0" (reload timer B2 with timer B2 underflow) should be set.

Figure 1.16.2. INVC0 Register

Three-phase PWM control register 1[1]



Symbol INVC1
Address 0309_{16}
When reset $0000\ 0000_2$

Bit symbol	Bit name	Function	RW
INV10	Timer A1, A2 and A4 start trigger select bit	0: Timer B2 underflow 1: Timer B2 underflow and write to timer B2	RW
INV11	Timer A1-1, A2-1, A4-1 control bit[2]	0: Three-phase mode 0[3] 1: Three-phase mode 1	RW
INV12	Dead time timer count source select bit	0 : f_1 1 : f_1 divided by 2	RW
INV13	Carrier wave detect flag[4]	0: Timer A output in odd timers (Count TAj1 register value) 1: Timer A output in even timers (Count TAj1 register value)	RO
INV14	Output polarity control bit	0 : Output waveform "L" active 1 : Output waveform "H" active	RW
INV15	Dead time disabled bit	0: Dead time enabled 1: Dead time disabled	RW
——	Reserved bit	Should set to "1"	RW
——	Reserved bit	Should set to "0"	RW

Notes :

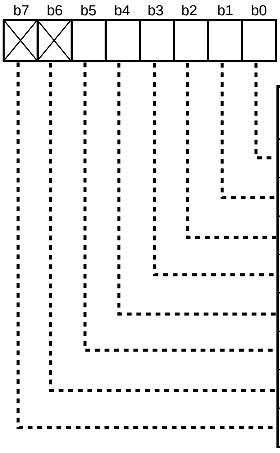
1. This register should be set after the PRC1 bit in the PRCR register is set to "1" (write enable).
2. The following table lists now the INV11 bit works.

Item	INV11 = 0	INV11 = 1
mode	Three-phase mode 0	Three-phase mode 1
TA11, TA21 and TA41 registers	Not used	Used
INV00 and INV01 bit	Disabled. ICTB2 counter is incremented whenever timer B2 underflows	Enabled
INV13 bit	Disabled	Enabled when INV11=1 and INV06=0

3. When setting the INV06 bit to "1" (sawtooth wave modulation mode), this bit is set to "0" (three-phase mode 0). Also, when setting the INV11 bit to "0", the PWCON bit is set to "0" (reload timer B2 with timer B2 underflow).
4. The INV13 bit is enabled only when setting INV06 bit to "1" (sawtooth wave modulation mode) and the INV11 bit to "1" (three-phase mode 1).

Figure 1.16.3. INVC1 Register

Three-phase output buffer register i (i=0, 1)[1]



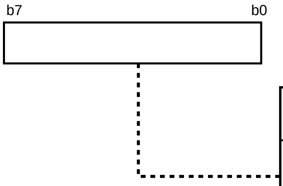
Symbol	Address	When reset
IDB0, IDB1	$030A_{16}$, $030B_{16}$	$XX11\ 1111_2$

Bit symbol	Bit name	Function	RW
DUi	U-phase output buffer i	Write output level 0: Active level 1: Non-active level When read, a value of three-phase shift register is read.	RW
DUBi	$\overline{U}$-phase output buffer i		RW
DVi	V-phase output buffer i		RW
DVBi	$\overline{V}$-phase output buffer i		RW
DWi	W-phase output buffer i		RW
DWBi	$\overline{W}$-phase output buffer i		RW
—	Nothing is assigned. When write, should set to "0". When read, its contents is "0".		–
—			–

Notes :

1. Values of the IDB0 and IDB1 registers are transferred to the three-phase output shift register by transfer trigger.
   After the transfer trigger occurs, the values written in the IDB0 register become each phase output signal first.
   Then values written in the IDB1 register on the falling edge of timers A1, A2 and A4 one-shot pulse become each phase output signal.

Dead time timer[1,2]



Symbol	Address	When reset
DTT	$030C_{16}$	Indeterminate

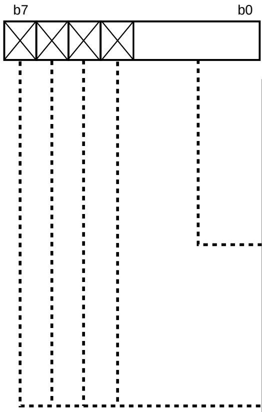
Function	Setting range	RW
As a setting value is n, after a start trigger occurs, the timer stops when counting n times of a count source selected by the INV12 bit. Phase of either positive or negative phase, which change from non-active level to active level, shifts on when the dead time timer stops.	1 to 255	WO

Notes :

1. The MOV instruction should be used to set this register.
2. The dead time timer is enabled when setting the INV15 bit to "0" (dead time enabled). No dead time period can be set when setting the INV15 bit to "1" (dead time disabled). The INV06 bit determines a start trigger of the DTT register in the INV06 bit.

Figure 1.16.4. IDB0, IDB1 and DTT Registers

Timer B2 interrupt generation frequency set counter



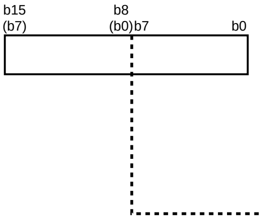
Symbol	Address	When reset
ICTB2	$030D_{16}$	XXXX ????$_2$

Function	Setting range	RW
When setting the INV01 bit to "0" (the ICTB2 counter increments whenever the timer B2 underflows) and a setting value is *n*, the timer B2 interrupt is generated whenever the *n*th timer B2 underflow occurs. When setting the INV01 bit to "1" (the NV00 bit selects count timing of the ICTB2 counter) and a setting value is *n*, the timer B2 interrupt is generated whenever the *n*th timer B2 underflow occurs by matching the condition selected in the INV00 bit. (Note 1)	1 to 15	WO
Nothing is assigned. When write, should set to "0".		—

Notes :

1. The MOV instruction should be used to set the ICTB2 register.
   When setting the INV01 bit to "1" and the TB2S bit to "0" (timer B2 counter stop), the ICTB2 register should be set.
   When setting the INV01 bit to "0" and the TB2S bit to "1" (timer B2 counter start), the ICTB2 register can be set.
   But, when the timer B2 underflows, avoid setting the ICTB2 register.

Timer Ai, Ai-1 register[1,2,3,4,5,6]



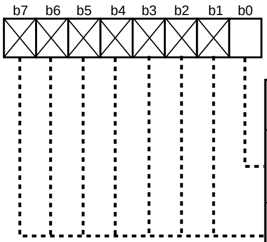
Symbol	Address	When reset
TA1, TA2, TA4	0349_{16} - 0348_{16}, $034B_{16}$ - $034A_{16}$, $034F_{16}$ - $034E_{16}$	Indeterminate
TA11, TA21, TA41	0303_{16} - 0302_{16}, 0305_{16} - 0304_{16}, 0307_{16} - 0306_{16}	Indeterminate

Function	Setting range	RW
As a setting value is *n*, after a start trigger occurs , a timer stops when *n*th count source is counted. Positive phase changes to negative phase, and vice versa, when the timers A1, A2 and A4 stop.	0000_{16} to $FFFF_{16}$	WO

Notes :

1. 16-bit data should be used for read and write.
2. When setting these registers to "0000_{16}", no counter starts and no timer Ai interrupt is generated.
3. The MOV instruction should be used to set these registers.
4. When the INV15 bit in the INVC1 register is set to "0" (dead timer enabled), a phase, which change from a non-active level to active level, changes when a dead time timer stops.
5. When setting the INV11 bit to "0" (three-phase mode 0), a value of the TAi register is transferred to the reload register by a timer Aj start trigger.
   When setting the INV11 bit to "1" (three-phase mode 1), value of the TAi1 register is transferred to the reload register. Then value of the TAj register is transferred to the reload register when a timer Ai start trigger occurs.
6. Avoid writing to these registers when the timer B2 underflows.

Timer B2 special mode register



Symbol	Address	When reset
TB2SC	$035E_{16}$	XXXX XXX0$_2$

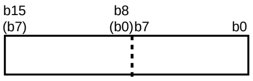
Bit symbol	Bit name	Function	RW
PWCOM	Timer B2 reload timing switching bit[1]	0 : Timer B2 underflow 1 : Timer A output in odd times	RW
Nothing is assigned. When write, should set to "0". When read, its content is "0."			–

Notes :

1. When setting the INV11 bit to "0" (three-phase mode 0) or the INV06 bit to "0" (timer B2 underflow), the PWCOM bit should be set to "0" (timer B2 underflow).

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

Timer B2 register[1]

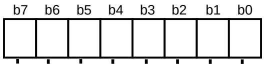


Symbol: TB2
Address: 0355_{16} - 0354_{16}
When reset: Indeterminate

Function	Setting range	RW
As a setting value is n, a count source is divided by n+1. Timers A1, A2 and A4 starts whenever an underflow occurs.	0000_{16} to $FFFF_{16}$	RW

Notes :
1. 16-bit data should be used for write and read.

Trigger select register

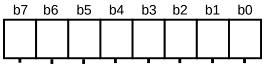


Symbol: TRGSR
Address: 0343_{16}
When reset: $0000\ 0000_2$

Bit symbol	Bit name	Function	RW
TA1TGL	Timer A1 event/trigger select bit	Should set to "01_2" (TB2 underflow) before using a V-phase output control circuit	RW
TA1TGH			RW
TA2TGL	Timer A2 event/trigger select bit	Should set to "01_2" (TB2 underflow) before using a W-phase output control circuit	RW
TA2TGH			RW
TA3TGL	Timer A3 event/trigger select bit	b1 b0 0 0 : Selects TA3IN pin input 0 1 : Selects TB2 overflow[1] 1 0 : Selects TA2 overflow[1] 1 1 : Selects TA4 overflow[1]	RW
TA3TGH			RW
TA4TGL	Timer A4 event/trigger select bit	Should set to "01_2" (TB2 underflow) before using a U-phase output control circuit	RW
TA4TGH			RW

Notes :
1. Overflow or underflow

Count start flag



Symbol: TABSR
Address: 0340_{16}
When reset: $0000\ 0000_2$

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

Figure 1.16.6. TB2, TRGSR and TABSR Registers

Timer Ai mode register (i=1, 2, 4)

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

Symbol	Address	When reset
TA1MR, TA2MR, TA4MR	0357$_{16}$, 0358$_{16}$, 035A$_{16}$	0000 0X00$_2$

Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	Should set to "10$_2$" (one-shot timer mode) with three-phase motor control timer function	RW
TMOD1			
MR0	Nothing is assigned. When write, should set to "0."		—
MR1	External trigger select bit	Disabled with three-phase motor control timer function	RW
MR2	Trigger select bit	Should set to "1"(selected by TRGSR register) with three-phase motor control timer function	RW
MR3	Should set to "0" in the three-phase motor control timer function		RW
TCK0	Count source select bit	b1 b0 0 0 : f$_1$ 0 1 : f$_8$ 1 0 : f$_{2n}$[1] 1 1 : f$_{C32}$	RW
TCK1			RW

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)".

Timer B2 mode register

b7	b6	b5	b4	b3	b2	b1	b0
			0			0	0

Symbol	Address	When reset
TA2MR	035D$_{16}$	00XX 0000$_2$

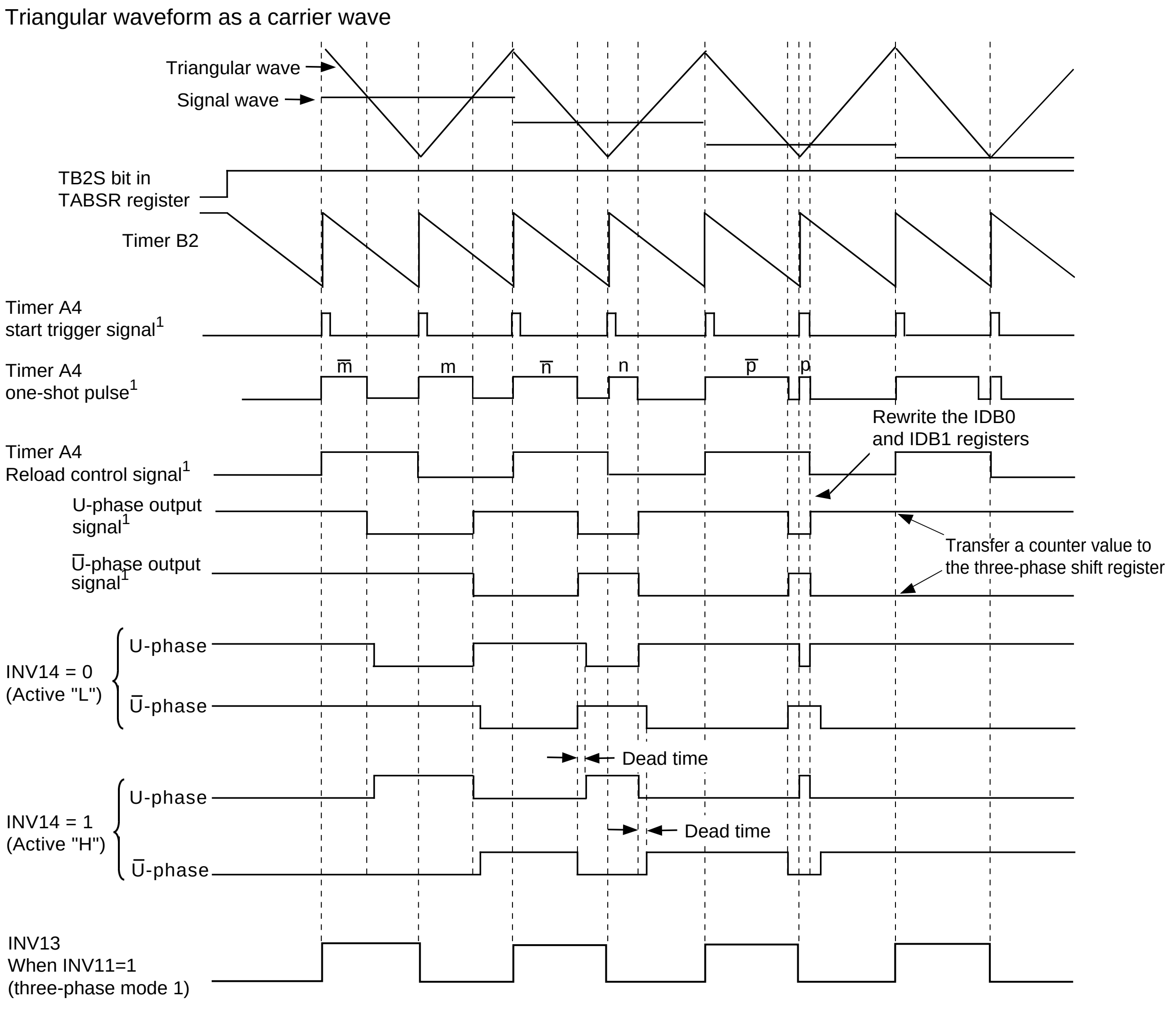
Bit symbol	Bit name	Function	RW
TMOD0	Operation mode select bit	Should set to "00$_2$" (timer mode) with three-phase motor control timer function	RW
TMOD1			
MR0	Disabled with three-phase motor control timer function. When write, should set to "0". When read, its content is indeterminate.		—
MR1			
MR2	Should set to "0" with three-phase motor control timer function		RW
MR3	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		RW
TCK0	Count source select bit	b1 b0 0 0 : f$_1$ 0 1 : f$_8$ 1 0 : f$_{2n}$[1] 1 1 : f$_{C32}$	RW
TCK1			RW

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)".

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

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, $\overline{U}$, V, $\overline{V}$, W, $\overline{W}$) 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.


Triangular waveform as a carrier wave
Triangular wave
Signal wave
TB2S bit in TABSR register
Timer B2
Timer A4 start trigger signal1
Timer A4 one-shot pulse1
m̄
m
n̄
n
p̄
p
Rewrite the IDB0 and IDB1 registers
Timer A4 Reload control signal1
U-phase output signal1
Ū-phase output signal1
Transfer a counter value to the three-phase shift register
INV14 = 0 (Active "L")
U-phase
Ū-phase
Dead time
INV14 = 1 (Active "H")
U-phase
Ū-phase
Dead time
INV13 When INV11=1 (three-phase mode 1)

Notes :

1. Internal signals. See Figure 1.16.1.

The above applies when INVC0 = 00XX11XX$_2$ and INVC1 = 010XXXX0$_2$ (X: Varies depending on each system.)

Examples of PWM output change are

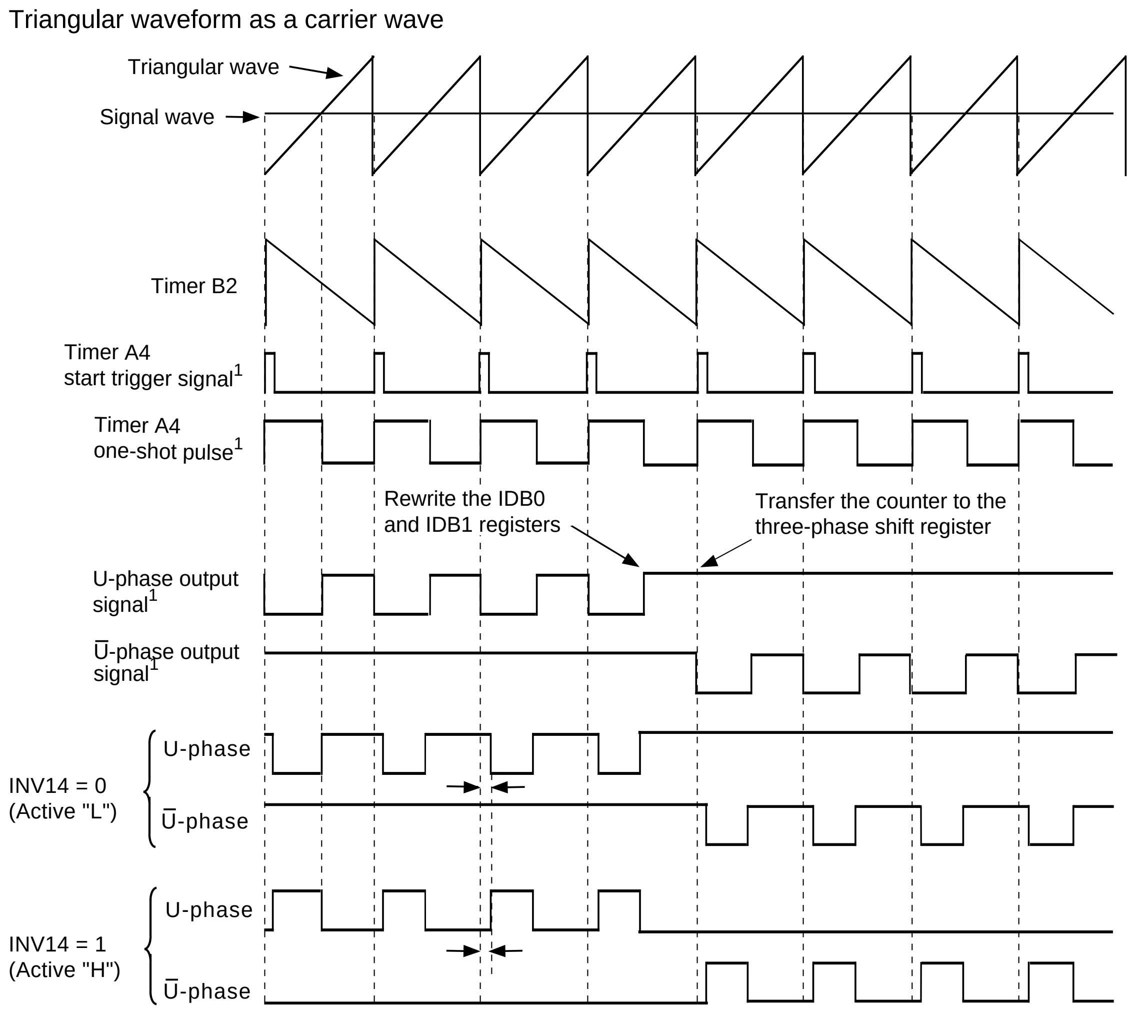
(1) When INV11=1 (three-phase mode 1)

- INV01=0 and ICTB2=2$_{16}$ (The timer B2 interrupt is generated whenever the second timer B2 underflow occurs) or INV01=1, INV00=1and ICTB2=1$_{16}$ (The timer B2 interrupt is generated whenever the timer B2 evenly underflows)
- Dealt value of the timer TA41=$\overline{m}$, TA4=m
  The TA4 and TA41 registers are changed whenever the timer B2 interrupt is generated.
  First time TA41=$\overline{n}$, TA4=n. Second time TA41=$\overline{p}$, TA4=p.
- Default value of the IDB0 and IDB1 registers
  DU0=1, DUB0=0, DU1=0, DUB1=1
  Changed to DU0=1, DUB0=0, DU1=1, DUB1=0 by the third timer B2 interrupt.

(2) When INV11=0 (three-phase mode 0)

- INV01=0, ICTB2=1$_{16}$ (The timer B2 interrupt is generated whenever the timer B2 underflows)
- Default value of the timer TA4=$\overline{m}$
  The TA4 register is changed whenever the timer B2 interrupt is generated.
  First time TA4=m. Second time TA4=$\overline{n}$. Third time TA4=n
  Fourth time TA=$\overline{p}$. Fifth time=p.
- Default value of the IDB0 and IDB1 registers
  DU0=1, DUB0=0, DU1=0, DUB1=1
  Changed to DU0=1, DUB0=0, DU1=1, DUB1=0 by the sixth timer B2 interrupt.

Figure 1.16.8. Trianglar Wave Modulation Operation



Notes :

1. Internal signals. See Figure 1.16.1.

The above applies when INVC0 = $01XX110X_2$ and INVC1 = $110XXX00_2$ (X: Varies depending on each system.)
The examples of PWM output change are
- ICTB2=*n* (The timer B2 interrupt is generated whenever the *n*th timer B2 underflows occur.)
- Default value of the IDB0 and IDB1 registers DU0=1, DUB0=0, DU1=0, DUB1=0
  Changed to DU0=1, DUB0=0, DU1=1, DUB1=1 by the timer B2 interrupt.

Figure 1.16.9. Sawtooth Wave Modulation Operation

Serial I/O

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

Each UARTi (i=0 to 4) has an exclusive timer to generate the transfer clock and operates independently.

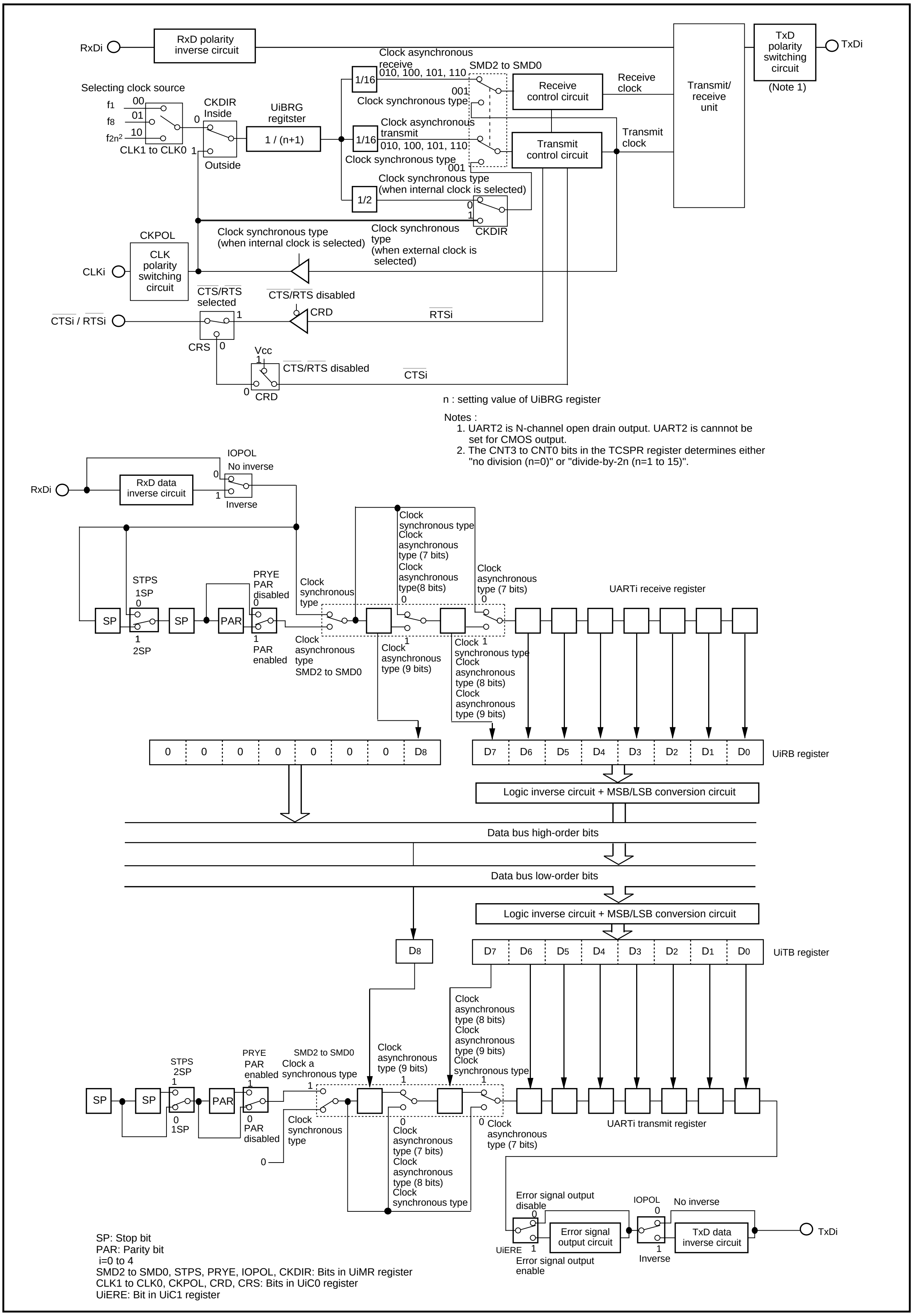
Figure 1.17.1 shows a UARTi block diagram.

UARTi 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 UARTi.

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



n : setting value of UiBRG register

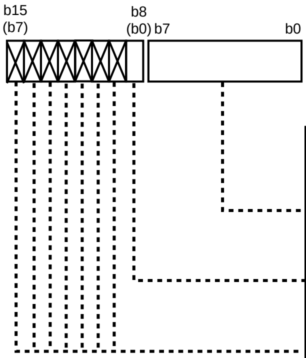
Notes :

1. UART2 is N-channel open drain output. UART2 is cannnot be set for CMOS output.
2. The CNT3 to CNT0 bits in the TCSPR register determines either "no division (n=0)" or "divide-by-2n (n=1 to 15)".

SP: Stop bit
PAR: Parity bit
i=0 to 4
SMD2 to SMD0, STPS, PRYE, IOPOL, CKDIR: Bits in UiMR register
CLK1 to CLK0, CKPOL, CRD, CRS: Bits in UiC0 register
UiERE: Bit in UiC1 register

Figure 1.17.1. UARTi Block Diagram

UARTi transmit buffer register (i=0 to 4)[1]

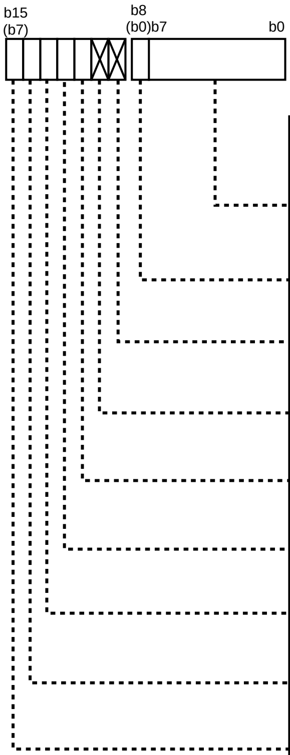


Symbol	Address	When reset
U0TB to U2TB	$036B_{16}$-$036A_{16}$, $02EB_{16}$-$02EA_{16}$, $033B_{16}$-$033A_{16}$	Indeterminate
U3TB, U4TB	$032B_{16}$-$032A_{16}$, $02FB_{16}$-$02FA_{16}$	Indeterminate

Bit symbol	Function	RW
——	Data to be transmitted (D_7 to D_0)	WO
——	Data to be transmitted (D_8)	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.

UARTi receive buffer register (i=0 to 4)



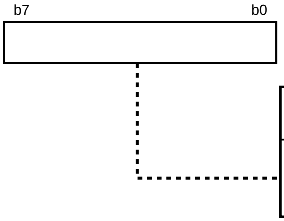
Symbol	Address	When reset
U0RB to U2RB	$036F_{16}$ - $036E_{16}$, $02EF_{16}$ - $02EE_{16}$, $033F_{16}$ - $033E_{16}$	Indeterminate
U3RB, U4RB	$032F_{16}$ - $032E_{16}$, $02FF_{16}$ - $02FE_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
——	——	Received data (D_7 to D_0)	RO
——	——	Received data (D_8)	RO
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
——			—
ABT	Arbitration lost detect flag[1]	0: Not detected (win) 1: Detected (lose)	RW
OER	Overrun error flag[2]	0: No overrun error 1: Overrun error occurs	RO
FER	Framing error flag[2]	0: No framing error 1: Framing error occurs	RO
PER	Parity error flag[2]	0: No parity error 1: Parity error occurs	RO
SUM	Error sum flag[2]	0: No error 1: Error occurs	RO

Notes :
1. The ABT bit can be set to "0" only.
2. When the SMD2 to SMD0 bits in the UiMR register is set to "000_2" (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)[1,2]

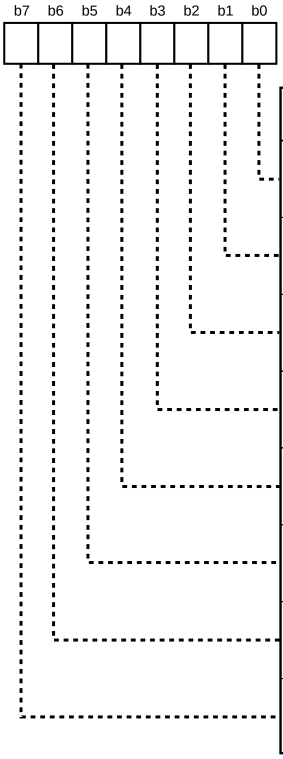


Symbol	Address	When reset
U0BRG to U4BRG	0369_{16}, $02E9_{16}$, 0339_{16}, 0329_{16}, $02F9_{16}$	Indeterminate

Function	Setting value	RW
As a setting value is n, the UiBRG register divides a count source by n+1	00_{16} to FF_{16}	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)



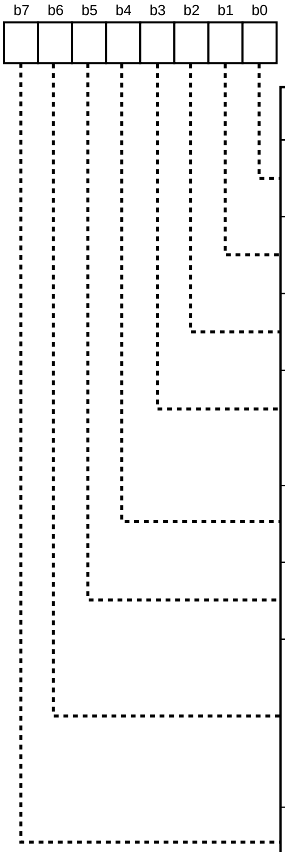
Symbol	Address	When reset
U0MR to U4MR	0368_{16}, $02E8_{16}$, 0338_{16}, 0328_{16}, $02F8_{16}$	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
SMD0	Serial I/O mode select bit	b2 b1 b0 0 0 0: Serial I/O disabled 0 0 1: Clock synchronous serial I/O mode 0 1 0: IIC mode 1 0 0: UART mode, transfer data 7 bits long 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
SMD1			RW
SMD2			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

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

Symbol	Address	When reset
U0C0 to U4C0	$036C_{16}$, $02EC_{16}$, $033C_{16}$, $032C_{16}$, $02FC_{16}$	$0000\ 1000_2$



Bit symbol	Bit name	Function	RW
CLK0	BRG count source select bit	b1 b0 0 0: Selects f_1 0 1: Selects f_8 1 0: Selects f_{2n}[2] 1 1: Avoid this setting	RW
CLK1			RW
CRS	$\overline{\text{CST}}$/$\overline{\text{RTS}}$ function select bit	Enabled when CRD=0 0 : Selects $\overline{\text{CTS}}$ function 1 : Selects $\overline{\text{RTS}}$ function	RW
TXEPT	Transmit register empty flag	0 : Data available in transmit register (during transmitting) 1 : No data available in transmit register (transmitting completed)	RO
CRD	$\overline{\text{CTS}}$/$\overline{\text{RTS}}$ disable bit	0 : $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function enabled 1 : $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function disabled	RW
NCH	Data output select bit[1]	0 : TxDi pin is CMOS output 1 : TxDi pin is N-channel open drain output	RW
CKPOL	CLK polarity select bit	0 : Data to be transmitted is output on the falling edge of the transfer clock and received data is input on the rising edge 1 : Data to be transmitted is output on the rising edge of the transfer clock and received data is input on the falling edge	RW
UFORM	Transfer format select bit	0 : LSB first 1 : MSB first	RW

Notes :

1. The $P7_0$/TxD2 pin is N-channel open drain output. $P7_0$/TxD2 pin cannot be set for CMOS output.
2. The CNT3 to CNT0 bits in the TRGSR register determines either "no division (n=0)" or "divide-by-2n (n=1 to 15)".

Figure 1.17.4. U0C0 to U4C0 Registers

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

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
U0C1 to U4C1	$036D_{16}$, $02ED_{16}$, $033D_{16}$, $032D_{16}$, $02FD_{16}$	0000 0010_2

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)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
U0SMR to U4SMR	0367_{16}, $02E7_{16}$, 0337_{16}, 0327_{16}, $02F7_{16}$	0000 0000_2

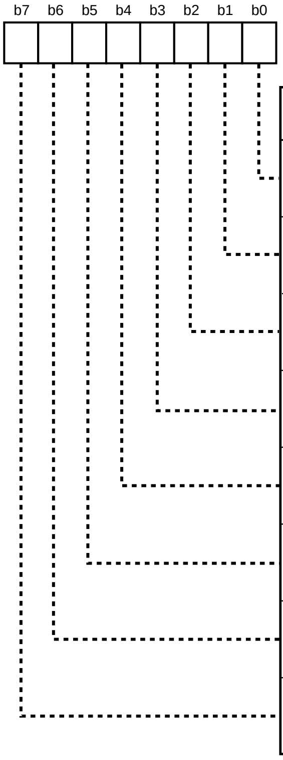
Bit symbol	Bit name	Function	RW
IICM	IIC mode select bit	0: Except IIC mode 1: IIC mode	RW
ABC	Arbitration lost detecting flag	0: Update per bit 1: Update per byte	RW
BBS	Bus busy flag	0: Stop condition detected 1: Start condition detected (Busy)	RW^1
LSYN	SCLL sync output enable bit	0: Disabled 1: Enabled	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

UARTi special mode register 2 (i=0 to 4)



Symbol	Address	When reset
U0SMR2 to U4SMR2	0366_{16}, $02E6_{16}$, 0336_{16}, 0326_{16}, $02F6_{16}$	$0000\ 0000_2$

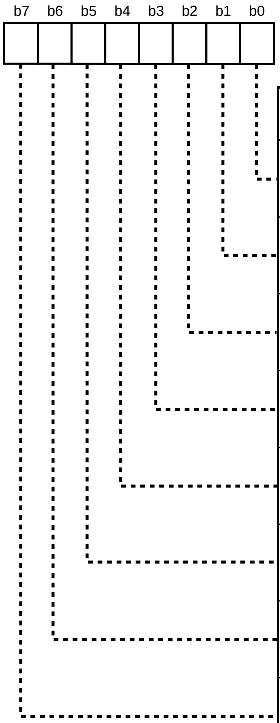
Bit symbol	Bit name	Function	RW
IICM2	IIC mode select bit 2	(Note 1)	RW
CSC	Clock synchronous bit	0: Disabled 1: Enabled	RW
SWC	SCL wait output bit	0: Disabled 1: Enabled	RW
ALS	SDA output stop bit	0: Disabled 1: Enabled	RW
STC	UARTi initialize bit	0: Disabled 1: Enabled	RW
SWC2	SCL wait output bit 2	0: Transfer clock 1: 0 output	RW
SDHI	SDA output inhibit bit	0: Enabled 1: Disabled (high-impedance)	RW
SU1HIM	External clock synchronous enable bit	(Note 2)	RW

Notes :
1. Refer to the "Special mode 1."
2. The external clock synchronous function can be selected in a combination with the SCLKDIV bit in the UiSMR register.

SCLKDIV bit in the UiSMR register	SU1HIM bit in the UiSMR2 register	External clock synchronous function that can be selected
0	0	No synchronization
0	1	Same division as an external clock
1	0 or 1	The external clock divided by 2

Figure 1.17.6. U0SMR2 to U4SMR2 Registers

UARTi special mode register 3 (i=0 to 4)



Symbol	Address	When reset
U0SMR3 to U4SMR3	0365_{16}, $02E5_{16}$, 0335_{16}, 0325_{16}, $02F5_{16}$	0000 0000_2

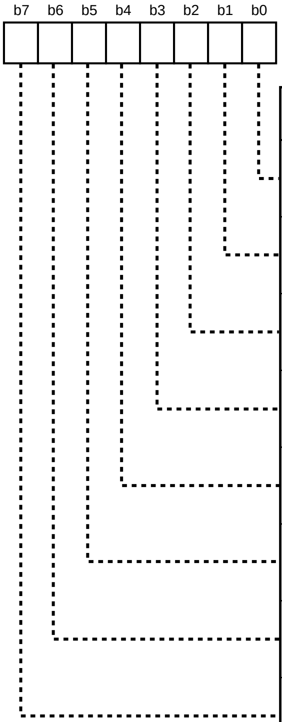
Bit symbol	Bit name	Function	RW
SSE	$\overline{SS}$ pin function enable bit[1]	0: $\overline{SS}$ function disabled 1: $\overline{SS}$ function enabled	RW
CKPH	Clock-phase set bit	0: No clock delay 1: Clock delay	RW
DINC	Serial input port set bit	0: Selects TxDi and RxDi (master mode) 1: Selects STxDi and SRxDi (slave mode)	RW
NODC	Clock output select bit	0: CLKi is CMOS output 1: CLKi is N-channel open drain output	RW
ERR	Fault error flag	0: No fault error 1: Fault error (Note 2)	RW
DL0	SDAi digital delay time set bit[3,4]	b7 b6 b5 000 : No delay 001 : 1-to-2 cycles of BRG count source 010 : 2-to-3 cycles of BRG count source 011 : 3-to-4 cycles of BRG count source 100 : 4-to-5 cycles of BRG count source 101 : 5-to-6 cycles of BRG count source 110 : 6-to-7 cycles of BRG count source 111 : 7-to-8 cycles of BRG count source	RW
DL1			RW
DL2			RW

Notes :

1. $\overline{SS}$ function should be set after the CRD bit in the UiC0 register is set to "1" ($\overline{CTS}$/$\overline{RTS}$ function disabled).
2. The ERR bit is set to "0" when setting it to "0" by program. It remains unchanged if setting it to "1".
3. Digital delay is generated from a SDAi output by the DL2 to DL0 bits in IIC mode. These bis should set to "000_2" (no delay) except in the IIC mode.
4. When selecting the external clock, a delay approximately 100ns is added.

Figure 1.17.7. U0SMR3 to U4SMR3 Registers

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



Symbol	Address	When reset
U0SMR4 to U04SMR4	0364_{16}, $02E4_{16}$, 0334_{16}, 0324_{16}, $02F4_{16}$	$0000\ 0000_2$

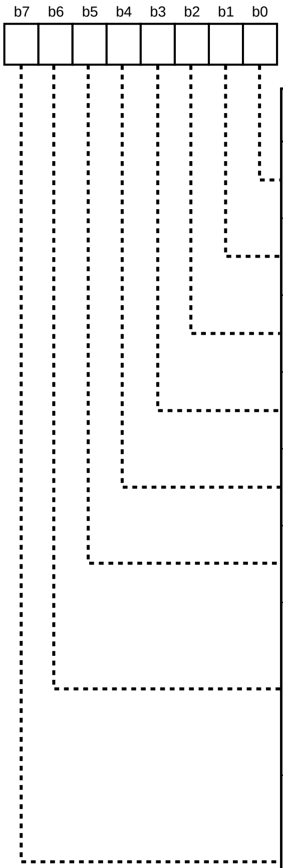
Bit symbol	Bit name	Function	RW
STAREQ	Start condition generate bit[1]	0: Clear 1: Start	RW
RSTAREQ	Restart condition generate bit[1]	0: Clear 1: Start	RW
STPREQ	Stop condition generate bit[1]	0: Clear 1: Start	RW
STSPSEL	SCL, SDA output select bit	0: Selects serial I/O circuit 1: Selects start/stop condition generation circuit	RW
ACKD	ACK data bit	0: ACK 1: NACK	RW
ACKC	ACK data output enable bit	0: Serial I/O data output 1: ACK data output	RW
SCLHI	SCL output stop enable bit	0: Disabled 1: Enabled	RW
SWC9	SCL wait output bit 3	0: SCL "L" hold disable 1: SCL "L" hold enable	RW

Notes :

1. When each condition is generated, this bit is set to "0". When a condition generation is incomplete, this bit is remains unchanged in "1".

Figure 1.17.8. U0SMR4 to U4SMR4 Registers

External interrupt request cause select register



Symbol	Address	When reset
IFSR	031F$_{16}$	0000 0000$_2$

Bit symbol	Bit name	Function	RW
IFSR0	INT0 interrupt polarity select bit[1]	0 : One edge 1 : Both edges	RW
IFSR1	INT1 interrupt polarity select bit[1]	0 : One edge 1 : Both edges	RW
IFSR2	INT2 interrupt polarity select bit[1]	0 : One edge 1 : Both edges	RW
IFSR3	INT3 interrupt polarity select bit[1]	0 : One edge 1 : Both edges	RW
IFSR4	INT4 interrupt polarity select bit[1]	0 : One edge 1 : Both edges	RW
IFSR5	INT5 interrupt polarity select bit[1]	0 : One edge 1 : Both edges	RW
IFSR6	UART0, 3 interrupt cause select bit	0 : UART3 bus conflict, start condition detect, stop condition detect, fault error detect 1 : UART0 bus conflict, start condition detect, stop condition detect, fault error detect	RW
IFSR7	UART1, 4 interrupt cause select bit	0 : UART4 bus conflict, start condition detect, stop condition detect, fault error detect 1 : UART1 bus conflict, start condition detect, stop condition detect, fault error detect	RW

Notes :

1.When selecting a level sense, this bit should be set to "0".
When selecting both edges, the POL bit in the INTiIC register (i = 0 to 5) should be set to "0" (falling edge).

Figure 1.17.9. IFSR Register

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 UARTi (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): $\frac{f_j}{2(m+1)}$ $f_j=f_1, f_8, f_{2n}$[1] m :setting value of the UiBRG register. 00_{16} to FF_{16}. • CKDIR bit is set to "1" (external clock is selected) : input from CLKi pin
Transmit/receive control	• Selectable from the $\overline{\text{CTS}}$ function, $\overline{\text{RTS}}$ function or $\overline{\text{CTS}}$/$\overline{\text{RTS}}$ function disabled
Transmit start condition	• To start transmitting, the following requirements should be met[2]: - 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 $\overline{\text{CTSi}}$ input level is in "L" when the $\overline{\text{CTS}}$ function is selected
Receive start condition	• To start receiving, the following requirements must be met[2]: - 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 UARTi transmit register (transfer starts) - The UiIRS bit is set to "1" (transmission completes) : when data transfer from the UARTi transmit register is completed • While receiving When data is transferred from the UARTi receive register to the UiRB register (reception completes)
Error detection	• Overrun error[3] 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-2n (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 CLKi 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 CLKi 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).

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	Overrun error flag
UiBRG	0 to 7	Set a baud rate
UiMR	SMD2 to SMD0	Set to "001$_2$"
	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 $\overline{\text{CTS}}$ or $\overline{\text{RTS}}$ when using one of them
	TXEPT	Transmit register empty flag
	CRD	Select the $\overline{\text{CTS}}$ or $\overline{\text{RTS}}$ 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

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

Port	Function	Setting value		
		PS0 register	PSL0 register	PD6 register
P6$_0$	$\overline{\text{CTS0}}$ input	PS0_0=0	-	PD6_0=0
	$\overline{\text{RTS0}}$ output	PS0_0=1	-	-
P6$_1$	CLK0 input	PS0_1=0	-	PD6_1=0
	CLK0 output	PS0_1=1	-	-
P6$_2$	RxD0 input	PS0_2=0	-	PD6_2=0
P6$_3$	TxD0 output	PS0_3=1	-	-
P6$_4$	$\overline{\text{CTS1}}$ input	PS0_4=0	-	PD6_3=0
	$\overline{\text{RTS1}}$ output	PS0_4=1	PSL0_4=0	-
P6$_5$	CLK1 input	PS0_5=0	-	PD6_5=0
	CLK1 output	PS0_5=1	-	-
P6$_6$	RxD1 input	PS0_6=0	-	PD6_6=0
P6$_7$	TxD1 output	PS0_7=1	-	-

Table 1.18.4. Pin Settings (Continued)

Port	Function	Setting value			
		PS1 register	PSL1 register	PSC register	PD7 register
P7$_0$[1]	TxD2 output	PS1_0=1	PSL1_0=0	PSC_0=0	-
P7$_1$[1]	RxD2 input	PS1_1=0	-	-	PD7_1=0
P7$_2$	CLK2 input	PS1_2=0	-	-	PD7_2=0
	CLK2 output	PS1_2=1	PSL1_2=0	PSC_2=0	-
P7$_3$	$\overline{\text{CTS2}}$ input	PS1_3=0	-	-	PD7_3=0
	$\overline{\text{RTS2}}$ output	PS1_3=1	PSL1_3=0	PSC_3=0	-

Notes :

1. N-channel open drain output

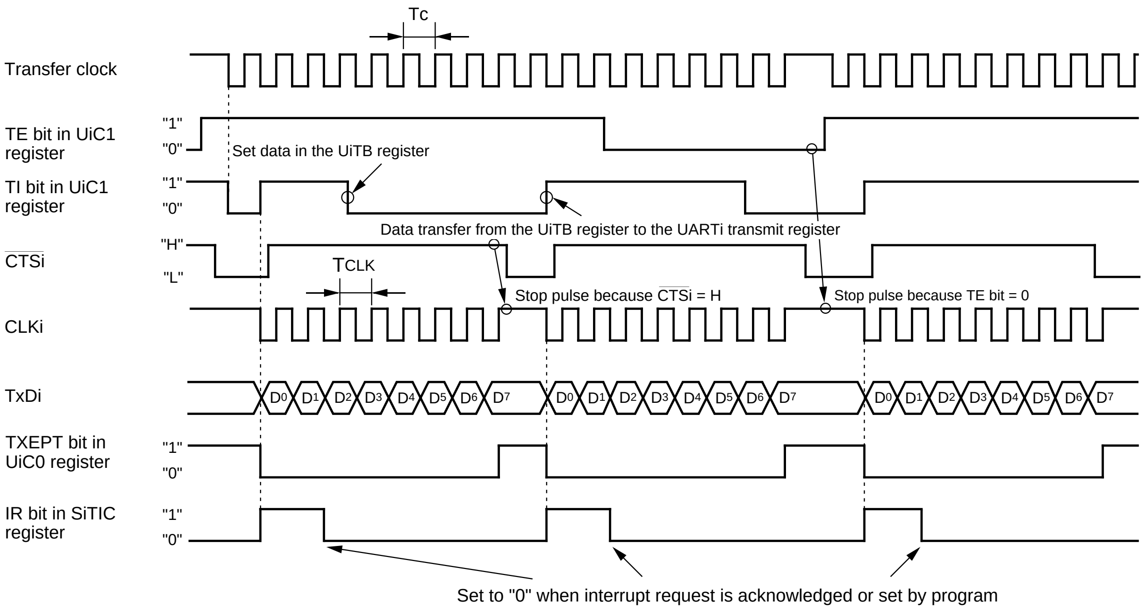
Table 1.18.5. Pin Settings (Continued)

Port	Function	Setting value		
		PS3 register[1]	PSL3 register	PD9 register[1]
P9$_0$	CLK3 input	PS3_0=0	-	PD9_0=0
	CLK3 output	PS3_0=1	-	-
P9$_1$	RxD3 input	PS3_1=0	-	PD9_1=0
P9$_2$	TxD3 output	PS3_2=1	PSL3_2=0	-
P9$_3$	$\overline{\text{CTS3}}$ input	PS3_3=0	PSL3_3=0	PD9_3=0
	$\overline{\text{RTS3}}$ output	PS3_3=1	-	-
P9$_4$	$\overline{\text{CTS4}}$ input	PS3_4=0	PSL3_4=0	PD9_4=0
	$\overline{\text{RTS4}}$ output	PS3_4=1	-	-
P9$_5$	$\overline{\text{CLK4}}$ input	PS3_4=0	PSL3_5=0	PD9_5=0
	$\overline{\text{CLK4}}$ output	PS3_5=1	-	-
P9$_6$	TxD4 output	PS3_6=1	-	-
P9$_7$	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.

(1) Transmit timing (with the internal clock)


Tc
Transfer clock
TE bit in UiC1 register
"1"
"0"
Set data in the UiTB register
TI bit in UiC1 register
"1"
"0"
Data transfer from the UiTB register to the UARTi transmit register
CTSi
"H"
"L"
TCLK
Stop pulse because CTSi = H
Stop pulse because TE bit = 0
CLKi
TxDi
D0 D1 D2 D3 D4 D5 D6 D7
TXEPT bit in UiC0 register
"1"
"0"
IR bit in SiTIC register
"1"
"0"
Set to "0" when interrupt request is acknowledged or set by program

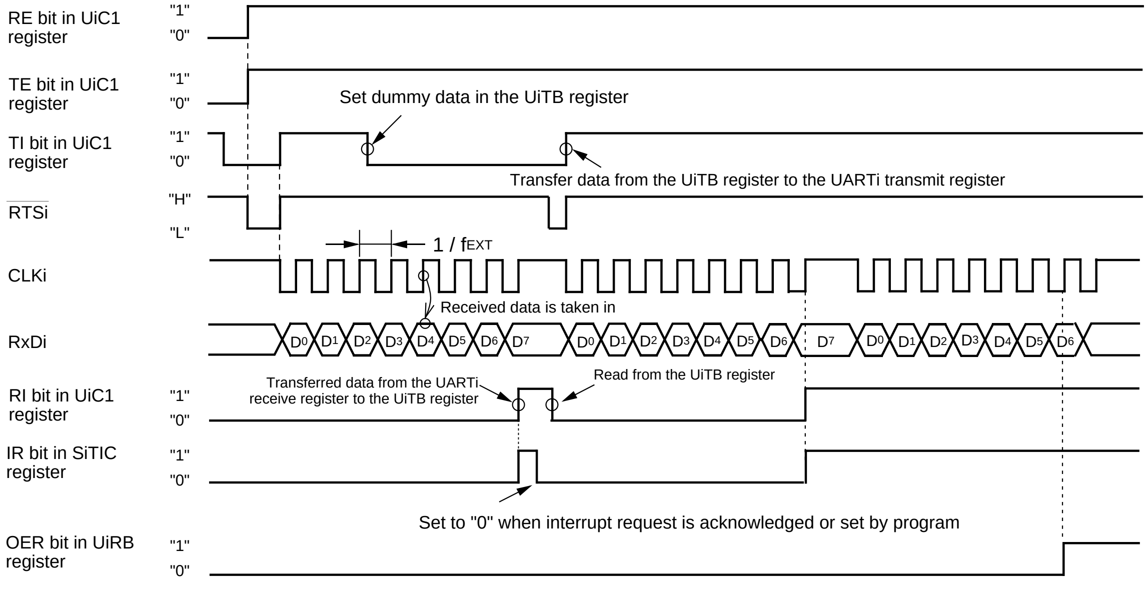
The above applies to the following settings:
- The CKDIR bit in the UiMR register is set to "0" (internal clock selected)
- The CRD bit in the UiC0 register is set to "0" (RTS/CTS function enabled) The CRS bit is set to "0" (CTS function selected)
- The CKPOL bit the in UiC0 register is set to "0" (data to be transmitted is output on the falling edge of the transfer clock)
- The UiRS bit in the UiC1 register is set to "0" (interrupt request generated due to a empty UiTB register)

$T_C = T_{CLK} = 2(m+1)/fj$

fj : a count source frequency set in the UiBRG register (f_1, f_8, f_{2n}[2])
m : setting value of the UiBRG register
i = 0 to 4

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

(2) Recieve timing (with the external clock)


RE bit in UiC1 register
"1"
"0"
TE bit in UiC1 register
"1"
"0"
Set dummy data in the UiTB register
TI bit in UiC1 register
"1"
"0"
Transfer data from the UiTB register to the UARTi transmit register
RTSi
"H"
"L"
1 / fEXT
CLKi
Received data is taken in
RxDi
D0 D1 D2 D3 D4 D5 D6 D7
RI bit in UiC1 register
"1"
"0"
Transferred data from the UARTi receive register to the UiTB register
Read from the UiTB register
IR bit in SiTIC register
"1"
"0"
Set to "0" when interrupt request is acknowledged or set by program
OER bit in UiRB register
"1"
"0"

The above applies to the following settings:
- The CKDIR bit in the UiMR register is set to "1" (external clock selected)
- The CRD bit in the UiC0 register is set to "0" (RTS/CTS fiction enabled) The CRS bit is set to "1" (RTS function selected)
- The CKPOL bit in the UiC0 register is set to "0" (Received data is input on the rising edge of the transfer clock)

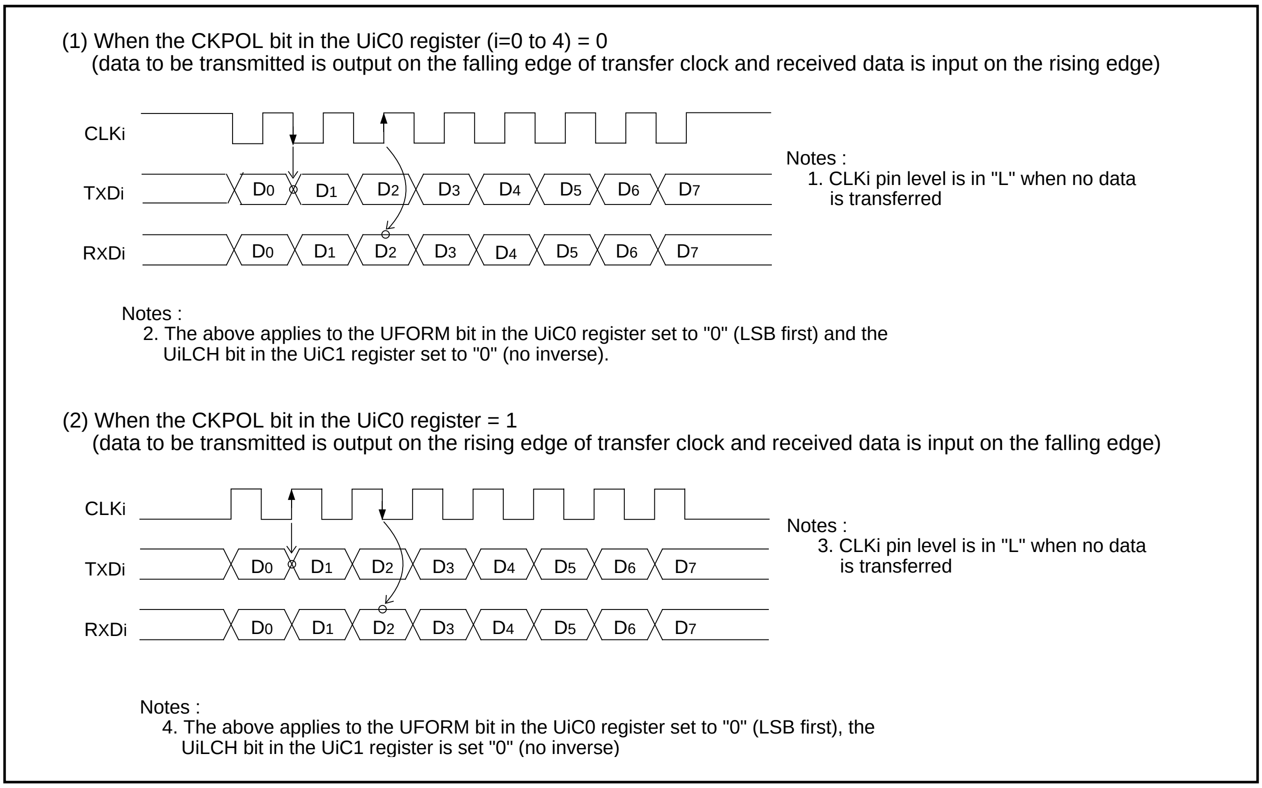
f_{EXT}: frequency of the external clock i=0 to 4

The following conditions are met when the CLKi pin input is in "H" before data in received:
- TE bit in the UiC1 register is set to "1" (transmit enabled)
- RE bit in the UiC1 register is set to "1" (recieve enabled)
- Dummy data in written to UiTB register

Figure 1.18.1. Transmit and receive Operation

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.


(1) When the CKPOL bit in the UiC0 register (i=0 to 4) = 0
(data to be transmitted is output on the falling edge of transfer clock and received data is input on the rising edge)

Notes :
1. CLKi pin level is in "L" when no data is transferred

Notes :
2. The above applies to the UFORM bit in the UiC0 register set to "0" (LSB first) and the UiLCH bit in the UiC1 register set to "0" (no inverse).

(2) When the CKPOL bit in the UiC0 register = 1
(data to be transmitted is output on the rising edge of transfer clock and received data is input on the falling edge)

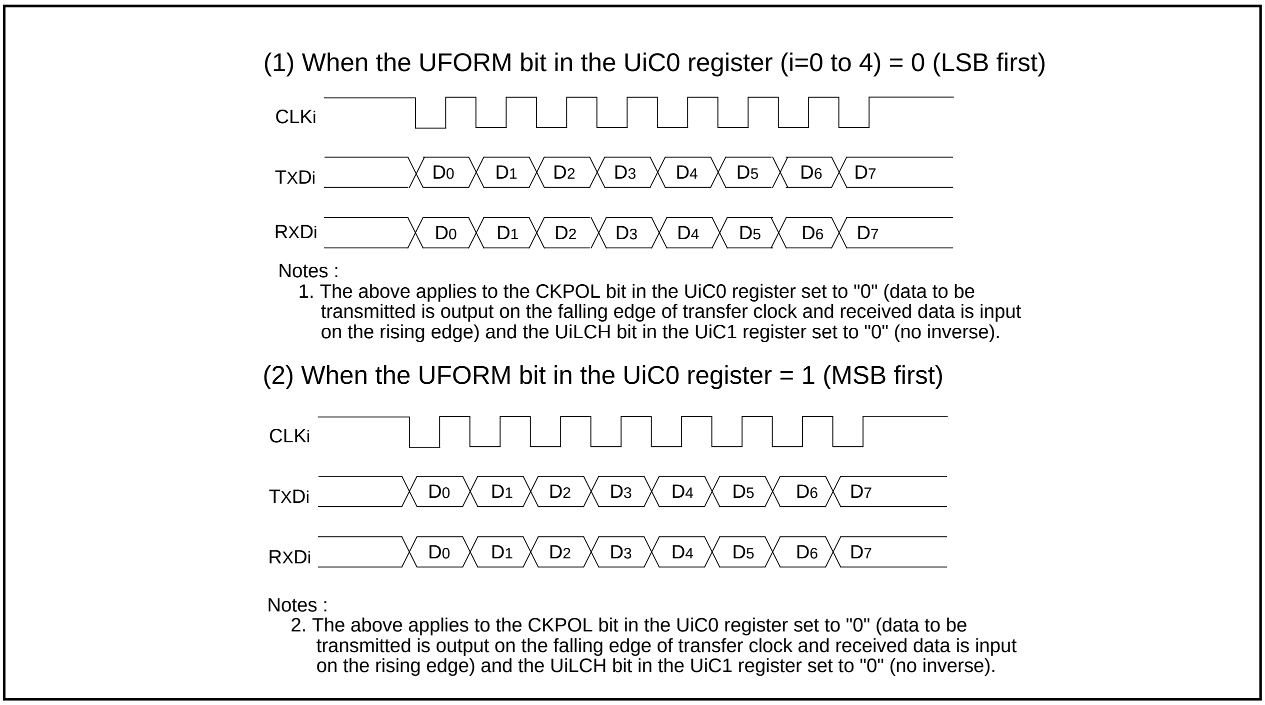
Notes :
3. CLKi pin level is in "L" when no data is transferred

Notes :
4. The above applies to the UFORM bit in the UiC0 register set to "0" (LSB first), the UiLCH bit in the UiC1 register is set "0" (no inverse)

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.


(1) When the UFORM bit in the UiC0 register (i=0 to 4) = 0 (LSB first)

Notes :
1. The above applies to the CKPOL bit in the UiC0 register set to "0" (data to be transmitted is output on the falling edge of transfer clock and received data is input on the rising edge) and the UiLCH bit in the UiC1 register set to "0" (no inverse).

(2) When the UFORM bit in the UiC0 register = 1 (MSB first)

Notes :
2. The above applies to the CKPOL bit in the UiC0 register set to "0" (data to be transmitted is output on the falling edge of transfer clock and received data is input on the rising edge) and the UiLCH bit in the UiC1 register set to "0" (no inverse).

Figure 1.18.3. Transfer Format

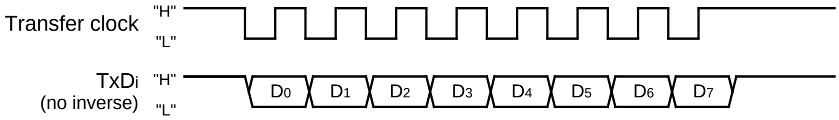
3. Continuous Receive Mode

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.

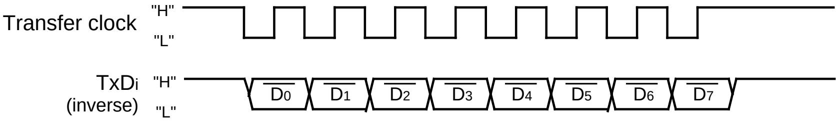
4. Serial Data Logic Inverse Function

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

(1) When the UiLCH bit in the UiC1 register (i=o to 4) = 0 (no inverse)



(2) When the UiLCH bit in the UiC1 register = 1 (inverse)



Notes :
1. The above applies to the CKPOL bit in the UiC0 register set to "0" (data to be transmitted is output on the falling edge and received data is input on the rising edge) and the UFORM bit in the UiC register set to "0" (LSB first).

Figure 1.18.4. Switching Serial Data Logic Switching

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	• character bit (transfer data) : selectable from 7 bits, 8 bits, or 9 bits • Start bit: 1 bit • Parity bit: selectable from Odd, even, or nothing • Stop bit: selectable from 1 bit or 2 bits
Transfer clock	• When the CKDIR bit in the UiMR register is set to "0" (internal clock is selected) : $f_j/16(m+1)$ $f_j = f_1, f_8, f_{2n}$[1] m:setting value of the UiBRG register 00_{16} to FF_{16} • When the CKDIR bit is set to "1" (external clock is selected) : $f_{EXT}/16(m+1)$ f_{EXT}: input clock from the CLKi pin
Transmit/receive control	• Selectable from $\overline{\text{CTS}}$ function, $\overline{\text{RTS}}$ function or $\overline{\text{CTS}}/\overline{\text{RTS}}$ function disabled
Transmit start condition	• To start transmitting, the following requirements should be met: - 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 transmit buffer register) - $\overline{\text{CTS}}$ input level is in "L" when $\overline{\text{CTS}}$ function is selected
Receive start condition	• To start receiving, the following requirements should be met: - The RE bit in the UiC1 register is set to "1" (receive enable) - The start bit is detected
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 UARTi transmit register (starts transfer) - The UiIRS bit is set to "1" (transmission completes) : when data transmission from the UARTi transfer register is completed • While receiving - When data is transferred from UARTi receive register to UiRB register (reception comples)
Error detection	• Overrun error[2] 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) • Framing error This error occurs when the number of stop bits set is not detected • 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 • Error sum flag This flag is set to "1" when any of an overrun, framing or parity errors occur
Selectable function	• LSB first or MSB first Whether a data is transmitted/received in bit 0 or in bit 7 can be selected •Serial data logic inverse Logic values of data to be transmitted and received data are inversed. The start bit and stop bit are not inversed •TxD, RxD I/O polarity switching TxD pin output and RxD pin input are inversed. All I/O data levels are also inversed.

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 indetermine. The IR bit in the SiRIC register remains unchanged in "0" (interrupt request).

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[1]
UiRB	0 to 8	Received data can be read[1]
	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 "100_2" When transfer data is 8 bits long, set to "101_2" When transfer data is 9 bits long, set to "110_2"
	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 reigster
	CRS	Select either $\overline{CTS}$ or $\overline{RTS}$ when using one of them
	TXEPT	Transfer register empty flag
	CRD	Select the $\overline{CTS}$ or $\overline{RTS}$ 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.

Table 1.19.3. Pin Settings in UART

Port	Function	Bit and setting value		
		PS0 register	PSL0 register	PD6 register
P60	$\overline{\text{CTS0}}$ input	PS0_0=0	–	PD6_0=0
	$\overline{\text{RTS0}}$ 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	$\overline{\text{CTS1}}$ input	PS0_4=0	–	PD6_3=0
	$\overline{\text{RTS1}}$ 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[1]	TxD2 output	PS1_0=1	PSL1_0=0	PSC_0=0	–
P71[1]	RxD2 input	PS1_1=0	–	–	PD7_1=0
P72	CLK2 input	PS1_2=0	–	–	PD7_2=0
P73	$\overline{\text{CTS2}}$ input	PS1_3=0	–	–	PD7_3=0
	$\overline{\text{RTS2}}$ output	PS1_3=1	PSL1_3=0	PSC_3=0	–

Notes :

1. N-channel open drain output

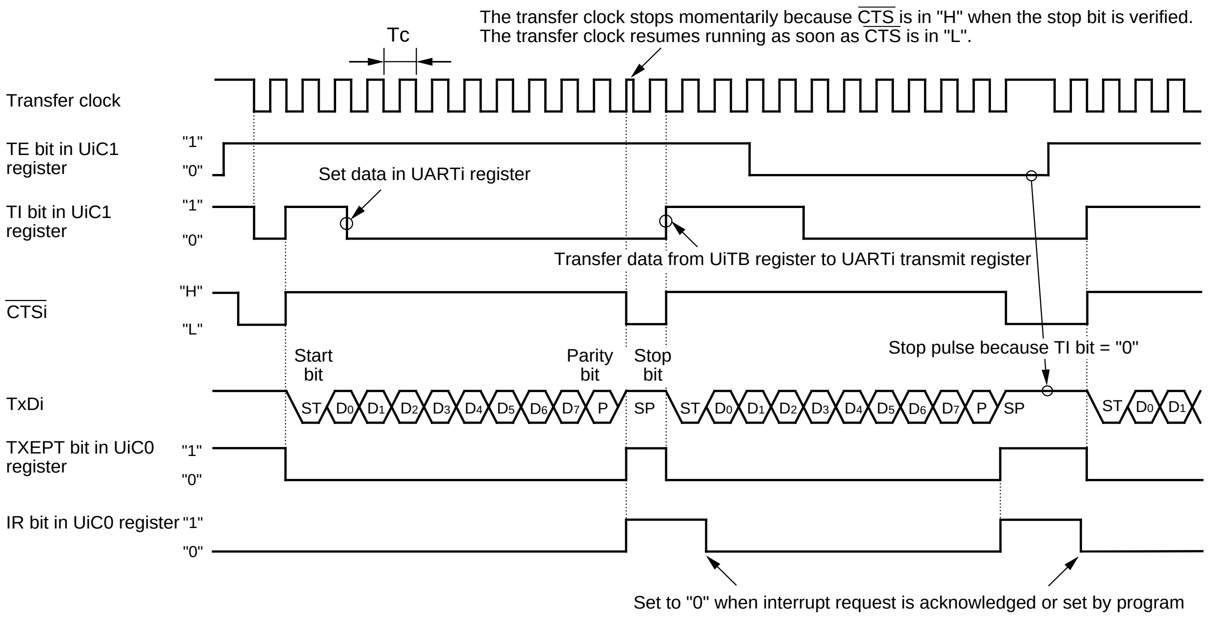
Table 1.19.5. Pin Settings (Continued)

Port	Function	Bit and setting value		
		PS3 register[1]	PSL3 register	PD9 register[1]
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	$\overline{\text{CTS3}}$ input	PS3_3=0	PSL3_3=0	PD9_3=0
	$\overline{\text{RTS3}}$ output	PS3_3=1	–	–
P94	$\overline{\text{CTS4}}$ input	PS3_4=0	PSL3_4=0	PD9_4=0
	$\overline{\text{RTS4}}$ 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.

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


The transfer clock stops momentarily because CTS is in "H" when the stop bit is verified.
The transfer clock resumes running as soon as CTS is in "L".
Tc
Transfer clock
TE bit in UiC1 register
"1"
"0"
Set data in UARTi register
TI bit in UiC1 register
"1"
"0"
Transfer data from UiTB register to UARTi transmit register
CTSi
"H"
"L"
Start bit
Parity bit
Stop bit
Stop pulse because TI bit = "0"
TxDi
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0 D1 D2 D3 D4 D5 D6 D7 P SP ST D0 D1
TXEPT bit in UiC0 register
"1"
"0"
IR bit in UiC0 register
"1"
"0"
Set to "0" when interrupt request is acknowledged or set by program

i=0 to 4

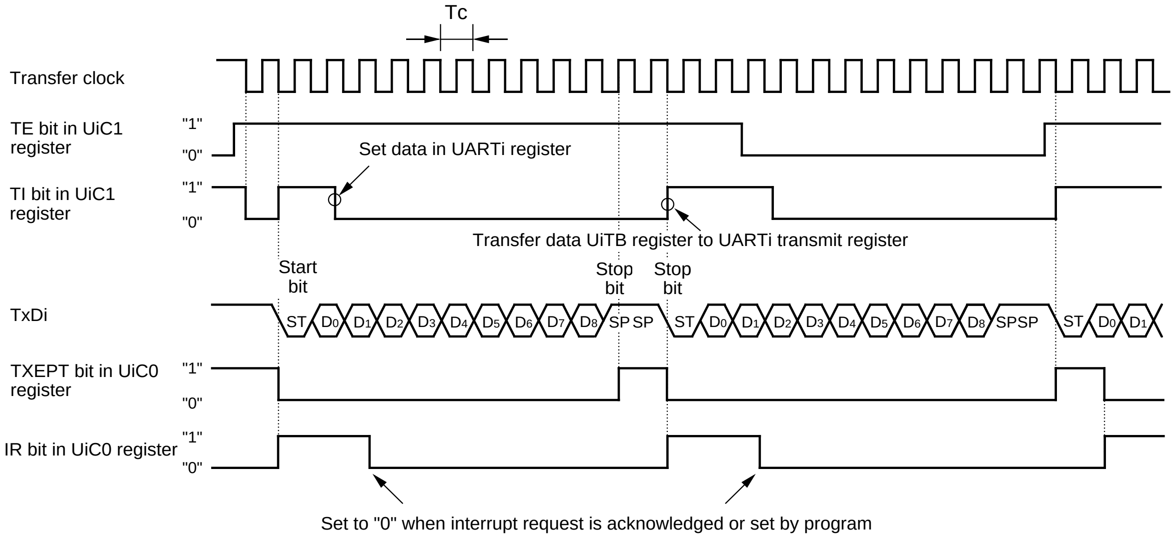
The above timing applies to the following settings :

- 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)

Tc = 16 (m + 1) / fj or 16 (m + 1) / fEXT
fj : a count source frequency set in the UiBRG register (f1, f8, f2n1
fEXT : 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 "
division (n=0)" or "divide-by-2n (n=1 to 15)".

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


Tc
Transfer clock
TE bit in UiC1 register
"1"
"0"
Set data in UARTi register
TI bit in UiC1 register
"1"
"0"
Transfer data UiTB register to UARTi transmit register
Start bit
Stop bit
Stop bit
TxDi
ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0 D1 D2 D3 D4 D5 D6 D7 D8 SP SP ST D0 D1
TXEPT bit in UiC0 register
"1"
"0"
IR bit in UiC0 register
"1"
"0"
Set to "0" when interrupt request is acknowledged or set by program

i=0 to 4

The above timing applies to the following settings :

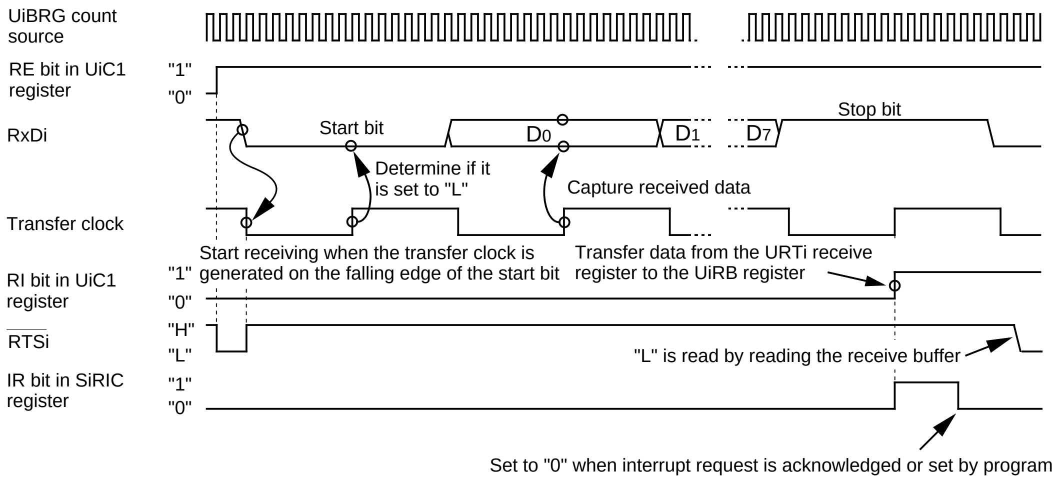
- The PRYE bit in the UiMR register is set to "0" (parity disabled)
- The SRPS bit in the UiMR register is set to "1" (2-stop bit)
- The CRD bit in the UiC0 register is set to "1" (CTS function disabled)
- The UiIRS bit in the UiC1 register is set to "1" (interrupt request is generated due to the UiTB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

Tc = 16 (m + 1) / fj or 16 (m + 1) / fEXT
fj : a count source frequency set in the UiBRG register (f1, f8, f2n1
fEXT : 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
division (n=0)" or "divide-by-2n (n=1 to 15)".

Figure 1.19.1. Transmit Operation

8-bit data receive timing (with parity disabled and 1 stop bit)


UiBRG count source
RE bit in UiC1 register
"1"
"0"
RxDi
Start bit
D0
D1
D7
Stop bit
Determine if it is set to "L"
Capture received data
Transfer clock
Start receiving when the transfer clock is generated on the falling edge of the start bit
Transfer data from the URTi receive register to the UiRB register
RI bit in UiC1 register
"1"
"0"
RTSi
"H"
"L"
"L" is read by reading the receive buffer
IR bit in SiRIC register
"1"
"0"
Set to "0" when interrupt request is acknowledged or set by program

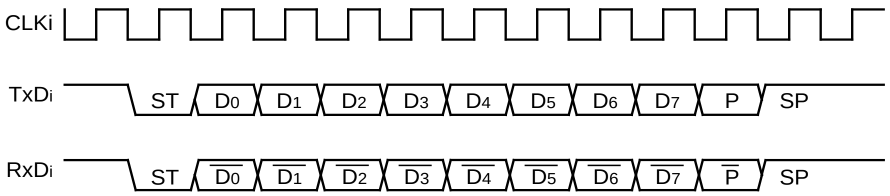
i=0 to 4
Notes :
1. The above applies to the PRYE bit in the UiMR register set to "0" (parity disabled), the SRPS bit in the UiMR register set to "0" (1 stop bit) and the CRS bit in the UiC0 register set to "1" (RTS function selected).

Figure 1.19.2. Receive Operation

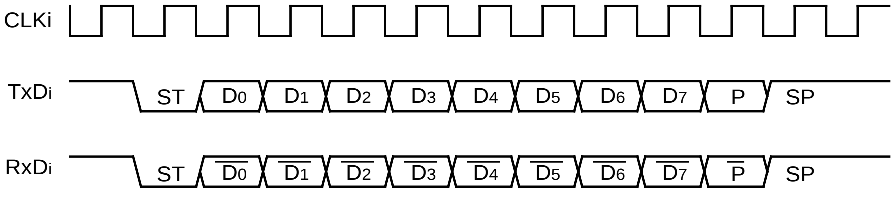
1. LSB First / MSB First Select Function

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.

(1) When the UFORM Bit in the UiC0 Register (i=0 to 4) = 0 (LSB first)



(1) When the UFORM Bit in the UiC0 Register = 1 (MSB first)



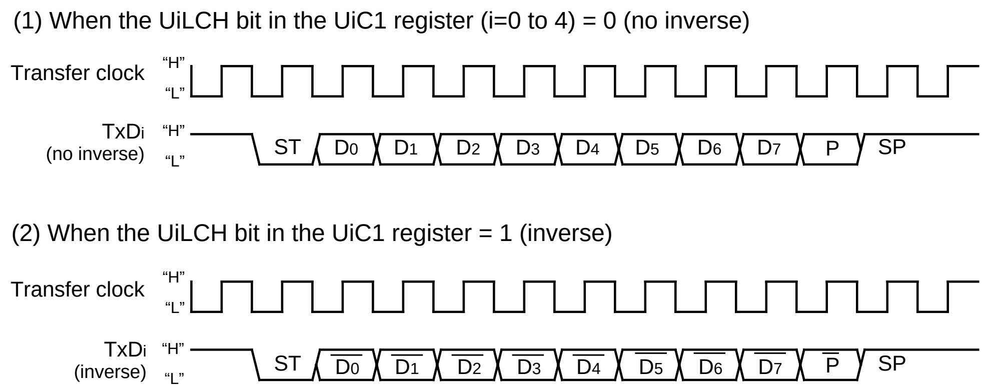
Notes :
1. The above applies to the CKPOL bit in the UiC0 register set to "0" (data to be transmitted is output on the falling edge of the transfer clock and received data is input on the rising edge) and the UiLCH bit in the UiC1 register set to "0" (no inverse).

ST : Start bit
P : Even parity
SP : Stop bit

Figure 1.19.3. Transfer Format

2. Serial Data Logic Inverse Function

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


(1) When the UiLCH bit in the UiC1 register (i=0 to 4) = 0 (no inverse)

(2) When the UiLCH bit in the UiC1 register = 1 (inverse)

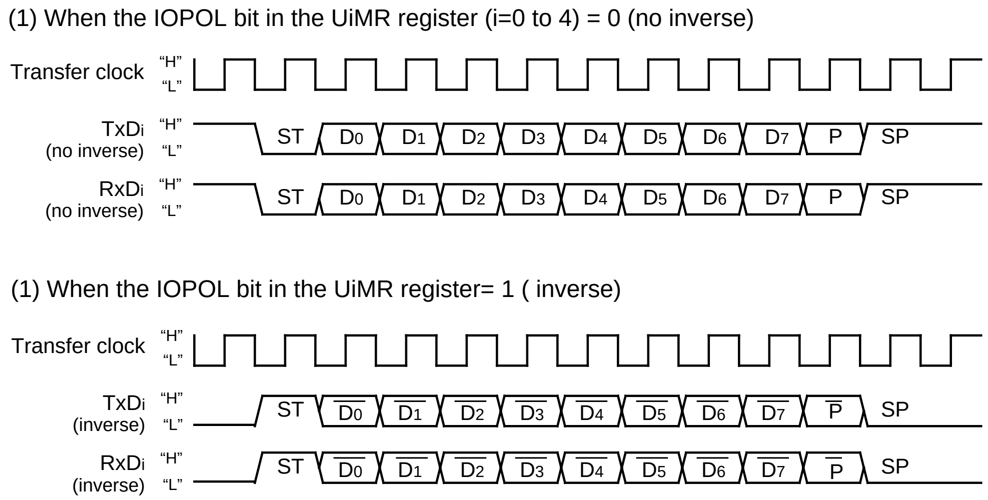
Notes :

1. The above applies to the UFORM bit in the UiC0 register set to "0" (LSB first) the STPS bit in the UiMR bit set to "0" (1 stop bit) and the PRYE bit set to "1" (parity enabled).

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


(1) When the IOPOL bit in the UiMR register (i=0 to 4) = 0 (no inverse)

(1) When the IOPOL bit in the UiMR register= 1 (inverse)

Notes :

1. The above applies to the UFORM bit in the UiC0 register set to "0" (LSB first), the STPS bit in the UiMR bit set to "0" (1 stop bit) and the PRYE bit set to "1" (parity enabled).

ST : Start bit
P : Even parity
SP : Stop bit

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

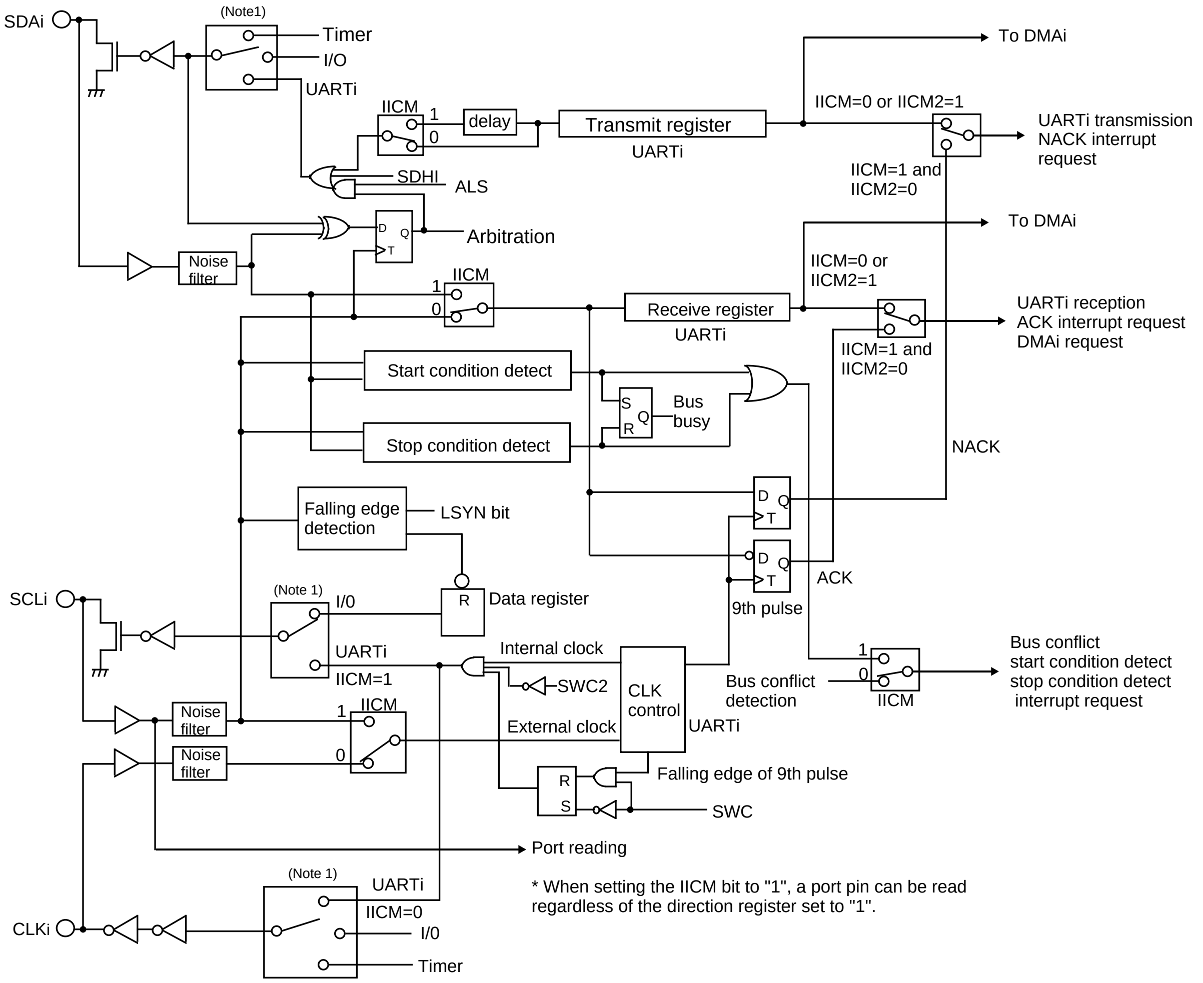
1. Special Mode 1 (IIC Mode)

IIC mode is a mode to communicate with external devices with a simplified I^2C . 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 "010_2" 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	• Arbitration lost An update timing of the ABT bit in the UiRB register can be selected Refer to the paragraph "• Arbitration" • 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" • Clock phase setting Selectable from clock delay or no clock delay Refer to the paragraph "• Transfer clock"



* When setting the IICM bit to "1", a port pin can be read regardless of the direction register set to "1".

i=0 to 4
Notes :
1. Set in the PSj (j=0,1,3), PSLj or PSC registers.

IICM : Bit in the UiSMR register
IICM2 : Bit in the UiSMR2 register

Figure 1.20.1. IIC Mode Block Diagram

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 "010_2"	
	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	

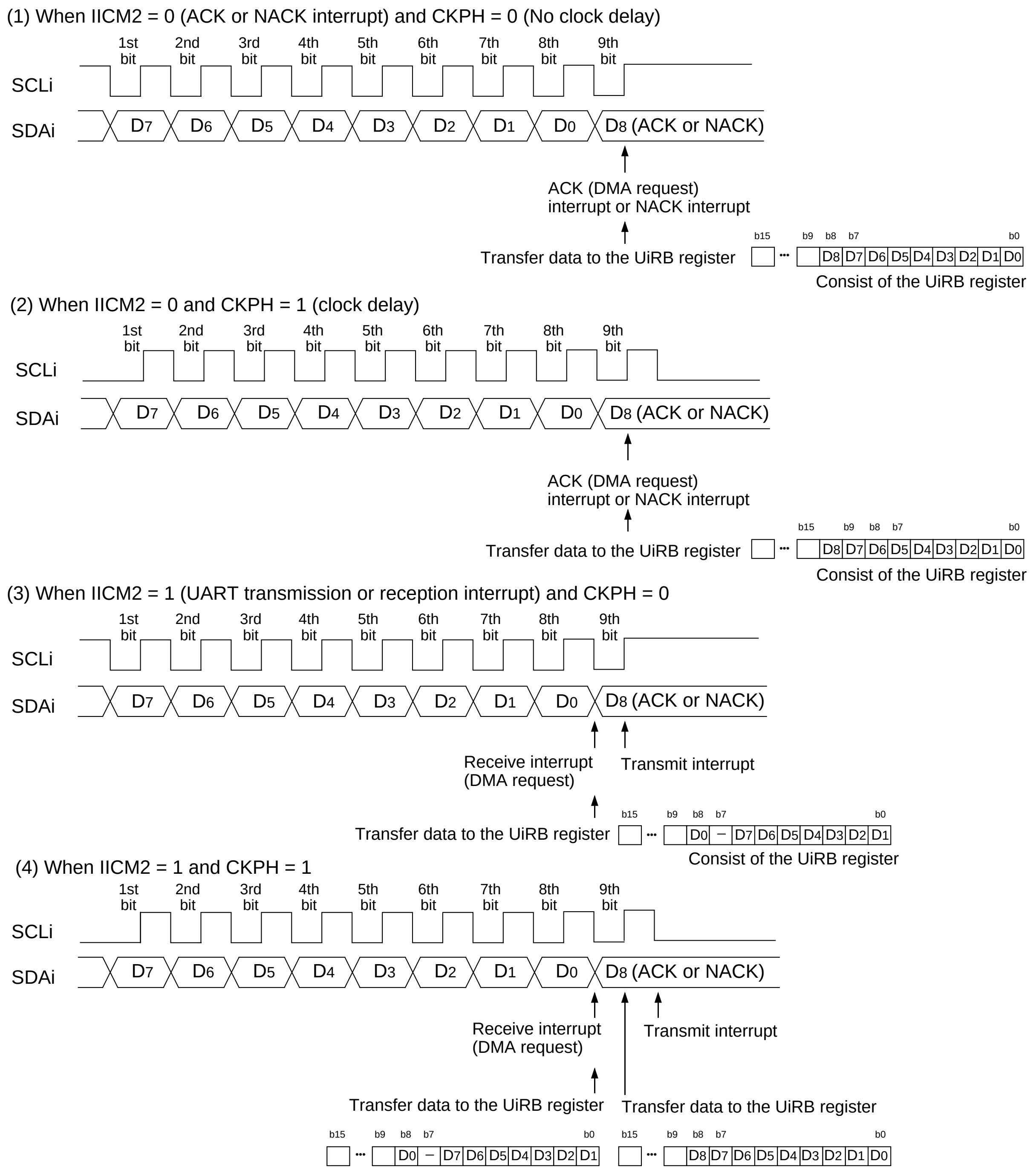
i=0 to 4

Table1.20.3. IIC Mode Functions

Function	Clock synchronous serial I/O mode (SMD2 to SMD0=001$_2$, IICM=0)	IIC mode (SMD2 to SMD0=010$_2$, 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[1] (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[1] (See Figure 1.20.2)	UARTi transmission Transmission start or transmission completed (selected by UiRS register)	No acknowlege 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)	Acknowlege 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			
P6$_3$, P6$_7$, P7$_0$, P9$_2$, P9$_6$ pin functions	TxDi output	SDAi I/O			
P6$_2$, P6$_6$, P7$_1$, P9$_1$, P9$_7$ pin functions	RxDi input	SCLi I/O			
P6$_1$, P6$_5$, P7$_2$, P9$_0$, P9$_5$ 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	Acknowlege 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[3]
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 :

1. The following procedures should be taken if changing how an interrupt is generated.
   (a) Disable an interrupt of a corresponding interrupt number.
   (b) Change how an interrupt is generated.
   (c) Set the IR bit of a corresponding interrupt number to "0" (interrupt disabled).
   (d) Set the IPL2 to IPL0 bits of a corresponding interrupt number.
2. An default value of a SDAi output should be set when the SMD2 to SMD0 bits in the UiMR register are set to "000$_2$" (serial I/O disabled).
3. Second data transfer to the UiRB register (on the rising edge of the ninth bit of SCLi).
4. First data transfer to the UiRB register (on the rising edge of the ninth bit of SCLi).


(1) When IICM2 = 0 (ACK or NACK interrupt) and CKPH = 0 (No clock delay)
1st bit
2nd bit
3rd bit
4th bit
5th bit
6th bit
7th bit
8th bit
9th bit
SCLi
SDAi
D7
D6
D5
D4
D3
D2
D1
D0
D8 (ACK or NACK)
ACK (DMA request) interrupt or NACK interrupt
Transfer data to the UiRB register
b15
b9
b8
b7
b0
D8 D7 D6 D5 D4 D3 D2 D1 D0
Consist of the UiRB register
(2) When IICM2 = 0 and CKPH = 1 (clock delay)
SCLi
SDAi
D8 (ACK or NACK)
ACK (DMA request) interrupt or NACK interrupt
Transfer data to the UiRB register
D8 D7 D6 D5 D4 D3 D2 D1 D0
Consist of the UiRB register
(3) When IICM2 = 1 (UART transmission or reception interrupt) and CKPH = 0
SCLi
SDAi
D8 (ACK or NACK)
Receive interrupt (DMA request)
Transmit interrupt
Transfer data to the UiRB register
D0 − D7 D6 D5 D4 D3 D2 D1
Consist of the UiRB register
(4) When IICM2 = 1 and CKPH = 1
SCLi
SDAi
D8 (ACK or NACK)
Receive interrupt (DMA request)
Transmit interrupt
Transfer data to the UiRB register
Transfer data to the UiRB register
D0 − D7 D6 D5 D4 D3 D2 D1
D8 D7 D6 D5 D4 D3 D2 D1 D0
Consist of the UiRB register
Consist of the UiRB register

i=0 to 4
IICM2 : Bit in the UiSMR2 register
CKPH : Bit in the UiSMR3 regiser

The above timing applies to the following setting :
- CKDIR bit in the UiMR register = 1 (select slave)

Figure 1.20.2. SCLi Timing

Table 1.20.4. Pin Settings in IIC Mode

Port	Function	Bit and setting value		
		PS0 register	PSL0 register	PD6 register
$P6_2$	SCL0 output	PS0_2=1	PSL0_2=0	PD6_2=0
$P6_3$	SDA0 output	PS0_3=1	-	PD6_3=0
$P6_6$	SCL1 output	PS0_6=1	PSL0_6=0	PD6_6=0
$P6_7$	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
$P7_0$[1]	SDA2 output	PS1_0=1	PSL1_0=0	PSC_0=0	PD7_0=0
$P7_1$[1]	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)

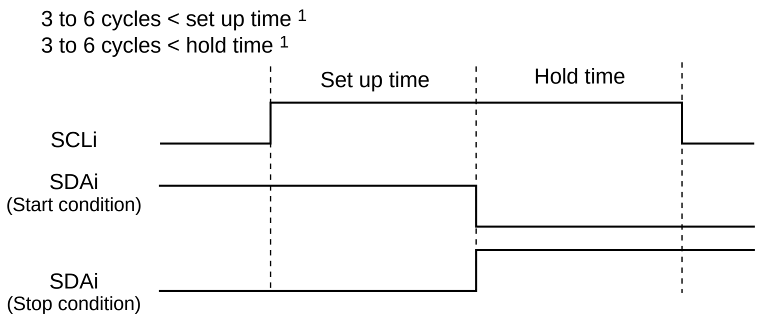
Port	Function	Bit and setting value		
		PS3 register[1]	PSL3 register	PD9 register[1]
$P9_1$	SCL3 output	PS3_1=1	PSL3_1=0	PD9_1=0
$P9_2$	SDA3 output	PS3_2=1	PSL3_2=0	PD9_2=0
$P9_6$	SCL4 output	PS3_6=1	-	PD9_6=0
$P9_7$	SDA4 output	PS3_7=1	PSL3_7=0	PD9_7=0

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 SCLi (i=0 to 4) pin is set to "H" and also the SDAi pin changes "H" to "L". The stop condition detect interrupt is generated when the SCLi pin is set to "H" and also the SDAi 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.



i=0 to 4
Notes :
1. The number of cycles means a main clock input generation frequency cycles (X_{IN})

Figure 1.20.3. Start Condition or Stop Condition Detection

• 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

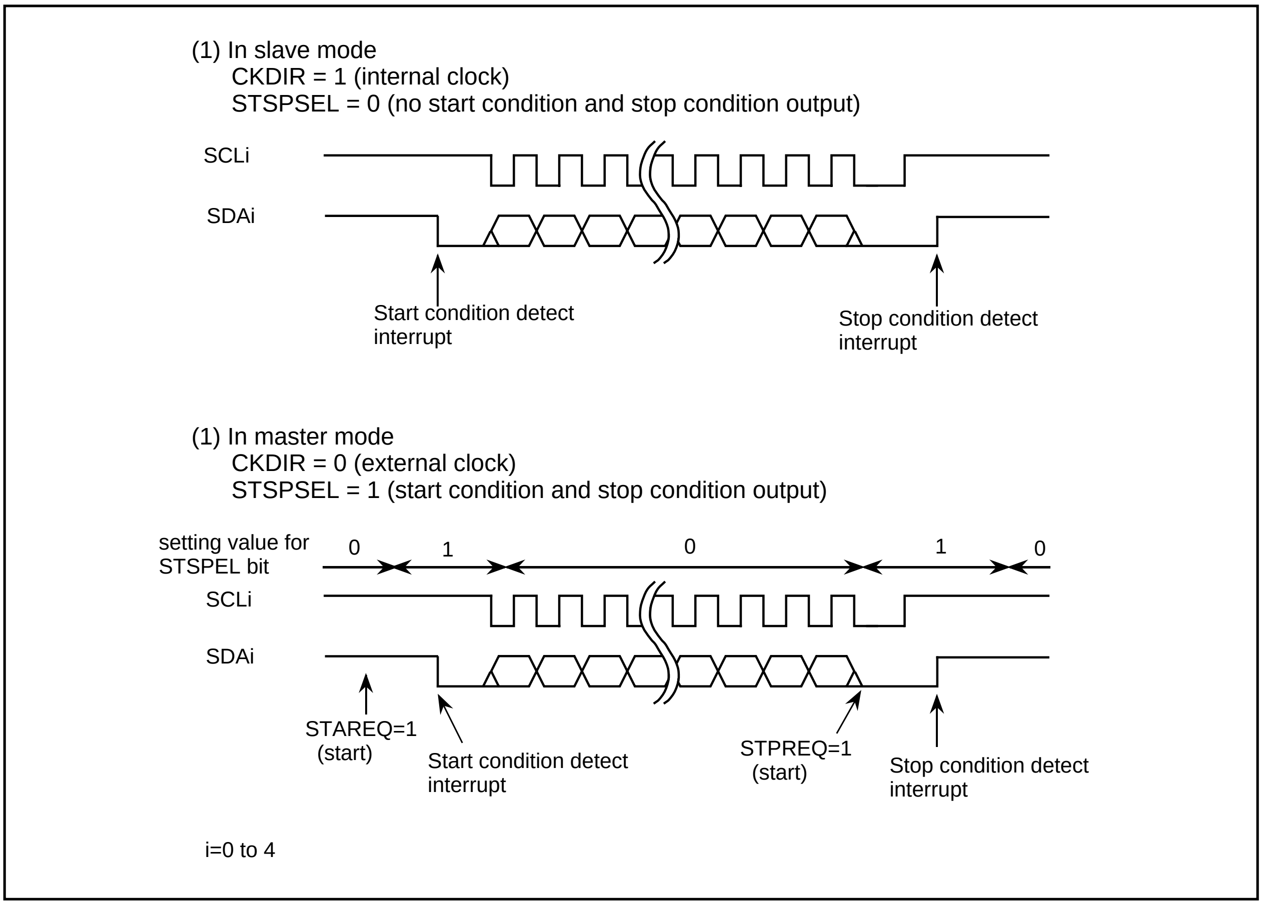

(1) In slave mode
CKDIR = 1 (internal clock)
STSPSEL = 0 (no start condition and stop condition output)
SCLi
SDAi
Start condition detect interrupt
Stop condition detect interrupt
(1) In master mode
CKDIR = 0 (external clock)
STSPSEL = 1 (start condition and stop condition output)
setting value for STSPEL bit
0
1
0
1
0
SCLi
SDAi
STAREQ=1 (start)
Start condition detect interrupt
STPREQ=1 (start)
Stop condition detect interrupt
i=0 to 4

Figure 1.20.4. STSPSEL Bit 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" form "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 (D_7 to D_0) in the UiTB register (i=0 to 4) are output in descending order from D_7. The ninth bit (D_8) 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 "000_2" (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 (D_7 to D_0) in the UiRB register. The ninth bit (D_8) is ACK or NACK.
When setting the IICM2 bit to "1", first to seventh bits (D_7 to D_1) of received data are stored into bits 6 to 0 in the UiRB register. Eighth bit (D_8) 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.

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 $\overline{SS}$i 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 00_{16} to FF_{16} 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 $\overline{CTS}$/$\overline{RTS}$ function disabled
Transmit start condition	•To start transmitting, the following requirement should be met[2] : - The TE bit in the UiC1 register is set to "1" (transmit enable) - The TI bit in the UiC1 register to "0" (data available in the UiTB register) - $\overline{CTS}$i input level is "L" when $\overline{CTS}$ function is selected
Receive start condition	• To start receiving, the following requirement should be met[2] : - The RE bit in the UiC1 register is set to "1" (receive enable) - The TE bit is set to "1" (receive enable) - 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 UARTi transmit register (transmit starting) - The UiRS register is set to "1" (transmit completed): when data transmission from UARTi transfer register is completed • While receiving When data is transferred from UARTi receive register to the UiRB register (receive completed)
Error detection	•Overrun error[3] 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 a the transfer clock can be selected when a transferred data is output and input • LSB first or 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 • TxD, RxD I/O polarity switching TxD pin output and RxD pin input are inversed. All I/O data levels are also inversed • Clock phase One of 4 combinations of transfer data polarity and phases can be read • $\overline{SS}$i input pin function Output pin is placed in a high-impedance state to avoid data conflict between a master and another masters or slave

Notes :

1. The CNT3 to CNT0 bits in the TRGSR register determine either "no division (n=0)" or "divide-by-2n (n=1 to 15)".
2. 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".
3. If an overrun error occurs, the UiRB register is indeterminate. The IR bit of SiRIC register does not change to "1" (interrupt request).

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 "001_2"
	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 recieve 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

Table 1.20.10. Pin settings in Special Mode 2

Port	Function	Bit and setting value		
		PS0 register	PSL0 register	PD6 register
$P6_0$	$\overline{SS0}$ input	PS0_0=0	–	PD6_0=0
$P6_1$	CLK0 input (slave)	PS0_1=0	–	PD6_1=0
	CLK0 output (master)	PS0_1=1	–	–
$P6_2$	RxD0 input (master)	PS0_2=0	–	PD6_2=0
	STxD0 input (slave)	PS0_2=1	PSL0_2=1	–
$P6_3$	TxD0 output (master)	PS0_3=1	–	–
	SRxD0 output (slave)	PS0_3=0	–	PD6_3=0
$P6_4$	$\overline{SS1}$ input	PS0_4=0	–	PD6_4=0
$P6_5$	CLK1 output (slave)	PS0_5=0	–	PD6_5=0
	CLK1 output (master)	PS0_5=1	–	–
$P6_6$	RxD1 input (master)	PS0_6=0	–	PD6_6=0
	STxD1 input (slave)	PS0_6=1	PSL0_6=1	–
$P6_7$	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
$P7_0$[1]	TxD2 output (master)	PS1_0=1	PSL1_0=0	PSC_0=0	–
	SRxD2 input (slave)	PS1_0=0	–	–	PD7_0=0
$P7_1$[1]	RxD2 input (master)	PS1_1=0	–	–	PD7_1=0
	STxD2 output (slave)	PS1_1=1	PSL1_1=1	–	–
$P7_2$	CLK2 input (slave)	PS1_2=0	–	–	PD7_2=0
	CLK2 output (master)	PS1_2=1	PSL1_2=0	PSC_2=0	–
$P7_3$	$\overline{SS2}$ 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[1]	PSL3 register	PD9 register[1]
$P9_0$	CLK3 input (slave)	PS3_0=0	–	PD9_0=0
	CLK3 output (master)	PS3_0=1	–	–
$P9_1$	RxD3 input (master)	PS3_1=0	–	PD9_1=0
	STxD3 output (slave)	PS3_1=1	PSL3_1=1	–
$P9_2$	TxD3 output (master)	PS3_2=1	PSL3_2=0	–
	SRxD3 input (slave)	PS3_2=0	–	PD9_2=0
$P9_3$	$\overline{SS3}$ input	PS3_3=0	PSL3_3=0	PD9_3=0
$P9_4$	SS4 input	PS3_4=0	PSL3_4=0	PD9_4=0
$P9_5$	CLK4 input (slave)	PS3_5=0	PSL3_5=0	PD9_5=0
	CLK4 output (master)	PS3_5=1	–	–
$P9_6$	TxD4 output (master)	PS3_6=1	–	–
	SRxD4 input (slave)	PS3_6=0	PSL3_6=0	PD9_6=0
$P9_7$	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.

• $\overline{\text{SSi}}$ Input Pin Function (i=0 to 4)

When the SSE bit in the UiSMR3 register to "1" ($\overline{\text{SS}}$ 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 $\overline{\text{SSi}}$ 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 $\overline{\text{SSi}}$ 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 $\overline{\text{SSi}}$ 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 $\overline{\text{SSi}}$ pin, serial communication is available due to a transmission. The master outputs the transfer clock. When a "L" signal is input to $\overline{\text{SSi}}$ 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 BCNiIC 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 "000_2" (serial I/O disabled).

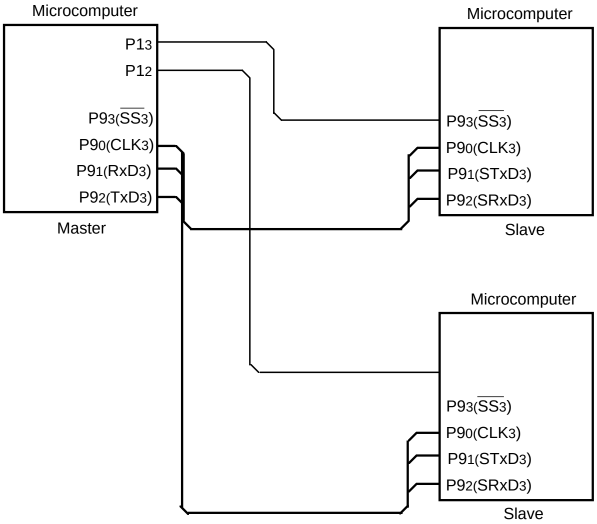


Figure 1.20.5. Serial Bus Communication Control with $\overline{\text{SS}}$ Pin

• 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 $\overline{\text{SS}}$i input pin to "H", output data is placed in a high-impedance state. When $\overline{\text{SS}}$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 $\overline{\text{SS}}$i input pin to "H", output data is placed in a high-impedance state. When the $\overline{\text{SS}}$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.

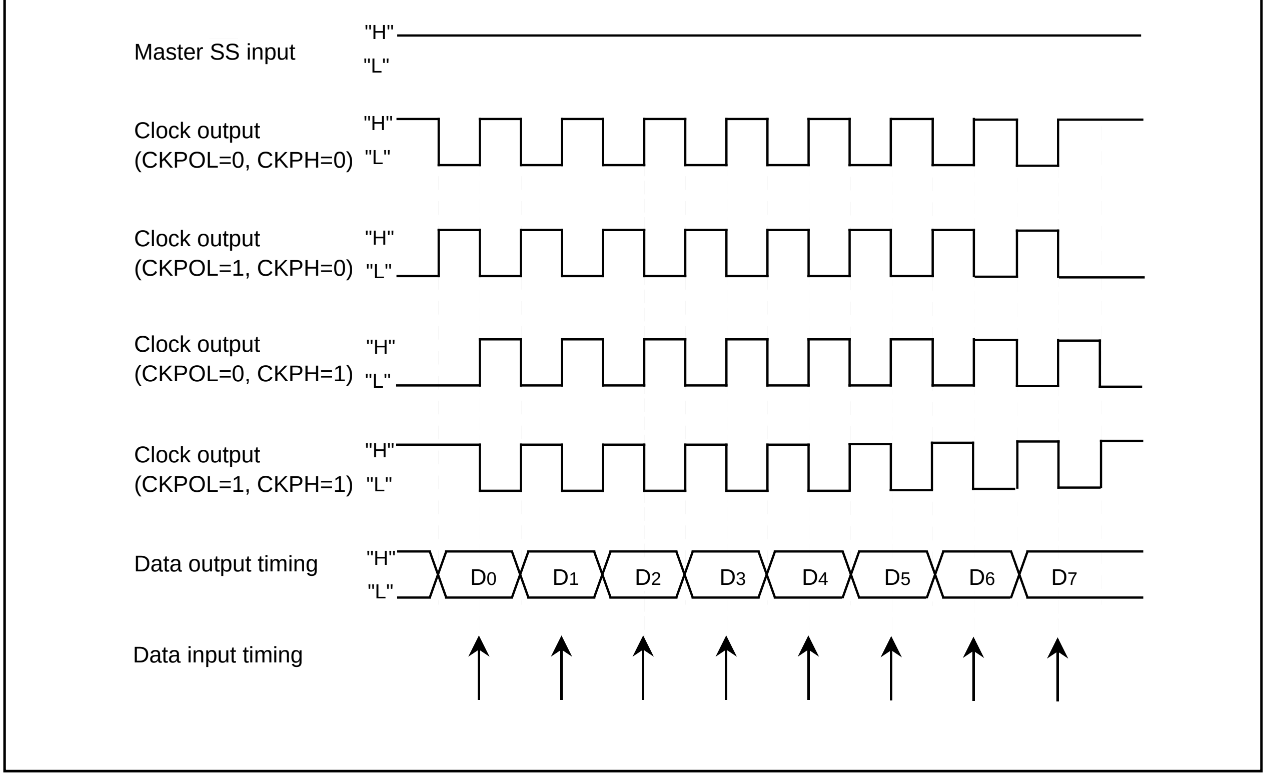
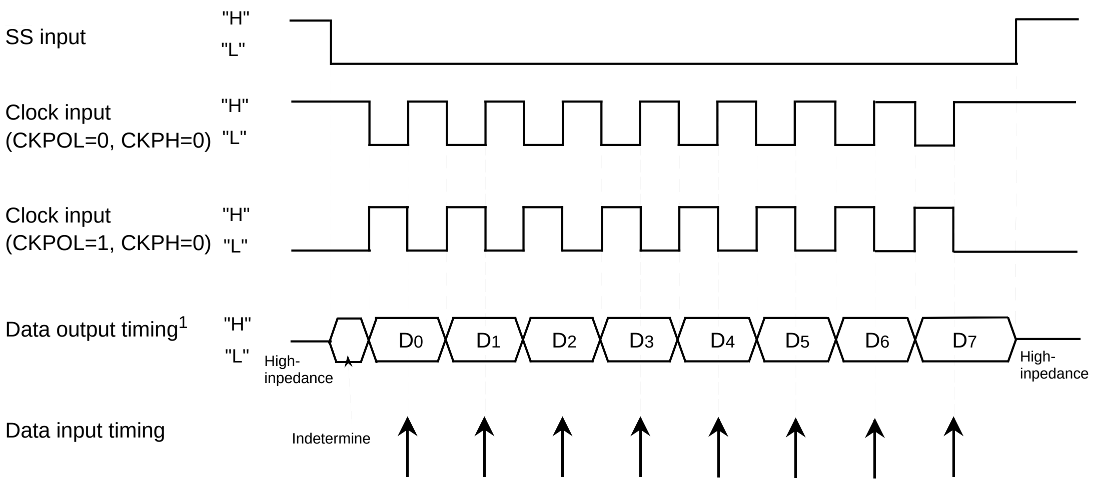

Master SS input
"H"
"L"
Clock output (CKPOL=0, CKPH=0)
"H"
"L"
Clock output (CKPOL=1, CKPH=0)
"H"
"L"
Clock output (CKPOL=0, CKPH=1)
"H"
"L"
Clock output (CKPOL=1, CKPH=1)
"H"
"L"
Data output timing
"H"
"L"
D0
D1
D2
D3
D4
D5
D6
D7
Data input timing

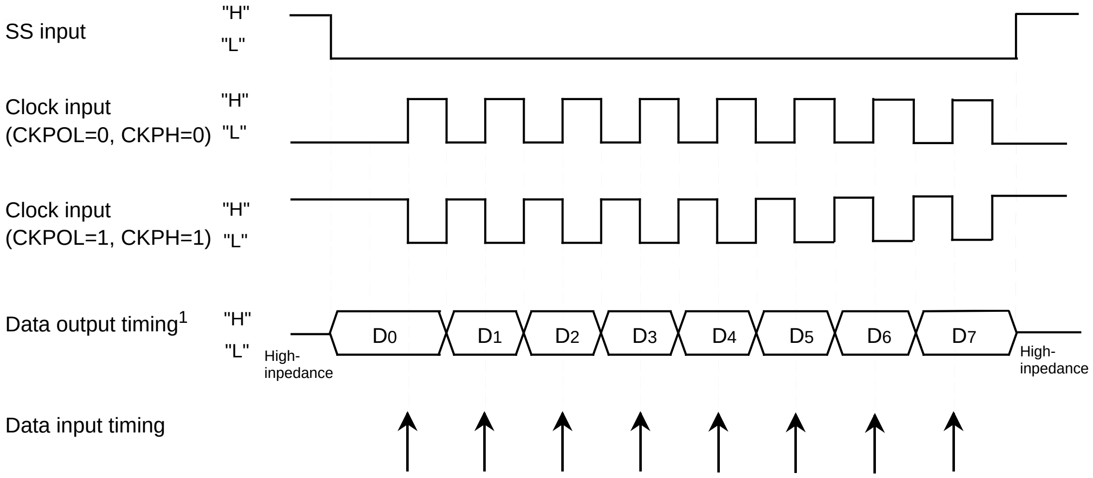
Figure 1.20.6. Transmit and Receive Timing in Master Mode (Internal Clock)



Notes :

1. UART2 output is an N-channel open drain and needs to be pulled up externally.

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



Notes :

1. UART2 output is an N-channel open drain and needs to be pulled up externally.

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

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.

Table1.20.13. 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 $\overline{\text{CTSi}}$ pin
Transmit/receive start condition	• When the external clock is set to "H", the following conditions should be met : - The TE bit in the UiC1 register is set to "1" (transmit enable) - The RE bit in the UiC1 register to "1" (receive enable) - The TI bit in the UiC1 register to "0" (Data in UiTB register) - $\overline{\text{CTSi}}$ input level is "L"
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 UARTi transmit register (starting transmit) - The UiIRS bit is set to "1" (Transmit completed): when data transmission from the UARTi transfer register is completed • While receiving When data is transferred from the UARTi receive register to the UiRB register (receive completed)
Error detection	Overrun error[1] This error occurs when reading a seventh bit of the next received data before reading the UiRB register

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).

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 "00$_{16}$"
UiMR	SMD2 to SMD0	Set to "001$_{2}$"
	CKDIR	Set to "1"
	IOPOL	Set to "0"
UiC0	CLK1 to CLK0	Set to "00$_{2}$"
	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

Table 1.20.15. Pin settings in CGI Mode

Port	Function	Bit and setting value		
		PS0 register	PSL0 register	PD6 register
$P6_0$	$\overline{\text{CTS0}}$ input[1]	PS0_0=0	–	PD6_0=0
$P6_1$	CLK0 input	PS0_1=0	–	PD6_1=0
$P6_2$	RxD0 input	PS0_2=0	–	PD6_2=0
$P6_3$	TxD0 output	PS0_3=1	–	–
$P6_4$	$\overline{\text{CTS1}}$ input	PS0_4=0	–	PD6_4=0
$P6_5$	CLK1 input	PS0_5=0	–	PD6_5=0
$P6_6$	RxD1 input	PS0_6=0	–	PD6_6=0
$P6_7$	TxD1 output	PS0_7=1	–	–

Notes :

1. $\overline{\text{CTS}}$ 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
$P7_0$[1]	TxD2 output	PS1_0=1	PSL1_0=0	PSC_0=0	–
$P7_1$[1]	RxD2 input	PS1_1=0	–	–	PD7_1=0
$P7_2$	CLK2 input	PS1_2=0	–	–	PD7_2=0
$P7_3$	$\overline{\text{RTS2}}$ input[2]	PS1_3=0	–	–	PD7_3=0

Notes :

1. N-channel open drain output
2. $\overline{\text{CTS}}$ input is used for trigger input.

Table 1.20.17. Pin settings (Continued)

Port	Function	Bit and setting value		
		PS3 register[1]	PSL3 register	PD9 register[1]
$P9_0$	CLK3 input	PS3_0=0	–	PD9_0=0
$P9_1$	RxD3 input	PS3_1=0	–	PD9_1=0
$P9_2$	TxD3 output	PS3_2=1	PSL3_2=0	–
$P9_3$	$\overline{\text{CTS3}}$ input[2]	PS3_3=0	PSL3_3=0	PD9_3=0
$P9_4$	$\overline{\text{CTS4}}$ input[2]	PS3_4=0	PSL3_4=0	PD9_4=0
$P9_5$	CLK4 input	PS3_5=0	PSL3_5=0	PD9_5=0
$P9_6$	TxD4 output	PS3_6=1	–	–
$P9_7$	RxD4 input	PS3_7=0	–	PD9_7=0

Notes :

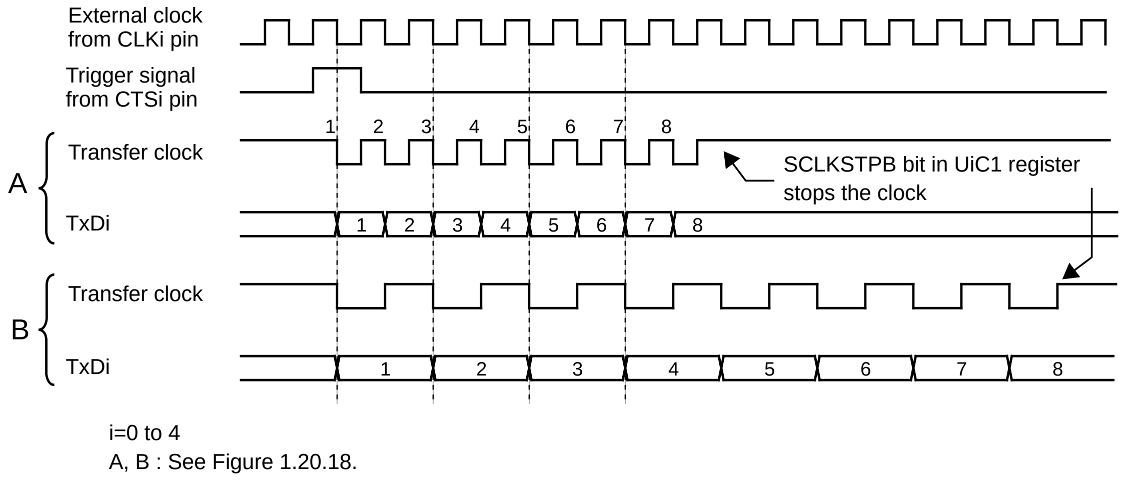
1. The PD9 and PS3 registers should be set with subsequent to the instruction that the PRC2 bit in the PRCR register is set to "1" (write enable). Avoid generating an interrupt and operating a DMA transfer between the instruction to set the PRC2 bit to "1" and the instruction to write to the PD9 and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

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


External clock from CLKi pin
Trigger signal from CTSi pin
A
Transfer clock
TxDi
B
Transfer clock
TxDi
1
2
3
4
5
6
7
8
SCLKSTPB bit in UiC1 register stops the clock

i=0 to 4
A, B : See Figure 1.20.18.

Figure1.20.9. Clock-Divided Synchronous Function

4. Special Mode 4 (IE Mode)

IE mode is a mode to communicate in UART mode among devices connenting 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.

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 "110_2"
	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

Table 1.20.21. Pin Settings in IE Mode

Port	Function	Bit and setting value		
		PS0 register	PSL0 register	PD6 register
$P6_1$	CLK0 input	PS0_1=0	–	PD6_1=0
	CLK0 output	PS0_1=1	–	–
$P6_2$	RxD0 input	PS0_2=0	–	PD6_2=0
$P6_3$	TxD0 output	PS0_3=1	–	–
$P6_5$	CLK1 input	PS0_5=0	–	PD6_5=0
	CLK1 output	PS0_5=1	–	–
$P6_6$	RxD1 input	PS0_6=0	–	PD6_6=0
$P6_7$	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
$P7_0$[1]	TxD2 output	PS1_0=1	PSL1_0=0	PSC_0=0	–
$P7_1$[1]	RxD2 input	PS1_1=0	–	–	PD7_1=0
$P7_2$	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[1]	PSL3 register	PD9 register[1]
$P9_0$	CLK3 input	PS3_0=0	–	PD9_0=0
	CLK3 output	PS3_0=1	–	–
$P9_1$	RxD3 input	PS3_1=0	–	PD9_1=0
$P9_2$	TxD3 output	PS3_2=1	PSL3_2=0	–
$P9_5$	CLK4 input	PS3_5=0	PSL3_5=0	PD9_5=0
	CLK4 output	PS3_5=1	–	–
$P9_6$	TxD4 output	PS3_6=1	–	–
$P9_7$	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.

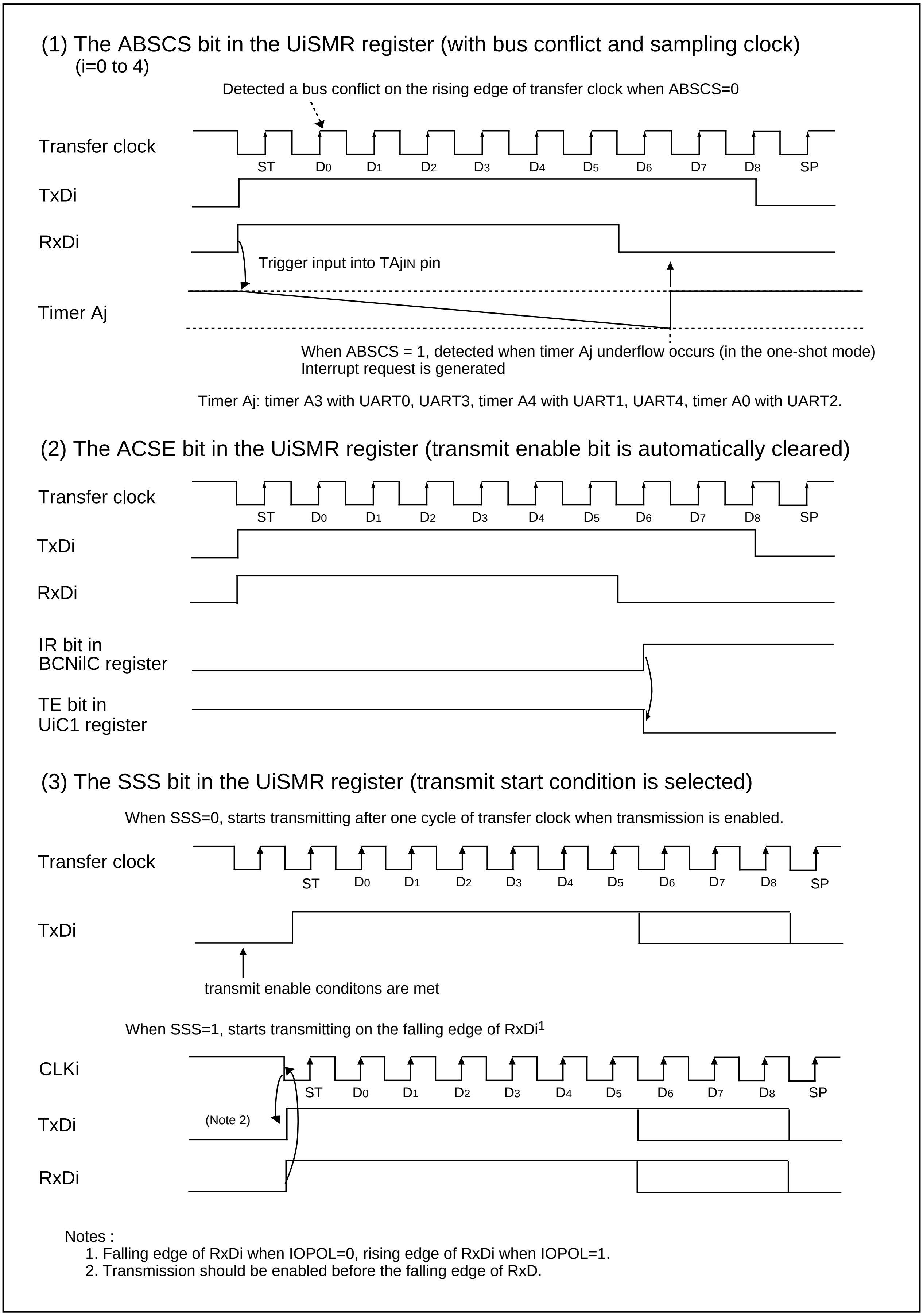
When an output level of the TxDi pin (i=0 to 4) differs from an input level of the RxDi pin, an interrupt request is generated.

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 detemined 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.


(1) The ABSCS bit in the UiSMR register (with bus conflict and sampling clock)
(i=0 to 4)
Detected a bus conflict on the rising edge of transfer clock when ABSCS=0
Transfer clock
ST
D0
D1
D2
D3
D4
D5
D6
D7
D8
SP
TxDi
RxDi
Trigger input into TAjIN pin
Timer Aj
When ABSCS = 1, detected when timer Aj underflow occurs (in the one-shot mode)
Interrupt request is generated
Timer Aj: timer A3 with UART0, UART3, timer A4 with UART1, UART4, timer A0 with UART2.
(2) The ACSE bit in the UiSMR register (transmit enable bit is automatically cleared)
Transfer clock
ST
D0
D1
D2
D3
D4
D5
D6
D7
D8
SP
TxDi
RxDi
IR bit in BCNiIC register
TE bit in UiC1 register
(3) The SSS bit in the UiSMR register (transmit start condition is selected)
When SSS=0, starts transmitting after one cycle of transfer clock when transmission is enabled.
Transfer clock
ST
D0
D1
D2
D3
D4
D5
D6
D7
D8
SP
TxDi
transmit enable conditons are met
When SSS=1, starts transmitting on the falling edge of RxDi1
CLKi
ST
D0
D1
D2
D3
D4
D5
D6
D7
D8
SP
TxDi
(Note 2)
RxDi

Notes :
1. Falling edge of RxDi when IOPOL=0, rising edge of RxDi when IOPOL=1.
2. Transmission should be enabled before the falling edge of RxD.

Figure 1.20.10. Bit Function Related Bus Conflict Detection

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.

Table1.20.24. SIM Mode Specifications

Item	Specification
Transfer data format	• Transfer data: 8-bit UART mode • One stop bit • In direct format: Parity Even, Data logic Direct, Transfer format LSB first • In inverse format: Parity Odd, Data logic Inverse, Transfer format MSB first
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_{16} to FF_{16} 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 • 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)
Receive start condition	To start receiving, the following requirements should be met • The RE bit in the UiC1 register is set to "1" (receive enable) • The start bit is detected
Interrupt request generation timing	While transmitting: • The UiRS bit is set to "1" (transmit is completed): when data transmission from the UARTi transfer register is completed While receiving • When data is transferred from the UARTi receive register to the UiRB register (receive completed)
Error detection	• Overrun error[1] This error occurs when an eighth bit of the next data before reading the UiRB bit is received • Flaming error This error occurs when the number of the stop bit set is not detected • Parity error This error occurs when the number of "1" in parity bit and character bits is different from the number set. • Error sum flag The SUM bit is set to "1" when an overrun error, flaming error or parity error occurs.

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-2n (n=1 to 15)".

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 "101$_2$"
	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

Table 1.20.26. Pin Settings in SIM Mode

Port	Function	Bit and setting value		
		PS0 register	PSL0 register	PD6 register
P6$_2$	RxD0 input	PS0_2=0	–	PD6_2=0
P6$_3$	TxD0 output	PS0_3=1	–	–
P6$_6$	RxD1 input	PS0_6=0	–	PD6_6=0
P6$_7$	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
P7$_0$[1]	TxD2 output	PS1_0=1	PSL1_0=0	PSC_0=0	–
P7$_1$[1]	RxD2 input	PS1_1=0	–	–	PD7_1=0

Notes :

1. N-channel open drain output

Table 1.20.28. Pin Settings (Continued)

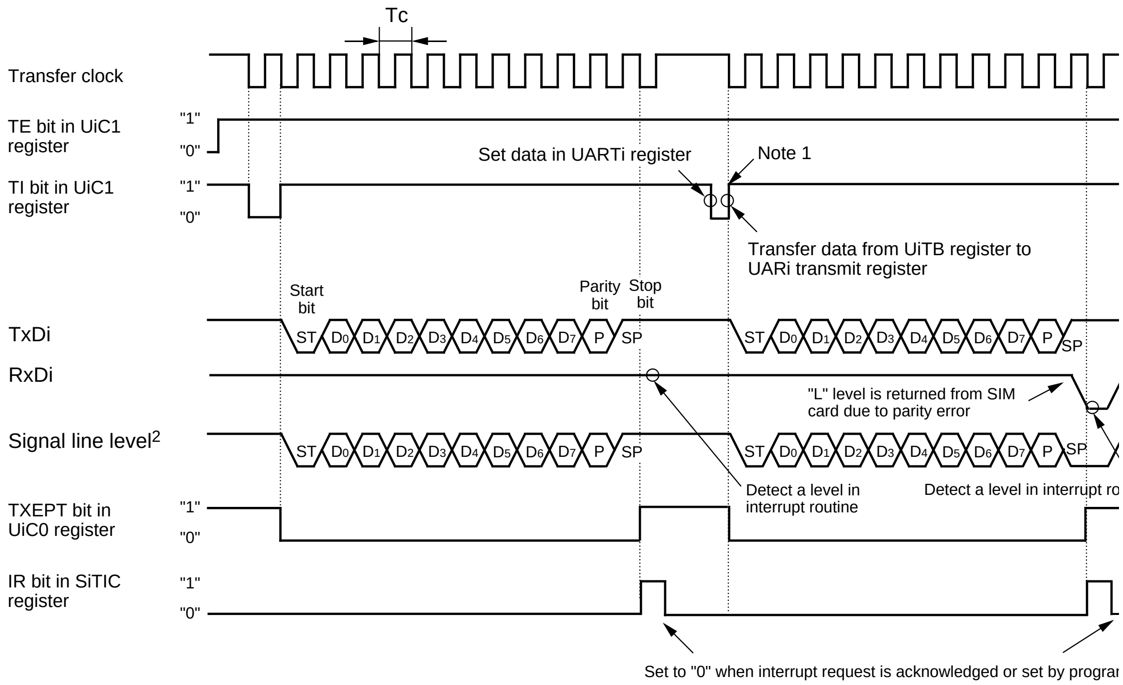
Port	Function	Bit and setting value		
		PS3 register[1]	PSL3 register	PD9 register[1]
P9$_1$	RxD3 input	PS3_1=0	–	PD9_1=0
P9$_2$	TxD3 output	PS3_2=1	PSL3_2=0	–
P9$_6$	TxD4 output	PS3_6=1	–	–
P9$_7$	RxD4 input	PS3_7=0	–	PD9_7=0

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.

(1) Ttransmit Timing


Tc
Transfer clock
TE bit in UiC1 register
"1"
"0"
TI bit in UiC1 register
"1"
"0"
Set data in UARTi register
Note 1
Transfer data from UiTB register to UARi transmit register
Start bit
Parity bit
Stop bit
TxDi
ST D0 D1 D2 D3 D4 D5 D6 D7 P SP
RxDi
"L" level is returned from SIM card due to parity error
Signal line level2
Detect a level in interrupt routine
Detect a level in interrupt ro
TXEPT bit in UiC0 register
"1"
"0"
IR bit in SiTIC register
"1"
"0"
Set to "0" when interrupt request is acknowledged or set by prograr

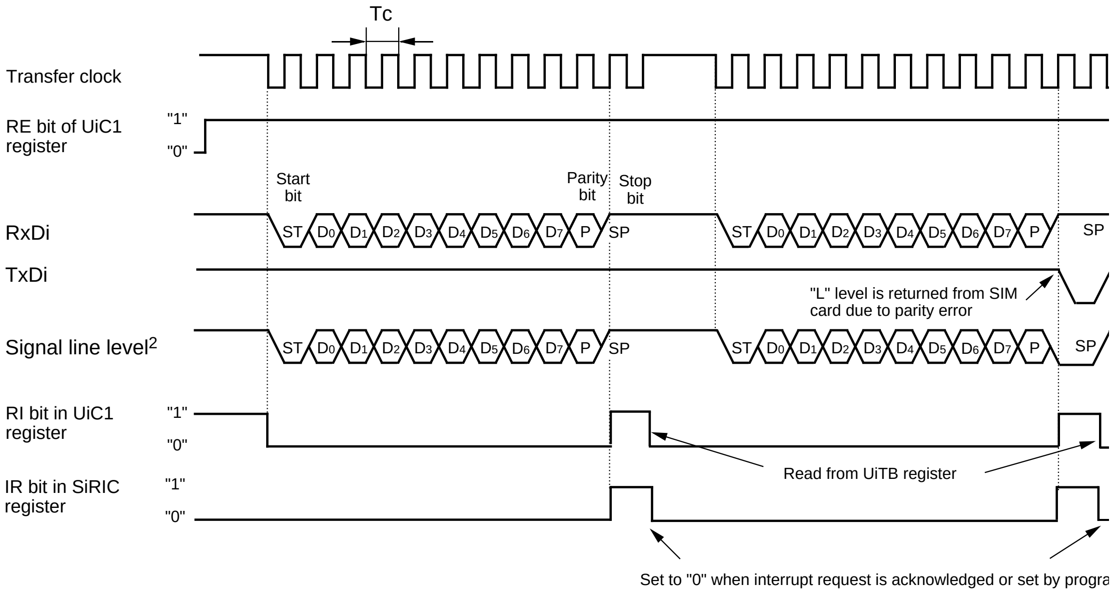
i=0 to 4

The avobe 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 UiIRS bit in the UiC1 register is set to "1" (interrupt request is generated due to transmission completed)

Tc = 16(m+1) / f_j
f_j : frequency of UiBRG count source (f_1, f_8, f_{2n}[3])
m : setting value of UiBRG

(1) Recive Timing



i=0 to 4

The avobe 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 UiIRS bit in the UiC1 register is set to "1" (interrupt request generated due to transmission completed)

Tc = 16(m+1) / f_j
f_j : frequency of UiBRG count source (f_1, f_8, f_{2n}[3])
m : setting value of UiBRG

Notes :
1. Transmission start when a BRG overflow occurs after a value is set to the UiTB register.
2. The same waveform is generated during transmitting and during receiving due to TxDi and RxDi connection.
3. The CNT3 to CNT0 bits in the TCSPR register determine either "no division (n=0)" or "divide-by-2n (n=1 to 15)".

Figure 1.20.11. SIM Interface Operation

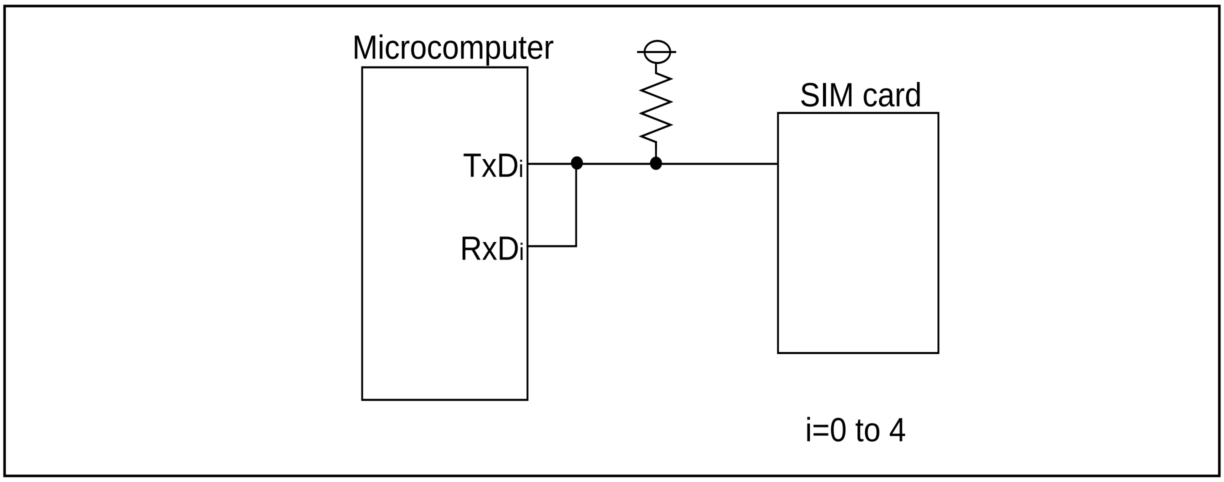


Figure 1.20.12. SIM Interface Connection

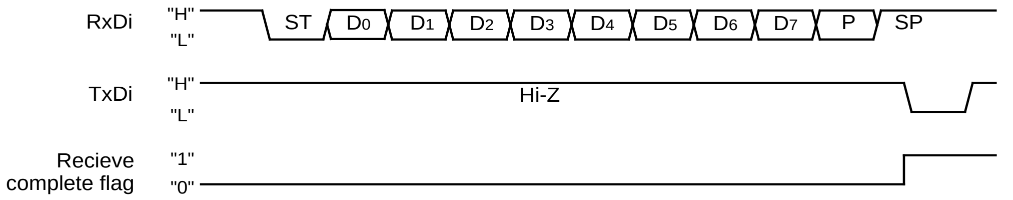
• Parity Error Signal

- 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.



Notes :
1. The above applies to the direct format conditions (PRY=1, UFORM=0, UiLCH=0).

ST : Start bit
P : Even parity
SP : Stop bit
i=0 to 4

Figure 1.20.13. Parity Error Signal Output Timing (LSB First)

• 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 D_0. When receiving data, received data are stored in the UiRB register, starting from D_0. 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 inversed and is transmitted with odd parity, starting from D_7. When receiving data, received data is logically inversed to be stored in the UiRB register, starting from D_7. Odd parity determines whether a parity error occurs.

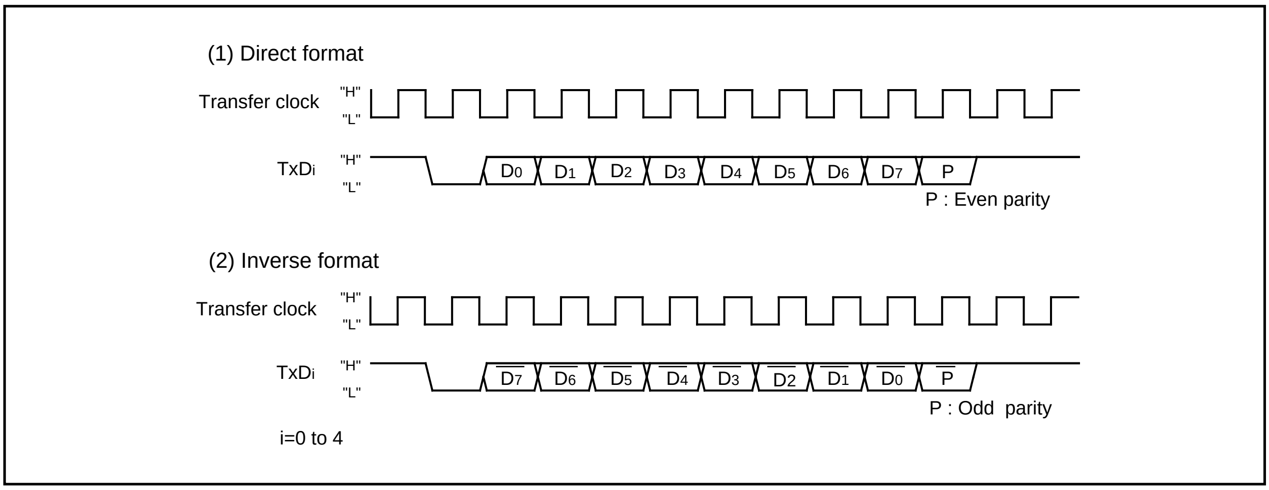


Figure 1.20.14. Parity Error Signal Output Timing (LSB First)

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 = $\frac{1}{\text{Tq clock cycle x Tq per bit}}$ --- Max 1 Mbps Tq clock cycle = $\frac{\text{BRP + 1}}{f_1}$ Tq 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.

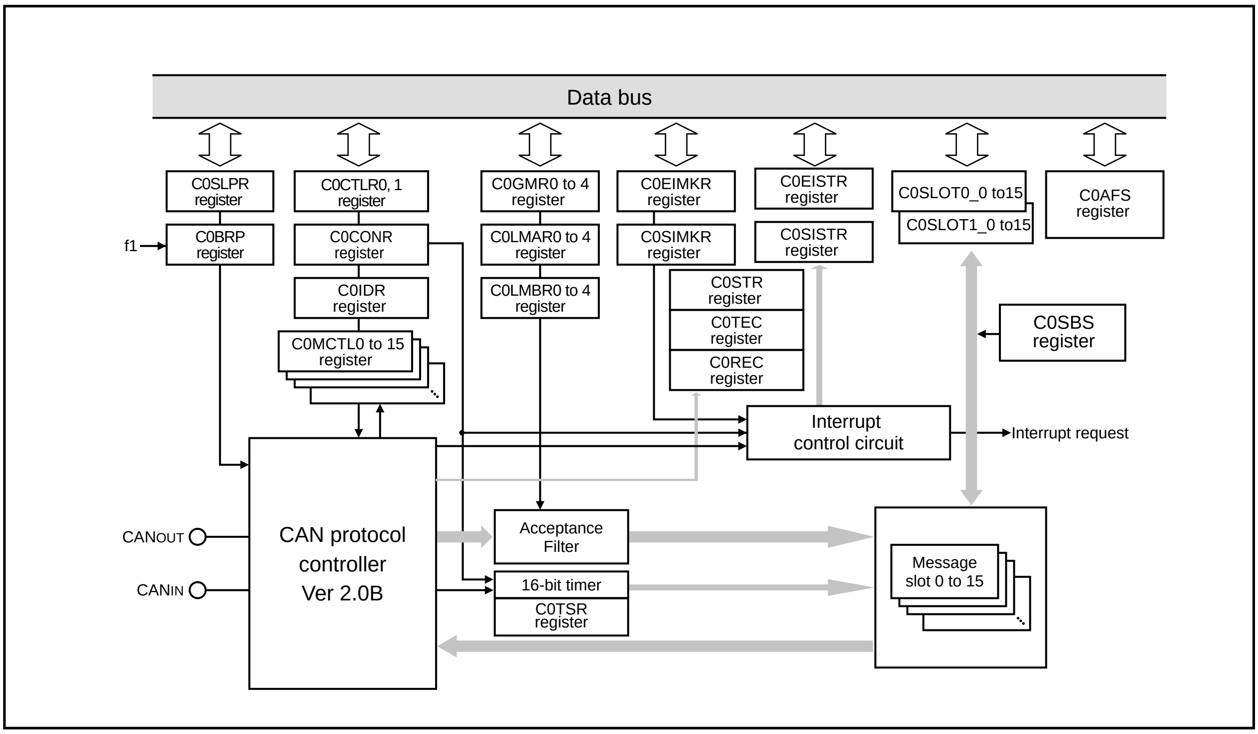


Figure 1.21.1. CAN Module Block Diagram

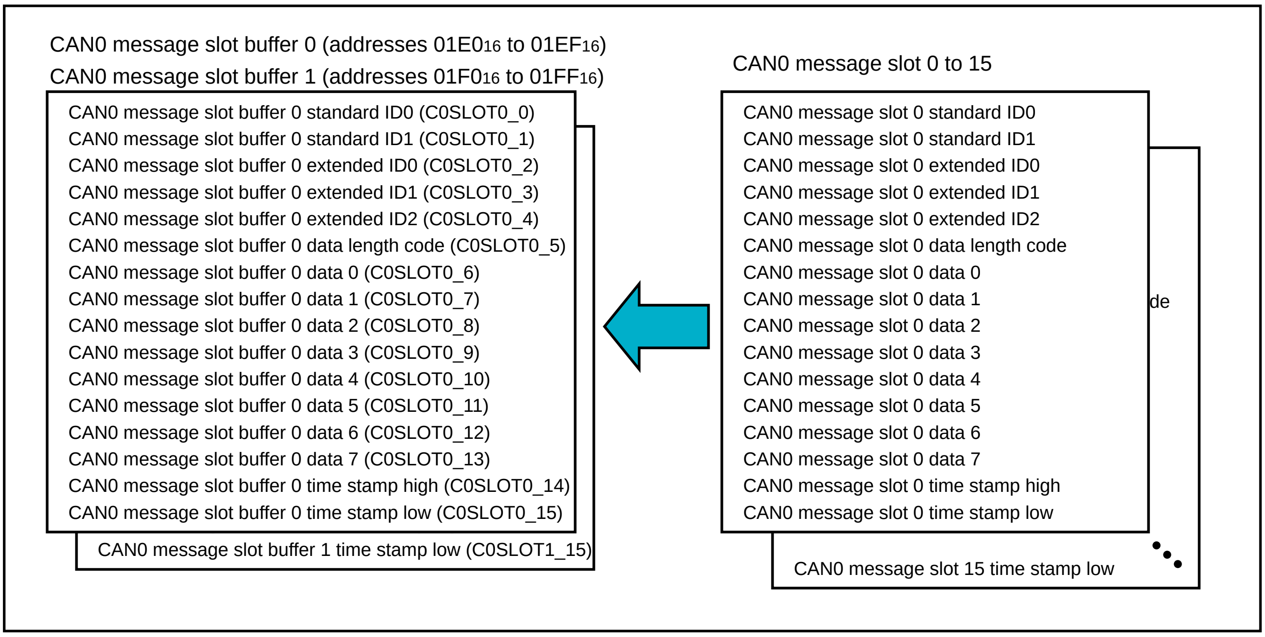

CAN0 message slot buffer 0 (addresses 01E016 to 01EF16)
CAN0 message slot buffer 1 (addresses 01F016 to 01FF16)
CAN0 message slot buffer 0 standard ID0 (C0SLOT0_0)
CAN0 message slot buffer 0 standard ID1 (C0SLOT0_1)
CAN0 message slot buffer 0 extended ID0 (C0SLOT0_2)
CAN0 message slot buffer 0 extended ID1 (C0SLOT0_3)
CAN0 message slot buffer 0 extended ID2 (C0SLOT0_4)
CAN0 message slot buffer 0 data length code (C0SLOT0_5)
CAN0 message slot buffer 0 data 0 (C0SLOT0_6)
CAN0 message slot buffer 0 data 1 (C0SLOT0_7)
CAN0 message slot buffer 0 data 2 (C0SLOT0_8)
CAN0 message slot buffer 0 data 3 (C0SLOT0_9)
CAN0 message slot buffer 0 data 4 (C0SLOT0_10)
CAN0 message slot buffer 0 data 5 (C0SLOT0_11)
CAN0 message slot buffer 0 data 6 (C0SLOT0_12)
CAN0 message slot buffer 0 data 7 (C0SLOT0_13)
CAN0 message slot buffer 0 time stamp high (C0SLOT0_14)
CAN0 message slot buffer 0 time stamp low (C0SLOT0_15)
CAN0 message slot buffer 1 time stamp low (C0SLOT1_15)
CAN0 message slot 0 to 15
CAN0 message slot 0 standard ID0
CAN0 message slot 0 standard ID1
CAN0 message slot 0 extended ID0
CAN0 message slot 0 extended ID1
CAN0 message slot 0 extended ID2
CAN0 message slot 0 data length code
CAN0 message slot 0 data 0
CAN0 message slot 0 data 1
CAN0 message slot 0 data 2
CAN0 message slot 0 data 3
CAN0 message slot 0 data 4
CAN0 message slot 0 data 5
CAN0 message slot 0 data 6
CAN0 message slot 0 data 7
CAN0 message slot 0 time stamp high
CAN0 message slot 0 time stamp low
de
CAN0 message slot 15 time stamp low

Figure 1.21.2. Message Slot Buffer and Message Slot

Table 1.21.2. Pin Settings

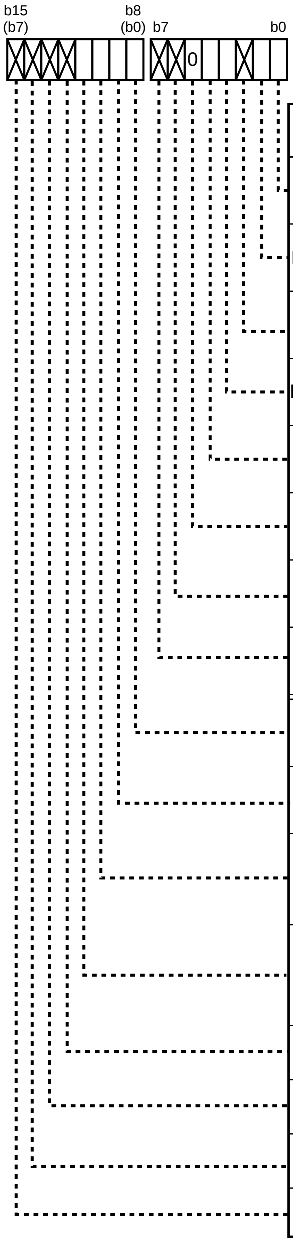
Port	Function	Bit and Setting				
		PS1, PS2 registers	PSL1, PSL2 registers	PSC register	IPS register	PD7, PD8 registers
P76	CANOUT	PS1_6=1	PSL1_6=0	PSC_6=1	–	–
P77	CANIN	PS1_7=0	–	–	IPS3=0	PD7_7=0
P82	CANOUT	PS2_2=1	PSL2_2=1	–	–	–
P83	CANIN	–	–	–	IPS3=1	PD8_3=0

CAN-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 "10010$_2$" (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 (C0CTLR0 Register)

CAN0 control register 0



Symbol	Address	When reset [1]
C0CTLR0	0201_{16} - 0200_{16}	XXXX 0000 XX01 0X01$_2$

Bit symbol	Bit name	Function	RW
RESET0	CAN reset bit 0	0: CAN module is not reset 1: CAN module is reset[2]	RW
LOOPBACK	Loop back mode select bit	0: Loop back function disabled 1: Loop back function enabled	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
BASICCAN	Basic CAN mode select bit	0 : Basic CAN mode function disabled 1 : Basic CAN mode function enabled	RW
RESET1	CAN reset bit 1	0: CAN module is not reset 1: CAN module is reset[2]	RW
——	Reserved bit	Should set to "0".	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–
TSPRE0	Time stamp prescaler select bit	b9 b8 0 0: CAN bus bit clock is selected 0 1: CAN bus bit clock divided by 2 is selected 1 0: CAN bus bit clock divided by 3 is selected 1 1: CAN bus bit clock divided by 4 is selected	RW
TSPRE1			RW
TSRESET	Time stamp counter reset bit	When writing "1" to this bit, the C0TSR register is set to "0000$_{16}$". After that, this bit is automatically set to "0". [3]	RW
ECRESET	Error counter reset bit	When writing "1" to this bit, the C0TEC and C0REC registers are set to "00$_{16}$". After that, this bit is automatically set to "0". [3]	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–
——			–
——			–

Notes :
1. Value is obtained by setting the SLEEP bit in the C0SLPR register to "1" (sleep mode exited) and supplying a clock to the CAN module after reset.
2. The RESET0 bit and RESET1 bit should be set to the same value simultaneously.
3. This bit can be set to "1" by program but cannot to "0".

Figure 1.21.3 C0CTLR0 Register

• RESET0 Bit and RESET1 Bit

When setting both RESET0 and RESET1 bits to "1", the CAN module is immediately reset regargless of ongoing CAN communication.

When the CAN module reset is completed, the C0TSR register is set to "0000_{16}". Also, the C0TEC and C0REC registers are set to "00_{16}" 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 CAN_{OUT} 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).

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_{16}". The TSRESET bit is automatically set to "0" after setting the C0TSR register to "0000_{16}".

ECRESET Bit

When setting the ECRESET bit to "1", the C0TEC and C0REC registers are set to "00_{16}". The CAN module forcibly enters an error active state.

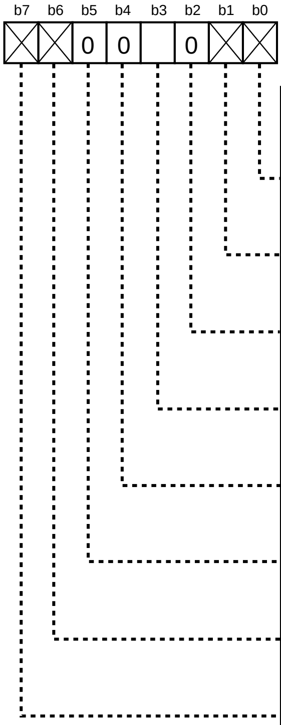
The ECRESET bit is automatically set to "0" after entering an error active state.

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 (C0CTLR1 Register)

CAN0 control register 1



Symbol	Address	When reset [1]
C0CTLR1	0241_{16}	$XX00\ 00XX_2$

Bit symbol	Bit name	Function	RW
——	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		–
——			–
——	Reserved bit	Should set to "0".	RW
BANKSEL	CAN0 bank switch bit	0 : Selects message slot control register 1 : Selects mask register	RW
——	Reserved bit	Should set to "0".	RW
——			RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–

Notes :

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

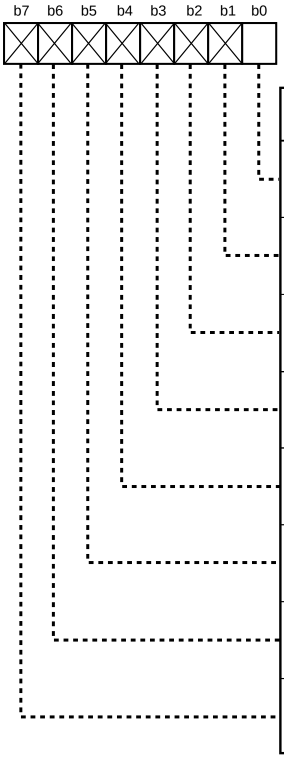
Figure 1.21.4. C0CTLR1 Register

BANKSEL Bit

The BANKSEL bit selects registers allocated to addresses 0220_{16} to $023F_{16}$.
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)

CAN0 sleep control register



Symbol: C0SLPR
Address: 0242$_{16}$
When reset: XXXX XXX0$_2$

Bit symbol	Bit name	Function	RW
SLEEP	Sleep mode control bit	0 : Enters sleep mode 1 : Exits sleep mode (Note 1)	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–
——			–
——			–
——			–
——			–
——			–

Notes:

1. The CAN configuration should be set up after CAN sleep mode is canceled. While the CAN module is in sleep mode, no SFR registers (addresses 01E0$_{16}$ to 0245$_{16}$) for the CAN, except this register, can be accessed.

Figure 1.21.5. C0SLPR Register

SLEEP Bit

When setting the SLEEP bit to "0", a clock stops running to enter sleep mode.

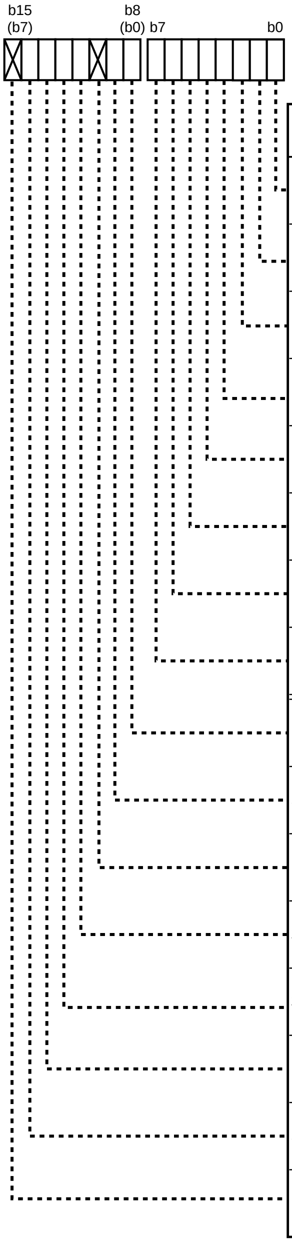
When setting the SLEEP bit to "1", a clock starts running to exit sleep mode.

Notes :

1. Sleep mode should be entered after setting the STATE_RESET bit to "1" (CAN module reset is completed).

4. CAN0 Status Register (C0STR Register)

CAN0 status register



Symbol	Address	When reset [1]
C0STR	0203_{16} - 0202_{16}	$X000\ 0X01\ 0000\ 0000_2$

Bit symbol	Bit name	Function	RW
MBOX0	Active slot determination bit	b3 b2 b1 b0 0 0 0 0 : Message slot 0 0 0 0 1 : Message slot 1 0 0 1 0 : Message slot 2 0 0 1 1 : Message slot 3 • • • • 1 1 0 1 : Message slot 13 1 1 1 0 : Message slot 14 1 1 1 1 : Message slot 15	RO
MBOX1			RO
MBOX2			RO
MBOX3			RO
TRMSUCC	Transmit completed status	0: Transmit incompleted 1: Transmit completed	RO
RECSUCC	Receive completed status	0: Receive incompleted 1: Receive completed	RO
TRMSTATE	Transmit status	0: Not transmitting 1: Transmitting	RO
RECSTATE	Receive status	0: Not receiving 1: Receiving	RO
STATE_RESET	CAN reset status	0: Resetting 1: CAN module reset is completed	RO
STATE_LOOPBACK	Loop back status	0: Except Loop back mode 1: Loop back mode	RO
——	Nothing is assigned. When write, set to "0". When read, its content is indeterminate.		—
STATE_BASICCAN	Basic CAN status	0: Except Basic CAN mode 1: Basic CAN mode	RO
STATE_BUSERROR	CAN bus error	0: No error occurred 1: Error occurred	RO
STATE_ERRPAS	Error passive status	0: No error passive state 1: Error passive state	RO
STATE_BUSOFF	Bus-off status	0: No bus-off state 1: Bus-off state	RO
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—

Notes :

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

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.

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 C0CTLR0 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 (C0CTLR0 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).

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

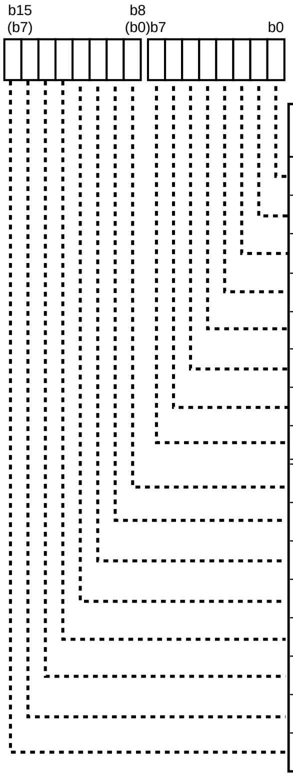
The STATE_BUSOFF bit is set to "1" when value of the C0TEC register exceeds 255 with the CAN module placed in a bus-off state.

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).

5. CAN0 Extended ID Register **(C0IDR Register)**

CAN0 extended ID register



Symbol	Address	When reset [1]
C0IDR	0205_{16} - 0204_{16}	0000_{16}

Bit symbol	Bit name	Function	RW
IDE15	Extended ID15 (message slot 15)	Standard or extended format is set by the corresponding message slot 0: Standard format 1: Extended format	RW
IDE14	Extended ID14 (message slot 14)		RW
IDE13	Extended ID13 (message slot 13)		RW
IDE12	Extended ID12 (message slot 12)		RW
IDE11	Extended ID11 (message slot 11)		RW
IDE10	Extended ID10 (message slot 10)		RW
IDE9	Extended ID9 (message slot 9)		RW
IDE8	Extended ID8 (message slot 8)		RW
IDE7	Extended ID7 (message slot 7)		RW
IDE6	Extended ID6 (message slot 6)		RW
IDE5	Extended ID5 (message slot 5)		RW
IDE4	Extended ID4 (message slot 4)		RW
IDE3	Extended ID3 (message slot 3)		RW
IDE2	Extended ID2 (message slot 2)		RW
IDE1	Extended ID1 (message slot 1)		RW
IDE0	Extended ID0 (message slot 0)		RW

Notes :

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

Figure 1.21.7. C0IDR Register

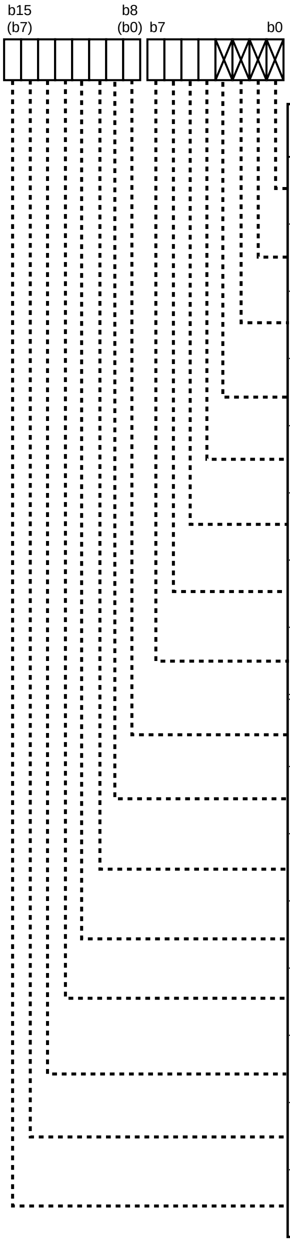
Bits in the C0IDR register determines a frame format in the message slot for each bit.
Standard format is selected when setting the above bit to "0".
Extended format is selected when setting the above bit to "1".

Notes :

1. 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)

CAN0 configuration register



Symbol	Address	When reset [1]
C0CONR	0207_{16} - 0206_{16}	0000 0000 0000 $XXXX_2$

Bit symbol	Bit name	Function	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–
——			–
——			–
SAM	Sampling number	0: Sampled once 1: Sampled three times	RW
PTS0	Propagation Time Segment	b7 b6 b5 0 0 0 : PTS = 1Tq 0 0 1 : PTS = 2Tq 0 1 0 : PTS = 3Tq 0 1 1 : PTS = 4Tq 1 0 0 : PTS = 5Tq 1 0 1 : PTS = 6Tq 1 1 0 : PTS = 7Tq 1 1 1 : PTS = 8Tq	RW
PTS1			RW
PTS2			RW
PBS10	Phase Buffer Segment 1	b10 b9 b8 0 0 0 : Avoid this setting 0 0 1 : PBS1 = 2Tq 0 1 0 : PBS1 = 3Tq 0 1 1 : PBS1 = 4Tq 1 0 0 : PBS1 = 5Tq 1 0 1 : PBS1 = 6Tq 1 1 0 : PBS1 = 7Tq 1 1 1 : PBS1 = 8Tq	RW
PBS11			RW
PBS12			RW
PBS20	Phase Buffer Segment 2	b13 b12 b11 0 0 0 : Avoid this setting 0 0 1 : PBS2 = 2Tq 0 1 0 : PBS2 = 3Tq 0 1 1 : PBS2 = 4Tq 1 0 0 : PBS2 = 5Tq 1 0 1 : PBS2 = 6Tq 1 1 0 : PBS2 = 7Tq 1 1 1 : PBS2 = 8Tq	RW
PBS21			RW
PBS22			RW
SJW0	reSynchronization Jump Width	b15 b14 0 0 : SJW = 1Tq 0 1 : SJW = 2Tq 1 0 : SJW = 3Tq 1 1 : SJW = 4Tq	RW
SJW1			RW

Notes :

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

PTS: Propagation Time Segment, PBS1: Phase Buffer Segment 1, PBS2: Phase Buffer Segment 2, SJW: reSynchronization Jump Width

Figure 1.21.8. C0CONR Register

SAM Bit

The SAM bit determines the number of smples of the CAN_{IN} 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 "001_2" or more.

PBS22 to PBS20 Bits

The PBS22 to PBS20 bits determine PBS2 wide. The PBS22 to PBS20 bits should be set to "001_2" or more.

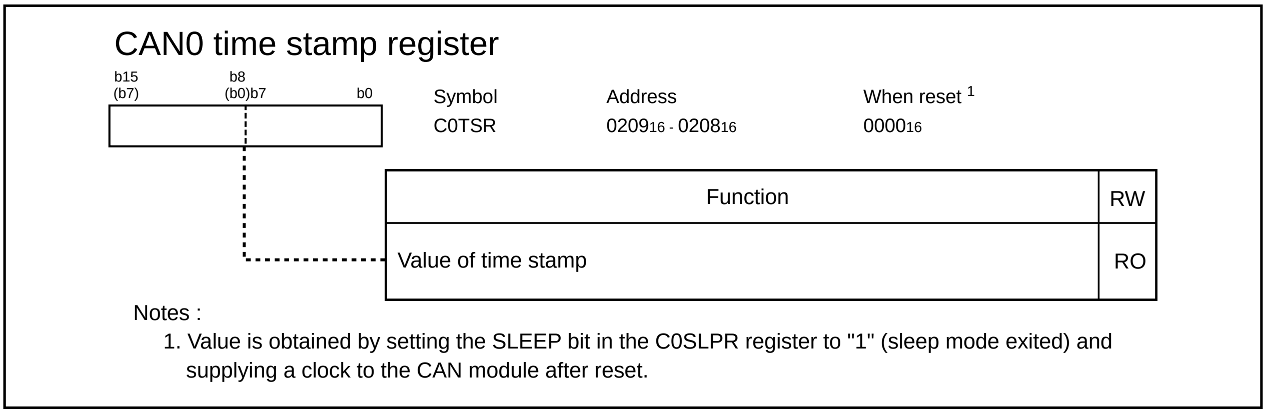
SJW1 to SJW0 Bits

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	Tq clock cycles (ns)	Tq 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)



Symbol	Address	When reset [1]
C0TSR	0209_{16} - 0208_{16}	0000_{16}

Function	RW
Value of time stamp	RO

Notes :

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

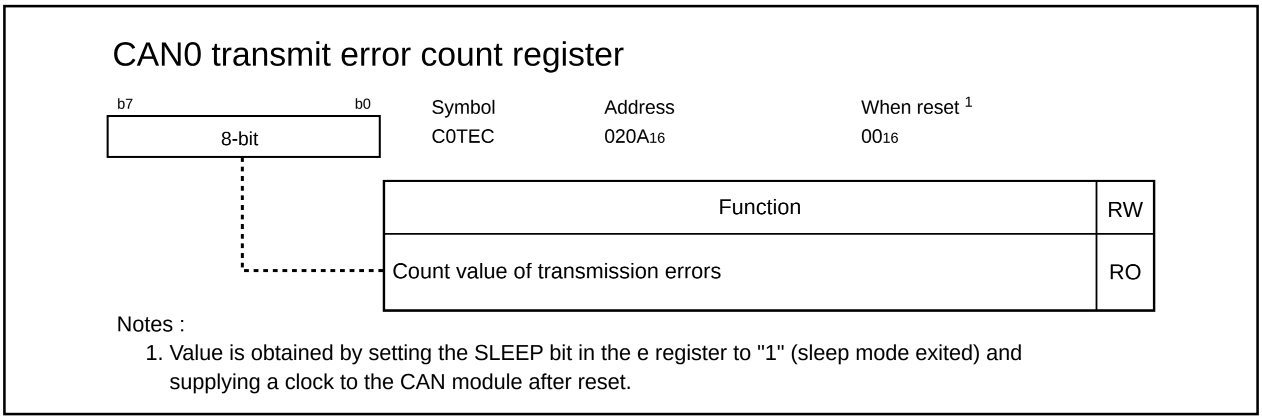
Figure 1.21.9. C0TSR Register

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 C0CTLR0 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 C0CTLR0 register are set to "0".
The C0TSR register is set to "0000_{16}" upon the following conditions:

- At the next count timing after the C0TSR register is set to "$FFFF_{16}$"
- 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)



Symbol	Address	When reset [1]
C0TEC	$020A_{16}$	00_{16}

Function	RW
Count value of transmission errors	RO

Notes :

1. Value is obtained by setting the SLEEP bit in the e register to "1" (sleep mode exited) and supplying a clock to the CAN module after reset.

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_{16}" when the CAN module is placed in an error active state again.

9. CAN0 Receive Error Count Register (C0REC Register)

CAN0 receive error count register

b7 8-bit b0

Symbol	Address	When reset [1]
C0REC	020B$_{16}$	00$_{16}$

Function	RW
Count value of reception error	RO

Notes :

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

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$_{16}$" the CAN module is placed in entering an error active state again.

10. CAN0 Baud Rate Prescaler (C0BPR Register)

CAN0 baud rate prescaler

b7 b0

Symbol	Address	When reset [1]
C0BRP	0217$_{16}$	01$_{16}$

Function	Setting range	RW
As a setting value is n, CPU clock is divided by (n+1)	01$_{16}$ to FF$_{16}$[2]	RW

Notes :

1. Value is obtained by setting the SLEEP bit in the C0SLPR register to "1" (sleep mode exited) and supplying a clock to the CAN module after reset.
2. Avoid setting to "00$_{16}$" (divide-by-1).

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.

$$\text{Tq clock cycle} = (\text{BRP}+1) / f_1$$

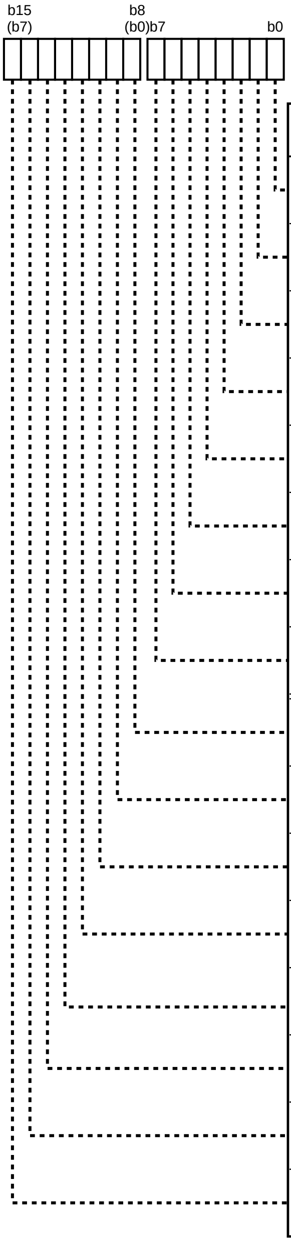
$$\text{Baud rate} = \frac{1}{\text{Tq clcok cycle x Tq per bit}}$$

$$\text{Tq per bit} = \text{SS} + \text{PTS} + \text{PBS1} + \text{PBS2}$$

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

11. CAN0 Slot Interrupt Status Register (C0SISTR Register)

CAN0 slot interrupt status register



Symbol	Address	When reset [1]
C0SISTR	$020D_{16}$ - $020C_{16}$	$0000\ 0000\ 0000\ 0000_2$

Bit symbol	Bit name	Function	RW
SIS15	Message slot 15 interrupt request status bit	Determines whether interrupt for a corresponding message slot is requested or not. 0: No interrupt requested 1: Interrupt requested (Note 2)	RW
SIS14	Message slot 14 interrupt request status bit		RW
SIS13	Message slot 13 interrupt request status bit		RW
SIS12	Message slot 12 interrupt request status bit		RW
SIS11	Message slot 11 interrupt request status bit		RW
SIS10	Message slot 10 interrupt request status bit		RW
SIS9	Message slot 9 interrupt request status bit		RW
SIS8	Message slot 8 interrupt request status bit		RW
SIS7	Message slot 7 interrupt request status bit		RW
SIS6	Message slot 6 interrupt request status bit		RW
SIS5	Message slot 5 interrupt request status bit		RW
SIS4	Message slot 4 interrupt request status bit		RW
SIS3	Message slot 3 interrupt request status bit		RW
SIS2	Message slot 2 interrupt request status bit		RW
SIS1	Message slot 1 interrupt request status bit		RW
SIS0	Message slot 0 interrupt request status bit		RW

Notes :

1. Value is obtained by setting the SLEEP bit in the C0SLPR register to "1" (sleep mode exited) and supplying a clock to the CAN module after reset.
2. Each bit can be set to "0" by program. When setting it to "1", the value before setting to "1" is remained unchanged.

Figure 1.21.13. C0SISTR Register

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[1].
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.

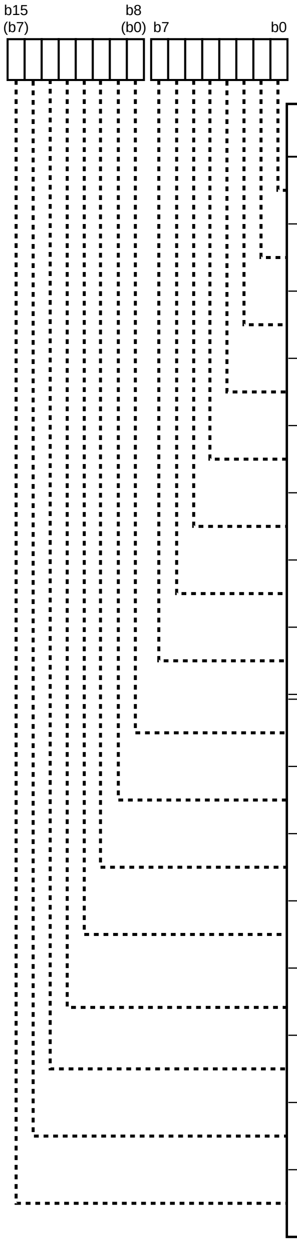
- **Message Slot Set for Reception**
  The SISi bit is set to "1" when the CAN module stores a received message in the message slot i after reception completes.

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.

12. CAN0 Slot Interrupt Mask Register (C0SIMKR Register)

CAN0 slot interrupt mask register



Symbol C0SIMKR
Address 0211_{16} - 0210_{16}
When reset [1] $0000\ 0000\ 0000\ 0000_2$

Bit symbol	Bit name	Function	RW
SIM15	Slot 15 interrupt request mask bit	Determines whether an interrupt request of the corresponding message slot is enabled or disabled. 0: Interrupt request masked (disabled) 1: Interrupt request enabled	RW
SIM14	Slot 14 interrupt request mask bit		RW
SIM13	Slot 13 interrupt request mask bit		RW
SIM12	Slot 12 interrupt request mask bit		RW
SIM11	Slot 11 interrupt request mask bit		RW
SIM10	Slot 10 interrupt request mask bit		RW
SIM9	Slot 9 interrupt request mask bit		RW
SIM8	Slot 8 interrupt request mask bit		RW
SIM7	Slot 7 interrupt request mask bit		RW
SIM6	Slot 6 interrupt request mask bit		RW
SIM5	Slot 5 interrupt request mask bit		RW
SIM4	Slot 4 interrupt request mask bit		RW
SIM3	Slot 3 interrupt request mask bit		RW
SIM2	Slot 2 interrupt request mask bit		RW
SIM1	Slot 1 interrupt request mask bit		RW
SIM0	Slot 0 interrupt request mask bit		RW

Notes :

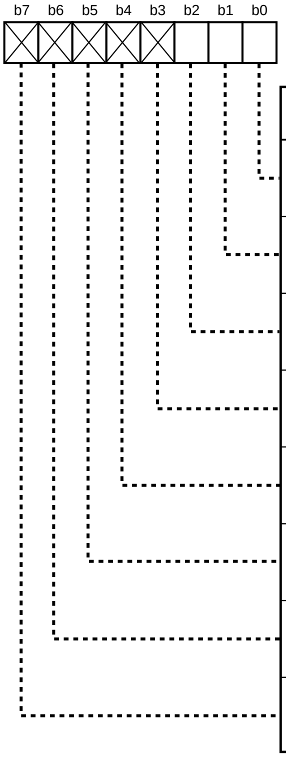
1. Value is obtained by setting the SLEEP bit in the C0SLPR register to "1" (sleep mode exited) and supplying a clock to the CAN module after reset.

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 SIMi 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)

CAN0 error interrupt mask register



Symbol	Address	When reset [1]
C0EIMKR	0214_{16}	XXXX X000$_2$

Bit symbol	Bit name	Function	RW
BOIM	Bus off interrupt mask bit	0: Interrupt request masked (disabled) 1: Interrupt request enabled	RW
EPIM	Error passive interrupt mask bit	0: Interrupt request masked (disabled) 1: Interrupt request enabled	RW
BEIM	CAN bus error interrupt mask bit	0: Interrupt request masked (disabled) 1: Interrupt request enabled	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–
——			–
——			–
——			–

Notes :

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

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

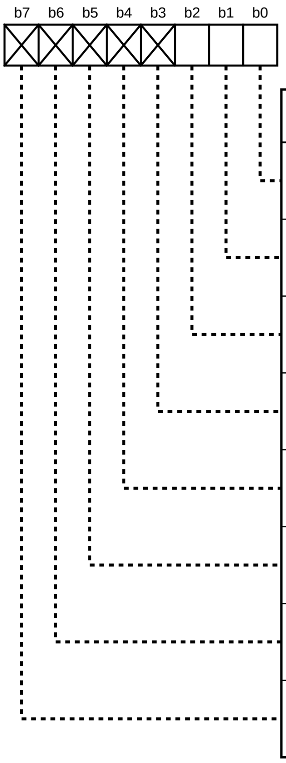
The EPIM bit determines whether an interrupt request is enabled or disabled when the CAN module is placed in an error passive state. When setting the EPIM bit to "1", an error passive interrupt request is enabled.

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.

14. CAN0 Error Interrupt Status Register (C0EISTR Register)

CAN0 error interrupt status register



Symbol	Address	When reset [1]
C0EISTR	0215$_{16}$	XXXX X000$_2$

Bit symbol	Bit name	Function	RW
BOIS	Bus-off interrupt status bit	0: No interrupt requested 1: Interrupt requested (Note 2)	RW
EPIS	Error-passive interrupt status bit	0: No interrupt requested 1: Interrupt requested (Note 2)	RW
BEIS	CAN bus error interrupt status bit	0: No interrupt requested 1: Interrupt requested (Note 2)	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–
——			–
——			–
——			–

Notes :

1. Value is obtained by setting the SLEEP bit in the C0SLPR register to "1" (sleep mode exited) and supplying a clock to the CAN module after reset.
2. Each bit can be set to "0" by program. When setting it to "1", the value before setting to "1" is remained unchanged.

Figure 1.21.16. C0EISTR Register

With the CAN interrupt, the C0EISTR register determines how an interrupt caused by occuring 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.[1]
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 :

1. 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)

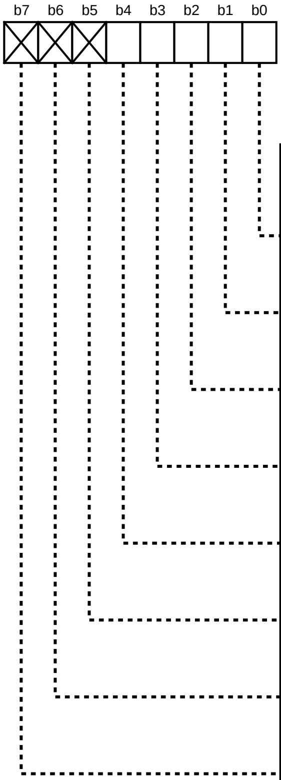
The C0GMRj, C0LMARj and C0LMBRj registers are used with the acceptance filter.
The C0GMRj register determines whether an identifier (ID) check for the message slots 0 to 13 is executed. The C0LMARj register determines whether an ID check for the massage slot 14 is executed. The C0LMBRj register determines 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.

CAN0 global mask register standard ID0[1]
CAN0 local mask register A standard ID0[1]
CAN0 local mask register B standard ID0[1]



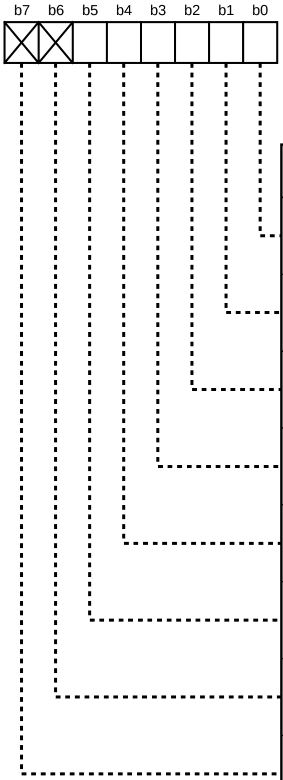
Symbol	Address	When reset [2]
C0GMR0	0228_{16}	XXX0 0000_2
C0LMAR0	0230_{16} [3]	XXX0 0000_2
C0LMBR0	0238_{16} [4]	XXX0 0000_2

Bit symbol	Bit name	Function	RW
SID6M	Standard ID6	0: No ID checked 1: ID checked	RW
SID7M	Standard ID7		RW
SID8M	Standard ID8		RW
SID9M	Standard ID9		RW
SID10M	Standard ID10		RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–
——			–

Notes :
1. This register can be accessed when the BANKSEL bit in the C0CTLR1 register is set to "1".
2. Value is obtained by setting the SLEEP bit in the C0SLPR register to "1" (sleep mode exited) and supplying the clock to the CAN module after reset.
3. The C0LMAR0 register shares the same address with the C0MCTL0 register.
4. The C0LMBR0 register shares the same address with the C0MCTL8 register.

Figure 1.21.17. C0GMR0, C0LMAR0 and C0LMBR0 Registers

CAN0 global mask register standard ID1[1]
CAN0 local mask register A standard ID1[1]
CAN0 local mask register B standard ID1[1]



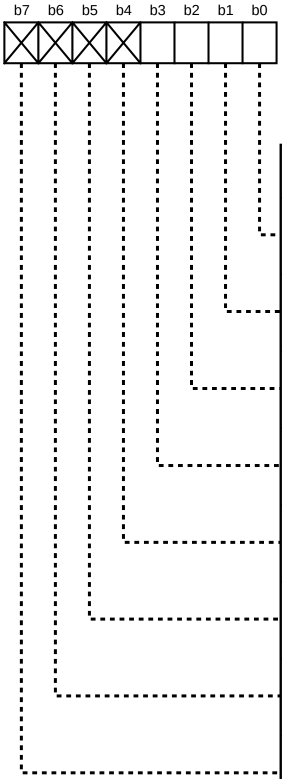
Symbol	Address	When reset[2]
C0GMR1	0229_{16}	$XX00\ 0000_2$
C0LMAR1	0231_{16}[3]	$XX00\ 0000_2$
C0LMBR1	0239_{16}[4]	$XX00\ 0000_2$

Bit symbol	Bit name	Function	RW
SID0M	Standard ID0	0: No ID checked 1: ID checked	RW
SID1M	Standard ID1		RW
SID2M	Standard ID2		RW
SID3M	Standard ID3		RW
SID4M	Standard ID4		RW
SID5M	Standard ID5		RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–

Notes :
1. This register can be accessed when the BANKSEL bit in the C0CTLR1 register is set to "1".
2. Value is obtained by setting the SLEEP bit in the C0SLPR register to "1" (sleep mode exited) and supplying the clock to the CAN module after reset.
3. The C0LMAR1 register shares the same address with the C0MCTL1 register.
4. The C0LMBR1 register shares the same address with the C0MCTL9 register.

Figure 1.21.18. C0GMR1, C0LMAR1 and C0LMBR1 Registers

CAN0 global mask register extended ID0 [1]
CAN0 local mask register A extended ID0 [1]
CAN0 local mask register B extended ID0 [1]



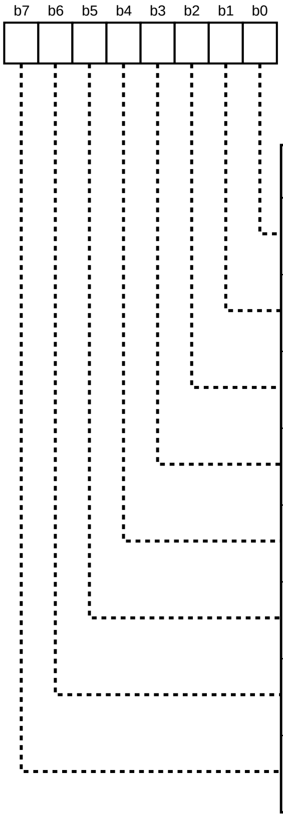
Symbol	Address	When reset [2]
C0GMR2	022A$_{16}$	XXXX 0000$_2$
C0LMAR2	0232$_{16}$ [3]	XXXX 0000$_2$
C0LMBR2	023A$_{16}$ [4]	XXXX 0000$_2$

Bit symbol	Bit name	Function	RW
EID14M	extended ID14	0: No ID checked 1: ID checked	RW
EID15M	extended ID15		RW
EID16M	extended ID16		RW
EID17M	extended ID17		RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
——			—
——			—
——			—

Notes :
1. This register can be accessed when the BANKSEL bit in the C0CTLR1 register is set to "1".
2. Value is obtained by setting the SLEEP bit in the C0SLPR register to "1" (sleep mode exited) and supplying a clock to the CAN module after reset.
3. The C0LMAR2 register shares the same address with the C0MCTL2 register.
4. The C0LMBR2 register shares the same address with the C0MCTL10 register.

Figure 1.21.19. C0GMR2, C0LMAR2 and C0LMBR2 Registers

CAN0 global mask register extended ID1 [1]
CAN0 local mask register A extended ID1 [1]
CAN0 local mask register B extended ID1 [1]



Symbol	Address	When reset [2]
C0GMR3	$022B_{16}$	$0000\ 0000_2$
C0LMAR3	0233_{16} [3]	$0000\ 0000_2$
C0LMBR3	$023B_{16}$ [4]	$0000\ 0000_2$

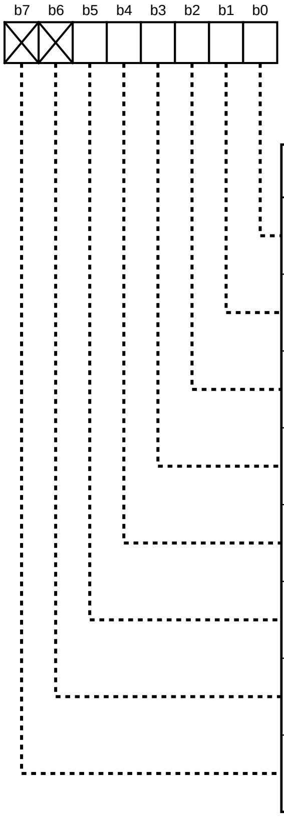
Bit symbol	Bit name	Function	RW
EID6M	extended ID6	0: No ID checked 1: ID checked	RW
EID7M	extended ID7		RW
EID8M	extended ID8		RW
EID9M	extended ID9		RW
EID10M	extended ID10		RW
EID11M	extended ID11		RW
EID12M	extended ID12		RW
EID13M	extended ID13		RW

Notes :

1. This register can be accessed when the BANKSEL bit in the C0CTLR1 register is set to "1".
2. Value is obtained by setting the SLEEP bit in the C0SLPR register to "1" (sleep mode exited) and supplying a clock to the CAN module after reset.
3. The C0LMAR3 register shares the same address with the C0MCTL3 register.
4. The C0LMBR3 register shares the same address with the C0MCTL11 register.

Figure 1.21.20. C0GMR3, C0LMAR3 and C0LMBR3 Registers

CAN0 global mask register extended ID2 [1]
CAN0 local mask register A extended ID2 [1]
CAN0 local mask register B extended ID2 [1]



Symbol	Address	When reset [2]
C0GMR4	022C$_{16}$	XX00 0000$_2$
C0LMAR4	0234$_{16}$ [3]	XX00 0000$_2$
C0LMBR4	023C$_{16}$ [4]	XX00 0000$_2$

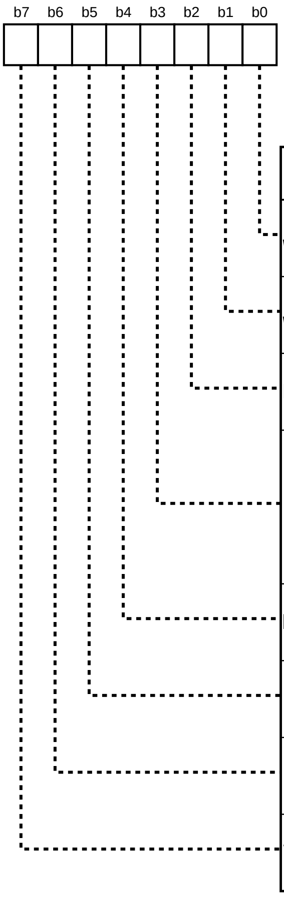
Bit symbol	Bit name	Function	RW
EID0M	extended ID0	0: No ID checked 1: ID checked	RW
EID1M	extended ID1		RW
EID2M	extended ID2		RW
EID3M	extended ID3		RW
EID4M	extended ID4		RW
EID5M	extended ID5		RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
——			—

Notes :
1. This register can be accessed when the BANKSEL bit in the C0CTLR1 register is set to "1".
2. Value is obtained by setting the SLEEP bit in the C0SLPR register to "1" (sleep mode exited) and supplying a clock to the CAN module after reset.
3. The C0LMAR4 register shares the same address with the C0MCTL4 register.
4. The C0LMBR4 register shares the same address with the C0MCTL12 register.

Figure 1.21.21. C0GMR4, C0LMAR4 and C0LMBR4 Registers

16. CAN0 Message Slot i Control Register (C0MCTLi Register)(i=0 to 15)

CAN0 message slot i control register (i=0 to 15)[1]



Symbol	Address	When reset [2]
C0MCTL0 to C0MCTL3	0230_{16}[3], 0231_{16}[3], 0232_{16}[3], 0233_{16}[3]	$0000\ 0000_2$
C0MCTL4 to C0MCTL7	0234_{16}[3], 0235_{16}, 0236_{16}, 0237_{16}	$0000\ 0000_2$
C0MCTL8 to C0MCTL11	0238_{16}[3], 0239_{16}[3], $023A_{16}$[3], $023B_{16}$[3]	$0000\ 0000_2$
C0MCTL12 to C0MCTL15	$023C_{16}$[3], $023D_{16}$, $023E_{16}$, $023F_{16}$	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
When receive, NEWDATA When transmit, SENTDATA	Receive complete flag Transmit complete flag	When transmitting: 0: Not transmitted[4] 1: Already transmitted When receiving: 0: Not received[4] 1: Already received	RW
When receive, INVALDATA When transmit, TRMACTIVE	Receiving flag Transmitting flag	When transmitting: 0: Stops transmitting 1: Accepts transmit request When receiving: 0: Stop receiving 1: Storing received data	RO
MSGLOST	Overwrite flag	0: No overrun error not occurred 1: Overrun error occurred (Note 4)	RW
REMACTIVE	Remote frame transmit/receive status flag	In modes except BasicCan mode (Note 4) 0: Data frame 1: Remote frame In BasicCan mode 0: Data frame 1: Remote frame	RW
RSPLOCK	Automatic answering disable mode select bit	0: Automatic answering of remote frame enabled 1: Automatic answering of remote frame disabled	RW
REMOTE	Remote frame set bit	0: Transmits/receives data frame 1: Transmits/receives remote frame	RW
RECREQ	Receive request bit	0: No receive requested 1: Receive requested	RW
TRMREQ	Transmit request bit	0: No transmit requested 1: Transmit requested	RW

Notes :

1. This register can be accessed when the BANKSEL bit in the C0CTLR1 register is set to "1".
2. Value is obtained by setting the SLEEP bit in the C0SLPR register to "1" (sleep mode exited) and supplying a clock to the CAN module after reset.
3. The C0MCTL0 to C0MCTL4 registers share addresses with the C0MAR0 to C0MAR4 registers.
4. Each bit can be set to "0" by program. When setting it to "1", the value before setting to "1" is remained unchanged.

Figure 1.21.22. C0MCTL0 to C0MCTL15 Registers

Table 1.21.4 C0MCTLi register(i = 0 to 15) Settings and Transmit/Receive Mode

Settings for the C0MCTLi register								Transmit/receive mode
TRMREC	RECREQ	REMOTE	RSPLOCK	REMACTIVE	MSGLOST	TRMACTIVE INVALDATA	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.)

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/INVALDATA 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 NEWDADA 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/INVALDATA Bit

The TRMACTIVE/INVALDATA bit indicates that the CAN module is transmitting or receiving a message and accessing the message slot i. The TRMACTIVE/INVALDATA 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.

INVALDATA : The INVALDATA 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 C0MCTL14 and C0MCTL15 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

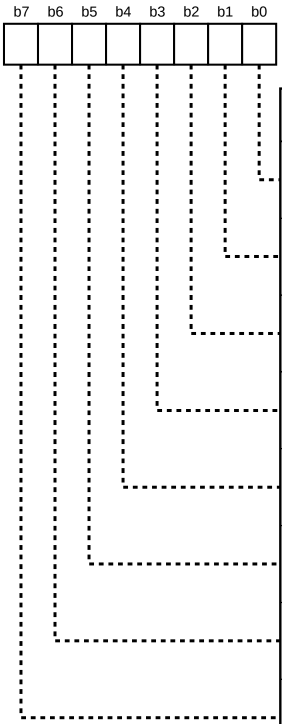
The RECREQ bit selects a transmit/receive mode shown in Table 1.21.4. The RECREQ bit should be set to "1" (receive requested) when receiving a data frame or a remote frame. The RECREQ bit should be set to "0" (no receive requested) when transmitting a data frame or a remote frame.
When automatically 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).

17. CAN0 Slot Buffer Select Register (C0SBS Register)

CAN0 slot buffer select register



Symbol	Address	When reset[2]
C0SBS	0240_{16}	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
SBS00	CAN0 message slot buffer 0 number select bit	b3 b2 b1 b0 0 0 0 0 : Message slot 0 0 0 0 1 : Message slot 1 0 0 1 0 : Message slot 2 0 0 1 1 : Message slot 3 ⋮ ⋮ (Note 1) 1 1 0 0 : Message slot 12 1 1 0 1 : Message slot 13 1 1 1 0 : Message slot 14 1 1 1 1 : Message slot 15	RW
SBS01			RW
SBS02			RW
SBS03			RW
SBS10	CAN0 message slot buffer 1 number select bit	b3 b2 b1 b0 0 0 0 0 : Message slot 0 0 0 0 1 : Message slot 1 0 0 1 0 : Message slot 2 0 0 1 1 : Message slot 3 ⋮ ⋮ (Note 1) 1 1 0 0 : Message slot 12 1 1 0 1 : Message slot 13 1 1 1 0 : Message slot 14 1 1 1 1 : Message slot 15	RW
SBS11			RW
SBS12			RW
SBS13			RW

Notes :

1. 16 CAN0 message slots are provided in total. Each message slot can be selected as a transmit or a receive slot.
2. Value is obtained by setting the SLEEP bit in the C0SLPR register to "1" (sleep mode exited) and supplying a clock to the CAN module after reset.

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 ($01E0_{16}$ to $01EF_{16}$).

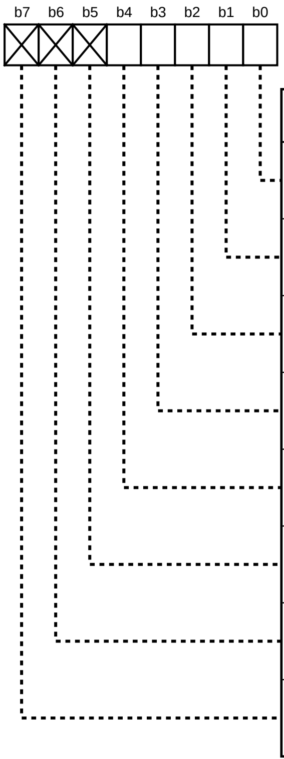
SBS13 to SBS10 Bits

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 ($01F0_{16}$ to $01FF_{16}$).

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.

CAN0 message slot buffer i standard ID0 (i=0,1)[1]



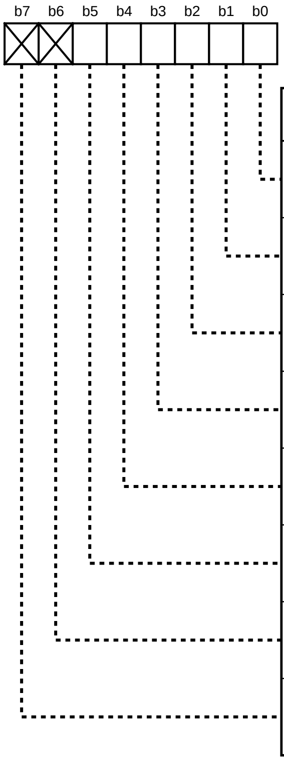
Symbol	Address	When reset
C0SLOT0_0, C0SLOT1_0	$01E0_{16}$, $01F0_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
SID6	Standard ID6	Read or write message slot j (j=0 to 15) standard ID6	RW
SID7	Standard ID7	Read or write message slot j standard ID7	RW
SID8	Standard ID8	Read or write message slot j standard ID8	RW
SID9	Standard ID9	Read or write message slot j standard ID9	RW
SID10	Standard ID10	Read or write message slot j standard ID10	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–
——			–

Notes :

1. The C0SBS register should select the message slot j to be accessed by this register.

CAN0 message slot buffer i standard ID1 (i=0,1)[1]



Symbol	Address	When reset
C0SLOT0_1, C0SLOT1_1	$01E1_{16}$, $01F1_{16}$	Indeterminate

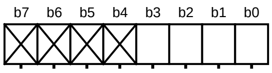
Bit symbol	Bit name	Function	RW
SID0	Standard ID0	Read or write message slot j (j=0 to 15) standard ID0	RW
SID1	Standard ID1	Read or write message slot j standard ID1	RW
SID2	Standard ID2	Read or write message slot j standard ID2	RW
SID3	Standard ID3	Read or write message slot j standard ID3	RW
SID4	Standard ID4	Read or write message slot j standard ID4	RW
SID5	Standard ID5	Read or write message slot j standard ID5	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–

Notes :

1. The C0SBS register should select the message slot j to be accessed by this register.

Figure 1.21.24. C0SLOT0_0, C0SLOT1_0, C0SLOT0_1 and C0SLOT1_1 Registers

CAN0 message slot buffer i extended ID0 (i=0,1) [1,2]

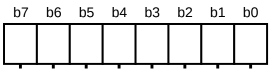


Symbol	Address	When reset
C0SLOT0_2, C0SLOT1_2	$01E2_{16}$, $01F2_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
EID14	extended ID14	Read or write message slot j (j=0 to 15) extended ID14	RW
EID15	extended ID15	Read or write message slot j extended ID15	RW
EID16	extended ID16	Read or write message slot j extended ID16	RW
EID17	extended ID17	Read or write message slot j extended ID17	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–
——			–
——			–

Notes :
1. When receive slot is standard ID format, the EID bits are indeterminate when storing a received data.
2. The C0SBS register should select the message slot j to be accessed by this register.

CAN0 message slot buffer i extended ID1 (i=0,1) [1,2]



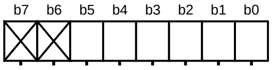
Symbol	Address	When reset
C0SLOT0_3, C0SLOT1_3	$01E3_{16}$, $01F3_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
EID6	extended ID6	Read or write message slot j (j=0 to 15) extended ID6	RW
EID7	extended ID7	Read or write message slot j extended ID 7	RW
EID8	extended ID8	Read or write message slot j extended ID 8	RW
EID9	extended ID9	Read or write message slot j extended ID 9	RW
EID10	extended ID10	Read or write message slot j extended ID 10	RW
EID11	extended ID11	Read or write message slot j extended ID 11	RW
EID12	extended ID12	Read or write message slot j extended ID 12	RW
EID13	extended ID13	Read or write message slot j extended ID 13	RW

Notes :
1. When receive slot is standard ID format, the EID bits are indeterminate when storing a received data.
2. The C0SBS register should select the message slot j to be accessed by this register.

Figure 1.21.25. C0SLOT0_2, C0SLOT1_2, C0SLOT0_3 and C0SLOT1_3 Registers

CAN0 message slot buffer i extended ID2 (i=0,1) [1,2]

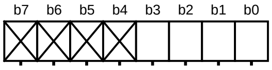


Symbol	Address	When reset
C0SLOT0_4, C0SLOT1_4	$01E4_{16}$, $01F4_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
EID0	extended ID0	Read or write message slot j (j=0 to 15) extended ID0	RW
EID1	extended ID1	Read or write message slot j extended ID1	RW
EID2	extended ID2	Read or write message slot j extended ID2	RW
EID3	extended ID3	Read or write message slot j extended ID3	RW
EID4	extended ID4	Read or write message slot j extended ID4	RW
EID5	extended ID5	Read or write message slot j extended ID5	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–

Notes :
1. When receive slot is standard ID format, the EID bits are indeterminate when storing a received data.
2. The C0SBS register should select the message slot j to be accessed by this register.

CAN0 message slot buffer i data length code (i=0,1) [1]



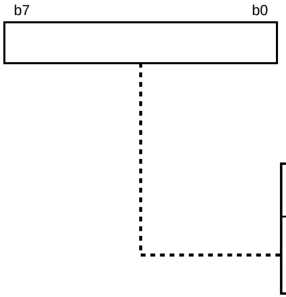
Symbol	Address	When reset
C0SLOT0_5, C0SLOT1_5	$01E5_{16}$, $01F5_{16}$	Indeterminate

Bit symbol	Bit name	Function	RW
DLC0	Data length set bit	Read or write message slot j (j=0 to 15) data length set bit	RW
DLC1			RW
DLC2			RW
DLC3			RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–
——			–
——			–

Notes :
1. The C0SBS register should select the message slot j to be accessed by this register.

Figure 1.21.26. C0SLOT0_4, C0SLOT1_4, C0SLOT0_5 and C0SLOT1_5 Registers

CAN0 message slot buffer i data m (i=0,1 k=0 to 7) [1]

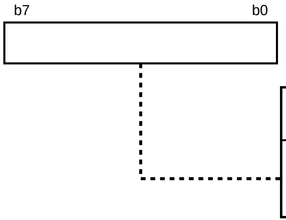


Symbol	Address	When reset
C0SLOT0_q(q=k+6,k=0 to 3)	01E616, 01E716, 01E816, 01E916	Indeterminate
C0SLOT0_q(q=k+6,k=4 to 7)	01EA16, 01EB16, 01EC16, 01ED16	Indeterminate
C0SLOT1_q(q=k+6,k=0 to 3)	01F616, 01F716, 01F816, 01F916	Indeterminate
C0SLOT1_q(q=k+6,k=4 to 7)	01FA16, 01FB16, 01FC16, 01FD16	Indeterminate

Function	Setting range	RW
Read or write message slot j data k (j=0 to 15)	00_{16} to FF_{16}	RW

Notes :
1. The C0SBS register should select data k in the message slot j to be accessed by this register.

CAN0 message slot buffer i time stamp high (i=0,1)[1]

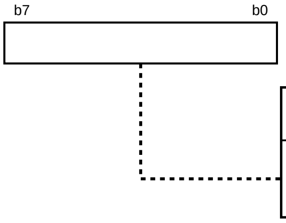


Symbol	Address	When reset
C0SLOT0_14, C0SLOT1_14	$01EE_{16}$, $01FE_{16}$	Indeterminate

Function	Setting range	RW
Read or write in message slot j time stamp high-order (j=0 to 15)	00_{16} to FF_{16}	RW

Notes :
1. The COSBS register should select the time stamp high-order in the message slot j to be accessed by this register.

CAN0 message slot buffer i time stamp low (i=0,1) [1]



Symbol	Address	When reset
C0SLOT0_15, C0SLOT1_15	$01EF_{16}$, $01FF_{16}$	Indeterminate

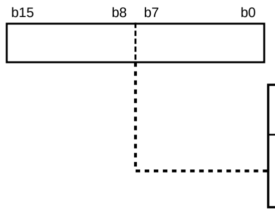
Function	Setting range	RW
Read or write in message slot j time stamp low-order (j=0 to 15)	00_{16} to FF_{16}	RW

Notes :
1. The C0SBS register should select the time stamp low-order in the message slot j to be accessed by this register.

Figure 1.21.27. C0SLOT0_6 to C0SLOT0_13, C0SLOT1_6 to C0SLOT1_13, C0SLOT0_14, C0SLOT1_14, C0SLOT0_15 and C0SLOT1_15 Registers

19. CAN0 Acceptance Filter Support Register

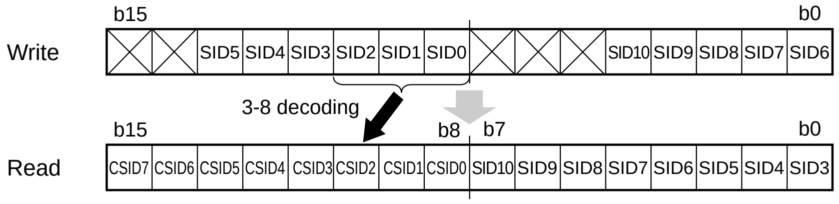
CAN0 acceptance filter support register



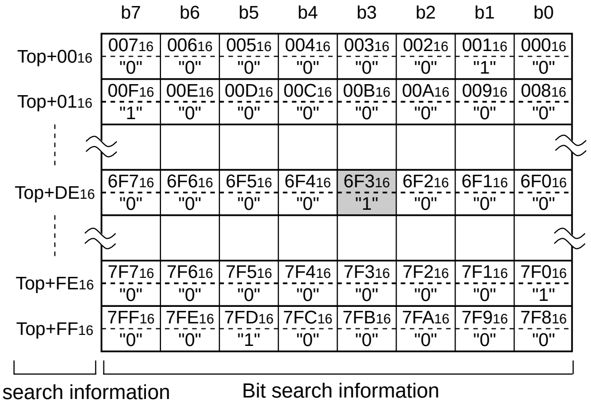
Symbol	Address	When reset[1]
C0AFS	0245_{16} - 0244_{16}	0100_{16}

Function	Setting range	RW
Generates data to determine received data	0000_{16} to $FFFF_{16}$	RW

Notes:
1. Value is obtained by setting the SLEEP bit in the C0SLPR register to "1" (sleep mode exited) and supplying a clock to the CAN module after reset.



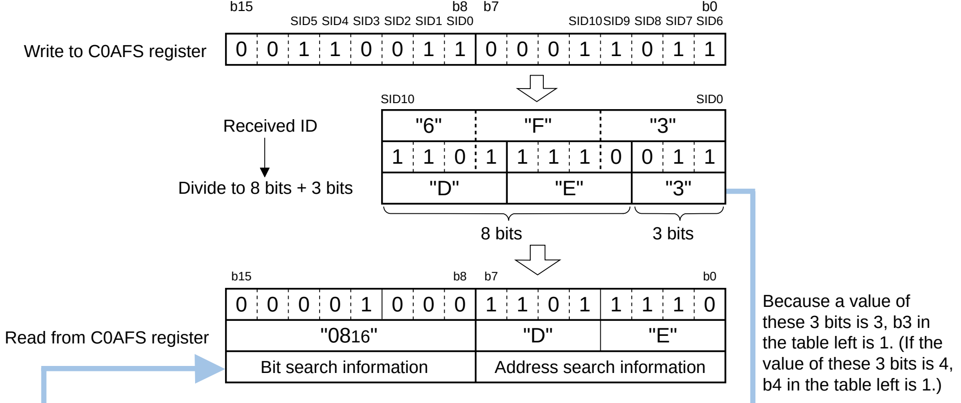
Data to search a data table is generated with a received ID of standard formated. The table search with this data determines whether a received ID is valid or not.



	b7	b6	b5	b4	b3	b2	b1	b0
Top+00_{16}	007_{16} "0"	006_{16} "0"	005_{16} "0"	004_{16} "0"	003_{16} "0"	002_{16} "0"	001_{16} "1"	000_{16} "0"
Top+01_{16}	$00F_{16}$ "1"	$00E_{16}$ "0"	$00D_{16}$ "0"	$00C_{16}$ "0"	$00B_{16}$ "0"	$00A_{16}$ "0"	009_{16} "0"	008_{16} "0"
⋮								
Top+DE_{16}	$6F7_{16}$ "0"	$6F6_{16}$ "0"	$6F5_{16}$ "0"	$6F4_{16}$ "0"	$6F3_{16}$ "1"	$6F2_{16}$ "0"	$6F1_{16}$ "0"	$6F0_{16}$ "0"
⋮								
Top+FE_{16}	$7F7_{16}$ "0"	$7F6_{16}$ "0"	$7F5_{16}$ "0"	$7F4_{16}$ "0"	$7F3_{16}$ "0"	$7F2_{16}$ "0"	$7F1_{16}$ "0"	$7F0_{16}$ "1"
Top+FF_{16}	$7FF_{16}$ "0"	$7FE_{16}$ "0"	$7FD_{16}$ "1"	$7FC_{16}$ "0"	$7FB_{16}$ "0"	$7FA_{16}$ "0"	$7F9_{16}$ "0"	$7F8_{16}$ "0"

Address search information   Bit search information

When a received ID is "$6F3_{16}$"



Because a value of these 3 bits is 3, b3 in the table left is 1. (If the value of these 3 bits is 4, b4 in the table left is 1.)

Bit search information	b7 b3 b0	3 low-order bits of received ID
01_{16}	0 0 0 0 0 0 0 1	0_{16}
02_{16}	0 0 0 0 0 0 1 0	1_{16}
04_{16}	0 0 0 0 0 1 0 0	2_{16}
08_{16}	0 0 0 0 1 0 0 0	3_{16}
10_{16}	0 0 0 1 0 0 0 0	4_{16}
20_{16}	0 0 1 0 0 0 0 0	5_{16}
40_{16}	0 1 0 0 0 0 0 0	6_{16}
80_{16}	1 0 0 0 0 0 0 0	7_{16}

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.

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 C0CTLR0 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).

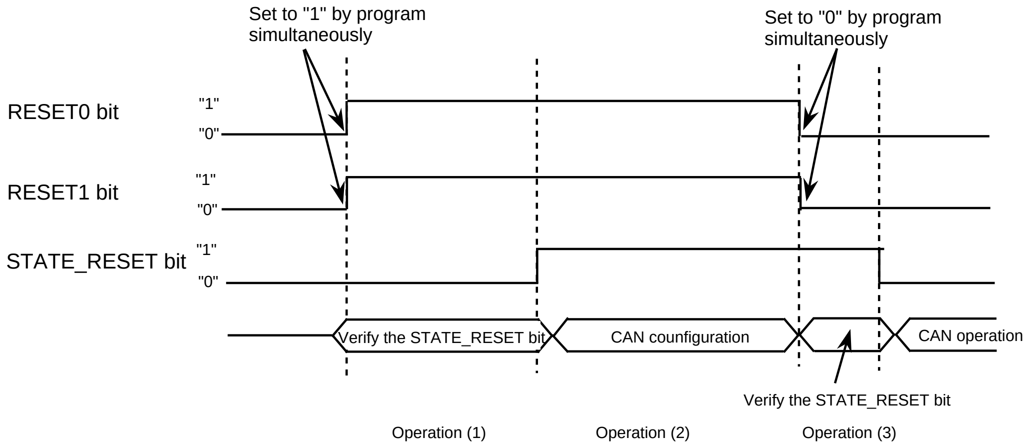

Set to "1" by program simultaneously
Set to "0" by program simultaneously
RESET0 bit
"1"
"0"
RESET1 bit
"1"
"0"
STATE_RESET bit
"1"
"0"
Verify the STATE_RESET bit
CAN counfiguration
CAN operation
Verify the STATE_RESET bit
Operation (1)
Operation (2)
Operation (3)

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 C0MCTLi 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 C0MCTLi 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.

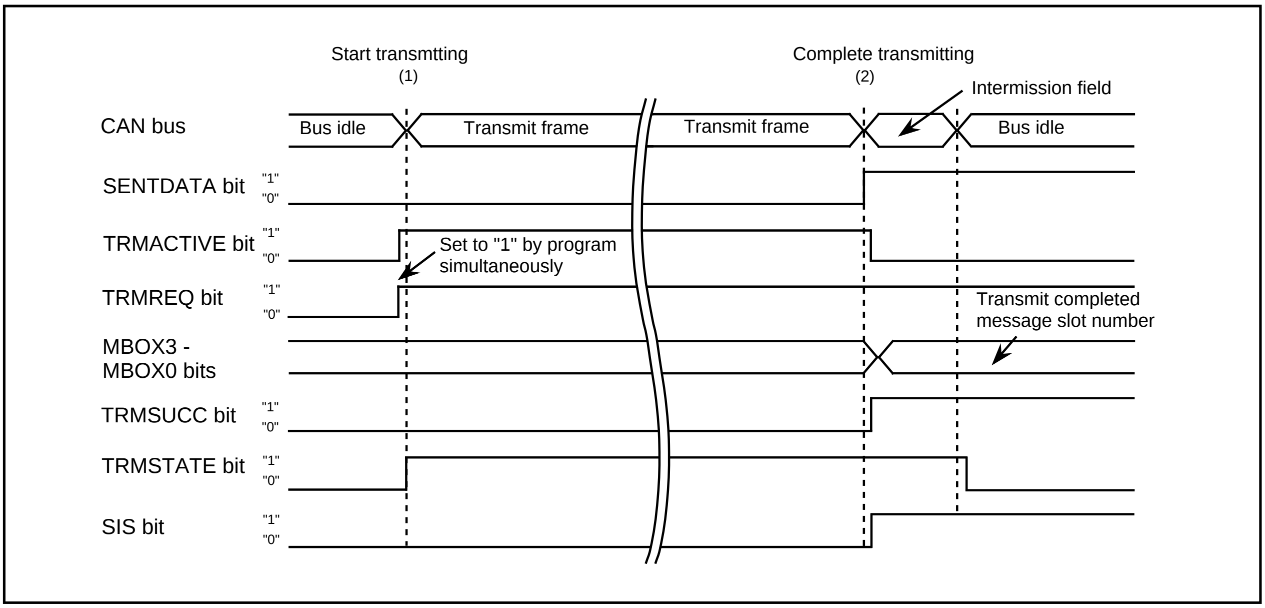


Figure 1.21.30 Operation Timing at CAN data frame transmission

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 INVALDTA 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 INVALDATA 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.

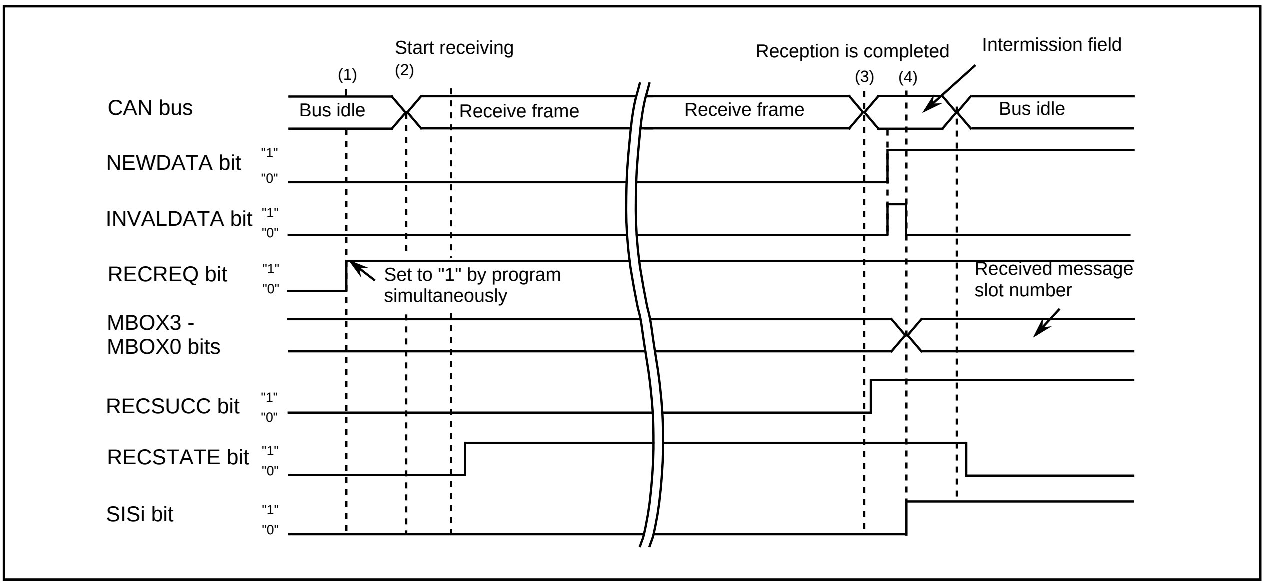


Figure 1.21.31 Operation Timing at CAN data frame reception

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.

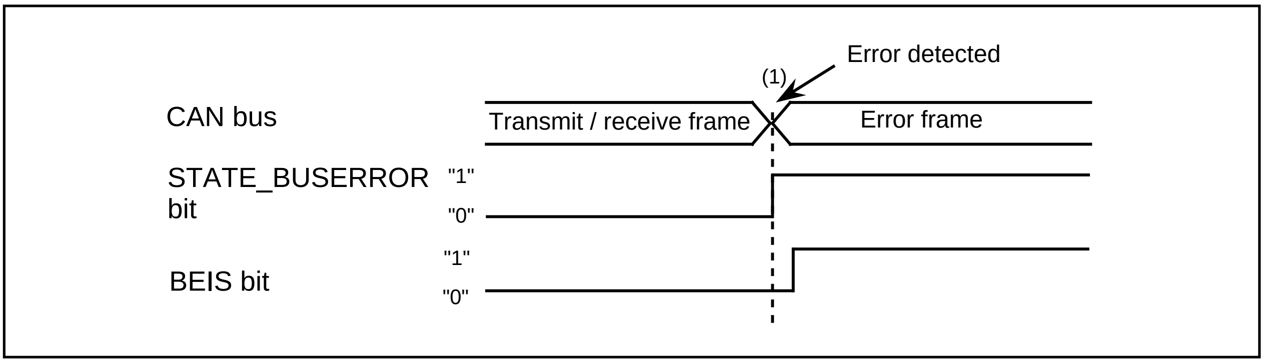


Figure 1.21.32. Operation timing when a CAN bus error occurred

CAN Interrupts

CANj 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 CANjR bits in the IIO9IR to IIO11IR registers are set to "1" (interrupt request).
If at least one of the CANjE bits in the IIO9IR to IIO11IR registers is set to "1", the IR bit in the corresponding CANjIC 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 CANjR and IR bits also remain unchanged.
Bits in the C0SISTR or C0EISTR register and CANjR 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 CANjR bits to acknowledge an interrupt. The
C0SIMKR or C0EIMKR register enables the C0SISTR or C0EISTR register to acknowledge an interrupt.

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.

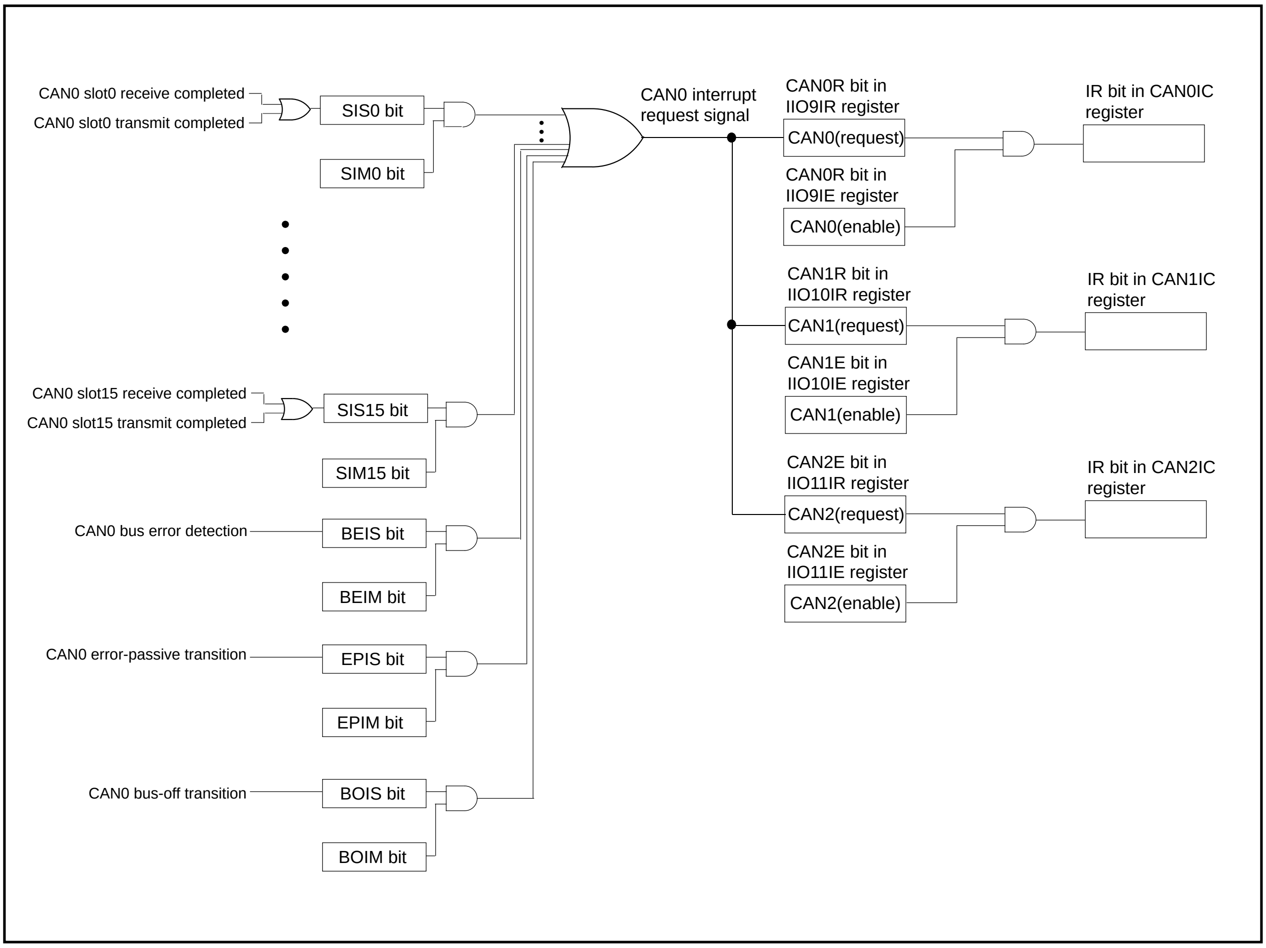

CAN0 slot0 receive completed
CAN0 slot0 transmit completed
SIS0 bit
SIM0 bit
CAN0 slot15 receive completed
CAN0 slot15 transmit completed
SIS15 bit
SIM15 bit
CAN0 bus error detection
BEIS bit
BEIM bit
CAN0 error-passive transition
EPIS bit
EPIM bit
CAN0 bus-off transition
BOIS bit
BOIM bit
CAN0 interrupt request signal
CAN0R bit in IIO9IR register
CAN0(request)
CAN0R bit in IIO9IE register
CAN0(enable)
IR bit in CAN0IC register
CAN1R bit in IIO10IR register
CAN1(request)
CAN1E bit in IIO10IE register
CAN1(enable)
IR bit in CAN1IC register
CAN2E bit in IIO11IR register
CAN2(request)
CAN2E bit in IIO11IE register
CAN2(enable)
IR bit in CAN2IC register

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[1] 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[1]	8 channels (3 channels)[2]	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[1]	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		

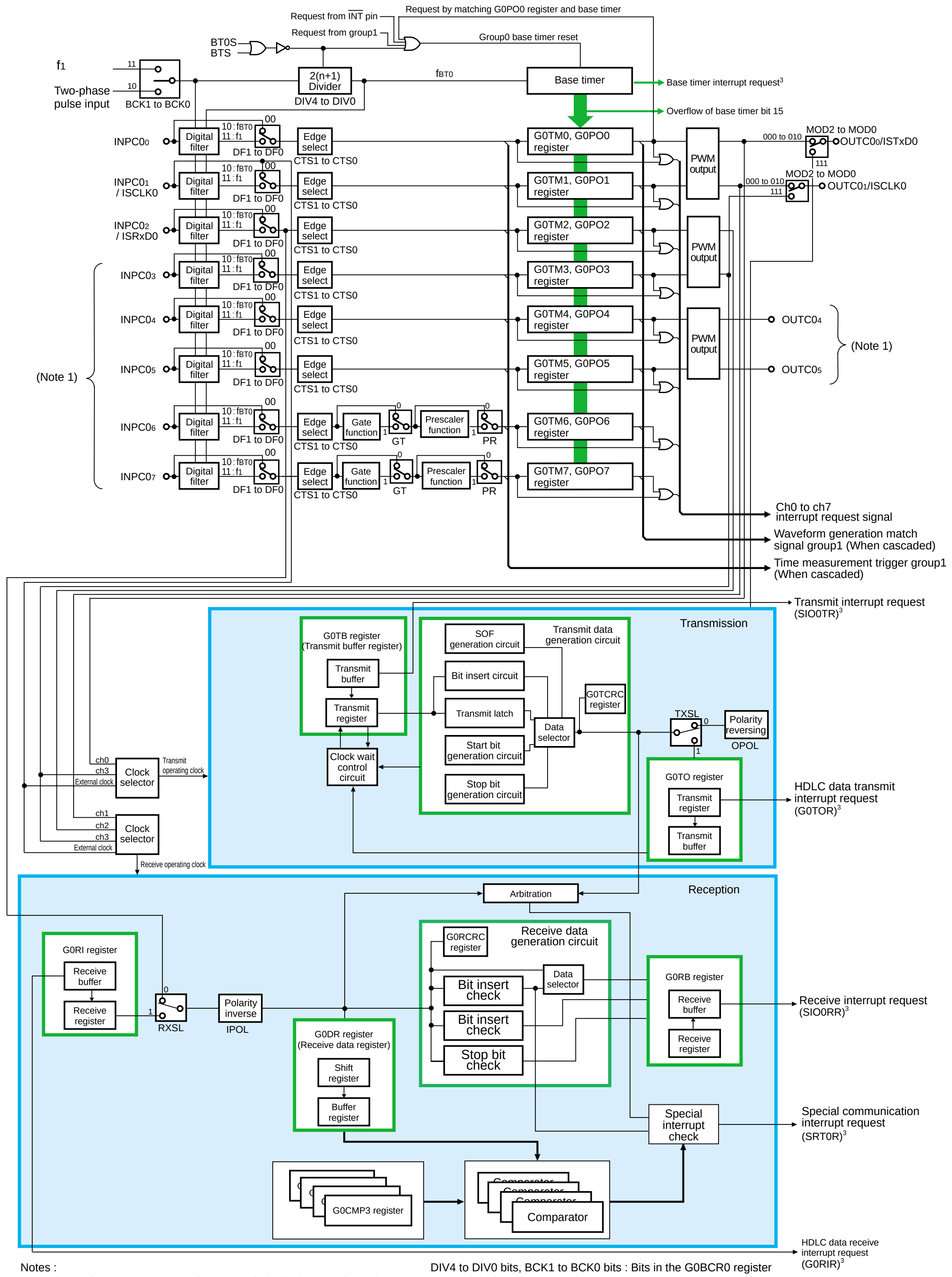
Notes :

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.

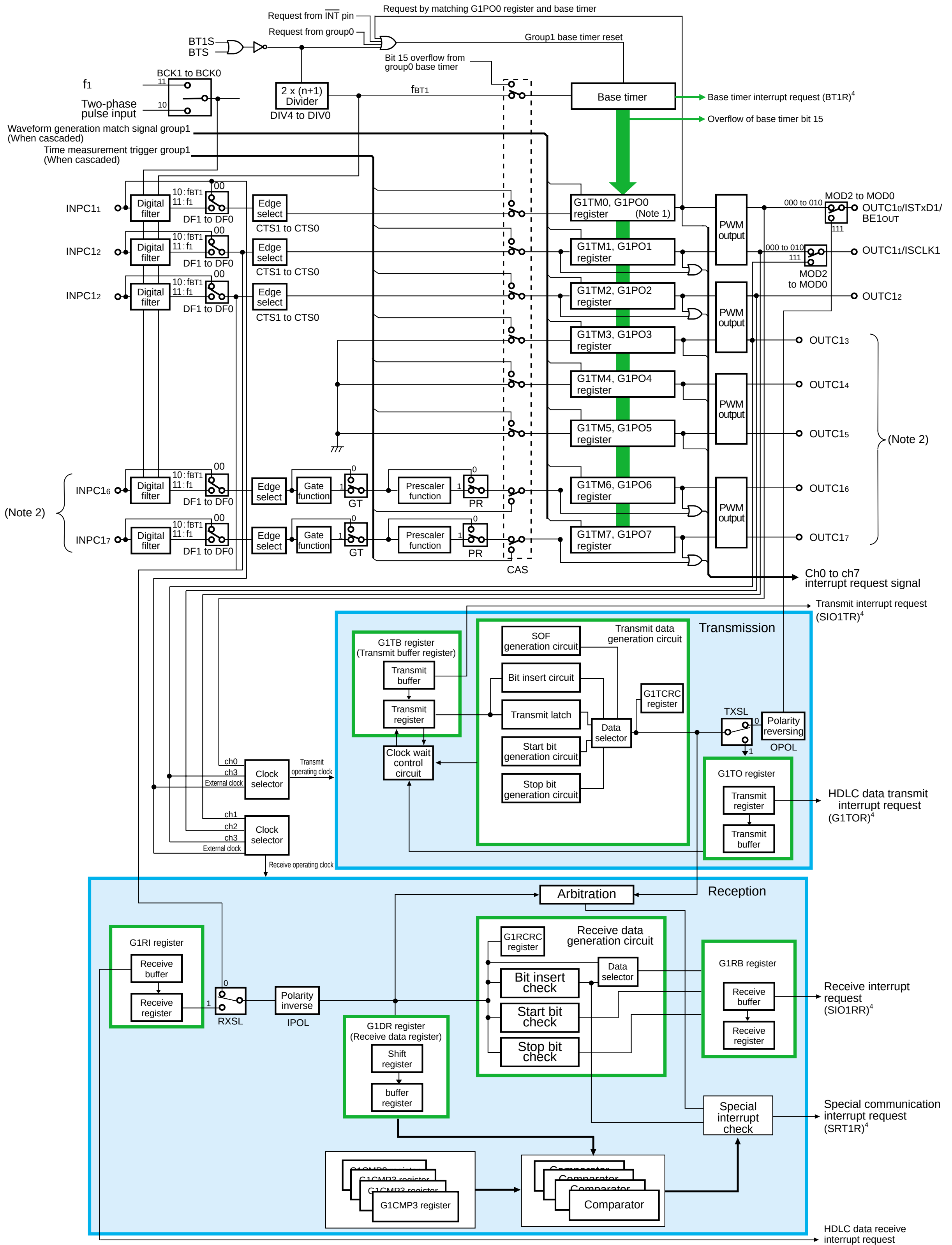
Figures 1.22.1 to 1.22.4 show block diagrams of the intelligent I/O group 0 to 3.



Notes :
1. These pins are not connected to external pins in the 100-pin package.
2. Each register is in a reset status after the G0BCR0 register allocates a clock.
3. See Figure 1.9.14.

DIV4 to DIV0 bits, BCK1 to BCK0 bits : Bits in the G0BCR0 register
BTS : Bit in the G0BCR1 register
BT0S : Bit in BTSR register
CTS1 to CTS0, DF1 to DF0, GT, PR : Bits in the G0TMCRj register (j = 0 to 7)
MOD2 to MOD0 : Bits in the G0POCRj register
TXSL, RXSL : Bits in the G0EMR register
OPOL, IPOL : Bits in the G0CR register

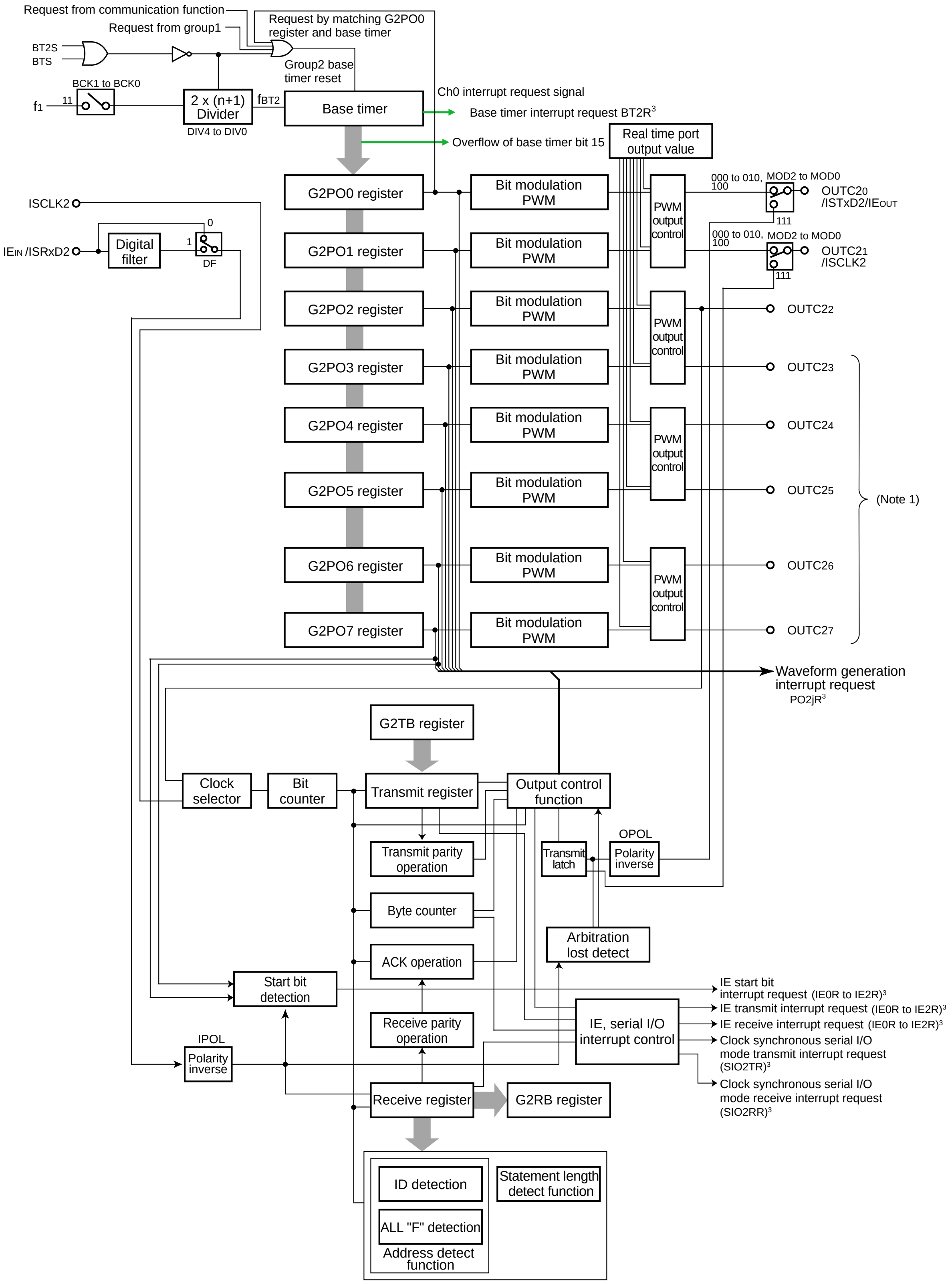
Figure 1.22.1. Intelligent I/O Group 0 Block Diagram



Notes :
1. G1TM0 register can be used in 32-bit cascade connections only.
2. These pins are not connected to external pins in the 100-pin package.
3. Each register is in a reset status after the G1BCR0 register allocates a clock.
4. See Figure 1.9.14.

BCK1 to BCK0 : Bits in the G1BCR0 register
CAS, BTS : Bits in the G1BCR1 register
BT1S : Bit in the BTSR register
DF1 to DF0, GT, PR : Bits in the G1TMCRj register (j = 0 to 7)
MOD2 to MOD0 : Bits in the G1POCRj register
TXSL, RXSL : Bits in the G1EMR register

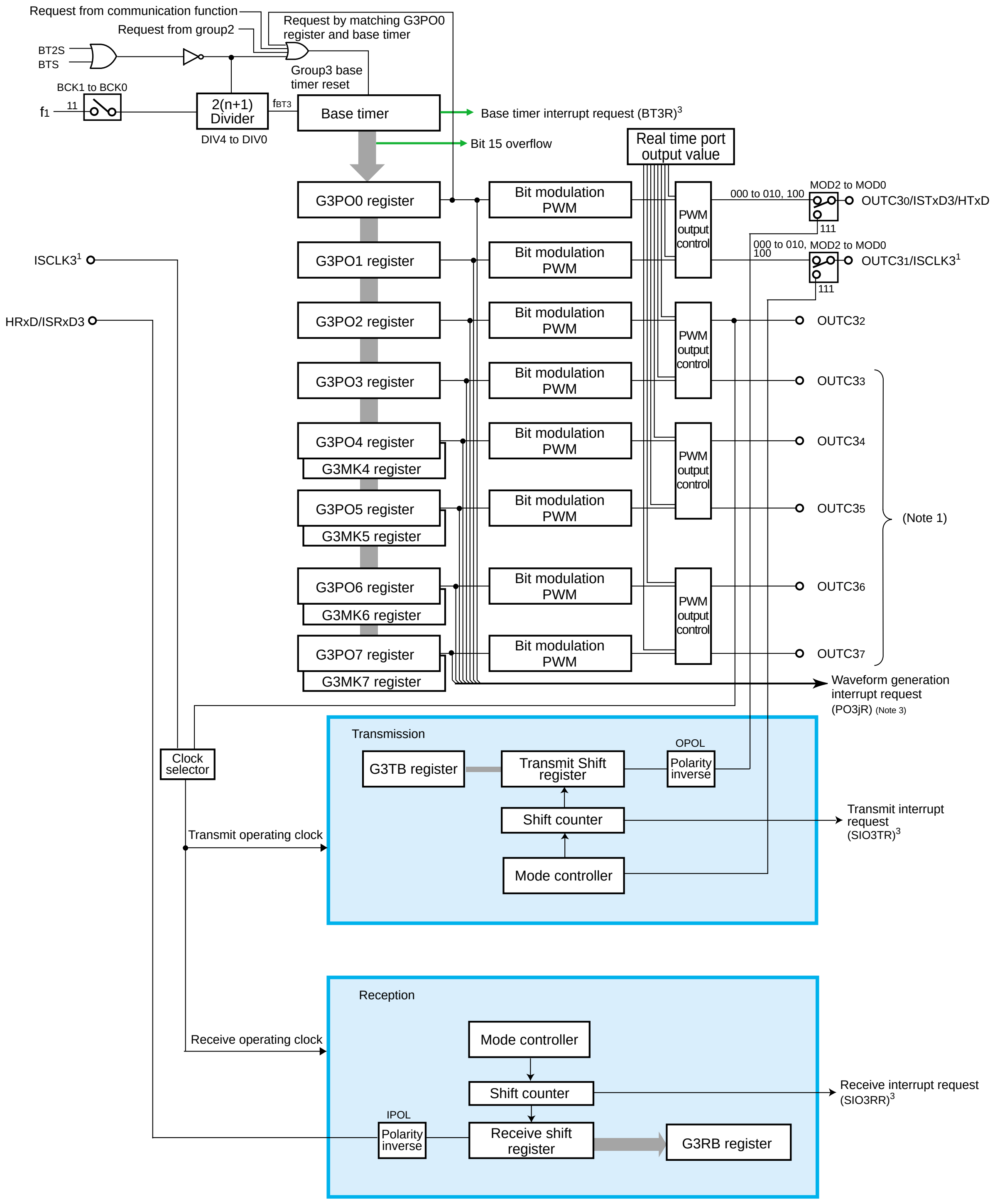
Figure 1.22.2. Intelligent I/O Group 1 Block Diagram



Notes :
1. These pins are not connected to external pins in the 100-pin package.
2. Each register is in a reset status after the G2BCR0 register allocates a clock.
3. See Figure 1.9.14.

DIV4 to DIV0, BCK1 to BCK0 : Bits in the G2BCR0 register
BTS : Bit in the G2BCR1 register
BT2S : Bit in the BTSR register
OPOL, IPOL : Bits in the G2CR register
DF : Bit in the IECR register
MOD2 to MOD0 : Bits in the G2POCRj register (j = 0 to 7)

Figure 1.22.3. Intelligent I/O Group 2 Block Diagram



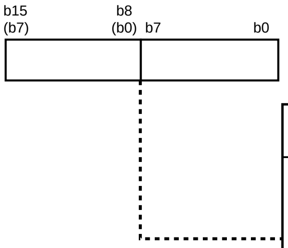
Notes :
1. These pins are not connected to external pins in the 100-pin package.
2. Each register is in a reset status after the G3BCR0 register allocates a clock.
3. See Figure 1.9.14.

DIV4 to DIV0, BCK1 to BCK0 : Bits in the G3BCR0 register
BTS : Bit in the G3BCR1 register
BT3S : Bit in the BTSR register
MOD2 to MOD0 : Bits in the G3POCRj register (j = 0 to 7)
OPOL, IPOL : Bits in the G3CR register

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.)

Group i base timer register (i=0 to 3)



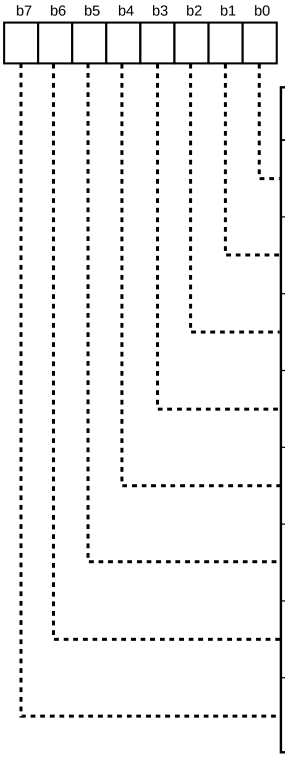
Symbol	Address	When reset
G0BT,G1BT	$00E1_{16}$ - $00E0_{16}$, 0121_{16} - 0120_{16}	Indeterminate
G2BT,G3BT	0161_{16} - 0160_{16}, $01A1_{16}$ - $01A0_{16}$	Indeterminate

Function	Setting range	RW
• When the base timer is operating When read, value of the counter can be read. When write, the counter starts counting from the value written. When the base timer is reset, this register is set to "0000_{16}".[1] • When the base timer is reset This register is set to "0000_{16}" but a value read is indeterminate. No value is written.[1]	0000_{16} to $FFFF_{16}$	RW

Notes :

1. Each base timer stops only when the BCK1 to BCK0 bits in the GiBCR0 register are set to "00_2" (count source clock stop). Each base timer operates when the BCK1 to BCK0 bits are set to other than "00_2". When both BTiS bit in the BTSR register and BTS bit in the GiBCR1 register are set to "0", the base timer continues to be reset to remain in a "no counting" state as setting "0000_{16}". When either the BTiS bit in the BTSR register or the BTS bit in the GiBCR1 register is set to "1", this state is cleared to start counting.

Group i base timer control register 0 (i=0 to 3)[1]



Symbol	Address	When reset
G0BCR0 to G3BCR0	$00E2_{16}$, 0122_{16}, 0162_{16}, $01A2_{16}$	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
BCK0	Count source select bit	b1b0 0 0: Clock stop 0 1: Avoid this setting 1 0: Two-phase input[2] 1 1: f_1	RW
BCK1			RW
DIV0	Count source division ratio select bit	A count divided source by 2x(n + 1) when a setting value is n (n = 0 to 31) b6 b5 b4 b3 b2 (n=0) 0 0 0 0 0 : Divide-by-2 (n=1) 0 0 0 0 1 : Divide-by-4 (n=2) 0 0 0 1 0 : Divide-by-6 : (n=30) 1 1 1 1 0 : Divide-by-62 (n=31) 1 1 1 1 1 : No division	RW
DIV1			RW
DIV2			RW
DIV3			RW
DIV4			RW
IT	Base timer interrupt select bit	0 : Bit 15 overflow 1 : Bit 14 overflow	RW

Notes :

1. When the CAS bit in the GiBCR1 register is set to "1" (32-bit time measurement, waveform generation function), the G0BCR0 register and G1BCR0 register should be set the same value.
2. This setting can be used when the UD1 to UD0 bits in the GjBCR1 register (j=0, 1) are set to "10_2" (two-phase signal processing mode) for the group 0 or 1. Avoid setting the BCK1 to BCK0 bits to "10_2" in other modes or for the group 2 or 3.

Figure 1.22.5. G0BT to G3BT Registers and G0BCR0 to G3BCR0 Registers

Group i base timer control register 1 (i=0,1)

b7	b6	b5	b4	b3	b2	b1	b0
				0			

Symbol	Address	When reset
G0BCR1, G1BCR1	$00E3_{16}$, 0123_{16}	$0000\ 0000_2$

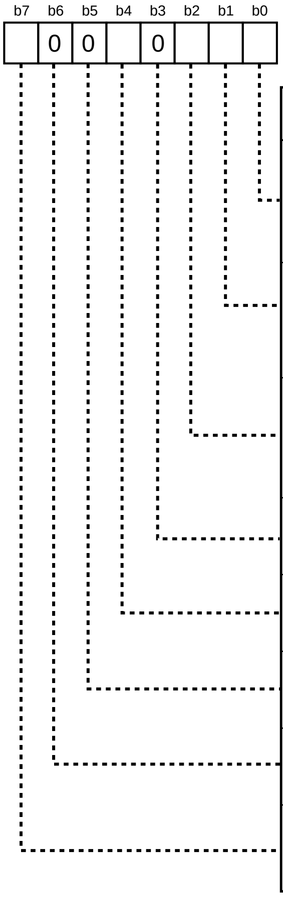
Bit symbol	Bit name	Function	RW
RST0	Base timer reset cause select bit 0	0: The base timer is not reset by synchronizing with a base timer reset 1: The base timer is reset by synchronizing with a base timer reset[1]	RW
RST1	Base timer reset cause select bit 1	0: The base timer is not reset by matching the GiPO0 register 1: The base timer is reset by matching the GiPO0 register[2]	RW
RST2	Base timer reset cause select bit 2	0: The base timer is not reset when an input to the $\overline{\text{INTi}}$ pin is "L" level 1: The base timer is reset when an input to the $\overline{\text{INTi}}$ pin is "L" level[3]	RW
——	Reserved bit	Should set to "0".	RW
BTS	Base timer start bit	0: Base timer is reset 1: Base timer starts counting	RW
UD0	Counter increment/ decrement control bit	b6b5 0 0 : Counter increment mode 0 1 : Counter increment/decrement mode 1 0 : Two-phase pulse signal processing mode 1 1 : Avoid this setting	RW
UD1			RW
CAS	Groups 0 and 1 cascaded function select bit	0: 16-bit time measurement, waveform generation function 1: 32-bit time measurement, waveform generation function[4]	RW

Notes :

1. In the group 0, the base timer is reset by synchronizing with the group 1 base timer. In the group 1, the base timer is reset by synchronizing with the group 0 base timer.
2. The base timer is reset after two clock cycles of f_{BTi} when it matches the GiPO0 register. (See Figure 1.22.13 about the GiPO0 register.) When setting the RST1 bit to "1", values of the GiPOj register (j=1 to7) to use for the waveform generation function and communication function should be set to smaller value than values of the GiPO0 register.
3. In the group 0, the base timer is reset when an input to the $\overline{\text{INT0}}$ pin is "L". In the group 1, the base timer is resets when an input to the $\overline{\text{INT1}}$ pin is "L".
4. When setting the CAS bit to "1" (32-bit time measurement, waveform generation function), the G0BCR1 register should be set to "81_{16}" and the G1BCR1 register be set to "$1000\ 0XX0_2$".
5. When each base timer starts counting in the group 0 and 1, the BTkS bit (k=0 to 1) in the BTSR register should be set to "0". Then the BTS bit should be set to "1".
6. When the base timers start counting in multiple groups simultaneously, the BTSR register should be used. The BTS bit should be set to "0".

Figure 1.22.6. G0BCR1 and G1BCR1 Registers

Group 2 base timer control register 1



Symbol	Address	When reset
G2BCR1	0163_{16}	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
RST0	Base timer reset cause select bit 0	0 : The base timer is not reset by synchronizing with a group 1 base timer reset 1 : The base timer is reset by synchronizing with a group 1 base timer reset	RW
RST1	Base timer reset cause select bit 1	0 : The base timer is not reset by matching the G2PO0 register 1 : The base timer is reset by matching the G2PO0 register[1]	RW
RST2	Base timer reset cause select bit 2	0 : The base timer is not reset by a reset request from the communication function 1 : The base timer is reset by a reset request from the communication function	RW
——	Reserved bit	Should set to "0".	RW
BTS	Base timer start bit[3,4]	0 : Base timer is reset 1 : Base timer starts counting	RW
——	Reserved bit	Should set to "0".	RW
——			RW
PRP	Parallel real-time port function select bit[2]	0 : RTP output mode 1 : Parallel RTP output mode	RW

Notes :

1. The base timer is reset after two clock cycles of f_{BT2} when it matches the G2PO0 register. (See Figure1.22.13 about the G2PO0 register.) When setting the RST1 bit to "1", values of the G2POi register (i=1 to 7) to use for the waveform generation function and communication function should be set to smaller value than values of the G2PO0 register.
2. The PRP bit is available when the RTP bit in the G2POCRi register is set to "1" (real-time port used).
3. When each base timer starts counting in the group 2, the BT2S bit in the BTSR register should be set to "0" (base timer reset). Then the BTS bit should be set to "1".
4. When the base timers start counting in multiple groups simultaneously, the BTSR register should be set. The BTS bit should be set to "0".

Figure 1.22.7. G2BCR1 Register

Group 3 base timer control register 1

b7	b6	b5	b4	b3	b2	b1	b0
	0	0		0			

Symbol	Address	When reset
G3BCR1	$01A3_{16}$	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
RST0	Base timer reset cause select bit 0	0 : The base timer is not reset by synchronizing with a group2 base timer reset 1 : The base timer is reset by synchronizing with a group2 base timer reset	RW
RST1	Base timer reset cause select bit 1	0 : The base timer is not reset by matching the G3PO0 register 1 : The base timer is reset by matching the G3PO0 register[1]	RW
RST2	Base timer reset cause select bit 2	0 : The base timer is not reset by a reset request from the communication function 1 : The base timer is reset by a reset request from the communication function	RW
——	Reserved bit	Should set to "0".	RW
BTS	Base timer start bit[3,4]	0 : Base timer is reset 1 : Base timer starts counting	RW
——	Reserved bit	Should set to "0".	RW
——			RW
PRP	Parallel real-time port function select bit[2]	0 : RTP output mode 1 : Parallel RTP output mode	RW

Notes :

1.The base timer is reset after two clock cycles of f_{BT3} when it matches the G3PO0 register. (See Figure 1.22.13 about the G3PO0 register.) When setting the RST1 bit to "1", values of the G3POi register (i=1 to 7) to use for the waveform generation function and communication function should be set to smaller value than values of the G3PO0 register.

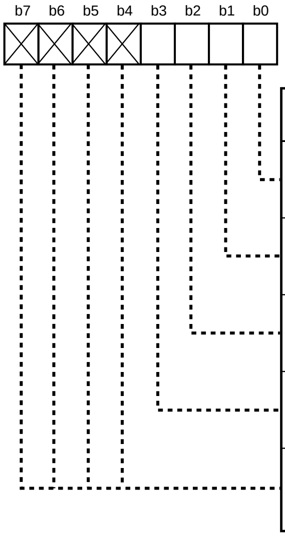
2. The PRP bit is available when the RTP bit in the G3POCRj register is set to "1" (real-time port used).

3. When the base timer starts counting in the group 3, the BT3S bit in the BTSR register should be set to "0" (base timer reset). Then the BTS bit should be set to "1".

4. When the base timers start counting in multiple groups simultaneously, the BTSR register should be used. The BTS bit should be set to "0".

Figure 1.22.8. G3BCR1 Register

Base timer start register[1,2]



Symbol	Address	When reset
BTSR	0164_{16}	$XXXX\ 0000_2$

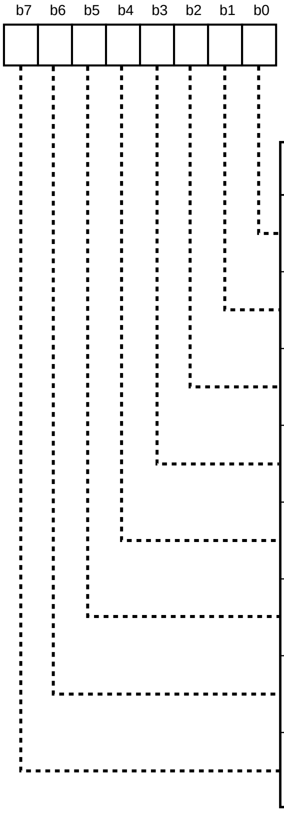
Bit symbol	Bit name	Function	RW
BT0S	Group 0 base timer start bit	0 : Base timer is reset 1 : Base timer starts counting	RW
BT1S	Group 1 base timer start bit	0 : Base timer is reset 1 : Base timer starts counting	RW
BT2S	Group 2 base timer start bit	0 : Base timer is reset 1 : Base timer starts counting	RW
BT3S	Group 3 base timer start bit	0 : Base timer is reset 1 : Base timer starts counting	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—

Notes :

1. The followings should be set first to use the intelligent I/O.
   (1) The G2BCR0 register should be set to provide the clock to the group 2 base timer.
   (2) The BT0S to BT3S bits in the BTSR register are set to "0" (base timer reset).
   (3) Other registers associated with the intelligent I/O should be set.
   The BTiS bit (i=0 to 3) starts counting the base timers in multiple groups simultaneously. When each base timer starts counting individually, the BTiS bit should be set to "0" to use the BTS bit in the GiBCR1 register.
2. When the base timers starts counting in multiple groups simultaneously (including the group 1 an 2 cascade connections), the following procedures should be taken. It is not required when each base timers starts individually.
   - The BCK1 to BCK0 bits and DIV4 to DIV0 bits in the GiBCR0 register (plural number out of i=0 to 3) in groups started simultaneously should be set to the same value.
   - After changing the BCK1 to BCK0 bits or DIV4 to DIV0 bits, the base timer should be started twice with the following procedures.
     (1) The BTiS bit in the BTSR register should be set to "1" (base timer starts counting).
     (2) The BTiS bit should be set to "0" (base timer stops counting) after one clock of f_{BTi}.
     (3) After additional one clock to f_{BTi}, the BTiS bit should be set to "1" (base timer starts counting).

Figure 1.22.9. BTSR Register

Group i time measurement control register j[1](i=0,1 / j=0 to 7)

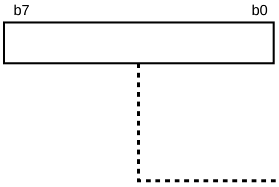


Symbol	Address	When reset
G0TMCR0 to G0TMCR3	$00D8_{16}$, $00D9_{16}$, $00DA_{16}$, $00DB_{16}$	$0000\ 0000_2$
G0TMCR4 to G0TMCR7	$00DC_{16}$, $00DD_{16}$, $00DE_{16}$, $00DF_{16}$	$0000\ 0000_2$
G1TMCR0 to G1TMCR3	0118_{16}, 0119_{16}, $011A_{16}$, $011B_{16}$	$0000\ 0000_2$
G1TMCR4 to G1TMCR7	$011C_{16}$, $011D_{16}$, $011E_{16}$, $011F_{16}$	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
CTS0	Time measurement trigger select bit	b1 b0 0 0 : No time measurement 0 1 : Rising edge 1 0 : Falling edge 1 1 : Both edges	RW
CTS1			RW
DF0	Digital filter function select bit	b3 b2 0 0 : No digital filter 0 1 : Avoid this setting 1 0 : f_{BTi} 1 1 : f_1	RW
DF1			RW
GT	Gate function select bit[2,4]	0 : Gate function not used 1 : Gate function used	RW
GOC	Gate function clear select bit[2,3,5]	0 : Not cleared 1 : The gate is cleared when the base timer matches the GiPOk register	RW
GSC	Gate function clear bit[2,3]	When setting the GSC bit to "1", a gate is cleared.	RW
PR	Prescaler function select bit[2]	0 : Not used 1 : Used	RW

Notes :
1. 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_{16}".
   When setting the CAS bit to "1" (32-bit time measurement function), the G0TMCRj and G1TMCRj registers should be set to same value.
2. This bit is in the GiTMCR6 and GiTMCR7 registers.
   All bits 4 to 7 in the GiTMCR0 to GiTMCR5 registers should set to "0".
3. These bits are available when setting the GT bit to "1".
4. 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.
5. 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)



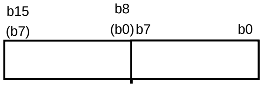
Symbol	Address	When reset
G0TPR6 to G0TPR7	$00E4_{16}$, $00E5_{16}$	$0000\ 0000_2$
G1TPR6 to G1TPR7	0124_{16}, 0125_{16}	$0000\ 0000_2$

Function	Setting range	RW
As the setting value is n, time is measured whenever a trigger input is counted by n+1[1]	00_{16} to FF_{16}	RW

Notes :
1. 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

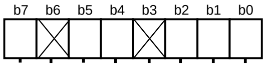
Group i time measurement register j (i=0,1 / j=0 to 7)



Symbol	Address	When reset
G0TM0 to G0TM2	$00C1_{16}$ - $00C0_{16}$, $00C3_{16}$ - $00C2_{16}$, $00C5_{16}$ - $00C4_{16}$	Indeterminate
G0TM3 to G0TM5	$00C7_{16}$ - $00C6_{16}$, $00C9_{16}$ - $00C8_{16}$, $00CB_{16}$ - $00CA_{16}$	Indeterminate
G0TM6 to G0TM7	$00CD_{16}$ - $00CC_{16}$, $00CF_{16}$ - $00CE_{16}$	Indeterminate
G1TM0 to G1TM2	0101_{16} - 0100_{16}, 0103_{16} - 0102_{16}, 0105_{16} - 0104_{16}	Indeterminate
G1TM3 to G1TM5	0107_{16} - 0106_{16}, 0109_{16} - 0108_{16}, $010B_{16}$ - $010A_{16}$	Indeterminate
G1TM6 to G1TM7	$010D_{16}$ - $010C_{16}$, $010F_{16}$ - $010E_{16}$	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[1] (i=0 to 1/ j=0 to 7)



Symbol	Address	When reset
G0POCR0 to G0POCR3	$00D0_{16}$, $00D1_{16}$, $00D2_{16}$, $00D3_{16}$	$0X00\ X000_2$
G0POCR4 to G0POCR7	$00D4_{16}$, $00D5_{16}$, $00D6_{16}$, $00D7_{16}$	$0X00\ X000_2$
G1POCR0 to G1POCR3	0110_{16}, 0111_{16}, 0112_{16}, 0113_{16}	$0X00\ X000_2$
G1POCR4 to G1POCR7	0114_{16}, 0115_{16}, 0116_{16}, 0117_{16}	$0X00\ X000_2$

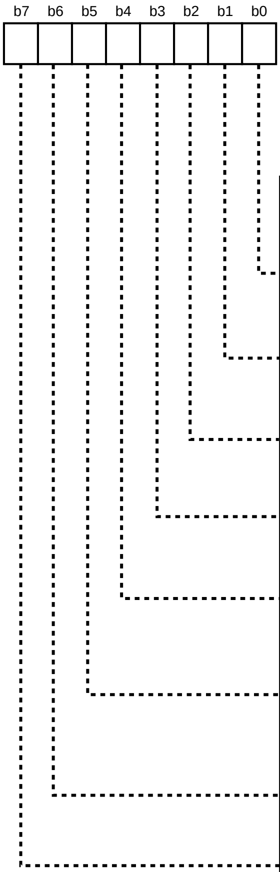
Bit symbol	Bit name	Function	RW
MOD0	Operation mode select bit	b2b1b0 0 0 0: Single waveform output mode 0 0 1: SR waveform output mode[2] 0 1 0: Phase-delayed waveform output mode 0 1 1: Avoid this setting 1 0 0: Avoid this setting 1 0 1: Avoid this setting 1 1 0: Avoid this setting[3] 1 1 1: Use communication function output[4]	RW
MOD1			RW
MOD2			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[5]	0: Output is not inversed 1: Output is inversed	RW

Notes :

1. 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 "00_{16}". When setting the CAS bit to "1" (32-bit waveform generation function), the G0POCRj and G1POCRj registers should set to the same value.
2. 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.
3. When receiving in UART mode of the group 0 and 1, the GiPOCR2 register should be set to "$0000\ 0110_2$".
4. This setting is enabled only for channel 0 and 1. When using ISTxDi, the GiPOCR0 register should set to "111_2". When using ISCLKi for output, the MOD2 to MOD0 bits in the GiPOCR1 register should set to"111_2". Avoid setting the MOD2 to MOD0 bits to "111_2" other than channels 0 and 1.
5. 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

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



Symbol	Address	When reset
G2POCR0 to G2POCR3	0150_{16}, 0151_{16}, 0152_{16}, 0153_{16}	$0000\ 0000_2$
G2POCR4 to G2POCR7	0154_{16}, 0155_{16}, 0156_{16}, 0157_{16}	$0000\ 0000_2$
G3POCR0 to G3POCR3	0190_{16}, 0191_{16}, 0192_{16}, 0193_{16}	$0000\ 0000_2$
G3POCR4 to G2POCR7	0194_{16}, 0195_{16}, 0196_{16}, 0197_{16}	$0000\ 0000_2$

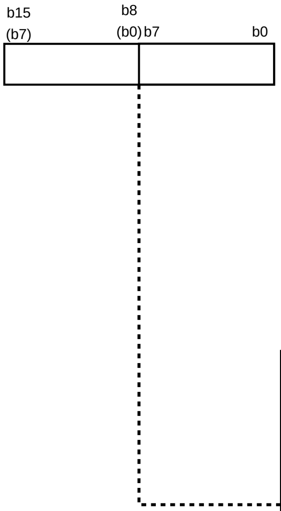
Bit symbol	Bit name	Function	RW
MOD0	Operation mode select bit[5]	b2b1b0 0 0 0: Single waveform output mode 0 0 1: SR waveform output mode[1] 0 1 0: Inverse waveform output mode 0 1 1: Avoid this setting 1 0 0: Bit modulation PWM mode 1 0 1: Avoid this setting 1 1 0: Avoid this setting 1 1 1: Use a communication function output[2]	RW
MOD1			RW
MOD2			RW
PRT	Parallel real-time port output trigger select bit[4]	0: Not triggered by the base timer matches the GiPO0 to GiPO7 register 1: Triggered by the base timer matches the GiPO0 to GiPO7 register	RW
IVL	Output initial value select bit	0: Outputs "L" as the initial value 1: Outputs "H" as the initial value	RW
RLD	GiPOj register value reload timing select bit	0: Reloads the GiPOj register when the CPU writes to counter 1: Reloads the GiPOj register when the base timer is reset	RW
RTP	Real-time port function select bit	0: Not used 1: Used (RTP output mode or parallel RTP output mode)	RW
INV	Inverse output function select bit[3]	0: Output is not inverted 1: Output is inverted	RW

Notes :

1. 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.
2. This setting is enabled only for channels 0 and 1 of the group 2 and 3. When using ISTxD2 or IEOUT, the MOD2 to MOD0 bits in the G2POCR0 register should be set to "111_2". When using ISCLK2 for an output, the MOD2 to MOD0 bits in the G2POCR1 register should be set to"111_2". Avoid setting the MOD2 to MOD0 bits to "111_2" other than channels 0 and 1.
   When using ISTxD3, the MOD2 to MOD0 bits in the G3POCR0 register should be set to "111_2". When using ISCLK3 for output, the MOD2 to MOD0 bits in the G3POCR1 register should be set to "111_2". Avoid setting the MOD2 to MOD0 bits to "111_2" other than channels 0 and 1.
3. The inverse output function is performed as a final step on a process of waveform generation. When setting the INV bit to "1" (output inversed), "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).
4. Enabled when the RPT bit is set to "1" (real-time port function used) and the PRP bit in the GiBCR1 register is set to "1" (parallel RTP output mode).
5. When setting the RTP bit to "1", value written to the MOD2 to MOD0 bits is ignored.

Figure 1.22.12. G2POCR0 to G2POCR7 and G3POCR0 to G3POCR7 Registers

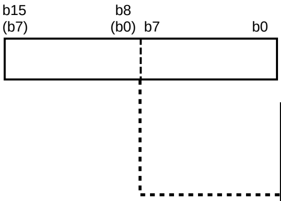
Group i waveform generation register j (i=0 to 3 / j=0 to 7)



Symbol	Address	When reset
G0PO0 to G0PO2	$00C1_{16}$-$00C0_{16}$, $00C3_{16}$-$00C2_{16}$, $00C5_{16}$-$00C4_{16}$	Indeterminate
G0PO3 to G0PO5	$00C7_{16}$-$00C6_{16}$, $00C9_{16}$-$00C8_{16}$, $00CB_{16}$-$00CA_{16}$	Indeterminate
G0PO6 to G0PO7	$00CD_{16}$-$00CC_{16}$, $00CF_{16}$-$00CE_{16}$	Indeterminate
G1PO0 to G1PO2	0101_{16}-0100_{16}, 0103_{16}-0102_{16}, 0105_{16}-0104_{16}	Indeterminate
G1PO3 to G1PO3	0107_{16}-0106_{16}, 0109_{16}-0108_{16}, $010B_{16}$-$010A_{16}$	Indeterminate
G1PO6 to G1PO7	$010D_{16}$-$010C_{16}$, $010F_{16}$-$010E_{16}$	Indeterminate
G2PO0 to G2PO2	0141_{16}-0140_{16}, 0143_{16}-0142_{16}, 0145_{16}-0144_{16}	Indeterminate
G2PO3 to G2PO5	0147_{16}-0146_{16}, 0149_{16}-0148_{16}, $014B_{16}$-$014A_{16}$	Indeterminate
G2PO6 to G2PO7	$014D_{16}$-$014C_{16}$, $014F_{16}$-$014E_{16}$	Indeterminate
G3PO0 to G3PO2	0181_{16}-0180_{16}, 0183_{16}-0182_{16}, 0185_{16}-0184_{16}	Indeterminate
G3PO3 to G3PO5	0187_{16}-0186_{16}, 0189_{16}-0188_{16}, $018B_{16}$-$018A_{16}$	Indeterminate
G3PO6 to G3PO7	$018D_{16}$-$018C_{16}$, $018F_{16}$-$018E_{16}$	Indeterminate

Function	Setting range	RW
• When the RLD bit in the GiPOCRj register should be set to "0" Value is reloaded into the GiPOj register to include it in output waveform, etc. • When the RLD bit should be set to "1" Value is reloaded when the base timer is reset. Value written can be read between writing the value and reloaded.	0000_{16} to $FFFF_{16}$	RW

Group 3 waveform generation mask register j (j=4 to 7)



Symbol	Address	When reset
G3MK4, G3MK5	0199_{16}-0198_{16}, $019B_{16}$-$019A_{16}$	Indeterminate
G3MK6, G3MK7	$019D_{16}$-$019C_{16}$, $019F_{16}$-$019E_{16}$	Indeterminate

Function	Setting range	RW
When setting the bit k (singular or plural number out of k=0 to 15) in this register to "1", the bit k of the group 3 base timer is masked to compare to the G3POj register.[1]	0000_{16} to $FFFF_{16}$	RW

Notes :

1. This is enabled in single-phase waveform output mode or phase-delayed waveform output mode. The G3MKi register should set to "0000_{16}" in other modes.

Figure 1.22.13. G0PO0 to G0PO7, G1PO0 to G1PO7, G2PO0 to G2PO7 and G3PO0 to G3PO7 Registers, G3MK4 to G3MK7 Registers

Group i function select register (i=0, 1)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
G0FS, G1FS	$00E7_{16}$, 0127_{16}	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
FSC0	Channel 0 time measurement/waveform generation function select bit	0 : Select the waveform generation function 1 : Select the time measurement function	RW
FSC1	Channel 1 time measurement/waveform generation function select bit		RW
FSC2	Channel 2 time measurement/waveform generation function select bit		RW
FSC3	Channel 3 time measurement/waveform generation function select bit		RW
FSC4	Channel 4 time measurement/waveform generation function select bit		RW
FSC5	Channel 5 time measurement/waveform generation function select bit		RW
FSC6	Channel 6 time measurement/waveform generation function select bit		RW
FSC7	Channel 7 time measurement/waveform generation function select bit		RW

Notes :

1. Channels 2, 3, 6 and 7 of the group 0 have no 16-bit waveform generation function.
   Channels 0, 3, 4 and 5 of the group 1 have no 16-bit time measurement function.
   When the CAS bit in the GiBCR1 register is set to "1" (32-bit time measurement function, waveform generation function), the G0SF and G1SF registers should be set to same value.

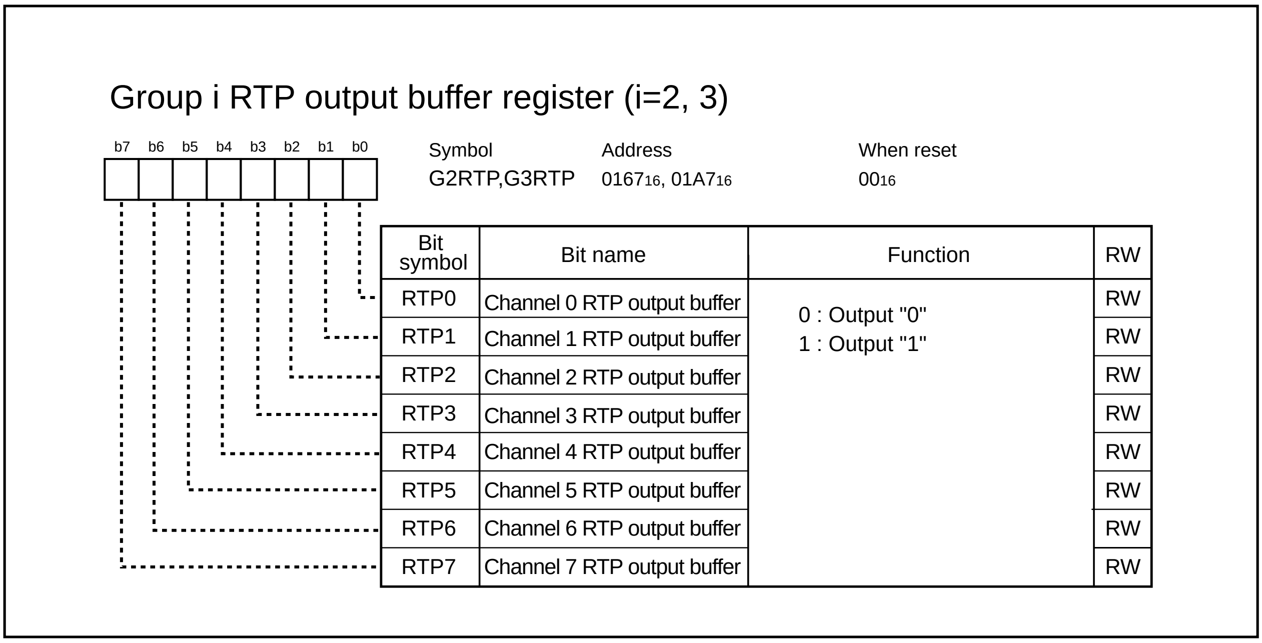
Group i function enable register (i=0 to 3)

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
G0FE to G3FE	$00E6_{16}$, 0126_{16}, 0166_{16}, $01A6_{16}$	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
IFE0	Channel 0 function enable bit	0 : Channel functions disabled 1 : Channel functions enabled	RW
IFE1	Channel 1 function enable bit		RW
IFE2	Channel 2 function enable bit		RW
IFE3	Channel 3 function enable bit		RW
IFE4	Channel 4 function enable bit		RW
IFE5	Channel 5 function enable bit		RW
IFE6	Channel 6 function enable bit		RW
IFE7	Channel 7 function enable bit		RW

Figure 1.22.14. G0FS and G1FS Registers and G0FE to G3FE Registers



Group i RTP output buffer register (i=2, 3)

Symbol	Address	When reset
G2RTP,G3RTP	0167$_{16}$, 01A7$_{16}$	00$_{16}$

Bit symbol	Bit name	Function	RW
RTP0	Channel 0 RTP output buffer	0 : Output "0" 1 : Output "1"	RW
RTP1	Channel 1 RTP output buffer		RW
RTP2	Channel 2 RTP output buffer		RW
RTP3	Channel 3 RTP output buffer		RW
RTP4	Channel 4 RTP output buffer		RW
RTP5	Channel 5 RTP output buffer		RW
RTP6	Channel 6 RTP output buffer		RW
RTP7	Channel 7 RTP output buffer		RW

Figure 1.22.15. G2RTP and G3RTP Registers

Base Timer (Group 0 to 3)

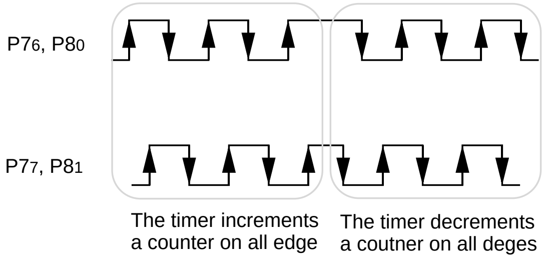
The base timer counts an internally generated count source with free-running.

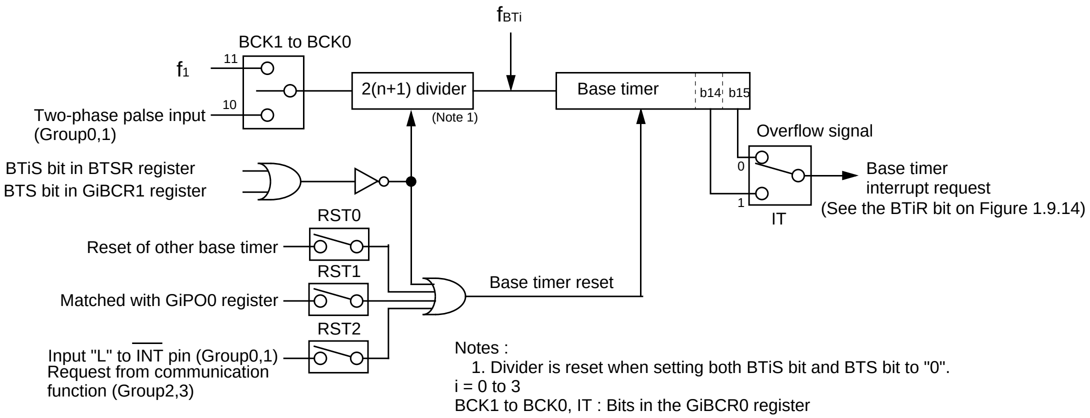
Table 1.22.2 lists specifications of the base timer. Figures 1.22.5 to 1.22.9 show registers associated with the base timer. Figure 1.22.16 shows a block diagram of the base timer. Figure 1.22.17 shows an example of the base timer (group 0, 1) in counter increment mode. Figure 1.22.18 shows an example 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 pules 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	• When the base timers individually start counting The BTS bit in the GiBCR1 register should be set to "1" (base timer starts counting) • When the base timers in multiple groups simultaneously started counting The BTiS bit in the BTSR register should be set to "1" (base timer starts counting)
Count stop condition	The BTiS bit in the BTSR 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 $\overline{INT0}$ pin Group 2: the $\overline{INT1}$ pin (4) Requested a reset from the communication function (group 2, 3)
Value for base timer reset	"0000_{16}"
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	• While the base timer is running, the GiBT register indicates a counter value • When the base timer is reset, a counter value is indeterminate
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	• Cascaded connection (group0, 1) The group 1 base timer increments its counter whenever the group 0 base timer overflows. (See Figure 1.22.19.) • Counter increment/decrement mode (group0, 1) After starting counting, the base timer increments the counter from "0000_{16}" until reaching "$FFFF_{16}$" to decrement the counter. The base timer increments the counter again when reaching the next "0000_{16}". (See Figure 1.22.18.)

Table 1.22.2. Base Timer Specifications (Continued)

Item	Specification
Selectable function	• Two-phase pulse processing mode (group 0, 1) In the group 0, two-phase pulses from $P7_6$ and $P7_7$ pins are counted In the group 1, two-phase pulses from $P8_0$ and $P8_1$ pins are counted as well (See Figure 1.22.20)  $P7_6$, $P8_0$ $P7_7$, $P8_1$ The timer increments a counter on all edge The timer decrements a coutner on all deges


f_{BTi}
BCK1 to BCK0
f_1
11
10
Two-phase palse input (Group0,1)
2(n+1) divider
(Note 1)
Base timer
b14
b15
Overflow signal
BTiS bit in BTSR register
BTS bit in GiBCR1 register
0
1
IT
Base timer interrupt request
(See the BTiR bit on Figure 1.9.14)
RST0
Reset of other base timer
RST1
Matched with GiPO0 register
Base timer reset
RST2
Input "L" to $\overline{INT}$ pin (Group0,1)
Request from communication function (Group2,3)

Notes :
1. Divider is reset when setting both BTiS bit and BTS bit to "0".
i = 0 to 3
BCK1 to BCK0, IT : Bits in the GiBCR0 register
RST2 to RST0 : Bits in the GiBCR1 register

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)

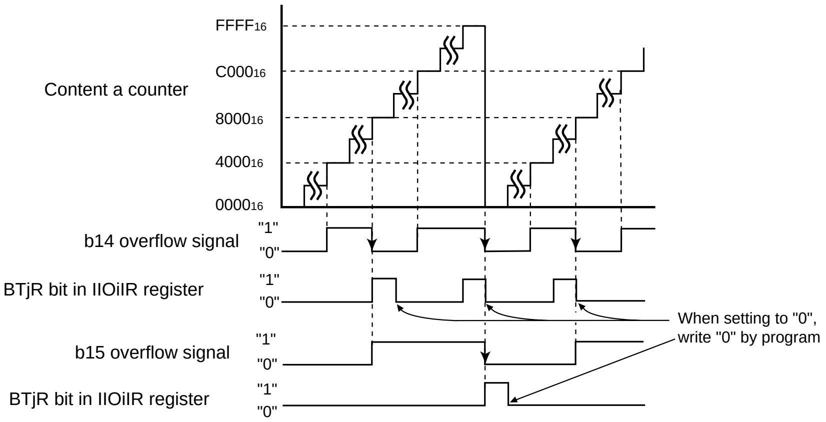
Register	Bit	Function
G2BCR0	-	Provides the clock to the BTSR register. Set to "0111 1111$_2$"
BTSR	-	Set to "0000 0000$_2$"
GiBCR0	BCK1 to BCK0	Select a count source
	DIV4 to DIV0	Select a divide ratio of a count source
	IT	Select the base timer interrupt
GiBCR1	RST2 to RST0	Select base timer reset 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 "000$_2$" (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.

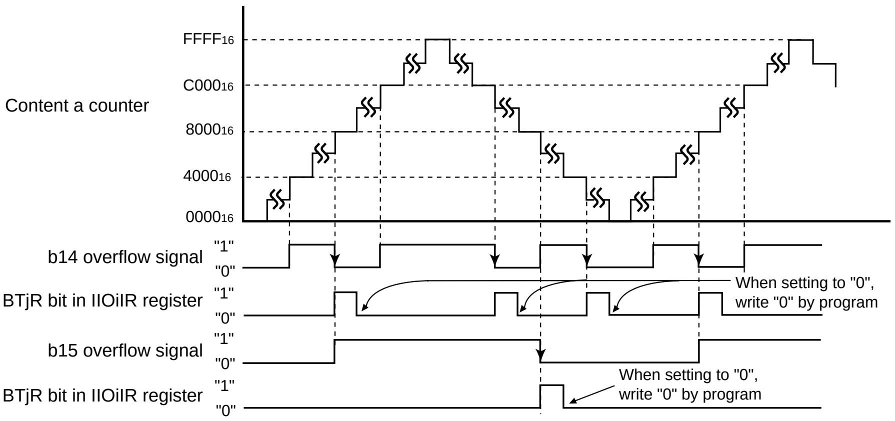


i=4,7,8,11 j=0 to 3 k=0,1

The above applies to the following conditions.

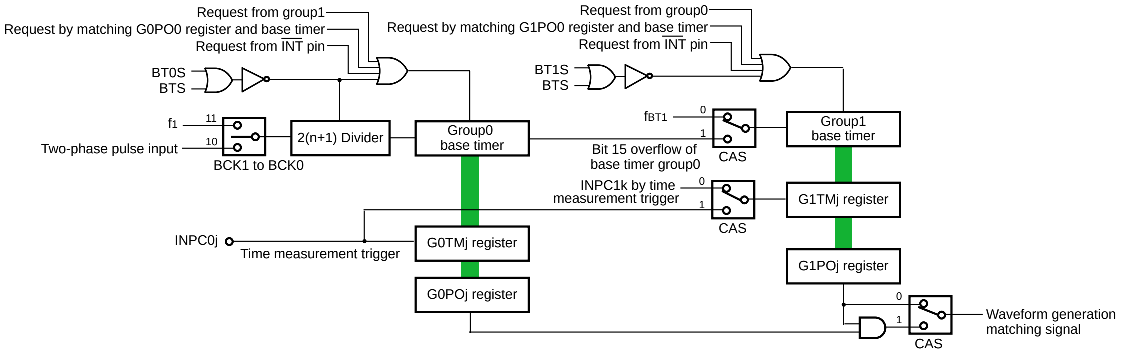
- The RST1 bit in the GkBCR1 register is set to "0" (the base timer is not reset by matching the GiPO0 register)
- The UD1 to UD0 bits in the GkBCR1 register are set to "00$_2$" (counter increment mode)

Figure 1.22.17. Counter Increment Mode (Group0, 1)



i=4,7,8,11 j=0 to 3

Figure 1.22.18. Counter Increment/Decrement Mode (Group 0,1)

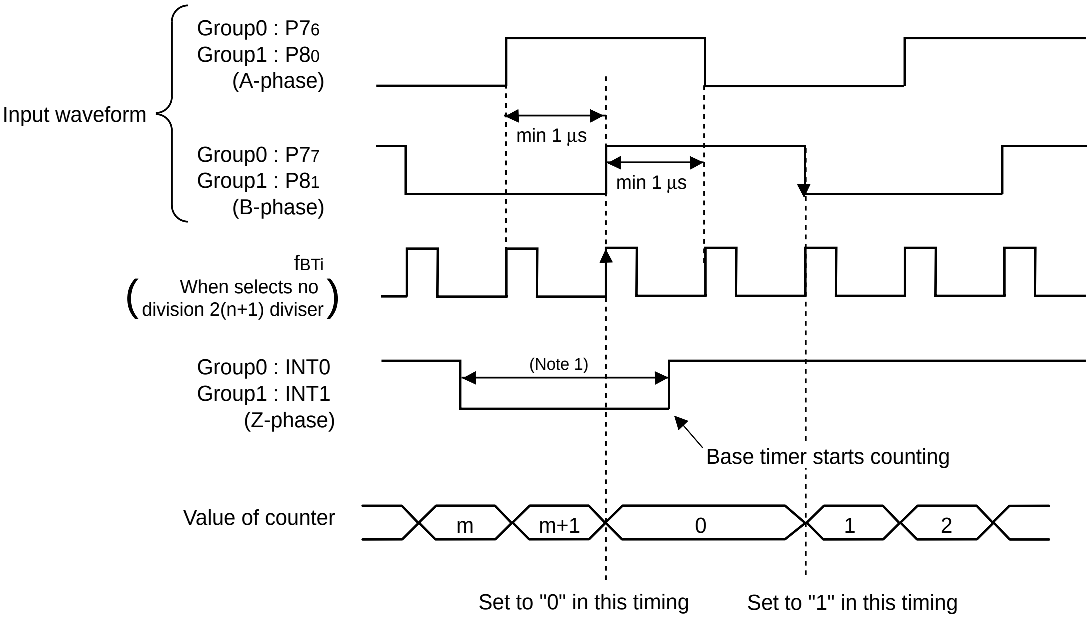


j=0 to 7, k=1,2,6,7

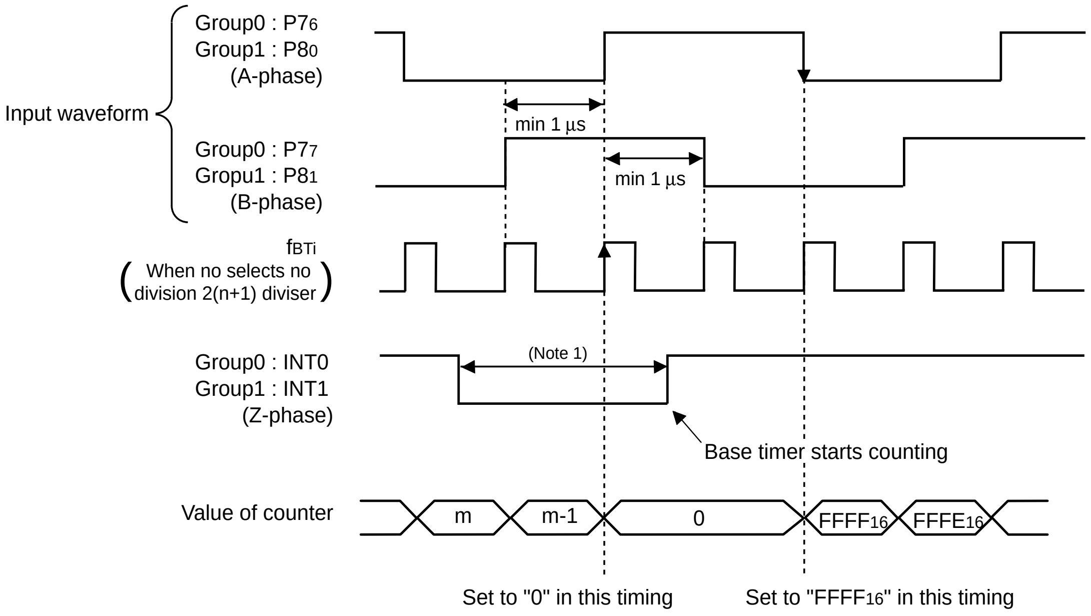
BT0S, BT1S : Bits in BTSR register
BTS, CAS : Bits in GiBCR1 register (i=0,1)
BCK1 to BCK0 : Bits in GiBCR0 register (i = 0,1)

Figure 1.22.19. Cascaded Connection

(1) When the base timer is reset while the base timer increments the counter



(2) When the base timer is reset while the base timer decrements the counter



i=0, 1

Notes :

1. More or equal to 1.5 fBTi clock cycles are required.

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[1]
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	•No prescaler : every a trigger is input •Prescaler (for channel 6 and channel 7) : every value of the GiTPRk register (k=6,7) +1 trigger input
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[1]	Trigger input pin
Selectable function	• Digital filter function The digital filter samples a trigger input level every f_1 or f_{BTi} to pass pulses matching a trigger input level three times • Cascaded connection function Group 0 and 1 are connected to operate as a 32-bit timer • 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 • 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

Notes :

1. The INPC0$_0$ to INPC0$_7$, INPC1$_1$ to INPC1$_2$ and INPC1$_6$ to INPC1$_7$ pins (INPC0$_0$ to INPC0$_7$ pins in cascaded connection)

Table 1.22.5. Pin settings for Time Measurement Function

Pin[2]	Bit and Setting		
	PS1, PS2, PS5, PS8, PS9 registers	PD7, PD8, PD11, PD14, PD15 registers	IPS register
P7$_4$/INPC1$_1$	PS1_4 = 0	PD7_4 = 0	IPS1 = 0
P7$_5$/INPC1$_2$	PS1_5 = 0	PD7_5 = 0	
P7$_6$/INPC0$_0$	PS1_6 = 0	PD7_6 = 0	IPS0 = 0
P7$_7$/INPC0$_1$	PS1_7 = 0	PD7_7 = 0	IPS0 = 0
P8$_0$/INPC0$_2$	PS2_0 = 0	PD8_0 = 0	IPS0 = 0
P11$_1$/INPC1$_1$[1]	PS5_1 = 0	PD11_1 = 0	IPS1 = 1
P11$_2$/INPC1$_1$[1]	PS5_2 = 0	PD11_2 = 0	
P14$_2$/INPC1$_6$[1]	PS8_2 = 0	PD14_2 = 0	-
P14$_3$/INPC1$_7$[1]	PS8_3 = 0	PD14_3 = 0	
P15$_0$/INPC0$_0$[1]	PS9_0 = 0	PD15_0 = 0	IPS0 = 1, IPS2 = 0
P15$_1$/INPC0$_1$[1]	PS9_1 = 0	PD15_1 = 0	
P15$_2$/INPC0$_2$[1]	-	PD15_2 = 0	
P15$_3$/INPC0$_3$[1]	-	PD15_3 = 0	IPS2 = 0
P15$_4$/INPC0$_4$[1]	PS9_4 = 0	PD15_4 = 0	
P15$_5$/INPC0$_5$[1]	PS9_5 = 0	PD15_5 = 0	
P15$_6$/INPC0$_6$[1]	-	PD15_6 = 0	
P15$_7$/INPC0$_7$[1]	-	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.

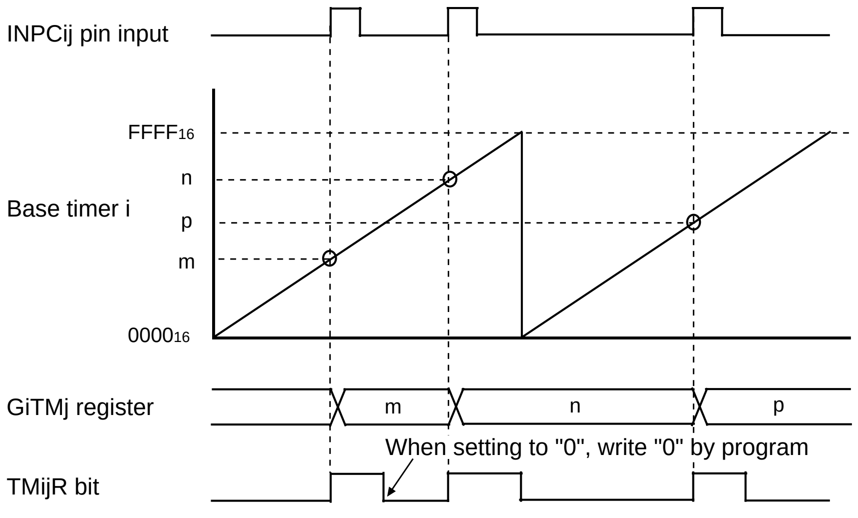
Table 1.22.6. Registers Setting Associated with the Time Measurement Function (group0, 1)

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.



i=0,1   j=0 to 7 (except j=1, 2, 6, 7 when i=1)
TMijR bit : Bits in the IIO0IIR to IIO8IR and IIO10IR to IIO11IR registers

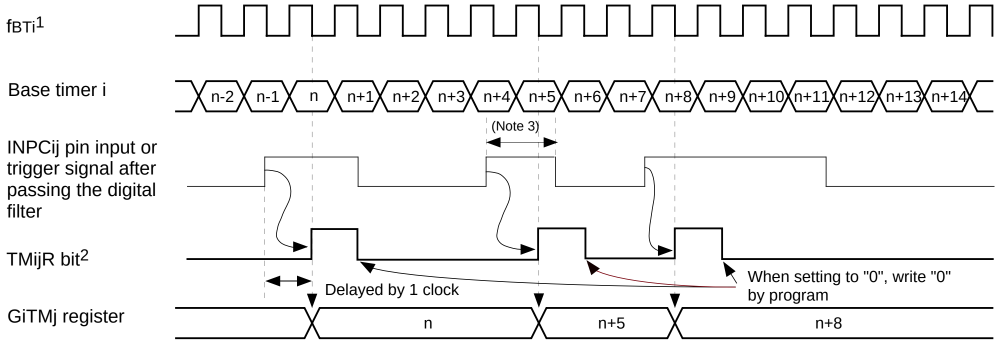
The above applies to the following condition.

- The CTS1 to CTS0 bits in the GiTMCRj registers are set to "01$_2$" (time measurement is trigger on the rising edge). The PR bit is set to "0" (no prescaler used) and the GT bit is set to"0" (no gate function used).
- All RTS2 to RTS0 bits in the GiBCR1 register are set to "0" (no base timer reset). The UD1 to UD0 bits are set to "00$_2$" (counter increment mode) and the CAS bit is set to "0" (16-bit time measurement).

When the base timer matches the GiPO0 register and is set to "0000$_{16}$" (setting the RST1 bit to "1" and the RST0 and RST2 bits to "0"), the base timer is set to "0000$_{16}$" after it reaches a setting value of the GiPO0 register+2.

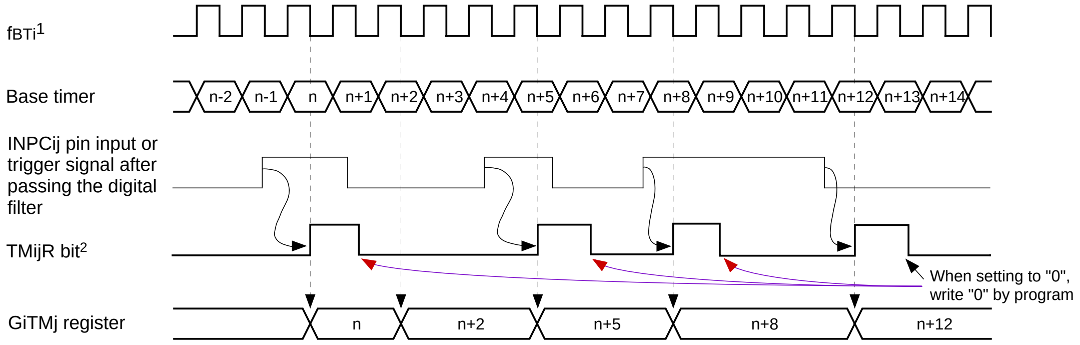
Figure 1.22.21. Time Measurement Function (1)

(a) When selecting the rising edge for timer measurement trigger
(The CTS1 to CTS0 bits in the GiTMCR register (i=0,1, j=0 to 7)=01_2)



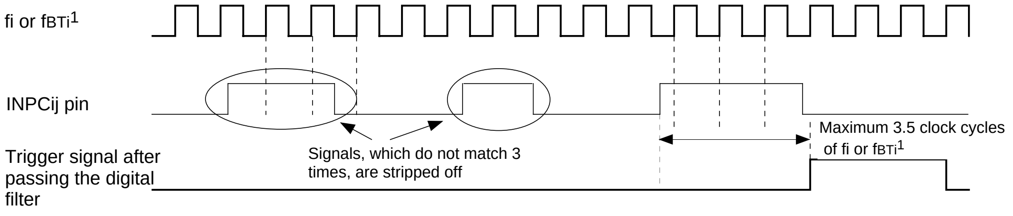
Notes :
1. If the CAS bit in the GiBCR1 register should be set to "1" (32-bit time measurement), the group 1 base timer increments its counter whenever the group 0 base timer overflows.
2. Bits in the IIO0IR to IIO8IR, IIO10IR to IIO11R registers. The TM0jR register when setting the CAS bit to "1".
3. For an input pulse to the INPCij pin, more or equal to 1.5 f_{BTi} clock period is required.

(b) When selecting both edges for timer measurement trigger
(The CTS1 to CTS0 bits=11_2)


fBTi1
Base timer
n-2
n-1
n
n+1
n+2
n+3
n+4
n+5
n+6
n+7
n+8
n+9
n+10
n+11
n+12
n+13
n+14
INPCij pin input or trigger signal after passing the digital filter
TMijR bit2
When setting to "0", write "0" by program
GiTMj register
n
n+2
n+5
n+8
n+12

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".
3. No interrupt is generated when the microcomputer receives input when the TMijR bit is in "H". Value of the base timer is stored into the GiTMj register.

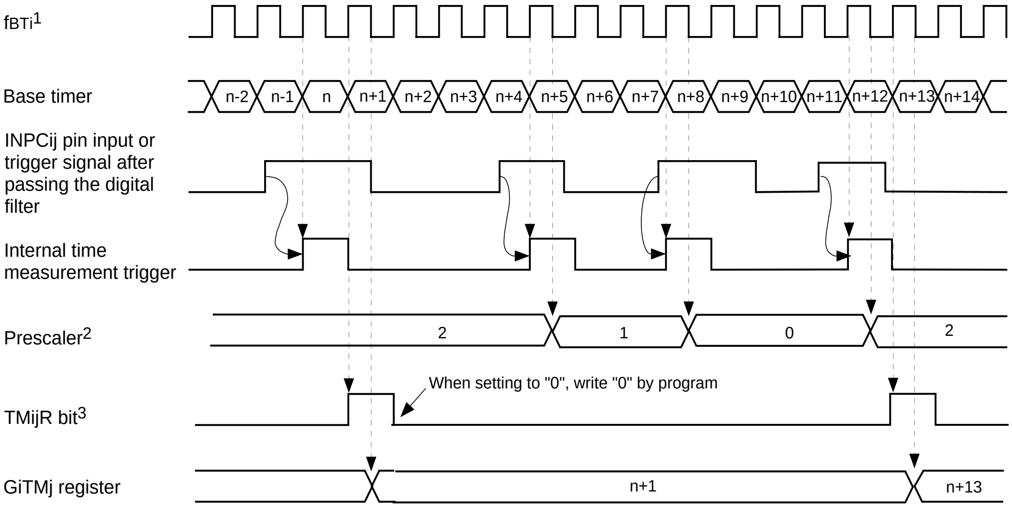
(c) Trigger signal when using digital filter
(The DF1 to DF0 bits in the GiTMCR register =01_2 or 11_2)


fi or fBTi1
INPCij pin
Trigger signal after passing the digital filter
Signals, which do not match 3 times, are stripped off
Maximum 3.5 clock cycles of fi or fBTi1

Notes :
1. f_{BTi} when the DF1 to DF0 bits are set to "10_2" and f_1 when to "11_2".

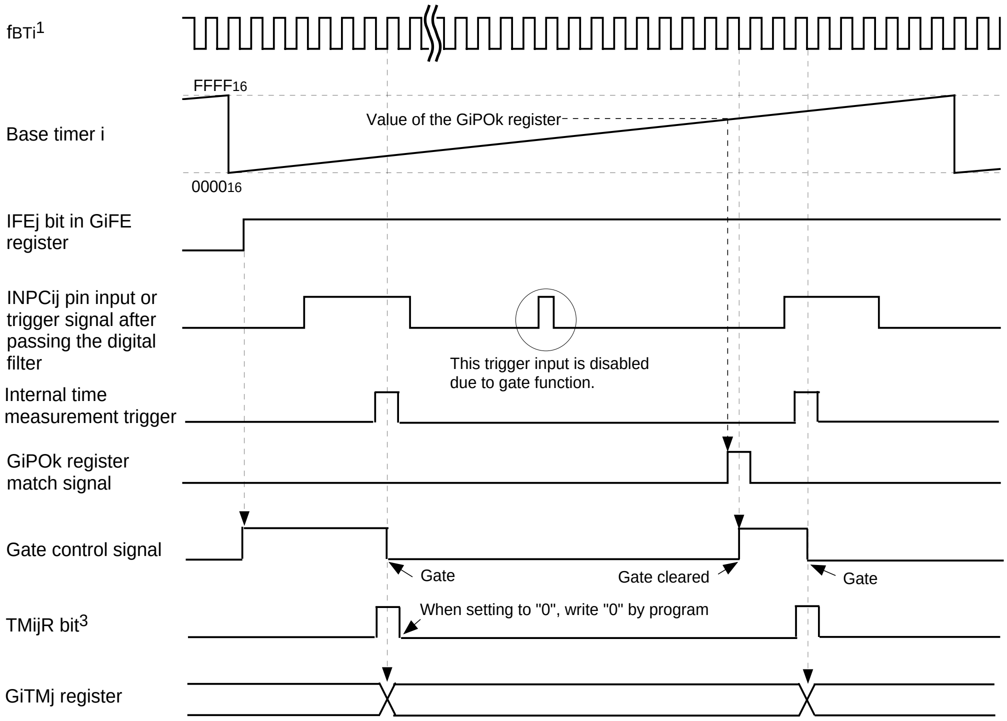
Figure 1.22.22. Time Measurement Function (2)

(a) When using the prescaler function
(When the GiTPRj register (i=0, 1, j=6, 7) $=02_{16}$, PR bit in the GiTMCR register=1)


fBTi1
Base timer
n-2
n-1
n
n+1
n+2
n+3
n+4
n+5
n+6
n+7
n+8
n+9
n+10
n+11
n+12
n+13
n+14
INPCij pin input or trigger signal after passing the digital filter
Internal time measurement trigger
Prescaler2
2
1
0
2
When setting to "0", write "0" by program
TMijR bit3
GiTMj register
n+1
n+13

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)


fBTi1
FFFF16
Base timer i
Value of the GiPOk register
000016
IFEj bit in GiFE register
INPCij pin input or trigger signal after passing the digital filter
This trigger input is disabled due to gate function.
Internal time measurement trigger
GiPOk register match signal
Gate control signal
Gate
Gate cleared
Gate
When setting to "0", write "0" by program
TMijR bit3
GiTMj register

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
$P6_4$/OUTC2_1	PS0_4 = 1	PSL0_4 = 1	-
$P7_0$/OUTC2_0	PS1_0 = 1	PSL1_0 = 0	PSC_0 = 1
$P7_1$/OUTC2_2	PS1_1 = 1	PSL1_1 = 0	PSC_1 = 1
$P7_3$/OUTC1_0[2]	PS1_3 = 1	PSL1_3 = 0	PSC_3 = 1
$P7_4$/OUTC1_1[2]	PS1_4 = 1	PSL1_4 = 0	PSC_4 = 1
$P7_5$/OUTC1_2[2]	PS1_5 = 1	PSL1_5 = 1	-
$P7_6$/OUTC0_0[2]	PS1_6 = 1	PSL1_6 = 0	PSC_6 = 1
$P7_7$/OUTC0_1[2]	PS1_7 = 1	-	-
$P8_1$/OUTC3_0	PS2_1 = 1	PSL2_1 = 1	-
$P8_2$/OUTC3_2	PS2_2 = 1	PSL2_2 = 0	-
$P9_2$/OUTC2_0	PS3_2 = 1	PSL3_2 = 1	-
$P11_0$/OUTC1_0[1,2]	PS5_0 = 1	-	-
$P11_1$/OUTC1_1[1,2]	PS5_1 = 1		
$P11_2$/OUTC1_2[1,2]	PS5_2 = 1		
$P11_3$/OUTC1_3[1,2]	PS5_3 = 1		
$P12_0$/OUTC3_0[1]	PS6_0 = 1	-	-
$P12_1$/OUTC3_1[1]	PS6_1 = 1		
$P12_2$/OUTC3_2[1]	PS6_2 = 1		
$P12_3$/OUTC3_3[1]	PS6_3 = 1		
$P12_4$/OUTC3_4[1]	PS6_4 = 1		
$P12_5$/OUTC3_5[1]	PS6_5 = 1		
$P12_6$/OUTC3_6[1]	PS6_6 = 1		
$P12_7$/OUTC3_7[1]	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
P13$_0$/OUTC2$_4$[1]	PS7_0 = 1	-	-
P13$_1$/OUTC2$_5$[1]	PS7_1 = 1		
P13$_2$/OUTC2$_6$[1]	PS7_2 = 1		
P13$_3$/OUTC2$_3$[1]	PS7_3 = 1		
P13$_4$/OUTC2$_0$[1]	PS7_4 = 1		
P13$_5$/OUTC2$_1$[1]	PS7_5 = 1		
P13$_6$/OUTC2$_2$[1]	PS7_6 = 1		
P13$_7$/OUTC2$_7$[1]	PS7_7 = 1		
P14$_0$/OUTC1$_4$[1,2]	PS8_0 = 1	-	-
P14$_1$/OUTC1$_5$[1,2]	PS8_1 = 1		
P14$_2$/OUTC1$_6$[1,2]	PS8_2 = 1		
P14$_3$/OUTC1$_7$[1,2]	PS8_3 = 1		
P15$_0$/OUTC0$_0$[1,2]	PS9_0 = 1	-	-
P15$_1$/OUTC0$_4$[1,2]	PS9_1 = 1		
P15$_4$/OUTC0$_4$[1,2]	PS9_4 = 1		
P15$_5$/OUTC0$_5$[1,2]	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[1]	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 "0000_{16}". Table 1.22.9 lists specifications of single-phase waveform mode. Figure 1.22.24 lists an example of single-phase waveform mode operation.

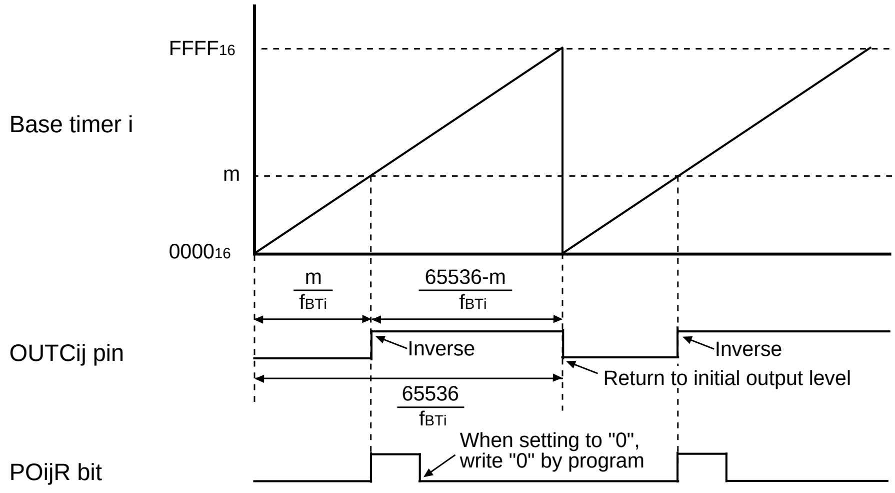
Table 1.22.9. Single-phase Waveform Output Mode Specifications

Item	Specification
Output waveform	• Free-running operation (the RST0 to RST2 bits in the GiBCR1 register (i=0 to 3) is set to "0" (no reset)) Cycle : $\frac{65536}{f_{BTi}}$ Default output level : $\frac{m}{f_{BTi}}$ Inverse level : $\frac{65536-m}{f_{BTi}}$ • 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 : $\frac{n+2}{f_{BTi}}$ Default output level : $\frac{m}{f_{BTi}}$ Inverse level : $\frac{n+2-m}{f_{BTi}}$ m : setting value of the GiPOj register (j=0 to 7), 0001_{16} to $FFFD_{16}$ n : setting value of the GiPO0 register, 0001_{16} to $FFFD_{16}$
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[1] pin	Pulse output
Selectable function	• Default value set function : Output level is set when waveform output starts • Inverse output function : Waveform level is inversed to output waveform from the OUTCij pin • Cascaded connection function : The groups 0 and 1 are connected to operate as a 32-bit timer

Notes :

1. The $OUTC0_0$, $OUTC0_1$, $OUTC0_4$, $OUTC0_5$, $OUTC1_0$ to $OUTC1_7$, $OUTC2_0$ to $OUTC2_7$ and $OUTC3_0$ to $OUTC3_7$ pins ($OUTC1_0$ to $OUTC1_7$ 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")

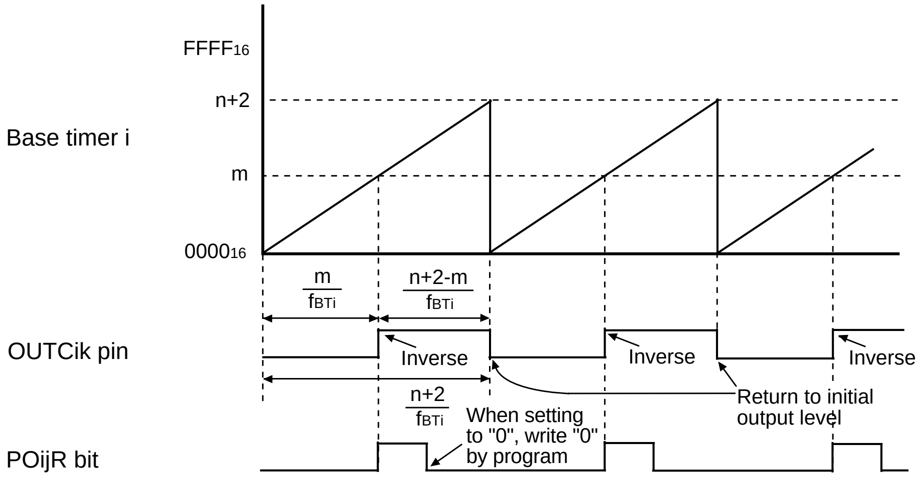


i=0 to 3, j=0 to 7 (j=0, 1, 4, 5 when i=0)
m : Setting value of the GiPOj register
POijR bit : Bits in the IIO0IR to IIO11IR registers

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 inversed).
- 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 "00_2" (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 "00_2" (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 inversed 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.

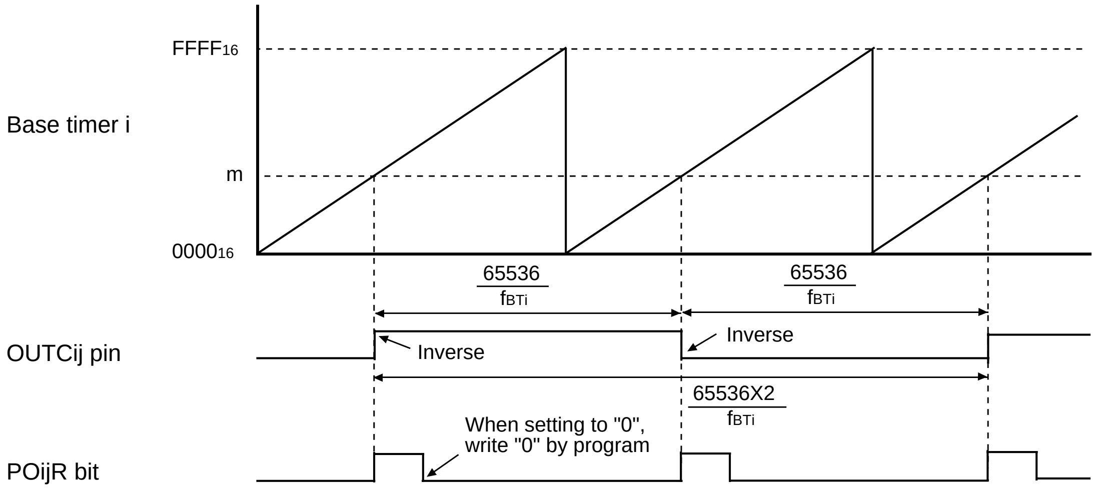
Table 1.22.10. Phase-delayed Waveform Output Mode Specifications

Item	Specification
Output waveform	• Free-running operation (all RST0 to RST2 bits in the GiBCR1 register (i=0 to 3) is set to "0" (no reset)) Cycle : $\frac{65536 \times 2}{f_{BTi}}$ "H" and "L" width : $\frac{65536}{f_{BTi}}$ • 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 : $\frac{2(n+2)}{f_{BTi}}$ "H" and "L" width : $\frac{n+2}{f_{BTi}}$ n : setting value of the GiPO0 register, 0001_{16} to $FFFD_{16}$
Waveform output start condition[1]	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[2] pin	Pulse output
Selectable function	• Default value set function : Output level is set when waveform output starts • Inverse output function : Waveform level is inversed to output waveform from the OUTCij pin • Cascaded connection function : The groups 0 and 1 are connected to operate as a 32-bit timer

Notes :

1. 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.
2. The $OUTC0_0$, $OUTC0_1$, $OUTC0_4$, $OUTC0_5$, $OUTC1_0$ to $OUTC1_7$, $OUTC2_0$ to $OUTC2_7$ and $OUTC3_0$ to $OUTC3_7$ pins ($OUTC1_0$ to $OUTC1_7$ 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")

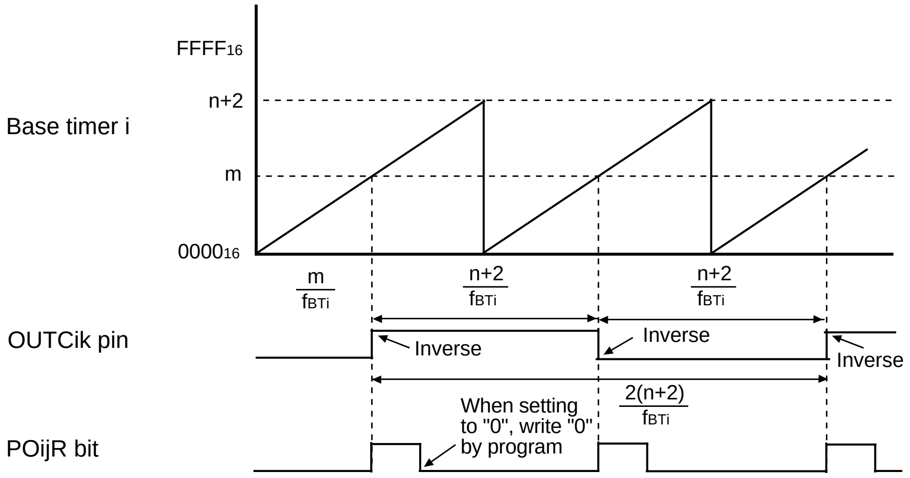


i=0 to 3, j=0 to 7 (j=0, 1, 4, 5 when i=0)
m : Setting value of the GiPOj register
POijR bit : Bits in the IIO0IR to IIO11IR registers

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 inversed).
- 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 "00_2" (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 inversed).
- The UD1 to UD0 bits are set to "00_2" (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.

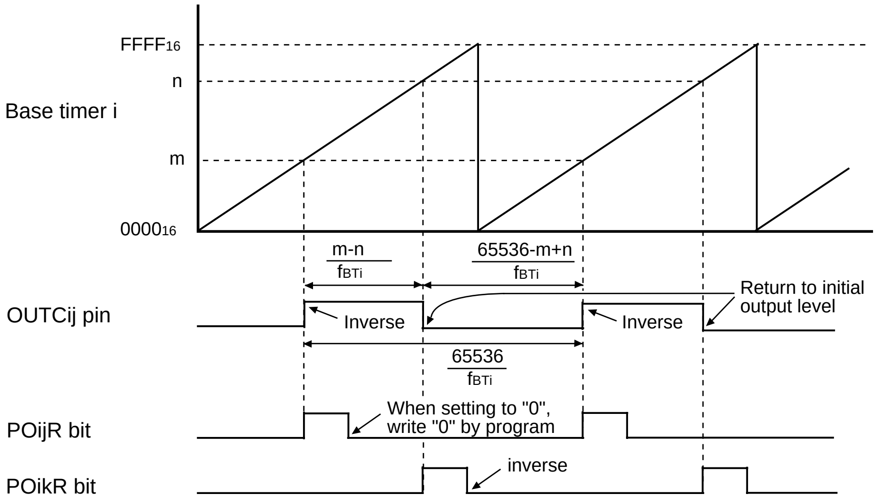
Table 1.22.11. SR Waveform Output Mode Specifications

Item	Specification
Output waveform	• Free-running operation (the RST0 to RST2 bits in the GiBCR1 register (i=0 to 3) is set to "0" (no reset)) Cycle : $\frac{65536}{f_{BTi}}$ Inverse level : $\frac{m-n}{f_{BTi}}$ • The base timer is reset when the base timer matches the GiPO0 register[1] (the RST1bit is set to "1" and both RST0 and RST2 bits is to "0") Cycle : $\frac{p+2}{f_{BTi}}$ Inverse level : $\frac{m-n}{f_{BTi}}$ 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_{16} to $FFFD_{16}$
Waveform output start condition[3]	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[4] pin	Pulse output
Selectable function	• Default value set function : Output level is set when waveform output starts • Inverse output function : Waveform level is inversed to output waveform from the OUTCij pin • Cascaded connection function : The groups 0 and 1 are connected to operate as a 32-bit timer

Notes :

1. When the GiPO0 register resets the base timer, the SR waveform generation function with channels 0 and 1 cannot be used.
2. The waveform generation register of odd channel should have greater value than the one of even channel has.
3. 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.
4. The $OUTC0_0$, $OUTC0_4$, $OUTC1_0$, $OUTC1_2$, $OUTC1_4$, $OUTC1_6$, $OUTC2_0$, $OUTC2_2$, $OUTC2_4$, $OUTC2_6$, $OUTC3_0$, $OUTC3_2$, $OUTC3_4$ and $OUTC3_6$ pins ($OUTC1_0$, $OUTC1_2$, $OUTC1_4$, $OUTC1_6$ 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")

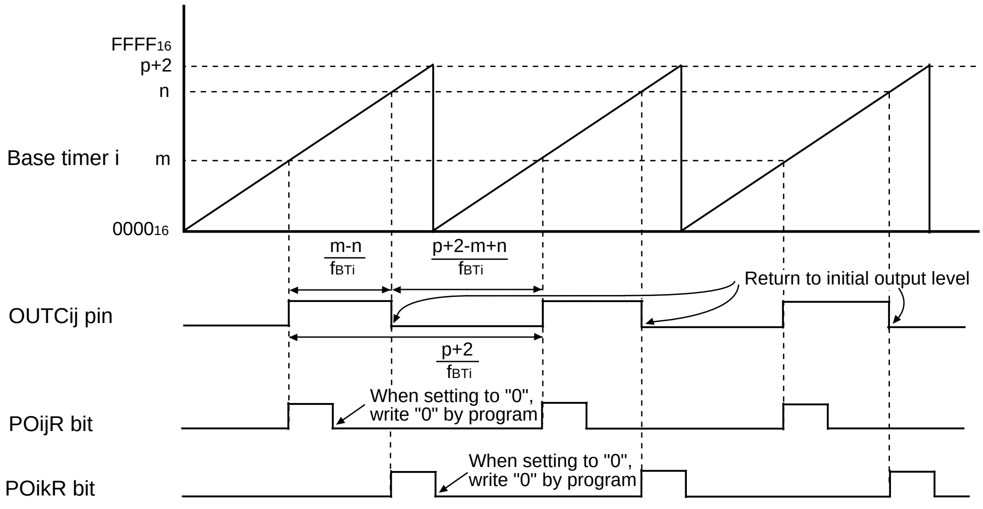


i=0 to 3, j=0, 2, 4, 6 (j=0, 4 when i=0) k=j+1
m : Setting value of the GiPOj register n: Setting value of the GiPOk register
POijR, POikR bits : Bits in the IIO0IR to IIO11IR registers

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 inversed).
- 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 "00_2" (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")


FFFF16
p+2
n
Base timer i
m
000016
m-n / fBTi
p+2-m+n / fBTi
Return to initial output level
OUTCij pin
p+2 / fBTi
When setting to "0", write "0" by program
POijR bit
When setting to "0", write "0" by program
POikR bit

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 inverse).
- The UD1 to UD0 bits are set to "00_2" (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)

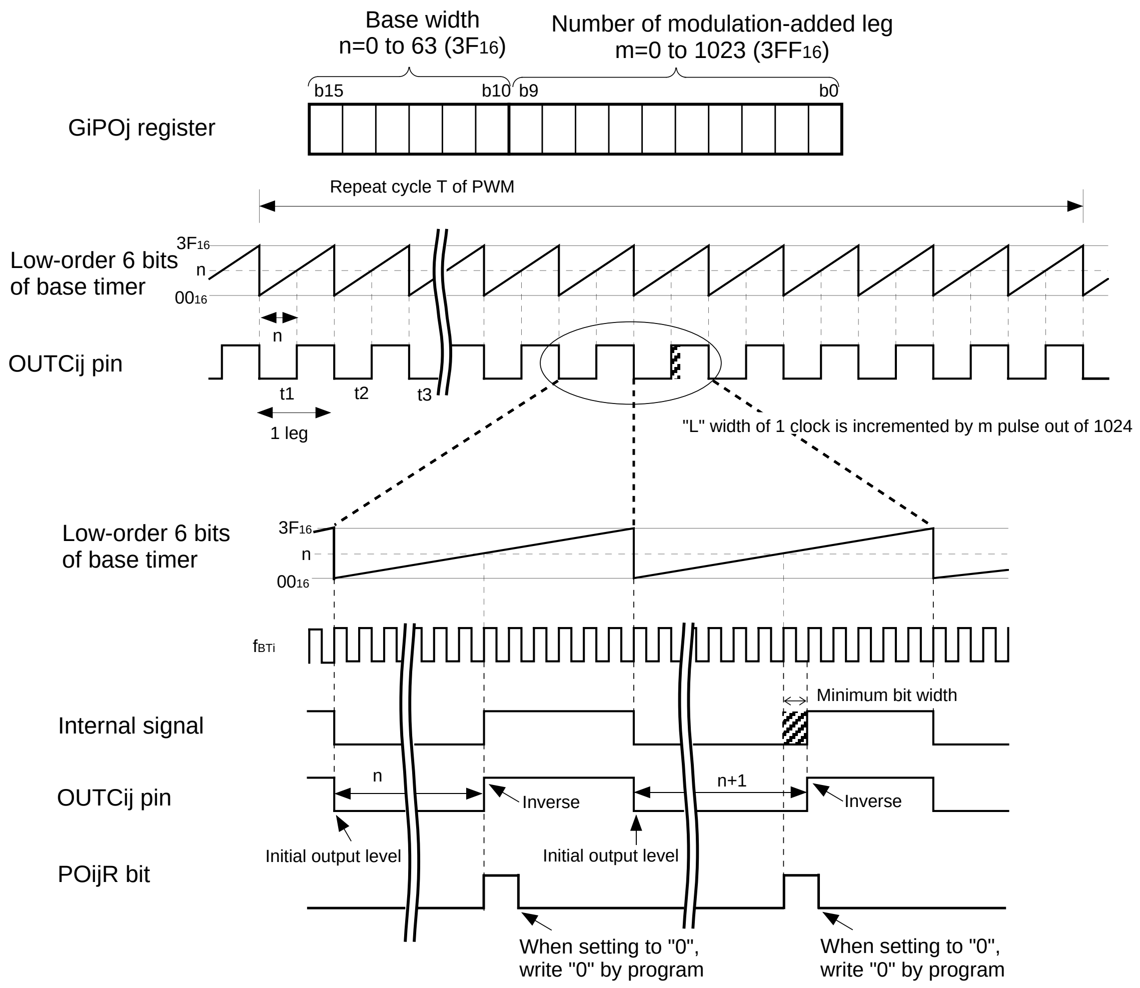
Output waveform "L" and "H" width can be changed in bit PWM output mode, "L" width become longer and "H" width is shorter by one cycle of f_{BTi} every 1024/m (m : low-order 10 bits in the GiPOj register).

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_{16} to $FFFD_{16}$
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	• Default value set function : Output level is set when waveform output starts • Inverse output function : Waveform level is inversed to output waveform from the OUTCij pin

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_2	none
00 0000 0001_2	t512
00 0000 0010_2	t256, t768
00 0000 0100_2	t128, t384, t640, t896
00 0000 1000_2	t64, t192, t320, t448, t576, t704, t832, t960
⋮	⋮
10 0000 0000_2	t1, t3, t5, t7, ••• t1019, t1021, t1023


(1) Free-running operation
(All RST2 to RST0 bits in the GiBCR1 register are set to "0")

The above applies to the following conditions.
- Both IVL and INV bits are set to "0".
- The RST2 to RST0 bits are set to "000_2".

i=2 to 3, j=0 to 7
POijR bit : Bits in the IIO3IR to IIO11IR registers

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_{16} to $FFFD_{16}$). (See Figure 1.9.14.)
OUTCij pin	RTP output
Selectable function	• Default value set function : Output level is set when waveform output starts. • Inverse output function : Waveform level is inversed to output waveform from the OUTCij pin

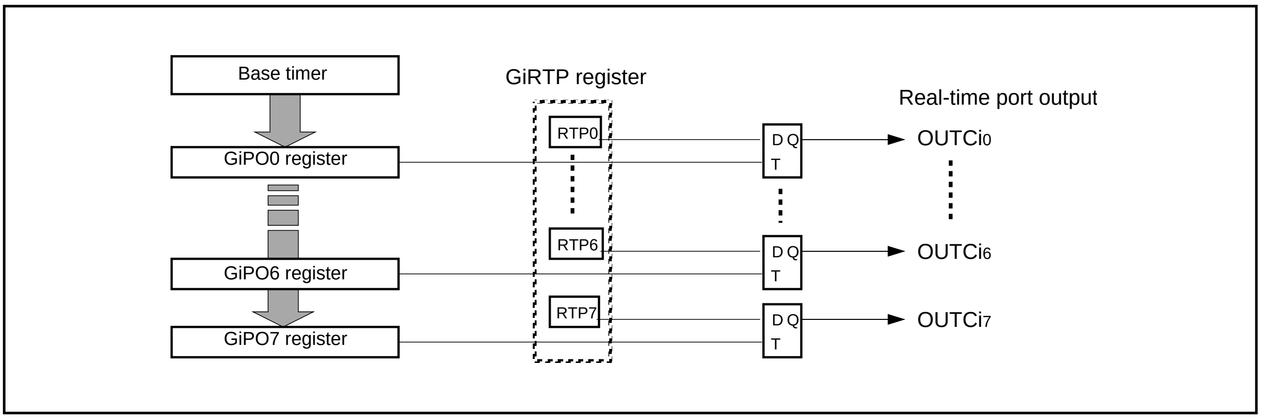
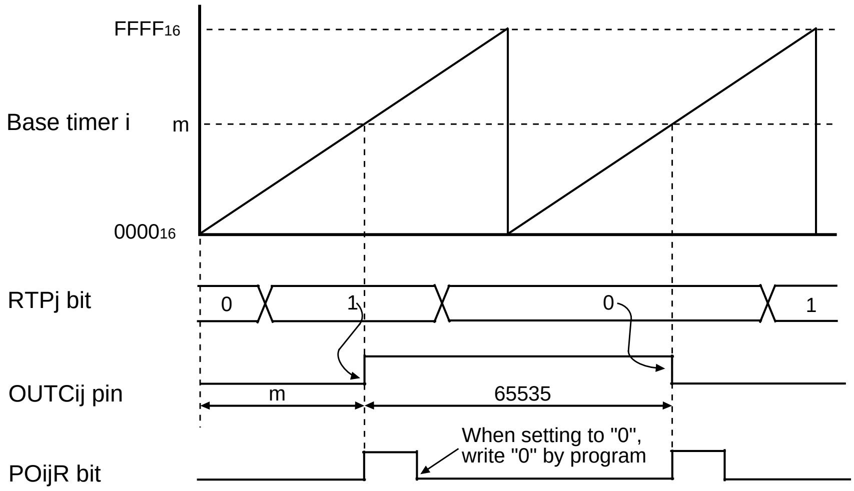


Figure 1.22.28. Real-time Port Output Function Block Diagram

(1) Free-running operation
(All RST2 to RST0 bits in the GiBCR1 register are set to "0")

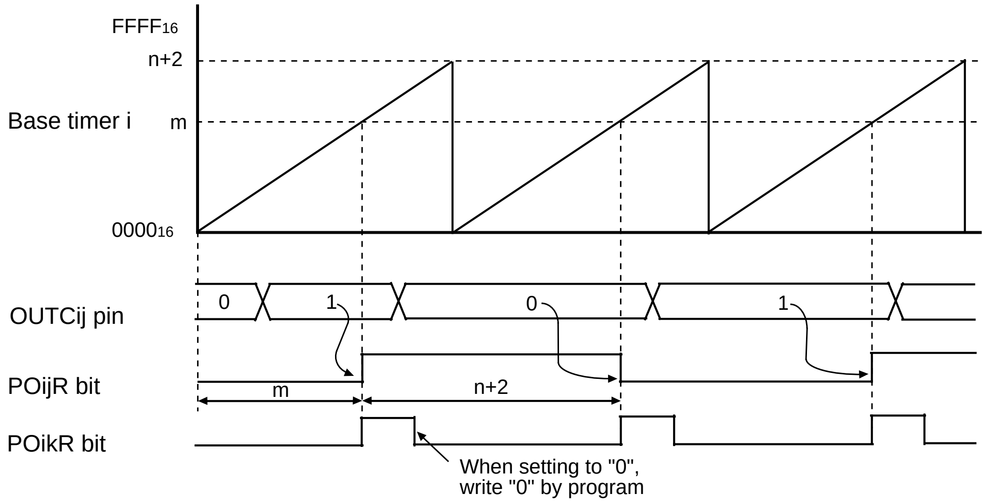


i=2 to 3, j=0 to 7
m : Setting value of the GiPOj register
POijR bit : Bits in the IIO3IR to IIO11IR registers

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_{16} to $FFFD_{16}$). (See Figure 1.9.14.)
OUTCij pin	RTP output
Selectable function	• Default value set function: Output level is set when waveform output starts. • Inverse output function: Waveform level is inversed to output waveform from the OUTCij pin

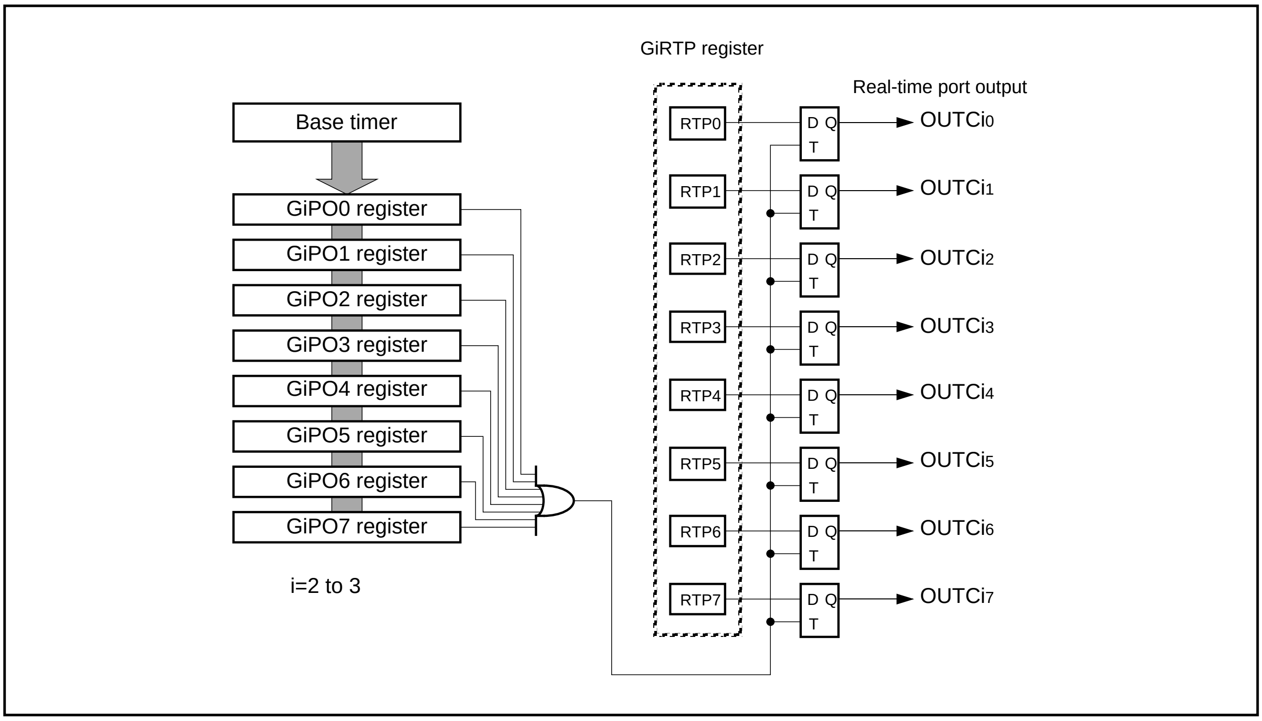
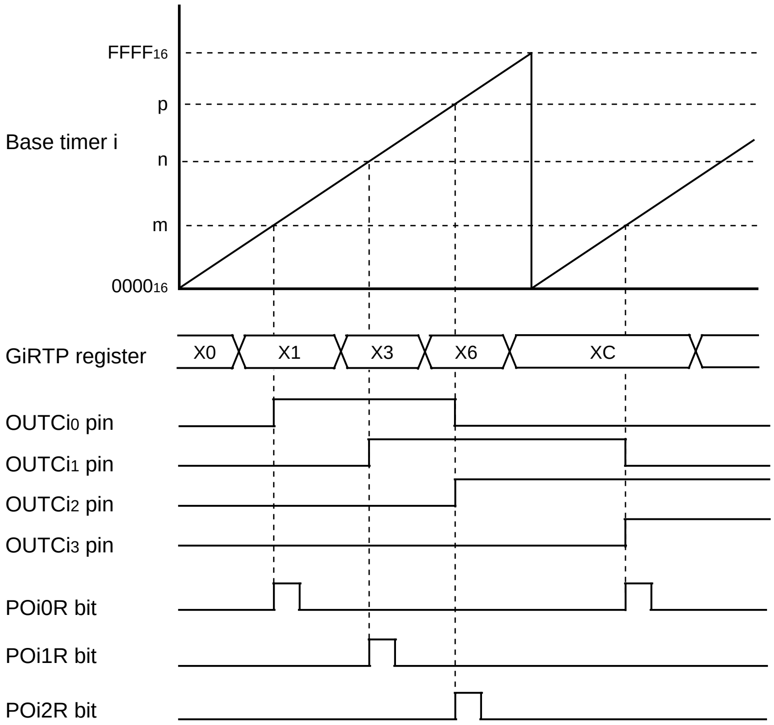


Figure 1.22.30. Parallel RTP Output Function Block Diagram

(1) Free-running operation



m : Setting value of the GiPO0 register i=2,3
n : Setting value of the GiPO1 register
p : Setting value of the GiPO2 register
POi0R, POi1R, POi2R bit : Bits in the IIO3IR to IIO11IR registers

The above applies to the following conditions.

- The IVL in the of 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).

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.

Group i receive input register (i=0,1)

b7 b0

Symbol	Address	When reset
G0RI,G1RI	00EC$_{16}$, 012C$_{16}$	Indeterminate

Function	Setting range	RW
Should set data that is input to a received data generation circuit	00$_{16}$ to FF$_{16}$	WO

Group i transmit output register (i=0,1)

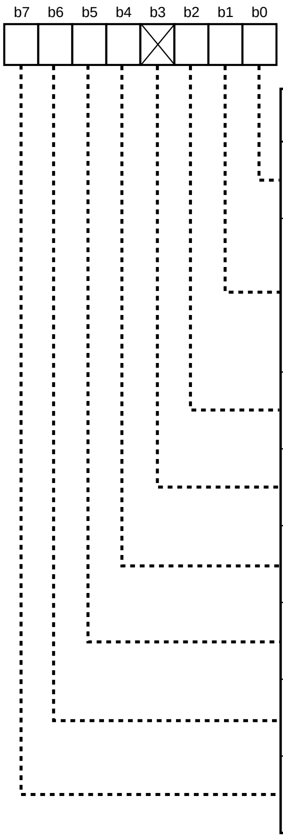
b7 b0

Symbol	Address	When reset
G0TO, G1TO	00EE$_{16}$, 012E$_{16}$	Indeterminate

Function	RW
Can read data that a transmit data generation circuit outputs	RO

Figure 1.22.32. G0RI to G1RI Registers and G0TO to G1TO Registers

Group i SI/O communication control register (i=0,1)



Symbol	Address	When reset
G0CR, G1CR	$00EF_{16}$, $012F_{16}$	$0000\ X000_2$

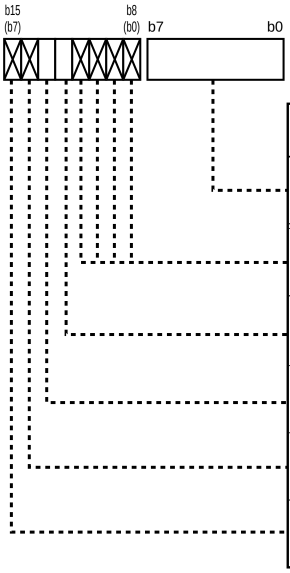
Bit symbol	Bit name	Function	RW
TI	Transmit buffer empty flag	0 : Data available in the GiTB register 1 : No data available in the GiTB register	RO
TXEPT	Transmit register empty flag	0 : Data available in the transmit register (during transmitting) 1 : No data available in the transmit register (transmit completed)	RO
RI	Receive complete flag	0 : No data available in the GiRB register 1 : Data available in the GiRB register	RO
——	Nothing is assigned. When write, should set to "0". When read, its contents is indeterminate.		—
TE	Transmit enable bit	0 : Transmit disable 1 : Transmit enable	RW
RE	Receive enable bit	0 : Receive disable 1 : Receive enable	RW
IPOL	ISRxD input polarity switch bit	0 : No inverse 1 : Inverse[1]	RW
OPOL	ISTxD output polarity switch bit	0 : No inverse 1 : Inverse[1]	RW

Notes :

1. This bit should be set to "1" in UART mode.

Figure 1.22.33. G0CR to G1CR Registers

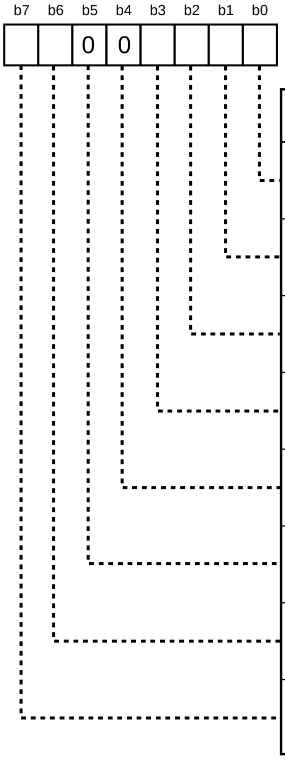
Group i SI/O receive buffer register (i=0,1)



Symbol	Address	When reset
G0RB, G1RB	00E9$_{16}$-00E8$_{16}$, 0129$_{16}$-0128$_{16}$	indeterminate

Bit symbol	Bit name	Function	RW
——	———	Received data	RW
——	Nothing is assigned. When read, its content is indeterminate.		–
OER	Overrun error flag	0 : No overrun error 1 : Overrun error found	RO
FER	Framing error flag	0 : No Framing error 1 : Framing error found	RO
——	Nothing is assigned. When read, its contents is indeterminate.		–
——	Nothing is assigned. When read, its contents is indeterminate.		–

Group i SI/O communication mode register (i=0,1)



Symbol	Address	When reset
G0MR,G1MR	00ED$_{16}$, 012D$_{16}$	0000 0000$_2$

Bit symbol	Bit name	Function	RW
GMD0	Communication mode select bit	b1 b0 0 0 : UART mode 0 1 : Clock synchronous serial I/O mode 1 0 : Special communication mode[1] 1 1 : HDLC data processing mode	RW
GMD1			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
——	Reserved bit	Should set to "0".	RW
——	Reserved bit	Should set to "0".	RW
UFORM	Transfer direction select bit	0 : LSB first 1 : MSB first	RW
IRS	Transmit interrupt cause select bit	0 : Transmit buffer is empty (TI=1) 1 : Transmit is completed (TXEPT=1)	RW

Notes :

1. Avoid this setting except for a car or motor vehicle.

Figure 1.22.34. G0RB to G1RB Registers and G0MR to G1MR Registers

Group i SI/O expansion mode register (i=0,1)[1]

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
G0EMR,G1EMR	$00FC_{16}$, $013C_{16}$	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
SMODE	Synchronous mode select bit	0 : Re-synchronous mode unused 1 : Re-synchronous mode	RW
CRCV	CRC initial value select bit	0 : "0000_{16}" is set 1 : "$FFFF_{16}$" is set	RW
ACRC	CRC initialization select bit	0 : Not reset 1 : reset[2]	RW
BSINT	Bit stuffing error interrupt select bit	0 : Not used 1 : Used	RW
RXSL	Receive source switch bit	0 : RxD pin 1 : GiRI register	RW
TXSL	Transmit source switch bit	0 : TxD pin 1 : GiTO register	RW
CRC0	CRC generation polynomial select bit	b7 b6 0 0 : X^8+X^4+X+1 0 1 : Avoid this setting 1 0 : $X^{16}+X^{15}+X^2+1$ 1 1 : $X^{16}+X^{12}+X^5+1$	RW
CRC1			RW

Notes :

1. This register is used in special communication mode or HDLC data processing mode. In clock synchronous serial I/O mode or UART mode, this register remains in a reset state or is set to "00_{16}".
2. Reset when matching a signal with the GiCMP3 register.

Group i SI/O expansion transmit control register (i=0,1)[1]

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
G0ETC,G1ETC	$00FF_{16}$, $013F_{16}$	$0000\ 0XXX_2$

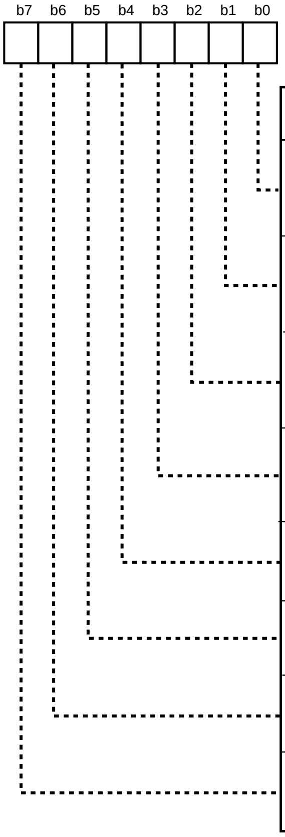
Bit symbol	Bit name	Function	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–
——			–
SOF	SOF transmit request bit	0 : No SOF transmit request 1 : SOF transmit request	RW
TCRCE	Transmit CRC enable bit	0 : Not used 1 : Used	RW
ABTE	Arbitration enable bit	0 : Not used 1 : Used	RW
TBSF0	Transmit bit stuffing "1" insert select bit	0 : "1" is not inserted 1 : "1" is inserted	RW
TBSF1	Transmit bit stuffing "0" insert select bit	0 : "0" is not inserted 1 : "0" is inserted	RW

Notes :

1. This register is used in special communication mode or HDLC data processing mode. In clock synchronous serial I/O mode or UART mode, this register remains in a reset state or is set to "00_{16}".

Figure 1.22.35. G0EMR to G1EMR Registers and G0ETC to G1ETC Registers

Group i SI/O expansion receive control register (i=0,1)[1]



Symbol	Address	When reset
G0ERC,G1ERC	$00FD_{16}$, $013D_{16}$	$0000\ 0000_2$

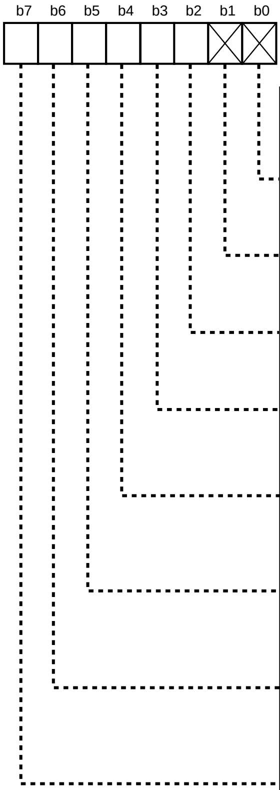
Bit symbol	Bit name	Function	RW
CMP0E	Data compare function 0 select bit	0 : The GiTB register is not compared (transmit data register) with the GiCMP0 register 1 : The GiTB register is compared (transmit data register) with the GiCMP0 register	RW
CMP1E	Data compare function 1 select bit	0 : The GiTB register is not compared (transmit data register) with the GiCMP1 register 1 : The GiTB register is compared (transmit data register) with the GiCMP1 register	RW
CMP2E	Data compare function 2 select bit	0 : The GiTB register is not compared (transmit data register) with the GiCMP2 register 1 : The GiTB register is compared (transmit data register) with the GiCMP2 register	RW
CMP3E	Data compare function 3 select bit	0 : The GiTB register is not compared (transmit data register) with the GiCMP3 register 1 : The GiTB register is compared (transmit data register) with the GiCMP3 register	RW
RCRCE	Receive CRC enable bit	0 : Not used 1 : Used	RW
RSHTE	Receive shift operation enable bit	0 : Receive shift operation disabled 1 : Receive shift operation enabled	RW
RBSF0	Receive bit stuffing "1" delete select bit	0 : "1" is not deleted 1 : "1" is deleted	RW
RBSF1	Receive bit stuffing "0" delete select bit	0 : "0" is not deleted 1 : "0" is deleted	RW

Notes :

1. This register is used in special communication mode or HDLC data processing mode. In clock synchronous serial I/O mode or UART mode, this register remains in a reset state or is set to "00_{16}".
2. When the ACRC bit in the GiEMR register is set to "1" (CRC reset function used), the CMP3E bit should be set to "1".

Figure 1.22.36. G0ERC to G1ERC Registers

Group i SI/O special communication interrupt detect register (i=0,1)[1]



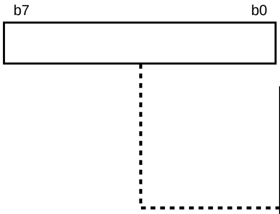
Symbol	Address	When reset
G0IRF,G1IRF	$00FE_{16}$, $013E_{16}$	0000 $00XX_2$

Bit symbol	Bit name	Function	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–
BSERR	Bit stuffing error detect flag	0 : Not detected 1 : Detected	RW
ABT	Arbitration lost detect flag	0 : Not detected 1 : Detected	RW
IRF0	Interrupt cause determination flag 0[2]	0 : The GiTB register (receive data register) does not match the GiCMP0 register 1 : The GiTB register (receive data register) matches the GiCMP0 register	RW
IRF1	Interrupt cause determination flag 1[2]	0 : The GiTB register (receive data register) does not match the GiCMP1 register 1 : The GiTB register (receive data register) matches the GiCMP1 register	RW
IRF2	Interrupt cause determination flag 2[2]	0 : The GiTB register (receive data register) does not match the GiCMP2 register 1 : The GiTB register (receive data register) matches the GiCMP2 register	RW
IRF3	Interrupt cause determination flag 3[2]	0 : The GiTB register (receive data register) does not match the GiCMP3 register 1 : The GiTB register (receive data register) matches the GiCMP3 register	RW

Notes :

1. This register is used in special communication mode or HDLC data processing mode. In clock synchronous serial I/O mode or UART mode, this register remains in a reset state or is set to "00_{16}".
2. The SRTiR bit in the IIO4IR register.

Group i transmit buffer (receive data) register (i=0,1)



Symbol	Address	When reset
G0TB,G0DR	$00EA_{16}$	Indeterminate
G1TB,G1DR	$012A_{16}$	Indeterminate

Function	RW
Should set data to be transmitted. In HDLC data processing mode, when reading the GiTB register, the receive data register is read. When writing to the GiTB register, the transmit buffer register is written. In HDLC data processing mode, value set to the GiRI register is transferred to the GiDR register.	RW

Figure 1.22.37. G0IRF to G1IRF registers and G0TB to G1TB Registers

Group i data compare register j (i=0,1/j=0 to 3)

Symbol	Address	When reset
G0CMP0 to G0CMP3	$00F0_{16}$, $00F1_{16}$, $00F2_{16}$, $00F3_{16}$	Indeterminate
G1CMP0 to G1CMP3	0130_{16}, 0131_{16}, 0132_{16}, 0133_{16}	Indeterminate

Function	Setting range	RW
Data to be compared	00_{16} to FF_{16}	RW

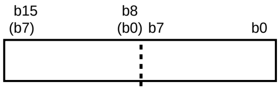
Notes :
1. When using the GiCMP0 register, the GiMSK0 register should be set. When using the GiCMP1 register, the GiMSK1 register should be set.

Group i data mask register j (i=0,1/j=0,1)

Symbol	Address	When reset
G0MSK0,G0MSK1	$00F4_{16}$, $00F5_{16}$	Indeterminate
G1MSK0,G1MSK1	0134_{16}, 0135_{16}	Indeterminate

Function	Setting range	RW
Masked data for received data Should write "0" to a bit uncompared.	00_{16} to FF_{16}	RW

Group i transmit CRC code register (i=0,1)

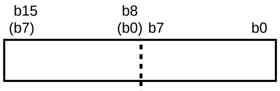


Symbol	Address	When reset
G0TCRC, G1TCRC	$00FB_{16}$-$00FA_{16}$, $013B_{16}$-$013A_{16}$	0000_{16}

Function	RW
Result of transmit CRC calculation[1]	RO

Notes :
1. Calculated result is initialized when the TCRCE bit in the GiETC register is set to "0" (unused). The CRCV bit in the GiEMR register selects an initial value.

Group i receive CRC code register (i=0,1)



Symbol	Address	When reset
G0RCRC, G1RCRC	$00F9_{16}$-$00F8_{16}$, 0139_{16}-0138_{16}	0000_{16}

Function	RW
Result of receive CRC calculation[1,2,3]	RO

Notes :
1. Calculated result is initialized when the TCRCE bit in the GiERC register is set to "0" (unused). It is also initialized when setting the ACRC bit to "1" (reset) and matching with the GiCMPj (j=0 to 3) register.
2. It is reset to an initial value set by the CRCV bit in the GiEMR register before a transmission starts.
3. When the RBSF bit in the GiERC register is set to "1" (0 deleted), preliminary calculation is required since a compared data is also moved to a 6-bit CRC calculation circuit.

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[1,2]	• When the CKDIR bit in the GiMR register (i=0, 1) is set to "0" (internal clock) : $\frac{f_{BTi}}{2(n+2)}$ n : setting value of the GiPO0 register, 0000_{16} to $FFFF_{16}$ – 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 • When the CKDIR bit is set to "1" (external clock) : input from the ISCLK1 pin
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. • The TE bit in the GiCR register is set to "1" (transmit enable) • The TI bit in the GiCR register is set to "0" (data in the GiTB register)
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. • The RE bit in the GiCR register is set to "1" (receive enable) • The TE bit is set to "1" (transmit enable) • The TI bit is set to "0" (data in the GiTB register)
Interrupt request	• While transmitting, the following condition can be selected to set the SIOiTR bit to "1" (see Figure 1.9.14) : – 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 – When the IRS bit is set to "1" (interrupt at reception completed) and data transfer from the transmit register is completed • 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)
Error detection	Overrun error[3] This error occurs when receiving the 8th bit of the next data before reading the GiRB register
Selectable function	• LSB first/MSB first Whether data is transmitted/received in bit 0 or in bit 7 can be selected • ISTxDi and ISRxDi I/O polarity inverse ISTxDi pin output and ISRxDi pin input levels are inversed

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 inversed), 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 "11_2"
	DIV4 to DIV0	Select a divide ratio of a count source
	IT	Set to "0"
GiBCR1	7 to 0	Set to "0001 0010_2"
GiPOCR0	7 to 0	Set to "0000 0111_2"
GiPOCR1	7 to 0	Set to "0000 0111_2"
GiPOCR3	7 to 0	Set to "0000 0010_2" [1]
GiPO0	15 to 0	Set a baud rate $\frac{f_{BTi}}{2 \text{ x } (\text{setting value} + 2)}$ = transfer clock frequency[1]
GiPO3	15 to 0	Set a smaller value than the GiPO0 register[1]
GiFS	FSC3,1,0	Set to "0"
GiFE	IFE3,1,0	Set to "1"
GiERC	7 to 0	Set to "0010 0000_2"
GiMR	GMD1 to GMD0	Set to "01_2"
	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[1]
		PS1 register	PSL1 register	PSC register	PD7 register	IPS register	
$P7_3$	ISTxD1 output	PS1_3 = 1	PSL1_3 = 0	PSC_3 = 1	-	-	G1POCR0
$P7_4$	ISCLK1 input	PS1_4 = 0	-	-	PD7_4 = 0	IPS1 = 0	-
	ISCLK1 output	PS1_4 = 1	PSL1_4 = 0	PSC_4 = 1	-	-	G1POCR1
$P7_5$	ISRxD1 input	PS1_5 = 0	-	-	PD7_5 = 0	IPS1 = 0	-
$P7_6$	ISTxD0 output	PS1_6 = 1	PSL1_6 = 0	PSC_6 = 0	-	-	G0POCR0
$P7_7$	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_2" (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	
$P8_0$	ISRxD0 input	PS2_0 = 0	PD8_2 = 0	IPS0 = 0	-

Table 1. 22. 20. Pin Settings (Continued)

Port name	Function	Bit and setting			register[1]
		PS5 register	PD11 register	IPS register	
P11$_0$	ISTxD1 output	PS5_0 = 1	-	-	G1POCR0
P11$_1$	ISCLK1 input	PS5_1 = 0	PD11_1 = 0	IPS1 = 1	-
	ISCLK1 output	PS5_1 = 1	-	-	G1POCR1
P11$_2$	ISRxD1 input	PS5_2 = 0	PD11_2 = 0	IPS1 = 1	-

Notes :

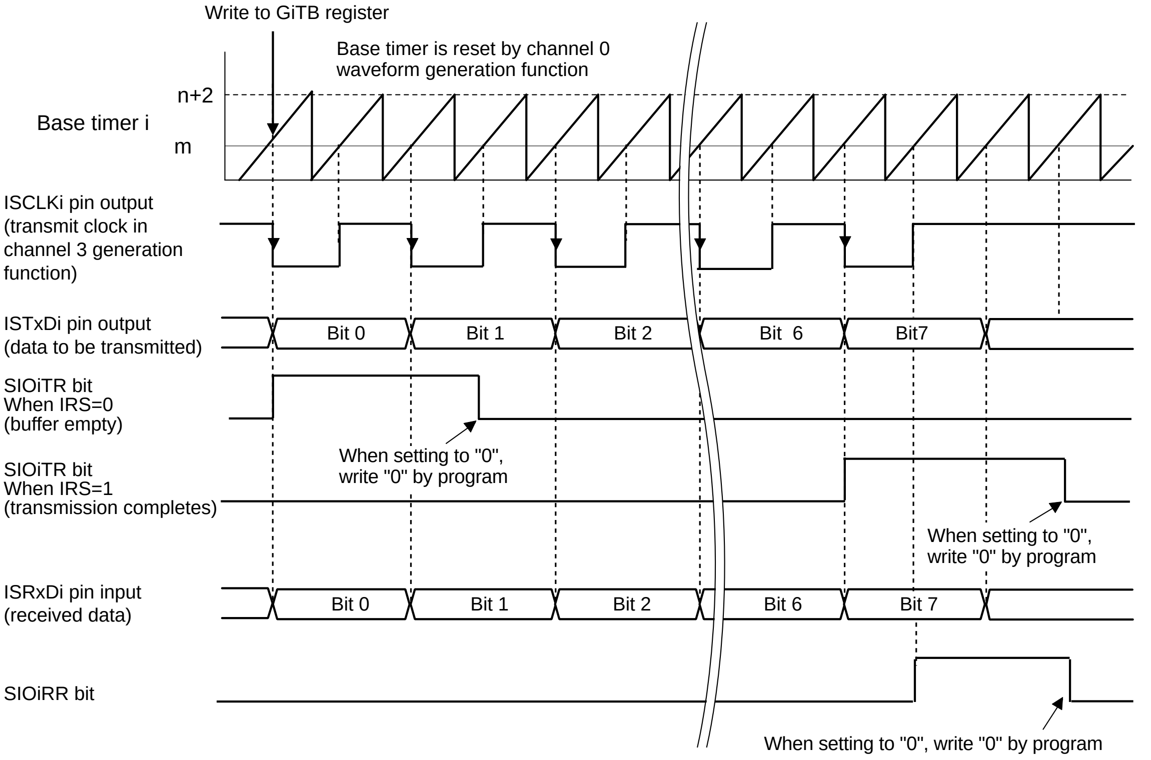
1. The MOD2 to MOD0 bits in the corresponding register should be set to "111$_2$" (output of the communication function is used).

Table 1. 22. 21. Pin Settings (Continued)

Port name	Function	Bit and setting			register[1]
		PS9 register	PD15 register	IPS register	
P15$_0$	ISTxD1 output	PS9_0 = 1	-	-	G0POCR0
P15$_1$	ISCLK1 input	PS9_1 = 0	PD15_2 = 0	IPS0 = 1	-
	ISCLK1 output	PS9_1 = 1	-	-	G0POCR1
P15$_2$	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 "111$_2$" (output of the communication function is used).



The above applies to the following condition.

- The CKDIR bit in the GiMR register is set to "0" (internal clock)
- The UFORM bit in the GiMR register is be set to "0" (LSB first)
- The IPOL and OPOL bits in the GiCR register are set to "0" (no inverse)

n : Setting value of the GiPO0 register
m : Setting value of the GiPO3 register
i : 0,1

SIOiTR bit : Bit in the IIOjIR register (j = 1,3)
SIOiRR bit : Bit in the IIOkIR register (k = 0,2)
IRS bit : Bit in the GiMR register

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	• Character Bit (transfer data) : 8 bits long • Start bit : 1 bit long • Stop bit : selectable from 1 bit or 2 bits long
Transfer clock[1,2]	• When the CKDIR bit in the GiMR register (i=0, 1) is set to "0" (internal clock) : $\frac{f_{BTi}}{2(n+2)}$ n : setting value of the GiPO0 register, 0000_{16} to $FFFF_{16}$. – The he GiPO0 register determines the baud rate and the transfer clock is generated in phase-delayed waveform output mode Transmit clock is generated with the channel 3 waveform generation function. Receive clock is generated with the channel 2 time measurement function.
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 • The TE bit in the GiCR register is set to "1" (transmit enable) • The TI bit in the GiCR register is set to "0" (data written to the GiTB register)
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 • The RE bit in the GiCR register is set to "1" (receive enable) • The start bit is detected
Interrupt request	• While transmitting, the following condition can be selected to set the SIOiTR bit to "1" (See Figure 1.9.14.) : – 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. – When the IRS bit is set to "1" (interrupt at reception completed) and data transfer from the transmit register is completed • 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)
Error detection	• Overrun error[3] This error occurs when receiving the last bit of the next data before reading the GiRB register • Flaming Error This error occurs when the number of the stop bits set is not detected
Selectable function	• Stop bit length The stop bit is selectable from 1 bit or 2 bits long • LSB first/MSB first Whether data is transmitted/received in bit 0 or in bit 7 can be selected

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 "11_2"
	DIV4 to DIV0	Select a divide ratio of a count source
	IT	Set to "0"
GiBCR1	7 to 0	Set to "0001 0010_2"
GiPOCR0	7 to 0	Set to "0000 0111_2"
GiPOCR2	7 to 0	Set to "0000 0110_2"
GiPOCR3	7 to 0	Set to "0000 0010_2"
GiTMCR2	7 to 0	Set to "0000 0010_2"
GiPO0	15 to 0	Set a baud rate $\frac{f_{BTi}}{2 \times (\text{setting value} + 2)}$ = transfer clock frequency
GiPO3	15 to 0	Set a smaller value than the GiPO0 register
GiFS	FSC3 to FSC0	Set to "0100_2"
GiFE	IFE3 to IFE0	Set to "1101_2"
GiMR	GMD1 to GMD0	Set to "00_{16}"
	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[1]
		PS1 register	PSL1 register	PSC register	PD7 register	IPS register	
$P7_3$	ISTxD1 output	PS1_3 = 1	PSL1_3 = 0	PSC_3 = 1	-	-	G1POCR0
$P7_5$	ISRxD1 input	PS1_5 = 0	-	-	PD7_5 = 0	IPS1 = 0	-
$P7_6$	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 "111_2" (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	
$P8_0$	ISRxD0 input	PS2_0 = 0	-	PD8_0 = 0	IPS0 = 0	-

Table 1.22.26. Pin Settings (Continued)

Port name	Function	Bit and setting			Register[1]
		PS5 register	PD11 register	IPS register	
$P11_0$	ISTxD1 output	PS5_0 = 1	-	-	G1POCR0
$P11_2$	ISRxD1 input	PS5_2 = 0	PD11_2 = 0	IPS1 = 1	-

Notes :

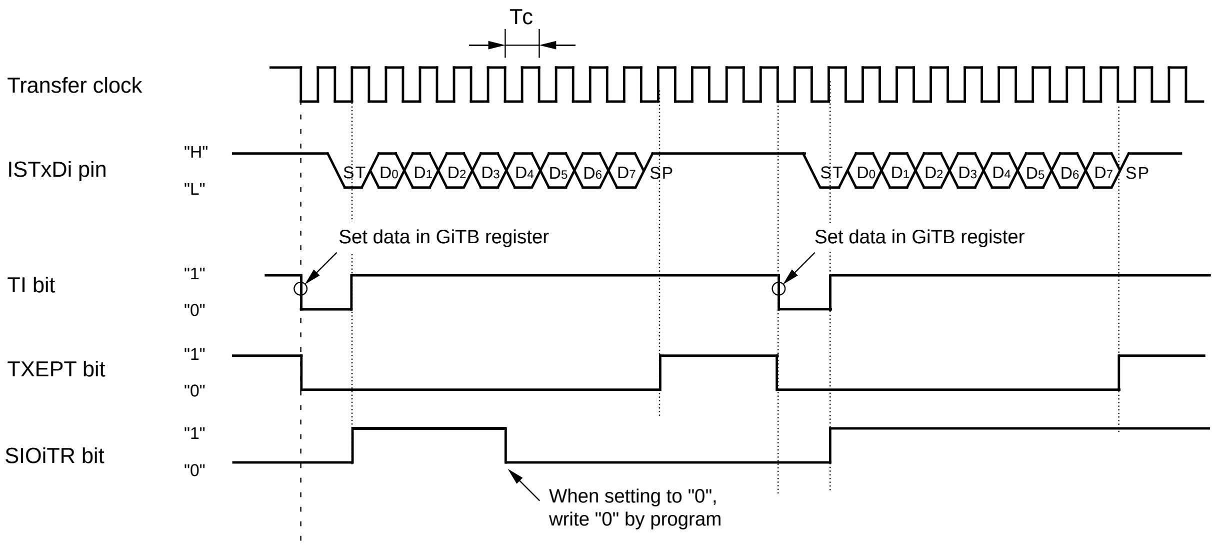
1. The MOD2 to MOD0 bits in the corresponding register should be set to "111_2" (output of the communication function is used).

Table 1.22.27. Pin Settings (Continued)

Port name	Function	Bit and setting			Register[1]
		PS9 register	PD15 register	IPS register	
$P15_0$	ISTxD0 output	PS9_0 = 1	-	-	G0POCR0
$P15_2$	ISRxD0 input	-	PD15_2 = 0	IPS0 = 1	-

Notes :

1. The MOD2 to MOD0 bits in the corresponding register should be set to "111_2" (output of the communication function is used).

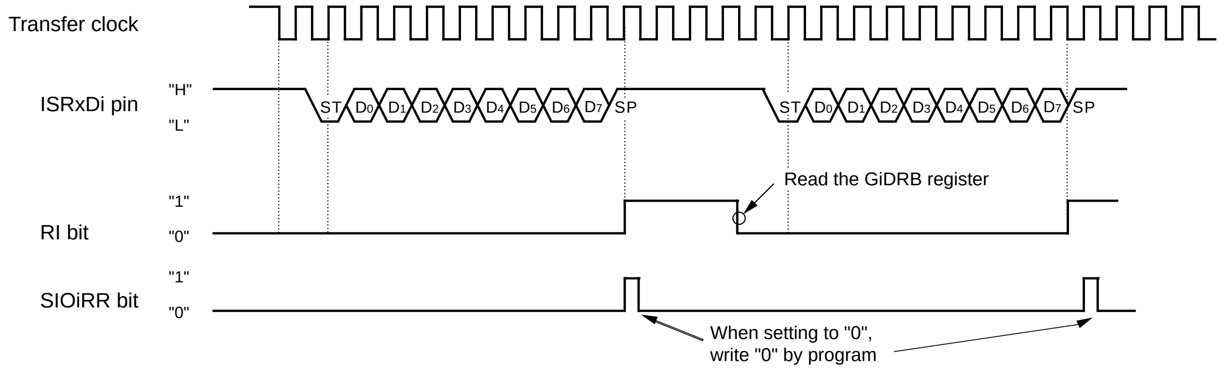


The above applies to the following condition.
- The STPS bit in the GiMR register is set to "0" (1 stop bit)
- The UFORM bit in the GiMR register is set to "0" (LSB first)
- The INV bits in the GiPOCR0 to GiPOCR7 registers are set to "0" (no inverse)

n : Setting value of the GiPO0 register
m : Setting value of the GiPO3 register
i : 0,1

TI, TXEPT bit : Bits in the GiCR register
SIOiTR bit : Bit in the IIOjIR register (j = 1,3)
IRS bit : Bit in the GiMR register

Figure 1.22.40. Transmit Operation



The above applies to the following condition.
- The STPS bit in the GiMR register is set to "0" (1 stop bit)
- The UFORM bit in the GiMR register is set to "0" (LSB first)
- The INV bits in the GiPOCR0 to GiPOCR7 registers are set to "0" (no inverse)

i : 0,1

SIOiRR bit : Bit in the IIOkIR register (k = 0,2)
RI bit : Bit in the GiCR register

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	• When the CKDIR bit in the GiMR register (i=0, 1)is set to "0" (internal clock) : $\frac{f_{BTi}}{n+2}$ n : setting value of the GiPO0 register 0000_{16} to $FFFF_{16}$ - 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 • When the RSHTE bit in the GiERC register is set to "1" (reception shifting operation enable) , the transfer clock is generated in the receiver
I/O method	• While transmitting Value set in the GiTB register is converted in HDLC data processing mode to transfer it to the GiTO register • 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).
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 "$7E_{16}$" 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_{16}" should be written in the GiMSKj register
CRC	The CRC1 to CRC0 bits are set to "11_2" ($X^{16}+X^{12}+X^5+1$) The CRV bit is set to "1" (set to $FFFF_{16}$) • 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).[1] • 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 "$7E_{16}$". The ACRC bit in the GiEMR register is set to "1" (CRC reset)[2]

Table 1.22.28. HDLC Processing Mode Specifications (Continued)

Item	Specification
Transmit start condition	The following conditions are required to start transmitting • The TE bit in the GiCR register is set to "1" (transmit enable) • Data written to the GiTB register are required
Receive start condition	The following conditions are required to start transmitting • The RE bit in the GiCR register is set to "1" (transmit enable) • Data written to the GiTB register
Interrupt request[3]	• While transmitting, (1) The following condition can be selected to set the GiTOR register bit to "1" – 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. – When the IRS bit is set to "1" (interrupt at reception completed) and data transfer from the transmit register is completed. (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". • While receiving, (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) (2) When received data is transferred to the receive register from the GiRI register, the GiRIR bit is set to "1". (3) When the GiTB register is compared to the GiCMPj register (j=0 to 3), the SRTiR bit is set to "1".

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_2"
GiPOCR0	7 to 0	Set to "0000 0000_2"
GiPOCR1	7 to 0	Set to "0000 0000_2"
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_2"
GiFE	IFE0 to IFE1	Set to "11_2"
GiMR	GMD1 to GMD0	Set to "11_2"
	CKDIR	Set to "0"
	UFORM	Set to "0"
	IRS	Select how the transmit interrupt is generated
GiEMR	7 to 0	Set to "1111 0110_2"
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_{16}" to abort processing
GiCMP2	7 to 0	Data to be compared
GiCMP3	7 to 0	Write "$7E_{16}$"
GiMSK0, GiMSK1	7 to 0	Write "01_{16}" 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

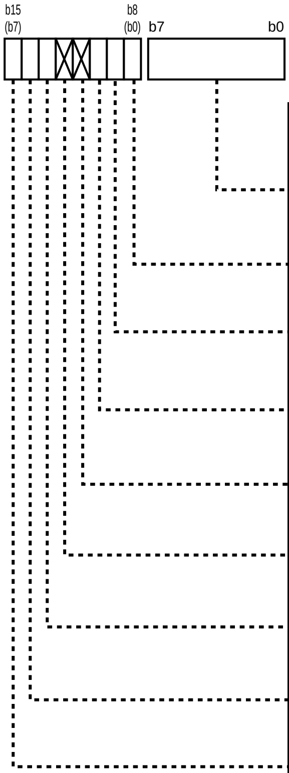
(2) Group 2 Communication Function

In the intelligent I/O group 2, variable clock synchronous serial I/O or IE Bus[1] 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.

Group 2 SI/O transmit buffer register



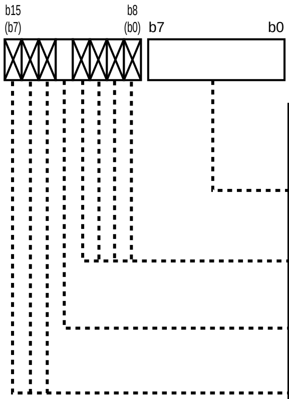
Symbol G2TB
Address 016D16-016C16
When reset Indeterminate

Bit symbol	Bit name	Function	RW
——	Transmit buffer	Data to be transmitted	RO
SZ0	Transfer bit length select bit	b10 b9 b8 0 0 0 : 8 bits long 0 0 1 : 1 bits long 0 1 0 : 2 bits long 0 1 1 : 3 bits long 1 0 0 : 4 bits long 1 0 1 : 5 bits long 1 1 0 : 6 bits long 1 1 1 : 7 bits long	RW
SZ1			RW
SZ2			RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–
A	ACK function select bit	0 : Adds no ACK bit 1 : Adds the ACK bit after a final transmit bit	RW
PC	Parity calculation continuing bit[1]	0 : Adds the parity bit after a transmit data to be transmitted 1 : Carries over a parity to the next transmit data	RW
P	Parity function select bit	0 : No parity 1 : Parity (even parity only)	RW

Notes :

1. When setting the PC bit to "1", the P bit should be set to "0".

Group 2 SI/O receive buffer register



Symbol G2RB
Address 016F16 - 016E16
When reset Indeterminate

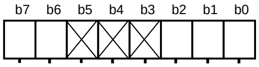
Bit symbol	Bit name	Function	RW
——	Receive buffer	Received data	RO
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
OER	Overrun error flag[1]	0 : No overrun error 1 : Overrun error found	RO
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–

Notes :

1. When the GMD1 to GMD0 bits in the G2MR register are set to "002" (communication unit reset) or the RE bit in the G2CR register be set to "0" (receive disable), the OER bit is set to "0".

Figure 1.22.42. G2TB and G2RB Register

Group 2 SI/O communication mode register



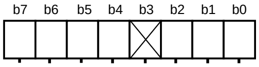
Symbol	Address	When reset
G2MR	$016A_{16}$	$00XX\ X000_2$

Bit symbol	Bit name	Function	RW
GMD0	Communication mode select bit	b1b0 0 0 : Communication unit is reset (The OER bit is set to "0")[1] 0 1 : Clock synchronous serial I/O mode[2] 1 0 : IE mode[2] 1 1 : Avoid this setting	RW
GMD1			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_2" (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_2" (clock synchronous serial I/O mode) or "10_2" (IE mode), while the base timer clock stops.

Group 2 SI/O communication control register



Symbol	Address	When reset
G2CR	$016B_{16}$	$0000\ X000_2$

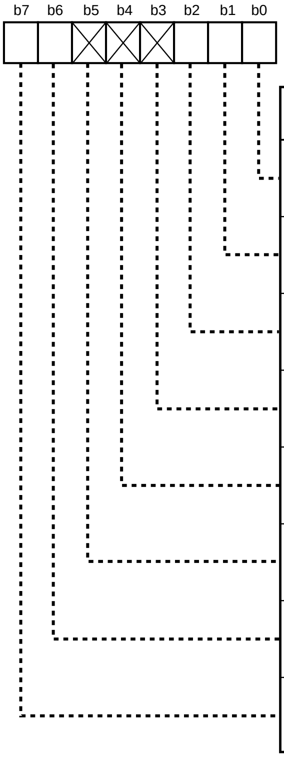
Bit symbol	Bit name	Function	RW
TE	Transmit enable bit	0 : Transmit disable 1 : Transmit enable	RW
TXEPT	Transmit register empty flag	0 : Data available in transmit register (during transmission) 1 : No data available in transmit register (transmission is completed)	RO
TI	Transmit buffer empty flag	0 : 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[1]	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[1]	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

Group 2 IE Bus control register

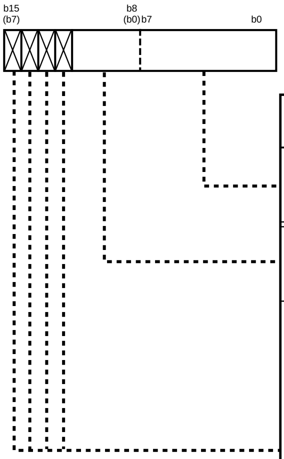


Symbol	Address	When reset
IECR	0172_{16}	$00XX\ X000_2$

Bit symbol	Bit name	Function	RW
IEB	IE Bus enable bit[1]	0 : IE Bus disabled[2] 1 : IE Bus enabled	RW
IETS	IE Bus transmit start request bit	0 : Transmission completes 1 : Transmission starts	RW
IEBBS	IE Bus busy flag	0 : Idle state 1 : Busy state (start condition detected)	RO
——	Nothing is assigned. When write, should set "0". When read, its content is indeterminate.		–
——			–
——			–
DF	Digital filter select bit	0 : No digital filter 1 : Digital filter	RW
IEM	IE Bus mode select bit	0 : Mode 1 1 : Mode 2	RW

Notes :
1. The IEB bit should be rewrite while the base timer clock stops.
2. When setting the IEB bit to "0", "0" should be hold for at least 1 cycle of the base timer. When setting the IEB bit to "1", the BCK1 to BCK0 bits in the G2BCR0 register should be set to "00_2" (clock stop) first.

Group 2 IE Bus address register

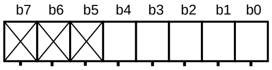


Symbol	Address	When reset
IEAR	0171_{16} - 0170_{16}	Indeterminate

Function	RW
Address data	RW
Address data	RW
Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.	–
	–
	–
	–

Figure 1.22.44. IECR and IEAR Registers

Group 2 IE Bus transmit interrupt cause determination register



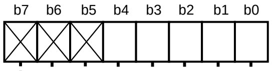
Symbol	Address	When reset
IETIF	0173$_{16}$	XXX0 0000$_2$

Bit symbol	Bit name	Function	RW
IETNF	Normal complete flag	0 : Completed in error 1 : Normally completed (Note 1)	RW
IEACK	ACK error flag	0 : No error 1 : Error found (Note 1)	RW
IETMB	Maximum transfer byte error flag	0 : No error 1 : Error found (Note 1)	RW
IETT	Timing error flag	0 : No error 1 : Error found (Note 1)	RW
IEABL	Arbitration lost flag	0 : No error 1 : Error found (Note 1)	RW
——	Nothing is assigned. When write, should set to "0". When read, its contents is indeterminate.		–
——			–
——			–

Notes :

1. This bit can be written to only "0" by program, but not to "1". When the IEB bit in the IECR register is set to "0" (IEBus disabled to use), this bit is set to "0".

Group 2 IE Bus receive interrupt cause determination register



Symbol	Address	When reset
IERIF	0174$_{16}$	XXX0 0000$_2$

Bit symbol	Bit name	Function	RW
IERNF	Normal completed flag	0 : Completed in error 1 : Normally completed (Note 1)	RW
IEPAR	Parity error flag	0 : No error 1 : Error found (Note 1)	RW
IERMB	Max. transfer byte error flag	0 : No error 1 : Error found (Note 1)	RW
IERT	Timing error flag	0 : No error 1 : Error found (Note 1)	RW
IERETC	Other cause receive completed flag	0 : No error 1 : Error found (Note 1)	RW
——	Nothing is assigned. When write, should set to "0". When read, its contents is indeterminate.		–
——			–
——			–

Notes :

1. This bit can be written to only "0" by program, but not to "1". When the IEB bit in the IECR register is set to "0" (IEBus disabled to use), this bit is set to "0".

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[1]	• When the CKDIR bit in the G2MR register is set to "0" (internal clock) : $\frac{f_{BT2}}{2(n+2)}$ n : setting value of the G2PO0 register 0000_{16} to $FFFF_{16}$ - 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 • When the CKDIR bit is set to "1" (external clock) : input from the ISCLK2 pin[2]
Transmit start condition	• To start transmitting, the following conditions are required : - The TE bit in the G2CR register is set to "1" (transmit enable) - Data written to the G2TB register
Receive start condition	• To start receiving, the following conditions are required : - The RE bit in the G2CR register should be set to "1" (receive enable) - The TE bit in the G2CR register should be set to "1" (transmit enable) - Data written to the G2TB register
Interrupt request	• While transmitting, the following condition can be selected to set the SIO2TR bit in the IIO6IR register to "1" (see Figure 1.9.14) : - 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 - When setting the IRS bit to "1" (interrupt at reception completed) and data transfer from the transmit register is completed • 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)
Error detection	Overrun error[3] This error occurs when receiving the 8th bit of the next data before reading the G2RB register
Selectable function	• LSB first/MSB first Whether data is transmitted/received in bit 0 or in bit 7 can be selected • ISTxD2 and ISRxD2 I/O polarity inverse ISTxD2 pin output and ISRxD2 pin input levels are inversed • Data transfer bit length Transfer bit is selectable from 1 bits to 8 bits

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 "11_2"
	DIV4 to DIV0	Select a divide ratio of a count source
	IT	Set to "0"
G2BCR1	7 to 0	Set to "0001 0010_2"
G2POCR0	7 to 0	Set to "0000 0111_2"
G2POCR1	7 to 0	Set to "0000 0111_2"
G2BCR2	7 to 0	Set to "0000 0010_2"
G2PO0	15 to 0	Set a baud rate $\frac{f_{BT2}}{2 \text{ x (setting value + 2)}}$ = transfer clock frequency
G2PO2	15 to 0	Set a smaller value than G2PO0 register
G2FE	IFE2 to IFE0	Set to "111_2"
G2MR	GMD1 to GMD0	Set to "01_2"
	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[2]
		PS1 register	PSL1 register	PSC register	PD7 register	IPS register	
P7$_0$[1]	ISTxD2 output	PS1_0 = 1	PSL1_0 = 0	PSC_0 = 1	-	-	G2POCR0
P7$_1$	ISRxD2 input	PS1_1 = 0	-	-	PD7_1 = 0	IPS5 to 4 = 00$_2$	-

Notes :

1. Output is N-channel open drain output.
2. The MOD2 to MOD0 bits in the corresponding register should be set to "111$_2$" (output of the communication function is used).

Table 1.22.33. Pin Settings (Continued)

Port name	Function	Bit and setting				Register[2]
		PS3 register	PSL3 register	PD9 register[1]	IPS register	
P9$_1$	ISRxD2 input	PS3_1=0	-	PD9_1=0	IPS5 to 4=01$_2$	-
P9$_2$	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 "111$_2$" (output of the communication function is used).

Table 1.22.34. Pin Settings (Continued)

Port name	Function	Bit and setting				Register[1]
		PS0 register	PSL0 Register	PD6 register	IPS register	
P6$_4$	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 "111$_2$" (output of the communication function is used).

Table 1.22.35. Pin Settings (Continued)

Port name	Function	Bit and setting			Register[1]
		PS7 register	PD13 register	IPS register	
P13$_4$	ISTxD2 output	PS7_4 = 1	-	-	G2POCR0
P13$_5$	ISRxD2 input	PS7_5 = 0	PD13_5 = 0	IPS5 to 4 = 10$_2$	-
P13$_6$	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 "111$_2$" (output of the communication function is used).

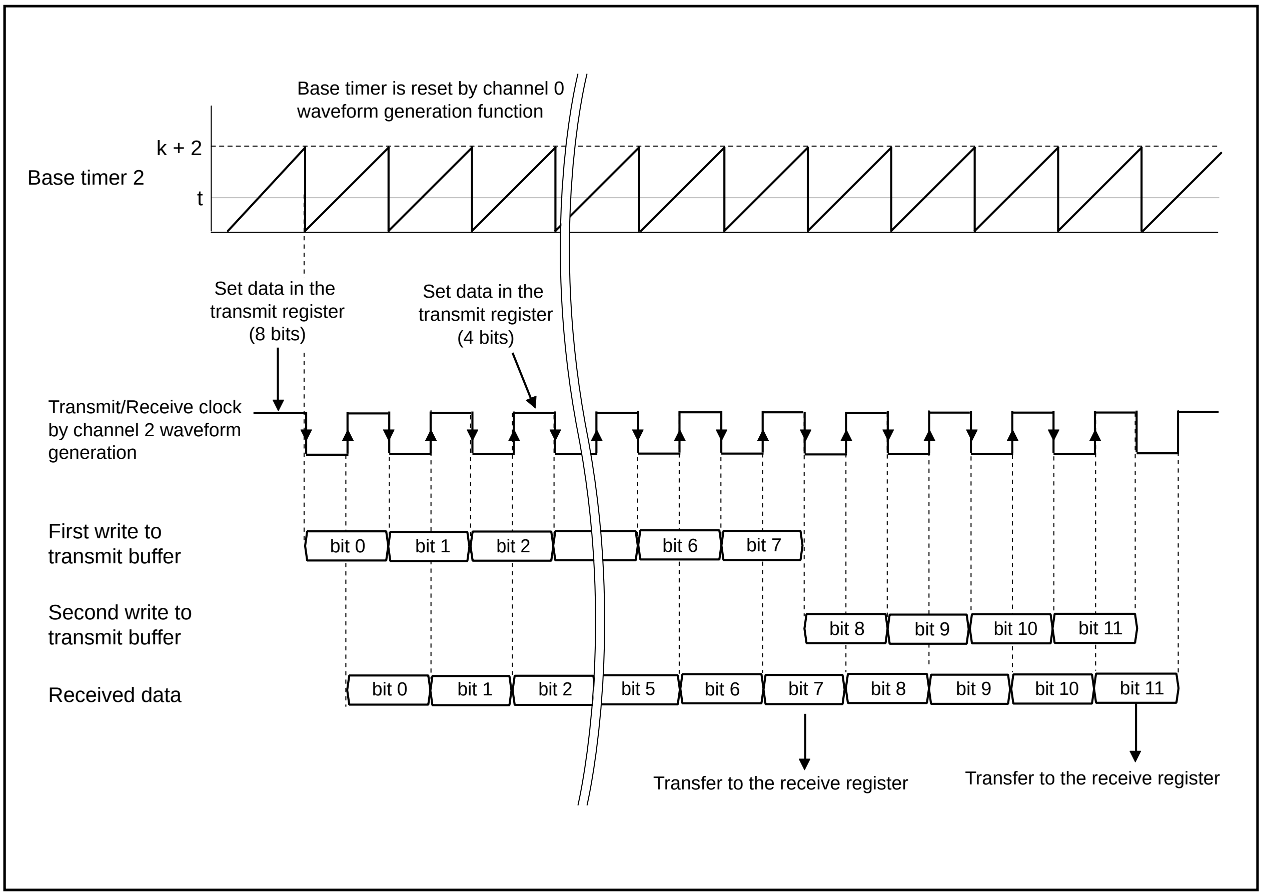

Base timer is reset by channel 0 waveform generation function
k + 2
Base timer 2
t
Set data in the transmit register (8 bits)
Set data in the transmit register (4 bits)
Transmit/Receive clock by channel 2 waveform generation
First write to transmit buffer
bit 0
bit 1
bit 2
bit 6
bit 7
Second write to transmit buffer
bit 8
bit 9
bit 10
bit 11
Received data
bit 0
bit 1
bit 2
bit 5
bit 6
bit 7
bit 8
bit 9
bit 10
bit 11
Transfer to the receive register
Transfer to the receive register

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	• Transfer data : Variable
Transfer clock	• When the CKDIR bit in the G2MR register is set to "0" (internal clock) : $\frac{f_{BT2}}{2(n+2)}$ n : setting value of the G2PO0 register, 0000_{16} to $FFFF_{16}$. – 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. – The G2PO2 register = (n+2)/2 • When the CKDIR bit is set to "1" (external clock) : input from the ISCLK2 pin[2]
Transmit start condition	To start transmitting, the following conditions are required : • The TE bit in the G2CR register is set to "1" (transmit enable) • Write data to G2TB register
Receive start condition	To start receiving, the following requirements must be met: • The RE bit in the G2CR register should be set to "1" (receive enable) • The TE bit in the G2CR register should be set to "1" (transmit enable) • Data written to the G2TB register
Interrupt request	• While transmitting, the following condition can be selected to set the SIO2TR bit in the IIO6R register to "1" (see Figure 1.9.14) : - 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) - When the IRS bit is set to "1" (interrupt at reception completed) and data transfer from the transmit register is completed • While receiving 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).
Error detection	Overrun error[3] This error occurs when receiving the 8th bit of the next data before reading the G2RB register
Selectable function	• LSB first/MSB first select Whether data is transmitted/received in bit 0 or in bit 7 can be selected • ISTxD2 and ISRxD2 I/O polarity inverse ISTxD2 pin output and ISRxD2 pin input levels are inversed • Data transfer bit length Transfer bit is selectable from 1 bit to 8 bits

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 "11_2"
	DIV4 to DIV0	Select a divide ratio of a count source
	IT	Set to "0"
G2BCR1	7 to 0	Set to "00010010_2"
G2POCR0 to G2POCR7	MOD2 to MOD0	Set to "111_2"
	PRT	Set to "0"
	IVL	Set to "0"
	RLD	Set to "0"
	RTP	Set to "0"
	INV	Set to "0"
G2PO0 to G2PO7	15 to 0	Set compared data for waveform generation
G2FE	7 to 0	Set a bit of corresponding channel to "1"
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[1]
		PS1 register	PSL1 register	PSC register	PD7 register	IPS register	
P7$_0$[2]	IEOUT output	PS1_0 = 1	PSL1_0 = 0	PSC_0 = 1	-	-	G2POCR0
P7$_1$[2]	IEIN input	PS1_1 = 0	-	-	PD7_1 = 0	IPS5 to 4 = 00$_2$	-

Notes :

1. The MOD2 to MOD0 bits in the G2POCR0 register should be set to "111$_2$".
2. Output is N-channel open drain.

Table 1.22.39. Pin Settings (Continued)

Port name	Function	Bit and setting				Register[1]
		PS3 register[2]	PSL3 register	PD9 register[2]	IPS register	
P9$_1$	IEIN input	PS3_1 = 0	-	-	IPS5 to 4 = 01$_2$	-
P9$_2$	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 "111$_2$".
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)

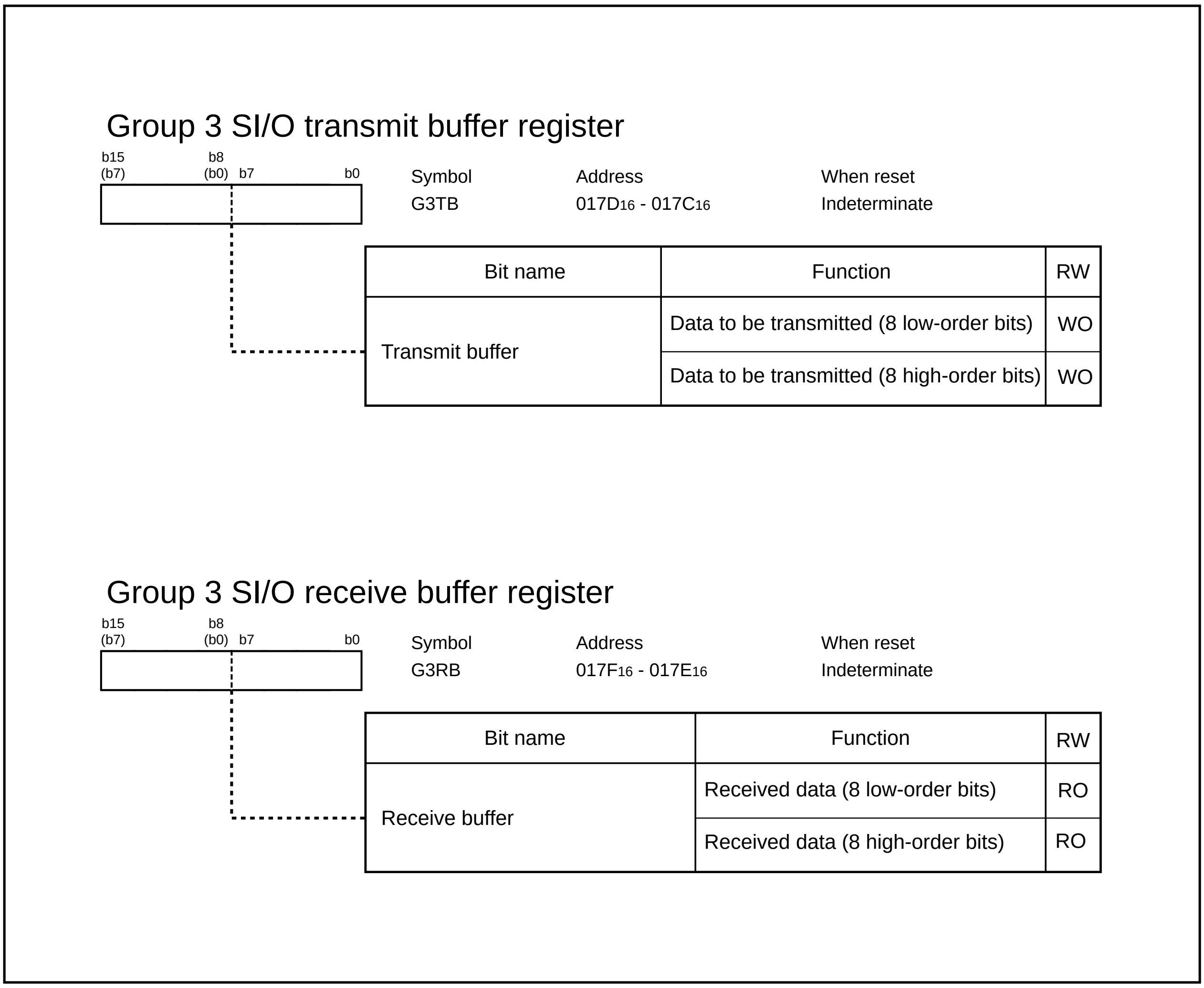
Port name	Function	Bit and setting			Register[1]
		PS7 register	PSL7 register	IPS register	
P13$_4$	IEOUT output	PS7_4 = 1	-	-	G2POCR0
P13$_5$	IEIN input	PS7_5 = 0	PD13_5 = 0	IPS5 to 4 = 10$_2$	-

Notes :

1. The MOD2 to MOD0 bits in the G2POCR0 register should be set to "111$_2$".

(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.



Group 3 SI/O transmit buffer register

Symbol	Address	When reset
G3TB	017D$_{16}$ - 017C$_{16}$	Indeterminate

Bit name	Function	RW
Transmit buffer	Data to be transmitted (8 low-order bits)	WO
	Data to be transmitted (8 high-order bits)	WO

Group 3 SI/O receive buffer register

Symbol	Address	When reset
G3RB	017F$_{16}$ - 017E$_{16}$	Indeterminate

Bit name	Function	RW
Receive buffer	Received data (8 low-order bits)	RO
	Received data (8 high-order bits)	RO

Figure 1.22.47. G3TB Register and G3RB Register

Group 3 SI/O communication mode register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
G3MR	017A$_{16}$	00XX 0000$_2$

Bit symbol	Bit name	Function	RW
GMD0	Communication mode select bit	b1 b0 0 0 : Communication unit is reset (The OER bit is set to "0") 0 1 : Clock synchronous serial I/O mode 1 0 : Avoid this setting 1 1 : Avoid this setting	RW
GMD1			RW
CKDIR	Internal/external clock select bit	0 : Internal clock 1 : External clock	RW
TLD	Transfer data length select bit	0 : 16 bits long 1 : 8 bits long	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

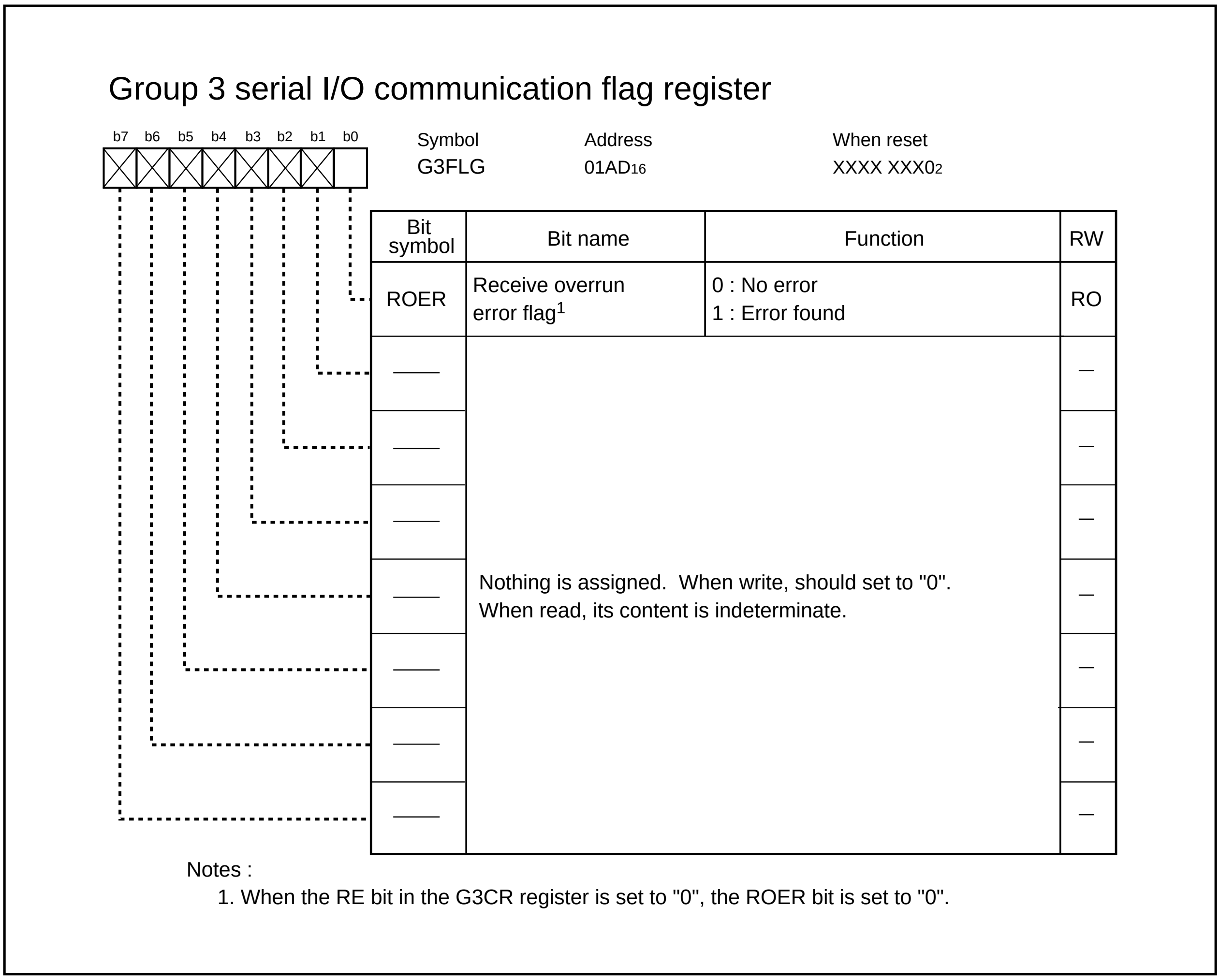
Group 3 SI/O communication control register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
G3CR	017B$_{16}$	0000 X000$_2$

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 G3TB register 1 : No data available in the G3TB register	RO
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
RE	Receive enable bit	0 : Receive disable 1 : Receive enable	RW
RI	Receive complete flag	0 : No data available in the G3RB register 1 : Data available in the G3RB register	RO
OPOL	ISTxD output polarity switch bit	0 : No inverse 1 : Inverse	RW
IPOL	ISRxD input polarity switch bit	0 : No inverse 1 : Inverse	RW

Figure 1.22.48. G3MR Register and G3CR Register



Group 3 serial I/O communication flag register

Symbol	Address	When reset
G3FLG	01AD$_{16}$	XXXX XXX0$_2$

Bit symbol	Bit name	Function	RW
ROER	Receive overrun error flag[1]	0 : No error 1 : Error found	RO
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
——			—
——			—
——			—
——			—
——			—
——			—

Notes :

1. When the RE bit in the G3CR register is set to "0", the ROER bit is set to "0".

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 $OUTC3_0$ to $OUTC3_2$ 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[1]	• When the CKDIR bit in the G3MR register is set to "0" (internal clock) : $\frac{f_{BT3}}{2(n+2)}$ n : setting value of the G3PO0 register 0000_{16} to $FFFF_{16}$ Baud rate is determined by the G3PO0 register and generated with the channel 2 waveform generation function and in phase-delayed waveform output mode • When setting the CKDIR bit to "1" (external clock) : input from the ISCLK3 pin
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. • The TE bit in the G3CR register is set to "1" (transmit enable) • The TI bit in the G3CR register is set to "0" (data written to the G3TB register)
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. • The TE bit should is set to "1" (transmit enable) • The RE bit in the G3CR register is set to "1" (receive enable) • The TI bit is set to "0" (data written to the G3TB register)
Interrupt request	• While transmitting, the following condition can be selected to set the SIO3TR bit in the IIO10R register to "1". (See Figure 1.9.14.) – 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). – 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 transimission starts in 8-bit clock clock synchronous serial I/O mode (setting the DLS bit to "1"). • While receiving 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. In 7.5 tranfer clock cycles, after data 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.)
Error detection	• Overrun error[3] 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.
Selectable function	• LSB first/MSB first : Whether data is transmitted/received in bit 0 or in bit 7 can be selected • ISTxD3 and ISRxD3 I/O polarity inverse : ISTxD3 pin output and ISRxD3 pin input levels are inversed.

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 "11_2"
	DIV4 to DIV0	Select a divide ratio of a count source
	IT	Set to "0"
G3BCR1	7 to 0	Set to "$0001\ 0010_2$"
G3POCR0	7 to 0	Set to "$0000\ 0111_2$"
G3POCR1	7 to 0	Set to "$0000\ 0111_2$"
G3POCR2	7 to 0	Set to "$0000\ 0010_2$"
G3PO0	15 to 0	Set a baud rate $\frac{f_{BT3}}{2 \times (\text{setting value} + 2)}$ = transfer clock frequency
G3PO2	15 to 0	Set a smaller value than the G3PO0 register
G3FE	7 to 0	Set to "$0000\ 0111_2$"
G3MR	GMD1 to GMD0	Set to "01_2"
	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[1]
		PS2 register	PSL2 register	PD8 register	IPS register	
P8$_1$	ISTxD3 output	PS2_1 = 1	PSL2_1 = 1	-	-	G3POCR0
P8$_2$	ISRxD3 input	PS2_2 = 0	-	PD8_2 = 0	IPS7 = 0	-

Notes :

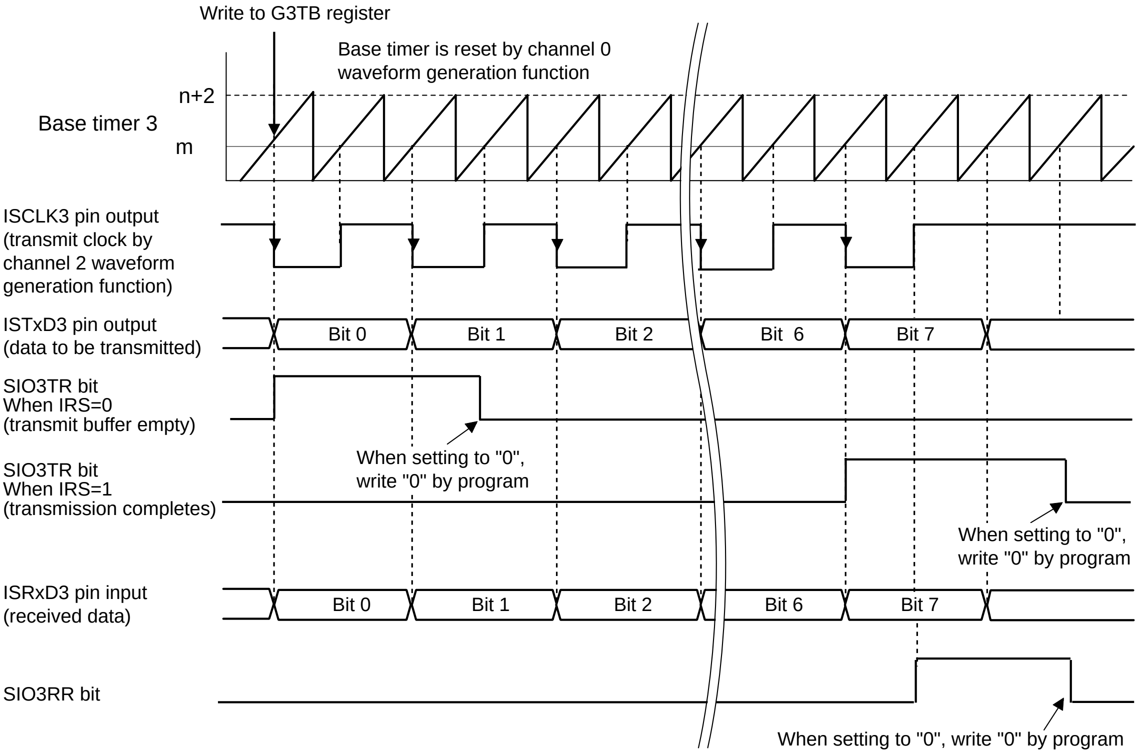
1. The MOD2 to MOD0 bits in the corresponding register should be set to "111_2" (output of the communication function is used).

Table 1.22.44. Pin Setting (Continued)

Port name	Function	Bit and setting			Register[1]
		PS6 register	PD12 register	IPS register	
P12$_0$	ISTxD3 output	PS6_0 =1	-	-	G3POCR0
P12$_1$	ISCLK3 input	PS6_1 = 0	PD12_1 = 0	-	-
	ISCLK3 output	PS6_1 = 1	-	-	G3POCR1
P12$_2$	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 "111_2" (output of the communication function is used).


Write to G3TB register
Base timer is reset by channel 0 waveform generation function
Base timer 3
n+2
m
ISCLK3 pin output (transmit clock by channel 2 waveform generation function)
ISTxD3 pin output (data to be transmitted)
Bit 0
Bit 1
Bit 2
Bit 6
Bit 7
SIO3TR bit When IRS=0 (transmit buffer empty)
When setting to "0", write "0" by program
SIO3TR bit When IRS=1 (transmission completes)
When setting to "0", write "0" by program
ISRxD3 pin input (received data)
Bit 0
Bit 1
Bit 2
Bit 6
Bit 7
SIO3RR bit
When setting to "0", write "0" by program

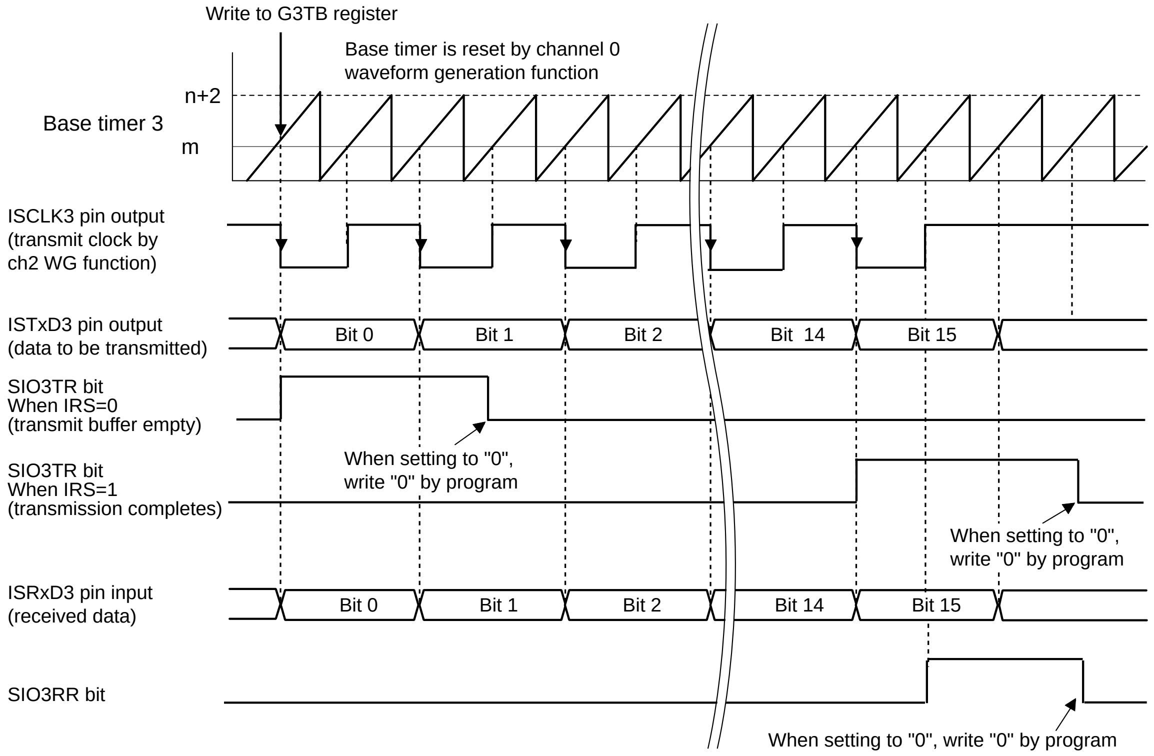
The above timing applies to the following settings.

- The CKDIR bit in the G3MR register is set to "0" (internal clock)
- The UFORM bit in the G3MR register is set to "0" (LSB first)
- The IPOL bit and OPOL bit in the G3CR register are set to "0" (no inverse)
- The TLD bit in the G3CR register is set to "1" (8 bits long)

n : Setting value of the G3PO0 register
m : Setting value of the G3PO2 register

SIO3TR bit : Bit in the IIO10IR register
SIO3RR bit : Bit in the IIO9IR register
IRS bit : Bit in the G3MR register

Figure 1. 22. 50. Transmit and Receive Operation (8-bit Length)


Write to G3TB register
Base timer is reset by channel 0 waveform generation function
n+2
Base timer 3
m
ISCLK3 pin output (transmit clock by ch2 WG function)
ISTxD3 pin output (data to be transmitted)
Bit 0
Bit 1
Bit 2
Bit 14
Bit 15
SIO3TR bit When IRS=0 (transmit buffer empty)
When setting to "0", write "0" by program
SIO3TR bit When IRS=1 (transmission completes)
When setting to "0", write "0" by program
ISRxD3 pin input (received data)
Bit 0
Bit 1
Bit 2
Bit 14
Bit 15
SIO3RR bit
When setting to "0", write "0" by program

The above timing applies to the following settings.

- The CKDIR bit in the G3MR register is set to "0" (internal clock)
- The UFORM bit in the G3MR register is set to "0" (LSB first)
- The IPOL bit and OPOL bit in the G3CR register are set to "0" (no inverse)
- The TLD bit in the G3CR register is set to "1" (16 bits long)

n : Setting value of the G3PO0 register
m : Setting value of the G3PO2 register

SIO3TR bit : Bit in the IIO10IR register
SIO3RR bit : Bit in the IIO9IR register
IRS bit : Bit in the G3MR register

Figure 1. 22. 51. Transmit and Receive Operation (16-bit Length)

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; AN_0 to AN_7, $AN0_0$ to $AN0_7$, $AN2_0$ to $AN2_7$, $AN15_0$ to $AN15_7$, ANEX0, ANEX1, $\overline{\mathrm{ADTRG}}$. 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, $AN15_0$ to $AN15_7$ pins are not provided.

Table 1.23.1. A-D Converter Specifications

Item	Specification
A-D conversion method	Successive approximation (capacitive coupling-amplifier)
Analog input voltage[1]	0V to AV_{CC} (V_{CC})
Operating clock Ø$_{AD}$[2]	f_{AD}, f_{AD}/2, f_{AD}/3 , f_{AD}/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	• Software trigger - The ADST bit in the ADiCON0 register (i = 0,1) is set to "1" (A-D conversion starts) by program - The PST bit in the AD0CON2 register is set to "1" (A-D0 and A-D1 start a conversion simultaneously) by program • External trigger (re-trigger enabled) When setting the ADST bit to "1" by program, $\overline{AD_{TRG}}$ pin input changes "H" to "L" • Hardware trigger (re-trigger enabled) When setting the ADST bit to "1" by program, one of the following interrupt is generated - Timer B2 interrupt request of the three-phase motor control timer functions (after the ICTB2 register completes counting) - Intelligent I/O interrupt request Group2 channel 1 (A-D0), group3 channel 1 (A-D1)
Conversion rate per pin	• Without the sample and hold function 8-bit resolution : 49 Ø$_{AD}$ cycles 10-bit resolution : 59 Ø$_{AD}$ cycles • With the sample and hold function 8-bit resolution : 28 Ø$_{AD}$ cycles 10-bit resolution : 33 Ø$_{AD}$ cycles

Notes:

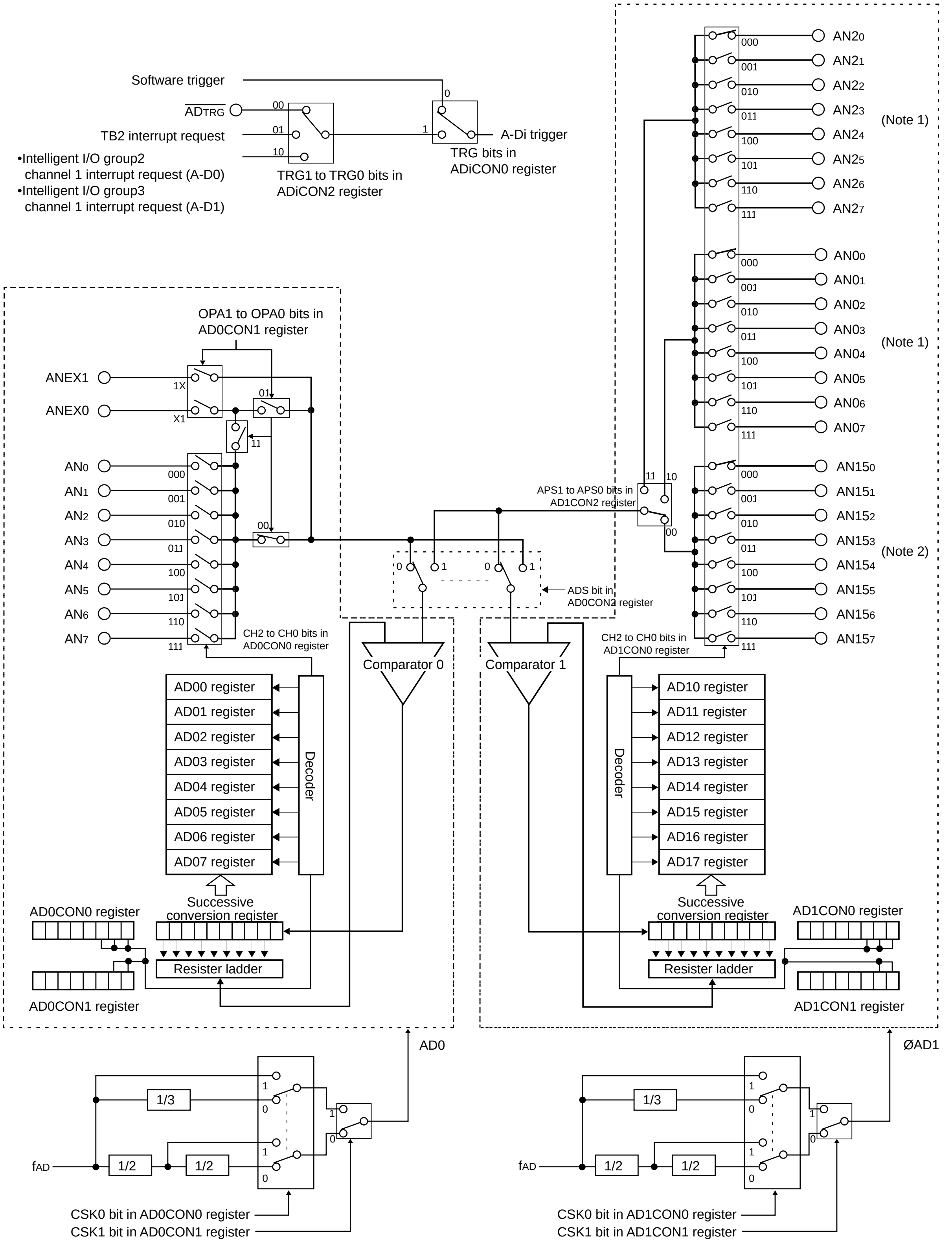
1. Analog input voltage does not vary whether the sample and hold function is used or not.
2. Ø$_{AD}$ frequency must be under 10 MHz.
   Without the sample and hold function, the Ø$_{AD}$ frequency should be 250kHz or more.
   With the sample and hold function, set the Ø$_{AD}$ frequency should be 1MHz or more.

Table 1.23.2. Difference between A-D0 and A-D1

Item	A-D0	A-D1
Analog input pins[1]	AN (AN$_0$ to AN$_7$)	Selectable from AN0 (AN0$_0$ to AN0$_7$), AN2 (AN2$_0$ to AN2$_7$), AN15 (AN15$_0$ to AN15$_7$)
Extended Analog input pins	ANEX0, ANEX1	Not provided
External op-amp[1]	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).



i=0,1

Notes:

1. These are available in single-chip mode.
2. These are available in the 144-pin package.

Figure 1.23.1. A-D Converter Block Diagram

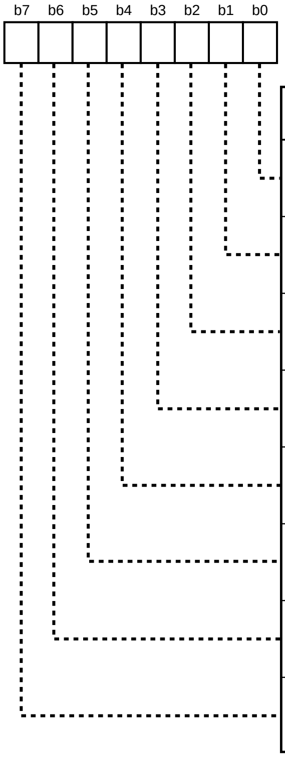
Table 1.23.3. Pin Settings

Port name	Function	Bit and setting			
		PD10, PD0, PD2, PD15, PD9[3] registers	PS3[3], PS9 registers	PSL3, IPS registers	PUR0, PUR3, PUR4 registers
P10$_0$	AN$_0$	PD10_0 = 0	-	-	PU30 = 0
P10$_1$	AN$_1$	PD10_1 = 0			
P10$_2$	AN$_2$	PD10_2 = 0			
P10$_3$	AN$_3$	PD10_3 = 0			
P10$_4$	AN$_4$	PD10_4 = 0			PU31 = 0
P10$_5$	AN$_5$	PD10_5 = 0			
P10$_6$	AN$_6$	PD10_6 = 0			
P10$_7$	AN$_7$	PD10_7 = 0			
P0$_0$	AN0$_0$[1]	PD0_0 = 0	-	-	PU00 = 0
P0$_1$	AN0$_1$[1]	PD0_1 = 0			
P0$_2$	AN0$_2$[1]	PD0_2 = 0			
P0$_3$	AN0$_3$[1]	PD0_3 = 0			
P0$_4$	AN0$_4$[1]	PD0_4 = 0			PU01 = 0
P0$_5$	AN0$_5$[1]	PD0_5 = 0			
P0$_6$	AN0$_6$[1]	PD0_6 = 0			
P0$_7$	AN0$_7$[1]	PD0_7 = 0			
P2$_0$	AN2$_0$[1]	PD2_0 = 0	-	-	PU04 = 0
P2$_1$	AN2$_1$[1]	PD2_1 = 0			
P2$_2$	AN2$_2$[1]	PD2_2 = 0			
P2$_3$	AN2$_3$[1]	PD2_3 = 0			
P2$_4$	AN2$_4$[1]	PD2_4 = 0			PU05 = 0
P2$_5$	AN2$_5$[1]	PD2_5 = 0			
P2$_6$	AN2$_6$[1]	PD2_6 = 0			
P2$_7$	AN2$_7$[1]	PD2_7 = 0			
P15$_0$	AN15$_0$[2]	PD15_0 = 0	PS9_0 = 0	IPS2 = 1	PU42 = 0
P15$_1$	AN15$_1$[2]	PD15_1 = 0	PS9_1 = 0		
P15$_2$	AN15$_2$[2]	PD15_2 = 0	-		
P15$_3$	AN15 $_3$[2]	PD15_3 = 0	-		
P15$_4$	AN15$_4$[2]	PD15_4 = 0	PS9_4 = 0		PU43 = 0
P15$_5$	AN15$_5$[2]	PD15_5 = 0	PS9_5 = 0		
P15$_6$	AN15$_6$[2]	PD15_6 = 0	-		
P15$_7$	AN15$_7$[2]	PD15_7 = 0	-		
P9$_5$	ANEX0	PD9_5 = 0	PS3_5 = 0	PSL3_5 = 1	PU27 = 0
P9$_6$	ANEX1	PD9_6 = 0	PS3_6 = 0	PSL3_6 = 1	
P9$_7$	$\overline{\text{AN}_{\text{TRG}}}$	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-D0 control register 0[1]



Symbol	Address	When reset
AD0CON0	0396_{16}	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
CH0	Analog input pin select bit[2,3,4]	b2 b1 b0 0 0 0 : AN_0 0 0 1 : AN_1 0 1 0 : AN_2 0 1 1 : AN_3 1 0 0 : AN_4 1 0 1 : AN_5 1 1 0 : AN_6 1 1 1 : AN_7	RW
CH1			RW
CH2			RW
MD0	A-D operation mode select bit 0[2]	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 or 1	RW
MD1			RW
TRG	Trigger select bit	0 : Software trigger 1 : External trigger, hardware trigger[5]	RW
ADST	A-D conversion start flag	0 : A-D conversion stops 1 : A-D conversion starts[5]	RW
CKS0	Frequency select bit[6]	0 : Selectable from $f_{AD}/3$ or $f_{AD}/4$ 1 : Selectable from $f_{AD}/1$ or $f_{AD}/2$	RW

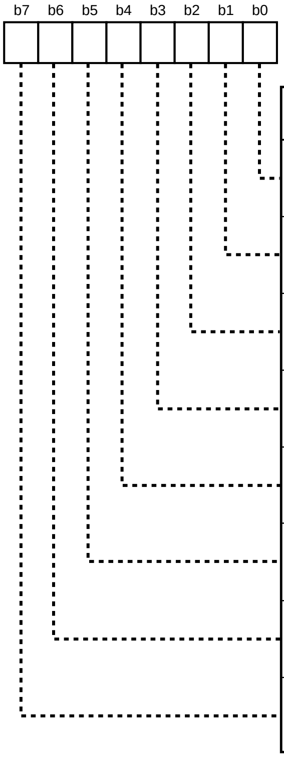
Notes:

1. When the AD0CON0 register is rewritten during the A-D conversion, the conversion result is indeterminate.
2. When changing an A-D operation mode, analog input pins should be set again.
3. This bit is disabled in single sweep mode, repeat sweep mode 0 and repeat sweep mode 1.
4. When the P10 pin is used for a analog input, the PSC_7 bit in the PSC register should be set to "1".
5. When setting the TRG bit to "1" (external trigger, hardware trigger), the TRG1 to TRG0 bits in the AD0CON2 register determine how to generate a trigger. Then the ADST bit should be set to "1"(A-D conversion starts) after setting the TRG bit to "1".
6. ϕ_{AD} frequency must be under 10 MHz. Combinations of CKS0 and CKS1 bits select ϕ_{AD}.

CKS0	CKS1	ϕ_{AD}
0	0	f_{AD} divided by 4
0	1	f_{AD} divided by 3
1	0	f_{AD} divided by 2
1	1	f_{AD}

Figure 1.23.2. AD0CON0 Register

A-D0 control register 1[1]



Symbol	Address	When reset
AD0CON1	0397$_{16}$	0000 0000$_2$

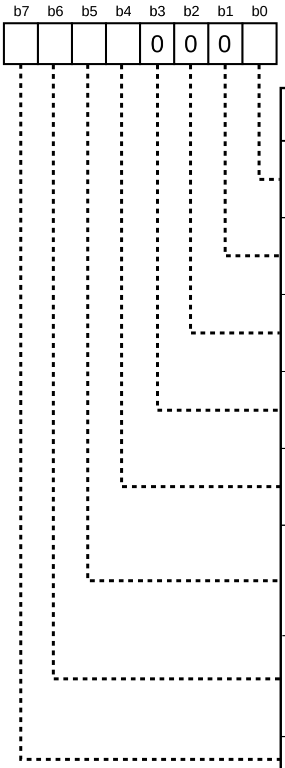
Bit symbol	Bit name	Function	RW
SCAN0	A-D sweep pin select bit[2]	b0 b1 0 0 : AN$_0$, AN$_1$ (AN$_0$) 0 1 : AN$_0$ to AN$_3$ (AN$_0$, AN$_1$) 1 0 : AN$_0$ to AN$_5$ (AN$_0$ to AN$_2$) 1 1 : AN$_0$ to AN$_7$ (AN$_0$ to AN$_3$)	RW
SCAN1			RW
MD2	A-D operation mode select bit 1	0 : Any modes other than repeat sweep mode 1 1 : Repeat sweep mode 1	RW
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency select bit[3]	0 : Selectable from f$_{AD}$/2 or f$_{AD}$/4 1 : Selectable from f$_{AD}$/1 or f$_{AD}$/3	RW
VCUT	V$_{REF}$ connection bit	0 : No V$_{REF}$ connection[4] 1 : V$_{REF}$ connection	RW
OPA0	External op-amp connection mode bit[5]	b6 b7 0 0 : ANEX0 and ANEX1 are not used[6] 0 1 : ANEX0 input is A-D converted 1 0 : ANEX1 input is A-D converted 1 1 : External op-amp connection mode	RW
OPA1			RW

Notes

1. When the AD0CON1 register is rewritten during the A-D conversion, the conversion result is indeterminate.
2. This bit is disabled in one-shot mode and repeat mode. Pins in parentheses are commonly used for A-D conversion when setting the MD2 bit to "1" (repeat sweep mode 1).
3. ϕ_{AD} frequency must be under 10 MHz. Combinations of the CKS0 and CKS1 bits select ϕ_{AD} (see the AD0CON0 register).
4. Avoid setting the VCUT bit to "0" during A-D conversion. This is a reference voltage for the AD-0. Nothing is related to the D-A conversion.
5. In single sweep mode and repeat sweep mode 0 or 1, the OPA1 to OPA0 bits cannot be set "01$_2$" (ANEX0 input is A-D converted) or "10$_2$" (ANEX1 input is A-D converted).
6. When setting the OPA1 to OPA0 bits to "00$_2$", the PSL3_5 bit in PSL3 register should be set to "0" (other than ANEX0) and the PSL3_6 bit be set to "0" (other than ANEX1).

Figure 1.23.3. AD0CON1 Register

A-D0 control register 2[1]



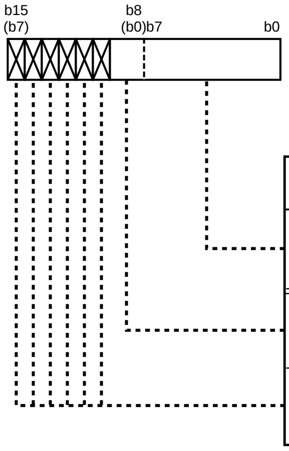
Symbol	Address	When reset
AD0CON2	0394$_{16}$	X000 0000$_2$

Bit Symbol	Bit name	Function	RW
SMP	A-D conversion method select bit	0 : Without sample and hold 1 : With sample and hold	RW
——	Reserved bit	Should set to "0"	RW
——			RW
——			RW
ADS	A-D channel replace select bit[2]	0 : Channel replace disable 1 : Channel replace enable[5]	RW
TRG0	External trigger request cause select bit	b6 b5 0 0 : Selects $\overline{\text{AD}_{\text{TRG}}}$ 0 1 : Selects timer B2 interrupt request of the three-phase motor control timer functions (after counting ICTB2 register) 1 0 : Selects intelligent I/O group 2 channel 1 interrupt 1 1 : Avoid this setting	RW
TRG1			RW
PST	Simultaneous start bit[2,3,4]	When setting this bit to "1", A-D0 and A-D1 starts a conversion simultaneously. When read, its content is indeterminate.	WO

Notes:

1. When the AD0CON2 register is rewritten during the A-D conversion, the conversion result is indeterminate.
2. Avoid setting this bit to "1" while either A-D0 or A-D1 is operating.
3. This bit is enabled when the TRG bit in the AD0CON0 register is set to "0" (software trigger). When the TRG bit is set to "1" (external trigger), avoid setting the PST bit to "1".
4. The A-D0 and A-D1 should be set to the same values.
5. When setting the the ADS bit to "1", avoid selecting single sweep mode or repeat sweep mode as an A-D operation mode.

A-D0 register i (i =0 to 7)



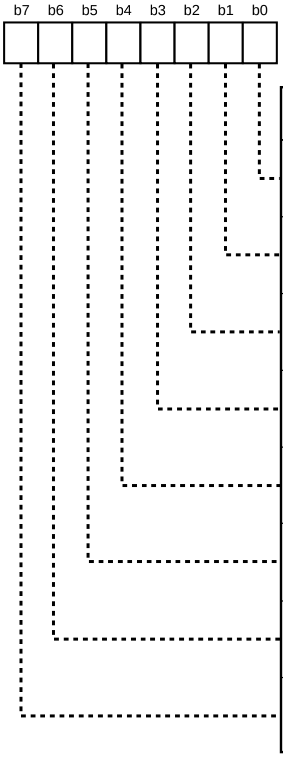
Symbol	Address	When reset
AD00 to AD02	0381$_{16}$ - 0380$_{16}$, 0383$_{16}$ - 0382$_{16}$, 0385$_{16}$ - 0384$_{16}$	Indeterminate
AD03 to AD05	0387$_{16}$ - 0386$_{16}$, 0389$_{16}$ - 0388$_{16}$, 038B$_{16}$ - 038A$_{16}$	Indeterminate
AD06 to AD07	038D$_{16}$ - 038C$_{16}$, 038F$_{16}$ - 038E$_{16}$	Indeterminate

Function	RW
8 low-order bits in an A-D conversion result	RO
In 10-bit mode : 2 high-order bits in an A-D conversion result In 8-bit mode : When read, its contents is indeterminate.	RO
Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.	—

Figure 1.23.4. AD0CON2 Register and AD00 to AD07 Registers

A-D1 control register 0[1]

Symbol	Address	When reset
AD1CON0	01D6$_{16}$	0000 0000$_2$



Bit symbol	Bit name	Function	RW
CH0	Analog input pin select bit[2,3,4,5,6]	b2 b1 b0 0 0 0 : ANi$_0$ 0 0 1 : ANi$_1$ 0 1 0 : ANi$_2$ 0 1 1 : ANi$_3$ 1 0 0 : ANi$_4$ 1 0 1 : ANi$_5$ 1 1 0 : ANi$_6$ 1 1 1 : ANi$_7$ (i=0, 2, 15)	RW
CH1			RW
CH2			RW
MD0	A-D operation mode select bit 0[2]	b4 b3 0 0 : One-shot mode 0 1 : Repeat mode 1 0 : Single sweep mode 1 1 : Repeat sweep mode 0 or 1	RW
MD1			RW
TRG	Trigger select bit	0 : Software trigger 1 : External trigger, hardware trigger[7]	RW
ADST	A-D conversion start flag	0 : A-D conversion stops 1 : A-D conversion starts[7]	RW
CKS0	Frequency select bit[8]	0 : Selectable from f$_{AD}$/3 or f$_{AD}$/4 1 : Selectable from f$_{AD}$/1 or f$_{AD}$/2	RW

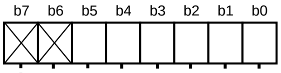
Notes:

1. When the AD1CON0 register is rewritten during the A-D conversion, the conversion result is indeterminate.
2. When changing an A-D operation mode, analog input pins should be set again.
3. This bit is disabled in single sweep mode, repeat sweep mode 0 and repeat sweep mode 1.
4. The APS1 to APS0 bit in the AD1CON2 register select i=0, 2 or 15.
5. i=0, 2 is available in single-chip mode only.
6. i=15 is available in the 144-pin package.
7. When setting the TRG bit to "1" (external trigger, hardware trigger), the TRG1 to TRG0 bits in the AD0CON2 register determine how to generate a trigger. Then the ADST bit should be set to "1" (A-D conversion started) after setting the TRG bit to "1".
8. ϕ_{AD} frequency must be under 10 MHz. Combinations of the CKS0 and CKS1 bits select ϕ_{AD}.

CKS0	CKS1	ϕ_{AD}
0	0	f$_{AD}$ divided by 4
0	1	f$_{AD}$ divided by 3
1	0	f$_{AD}$ divided by 2
1	1	f$_{AD}$

Figure 1.23.5. AD1CON0 Register

A-D1 control register 1[1]



Symbol	Address	When reset
AD1CON1	$01D7_{16}$	$XX00\ 0000_2$

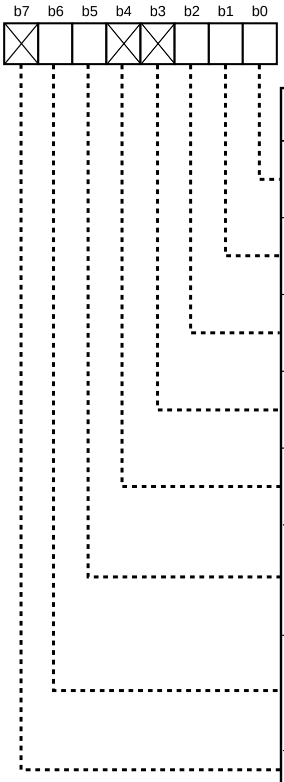
Bit symbol	Bit name	Function	RW
SCAN0	A-D sweep pin select bit[2,3,4]	b1 b0 0 0 : AN_{i0},AN_{i1} (AN_{i0}) 0 1 : AN_{i0} to AN_{i3} (AN_{i0},AN_{i1}) 1 0 : AN_{i0} to AN_{i5} (AN_{i0} to AN_{i2}) 1 1 : AN_{i0} to AN_{i7} (AN_{i0} to AN_{i3}) (i=0, 2, 15)	RW
SCAN1			RW
MD2	A-D operation mode select bit 1	0 : Any modes other than repeat sweep mode 1 1 : Repeat sweep mode 1	RW
BITS	8/10-bit mode select bit	0 : 8-bit mode 1 : 10-bit mode	RW
CKS1	Frequency select bit[5]	0 : Selectable from $f_{AD}/2$ or $f_{AD}/4$ 1 : Selectable from $f_{AD}/1$ or $f_{AD}/3$	RW
VCUT	VREF connection bit	0 : VREF not connection[6] 1 : VREF connection	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–

Notes:

1. When the AD1CON1 register is rewritten during the A-D conversion, the conversion result is indeterminate.
2. This bit is disabled in one-shot mode and repeat mode. Pins in parentheses are commonly used for the A-D conversion when setting the MD2 bit to "1" (repeat sweep mode 1).
3. The APS1 to APS0 bits in the AD1CON2 register select i=0, 2 or 15.
4. i=15 is available in the 144-pin package.
5. ϕ_{AD} frequency must be under 10 MHz. Combinations of the CKS0 and CKS1 bits select ϕ_{AD} (see the AD1CON0 register).
6. Avoid setting the VCUT bit to "0" during A-D conversion. This is a reference voltage for the AD-1. Nothing is related to the D-A conversion.

Figure 1.23.6. AD1CON1 Register

A-D1 control register 2[1]



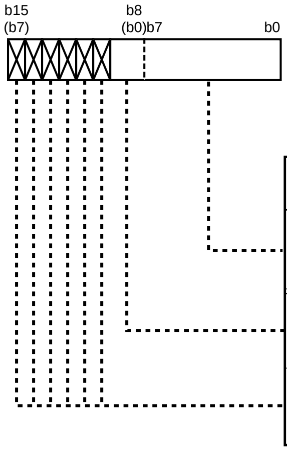
Symbol	Address	When reset
AD1CON2	01D4$_{16}$	X00X X000$_2$

Bit Symbol	Bit name	Bit name	RW
SMP	A-D conversion method select bit	0 : Without sample and hold 1 : With sample and hold	RW
APS0	Analog input port select bit	b2 b1 0 0 : AN15$_0$ to AN15$_7$[2] 0 1 : Avoid this setting 1 0 : AN0$_0$ to AN0$_7$[3] 1 1 : AN2$_0$ to AN2$_7$[3]	RW
APS1			RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–
——			–
TRG0	External trigger request cause select bit	b6 b5 0 0 : Selects $\overline{\text{AD}}_{\text{TRG}}$ 0 1 : Selects timer B2 interrupt request of the three-phase motor control timer functions (after counting ICTB2 register) 1 0 : Intelligent I/O group 3 channel 1 interrupt is selected 1 1 : Avoid this setting	RW
TRG1			RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		–

Notes:

1. When the AD1CON2 register is rewritten during the A-D conversion, the conversion result is indeterminate.
2. AN15$_0$ to AN15$_7$ are available in the 144-pin package.
3. AN00 to AN07, AN20 to AN27 are available in single-chip mode only.

A-D1 register j (j=0 to 7)



Symbol	Address	When reset
AD10 to AD12	01C1$_{16}$ - 01C0$_{16}$, 01C3$_{16}$ - 01C2$_{16}$, 01C5$_{16}$ - 01C4$_{16}$	Indeterminate
AD13 to AD15	01C7$_{16}$ - 01C6$_{16}$, 01C9$_{16}$ - 01C8$_{16}$, 01CB$_{16}$ - 01CA$_{16}$	Indeterminate
AD16 to AD17	01CD$_{16}$ - 01CC$_{16}$, 01CF$_{16}$ - 01CE$_{16}$	Indeterminate

Function	RW
8 low-order bits in an A-D conversion result	RO
In 10-bit mode : 2 high-order bits in an A-D conversion result In 8-bit mode : When read, its content is indeterminate	RO
Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.	–

Figure 1.23.7. AD1CON2 Register and AD10 to AD17 Registers

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) • The ADST bit in the ADiCON0 register is set to "1" (A-D conversion starts) by program • The PST bit in the AD0CON2 register is set to "1" (A-D0 and A-D1 start the A-D conversion simultaneously) by program (b) When the TRG bit is set to "1" (external trigger, hardware trigger) • The $\overline{AD_{TRG}}$ input pin changes "L" to "H" after the ADST bit is set to "1" by program • One of the following interrupt is generated after the ADST bit is set to "1" by program - Timer B2 interrupt request of three-phase motor control timer functions after the ICTB2 register completes counting - Intelligent I/O interrupt request Group2 channel 1(A-D0), group3 channel 1 (A-D1)
Stop condition	• A-D conversion is completed (When internal trigger is selected, the ADST bit is set to "0") • The ADST bit is set to "0" (A-D conversion stops) by program
Interrupt request generation timing	A-D conversion is completed
Input pin	Selectable from AN_0 to AN_7, ANEX0 or ANEX1 Selectable from AN_{j0} to AN_{j7} (j = 0, 2, 15)
Reading of A-D converter result	Read the ADik 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_0 to AN_7, ANEX0 or ANEX1 Selectable from AN_{j0} to AN_{j7} (j = 0, 2, 15), ANEX0, ANEX1
Reading of A-D converter result	Read the ADik registers (k = 0 to 7) corresponding to a selected pin

(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	• A-D conversion is completed (When internal trigger is selected, the ADST bit is set to "0") • The ADST bit is set to "0" (A-D conversion stops) by program
Interrupt request generation timing	Sweeping is completed
Input pin	Selectable from AN_0 to AN_1 (2 pins), AN_0 to AN_3 (4 pins), AN_0 to AN_5 (6 pins), or AN_0 to AN_7 (8 pins) Selectable from ANj_0 to ANj_1 (2 pins), ANj_0 to ANj_3 (4 pins), ANj_0 to ANj_5 (6 pins), or ANj_0 to ANj_7 (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 AN_0 to AN_1 (2 pins), AN_0 to AN_3 (4 pins), AN_0 to AN_5 (6 pins), or AN_0 to AN_7 (8 pins) Selectable from ANj_0 to ANj_1 (2 pins), ANj_0 to ANj_3 (4 pins), ANj_0 to ANj_5 (6 pins), or ANj_0 to ANj_7 (8 pins) (j = 0, 2, 15)
Reading of A-D converter result	Read the ADik registers (k = 0 to 7) corresponding to selected pins

(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 ANj$_0$ is selected (j =none, 0, 2, 15), analog voltage is converted to a digital form as the following order : ANj$_0$ → ANj$_1$ → ANj$_0$ → ANj$_2$ → ANj$_0$ → ANj$_3$ .... 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	ANj$_0$ to ANj$_7$
Pins to be put emphasis	Selectable from AN$_0$ to AN$_1$ (2 pins), AN$_0$ to AN$_3$ (4 pins), AN$_0$ to AN$_5$ (6 pins), or AN$_0$ to AN$_7$ (8 pins) Selectable from ANj$_0$ to ANj$_1$ (2 pins), ANj$_0$ to ANj$_3$ (4 pins), ANj$_0$ to ANj$_5$ (6 pins), or ANj$_0$ to ANj$_7$ (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 Ø$_{AD}$ cycles are achieved with an 8-bit resolution and 33 Ø$_{AD}$ 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.

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[1]	TRG1 to TRG0 = 00_2	Falling edge of the $\overline{\text{ADTRG}}$ input signal
	TRG1 to TRG0 = 01_2	Timer B2 interrupt request of three-phase motor control timer functions (after the ICTB2 register completes counting)
	TRG1 to TRG0 = 10_2	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.

ANj (j = 0 to 7) input is converted to a digital form in the A-D1 and the conversion result is stored into the AD1j register. AN0j, AN2j and AN15j inputs are converted to digital forms in the A-D0 and the conversion results are stored into the AD0j 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_2" (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.

(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 AD0CON1 register are set to "11_2" (external op-amp connection), voltage input to the AN_0 to AN_7 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 ope-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 $P9_5$ (AN_0)	Not used
1	0	Not used	Analog input to $P9_6$ (AN_1)
1	1	Output to external op-amp	Input from external op-amp

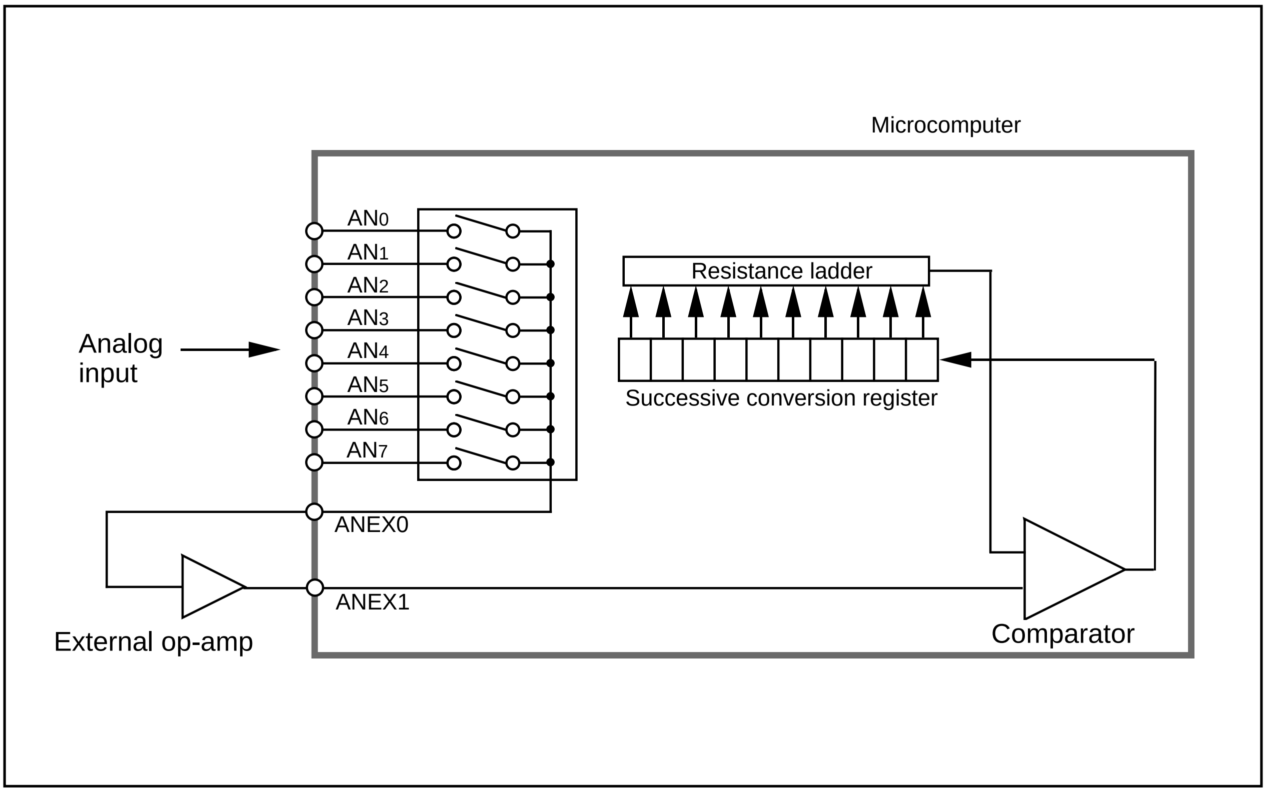


Figure 1.23.8. External Ope-amp Connection

(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 affects 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.

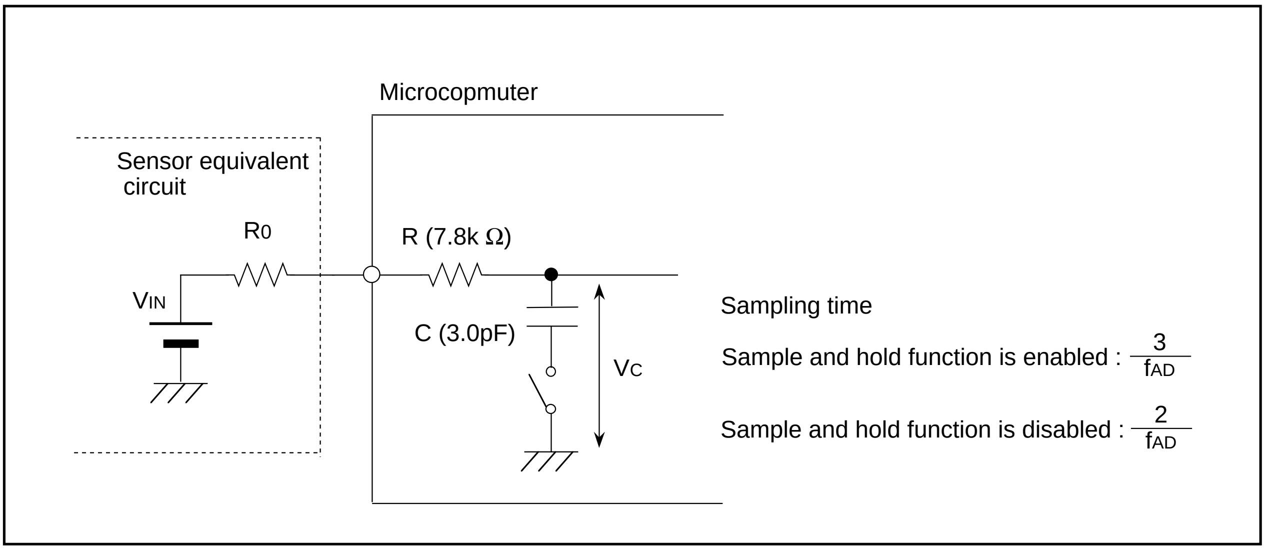

Microcopmuter
Sensor equivalent circuit
R0
R (7.8k Ω)
VIN
C (3.0pF)
VC
Sampling time
Sample and hold function is enabled : $\frac{3}{f_{AD}}$
Sample and hold function is disabled : $\frac{2}{f_{AD}}$

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 ADij register (i=0,1, j = 0 to 7) when the A-D conversion result is stored into the ADij register, wrong A-D conversion value is stored into the ADij register. This event occur 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 ADij register. (The IR bit in the ADiIC 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.

(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 ADij register (j=0 to 7), which does not perform the A-Di conversion, may also be indeterminate.

When A-Di is forcibly terminated, avoid using value of all ADij 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 ADi as a survivor normally completes the A-D conversion. Values of the ADij registers that does not perform the A-D conversion remain unchanged.

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 DAi registers (i=0,1). The DAiE bit in the DACON register should be set to output a D-A conversion result. The DAiE 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.

$$V = \frac{V_{REF} \text{ x } n}{256} \quad (n = 0 \text{ to } 255)$$

V_{REF} : 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 DAi register is set to "00_{16}" and the DAiE 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[1]	PS3 register[1]	PSL3 register
$P9_3$	DA_0 output	PD9_3=0	PS3_3=0	PSL3_3=1
$P9_4$	DA_1 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.

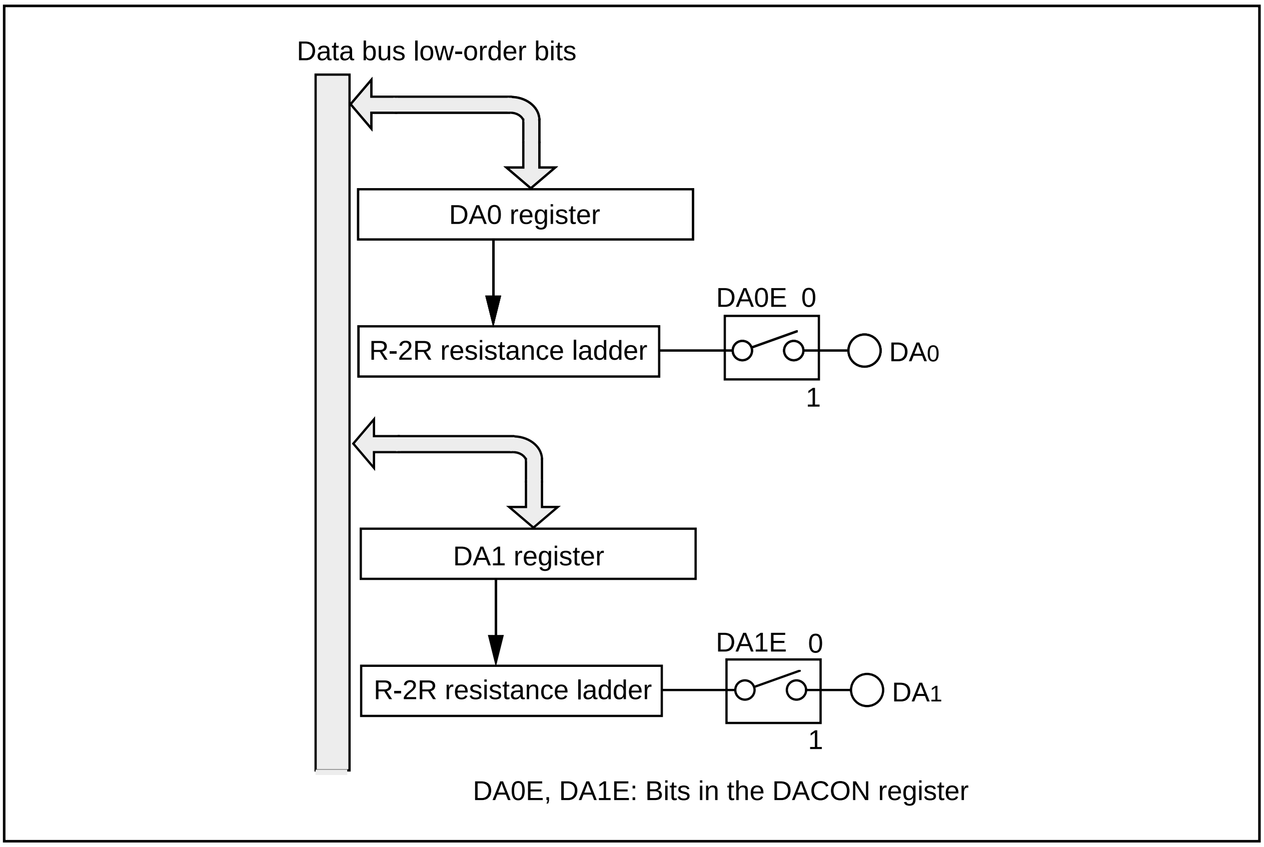


Figure 1.24.1. D-A Converter

D-A control register

b7 b6 b5 b4 b3 b2 b1 b0

Symbol	Address	When reset
DACON	039C$_{16}$	XXXX XX00$_2$

Bit symbol	Bit name	Function	RW
DA0E	D-A0 output enable bit	0 : Output disabled 1 : Output enabled	RW
DA1E	D-A1 output enable bit	0 : Output disabled 1 : Output enabled	RW
——		Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.	—
——			—
——			—
——			—
——			—
——			—

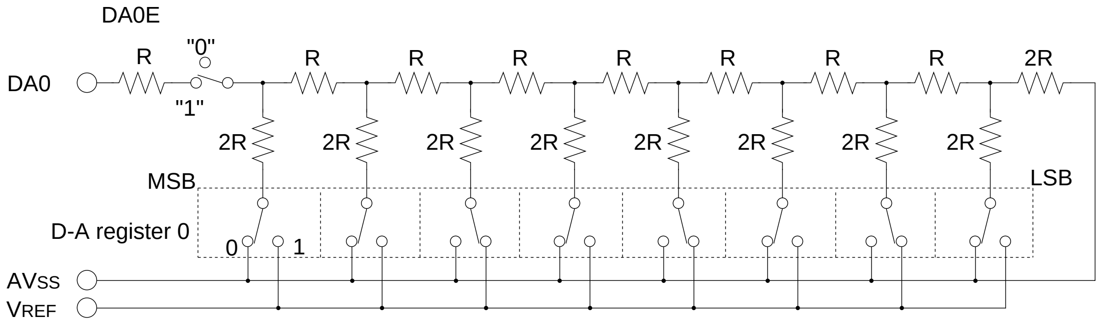
D-A register i (i=0, 1)

b7 b0

Symbol	Address	When reset
DA0, DA1	0398$_{16}$, 039A$_{16}$	Indeterminate

Function	Setting range	RW
Output value of the D-A conversion	00$_{16}$ to FF$_{16}$	RW

Figure 1.24.2. DACON Register, DA0 and DA1 Registers



Notes:
1. The above applies to a condition that the D-A register is set to "2A$_{16}$".
2. This circuit is the same as in the D-A1.
3. To save power dissipationwhen the D-A converter is not used, the DAiE bit (i=0, 1) should be set to "0" (output inhibited) and the DAi register be set to "00$_{16}$" to prevent current flowing to the R-2R resistance.
4. It is not related to the VCUT bit in the AD0CON1 and AD1CON1 registers.

Figure 1.24.3. D-A converter Equivalent Circuit

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.

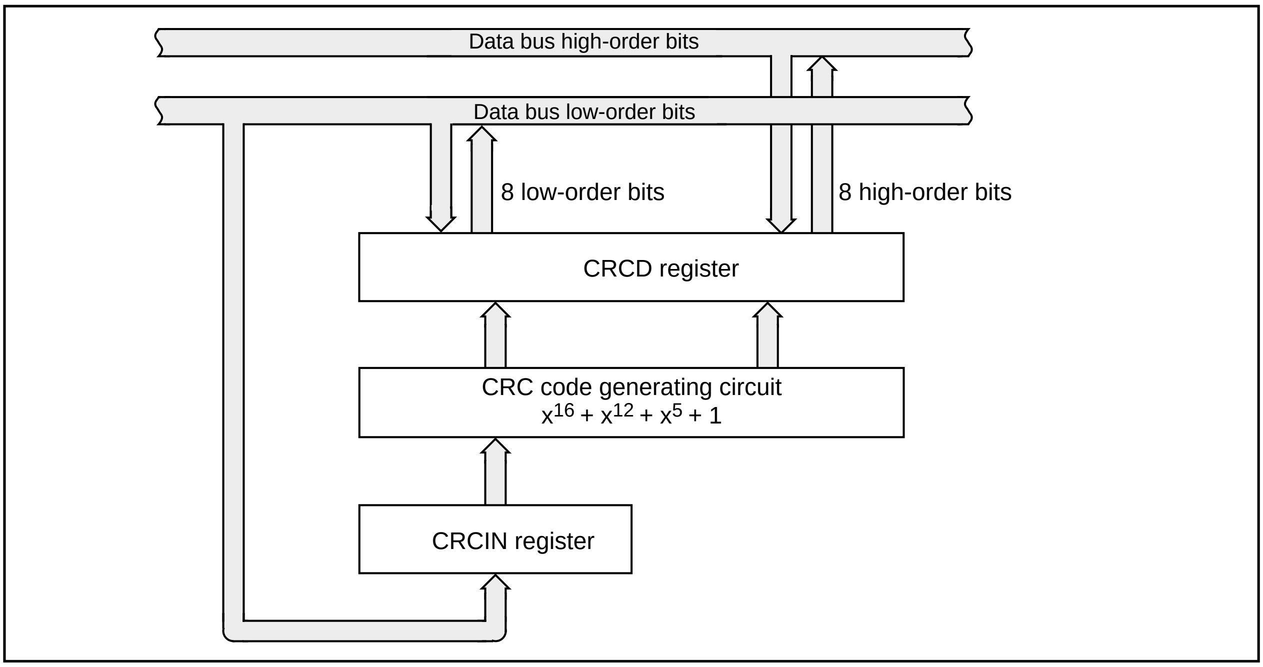
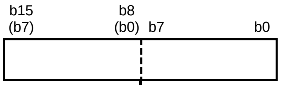


Figure 1.25.1. CRC Calculation Block Diagram

CRC data register



Symbol	Address	When reset
CRCD	$037D_{16}$- $037C_{16}$	Indeterminate

Function	Setting range	RW
When writing data to the CRCIN register after writing a default value to the CRCD register, the CRC code can be read from the CRCD register. A bit position of a default value is inversed. The value inversed a bit position can be read as the CRC code.	0000_{16} to $FFFF_{16}$	RW

CRC input register

Symbol	Address	When reset
CRCIN	$037E_{16}$	Indeterminate

Function	Setting range	RW
Data input. Inverse a bit position of data.	00_{16} to FF_{16}	RW

Figure 1.25.2. CRCD Register and CRCIN Register

CRC calculation and setup procedure to generate CRC code for "$80C4_{16}$"

○ CRC calculation for M16C

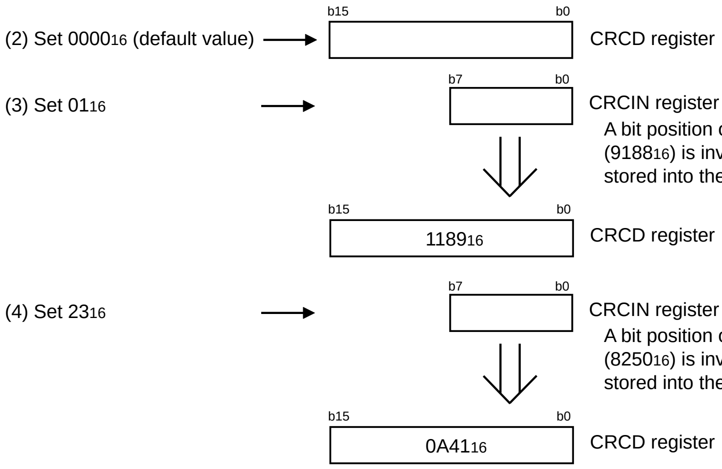
CRC code : reminder of a division, $\frac{\text{value that is inversed a bit position of a value written in CRCIN register}}{\text{generation polynomial}}$

Generation polynomial : $X^{16} + X^{12} + X^5 + 1$ ($1\ 0001\ 0000\ 0010\ 0001_2$)

○ Setting steps

(1) Inverse a bit position of "$80C4_{16}$" in byte units by program

"80_{16}" → "01_{16}", "$C4_{16}$" → "23_{16}"



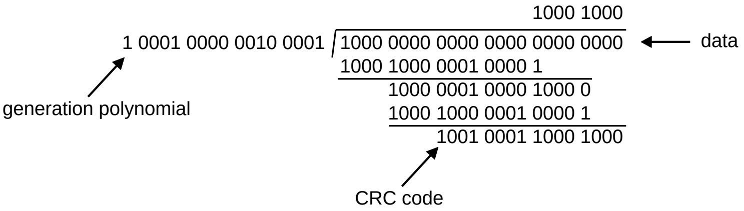
A bit position of the CRC code for "80_{16}" (9188_{16}) is inversed to "1189_{16}", which is stored into the CRCD register after 2 cycles.

A bit position of the CRC code for "$80C4_{16}$" (8250_{16}) is inversed to "$0A41_{16}$", which is stored into the CRCD register after 2 cycles.

○ Details of CRC calculation

As shown in (3) above, a bit position of a value set in the CRCIN register "01_{16}" (00000001_2) is inversed to become "10000000_2".

Add "$1000\ 0000\ 0000\ 0000\ 0000\ 0000_2$", as "$10000000_2$" plus 16 digits, to "$0000_{16}$" 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

"$0001\ 0001\ 1000\ 1001_2$", inversed a bit position of "$1001\ 0001\ 1000\ 1000_2$ (9188_{16})" as a remainder, can be read from the CRCD register.

When going on to (4) above, "23_{16} (00100011_2)" written in the CRCIN register is inversed to become "11000100_2".

Add "$1100\ 0100\ 0000\ 0000\ 0000\ 0000_2$", as "$11000100_2$" plus 16 digits, to "$1001\ 0001\ 1000\ 1000_2$" as a remainder of (3) left in the CRCD register to perform the modulo-2 division.

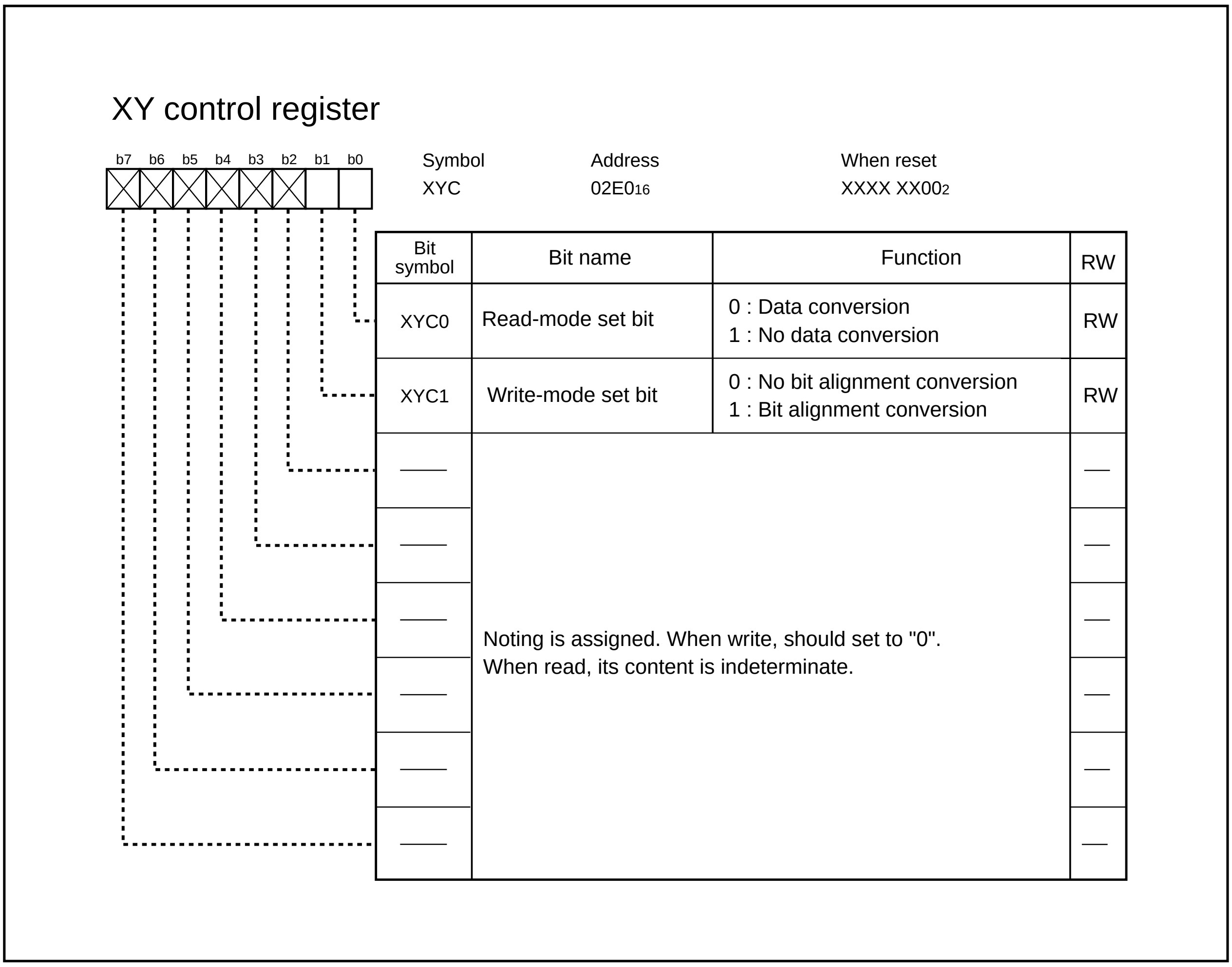
"$0000\ 1010\ 0100\ 0001_2$ ($0A41_{16}$)", inversed a bit position of the remainder, can be read from CRCD register.

Figure 1.25.3. CRC Calculation

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.



Symbol	Address	When reset
XYC	02E0$_{16}$	XXXX XX00$_2$

Bit symbol	Bit name	Function	RW
XYC0	Read-mode set bit	0 : Data conversion 1 : No data conversion	RW
XYC1	Write-mode set bit	0 : No bit alignment conversion 1 : Bit alignment conversion	RW
——	Noting is assigned. When write, should set to "0". When read, its content is indeterminate.		—
——			—
——			—
——			—
——			—
——			—

Figure 1.26.1. XYC Register

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.

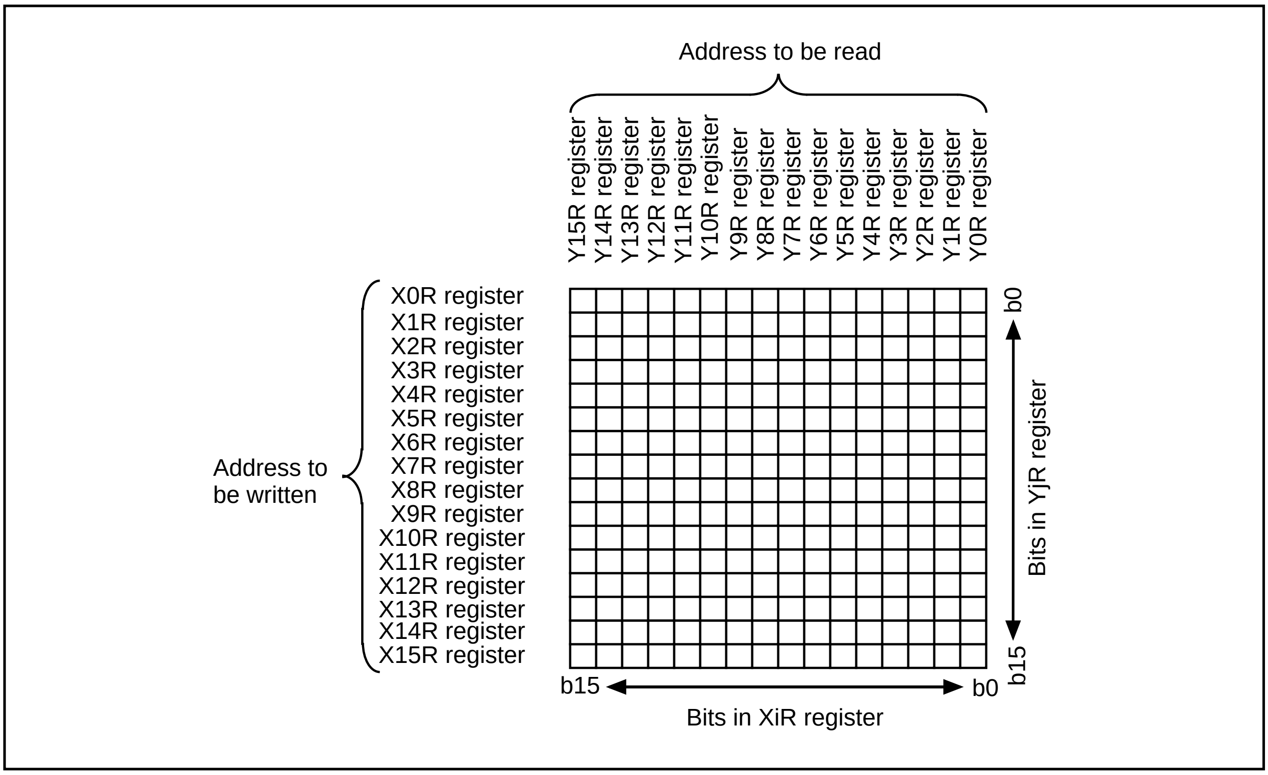


Figure 1.26.2. Conversion Table when Setting the XYC0 Bit to "0"

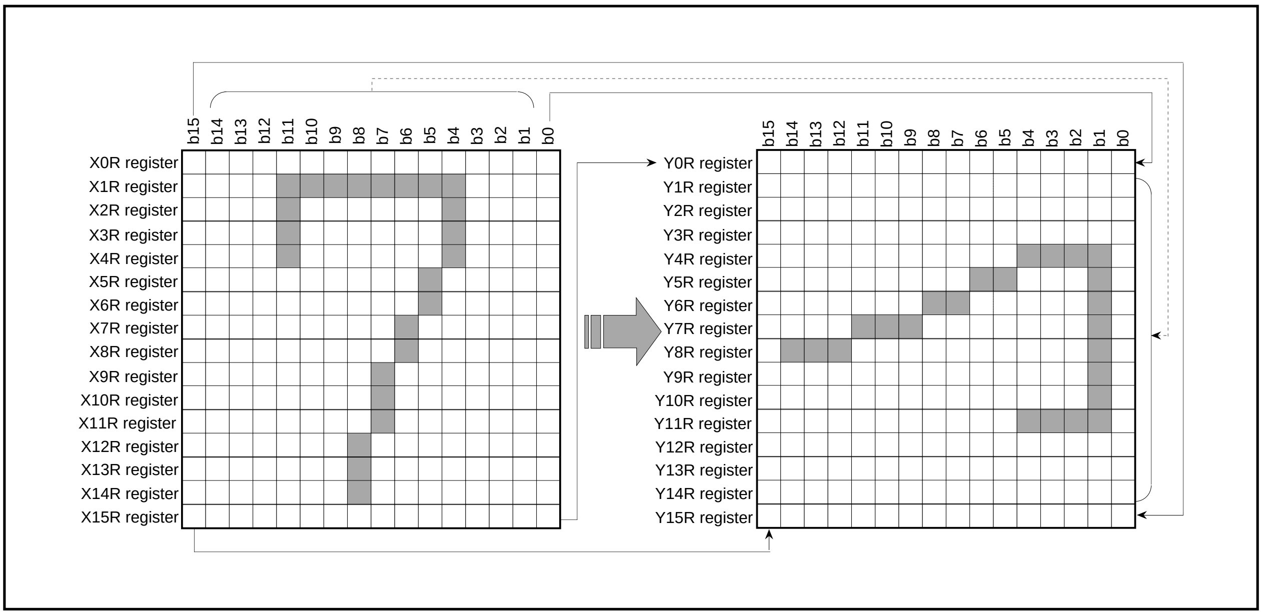


Figure 1.26.3. X-Y Conversion

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."

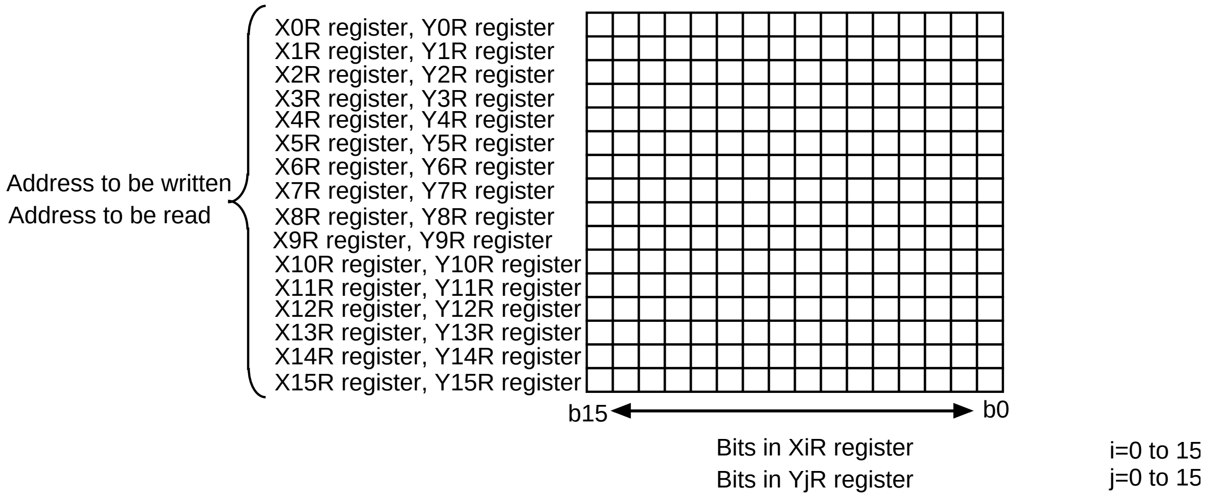

Address to be written
Address to be read
X0R register, Y0R register
X1R register, Y1R register
X2R register, Y2R register
X3R register, Y3R register
X4R register, Y4R register
X5R register, Y5R register
X6R register, Y6R register
X7R register, Y7R register
X8R register, Y8R register
X9R register, Y9R register
X10R register, Y10R register
X11R register, Y11R register
X12R register, Y12R register
X13R register, Y13R register
X14R register, Y14R register
X15R register, Y15R register
b15
b0
Bits in XiR register
Bits in YjR register
i=0 to 15
j=0 to 15

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 inversed to write.

Figure 1.26.5 shows a conversion table when setting the XYC1 bit to "1".

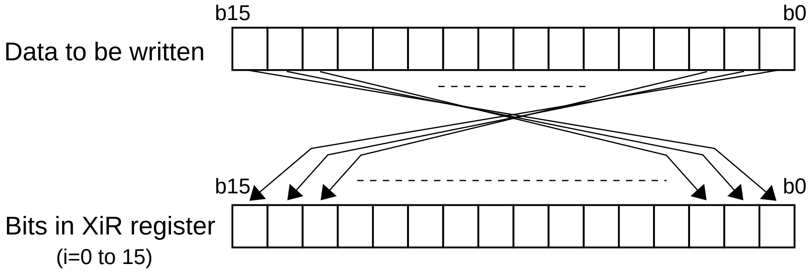


Figure 1.26.5. Conversion Table when Setting the XYC1 Bit to "1"

DRAMC

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	$\overline{\mathrm{CAS}}$ before $\overline{\mathrm{RAS}}$ 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 $\overline{\mathrm{RAS}}$, $\overline{\mathrm{CASL}}$, $\overline{\mathrm{CASH}}$ and $\overline{\mathrm{DW}}$ signal operations. Figure 1.27.1 shows the DRAMCONT register and REFCNT register.

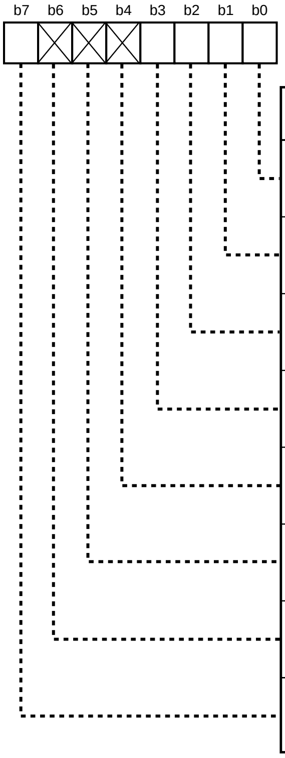
Table 1.27.2. DRAMC-associated Pins

Port	Bus for device access except DRAM[1]	Bus for DRAM access
P0	D_0 to D_7	D_0 to D_7
P1	D_8 to D_{15}	D_8 to D_{15}[2]
P3	A_8 to D_{15}	MA_0 to MA_7
$P4_0$ to $P4_4$	A_{16} to D_0	MA_8 to MA_{12}
$P5_0$	$\overline{\mathrm{WRL}}$ / $\overline{\mathrm{WR}}$	$\overline{\mathrm{CASL}}$
$P5_1$	$\overline{\mathrm{WRH}}$ / $\overline{\mathrm{BHE}}$	$\overline{\mathrm{CASH}}$
$P5_2$	$\overline{\mathrm{RD}}$	$\overline{\mathrm{DW}}$
$P5_6$	ALE	$\overline{\mathrm{RAS}}$

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" ($\overline{\mathrm{RD}}$/$\overline{\mathrm{WRL}}$/$\overline{\mathrm{WRH}}$ in R/W mode).

DRAM Control register



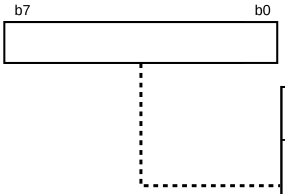
Symbol	Address	When reset
DRAMCONT	0040_{16}	Indeterminate[1]

Bit symbol	Bit name	Function	RW
WT	Wait select bit[2]	0 : 2 waits 1 : 1 wait	RW
AR0	DRAM space select bit	b3 b2 b1 0 0 0 : DRAM disabled 0 0 1 : Avoid this setting 0 1 0 : 0.5MB 0 1 1 : 1MB 1 0 0 : 2MB 1 0 1 : 4MB 1 1 0 : 8MB 1 1 1 : Avoid this setting	RW
AR1			RW
AR2			RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
——			—
——			—
SREF	Self-refresh mode bit[3]	0 : Self-refresh OFF 1 : Self-refresh ON	RW

Notes:

1. After reset, content of this register is indeterminate. The DRAM starts operating when writing data to this register.
2. The number of cycles with two waits is 3-2-2. With one wait, it is 2-1-1.
3. Refer to the paragraph "2. Self-refresh" about a setup procedure for the SREF bit.
   When setting the SREF bit to "1", both $\overline{RAS}$ and $\overline{CAS}$ output "L". When attaching external devices except DRAM, the $\overline{WR}$ signal outputs "L".
4. The DS register determines how long a data bus is . With a 8-bit bus, $\overline{CASH}$ is indeterminate.

DRAM refresh interval set register



Symbol	Address	When reset
REFCNT	0041_{16}	Indeterminate

Function	Setting range	RW
As the setting value is n, Refresh interval = CPU clock cycle X (n + 1) X 32	00_{16} to FF_{16}	RW

Figure 1.27.1 DRAMCONT Register and REFCNT Register

DRAMC is unavailable when the PM11 to PM10 bits in the PM1 register is set to "11_2" (mode 3). The PM11 to PM10 bits should be set to "00_2," "01_2" or "10_2" (mode 0 to 2). For a 16-bit DRAM data bus, the PM02 bit in the PM0 register should be set to "1" ($\overline{\text{RD}}$/$\overline{\text{WRH}}$/$\overline{\text{WRL}}$).
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 A_8 to A_{20}. Figure 1.27.2 shows an output format for multiplexed addresses.

Refresh

1. Refresh

Refresh method is $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ 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:

Value of the REFCNT register (00_{16} to FF_{16}) = refresh interval time / (CPU clock frequency X 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.

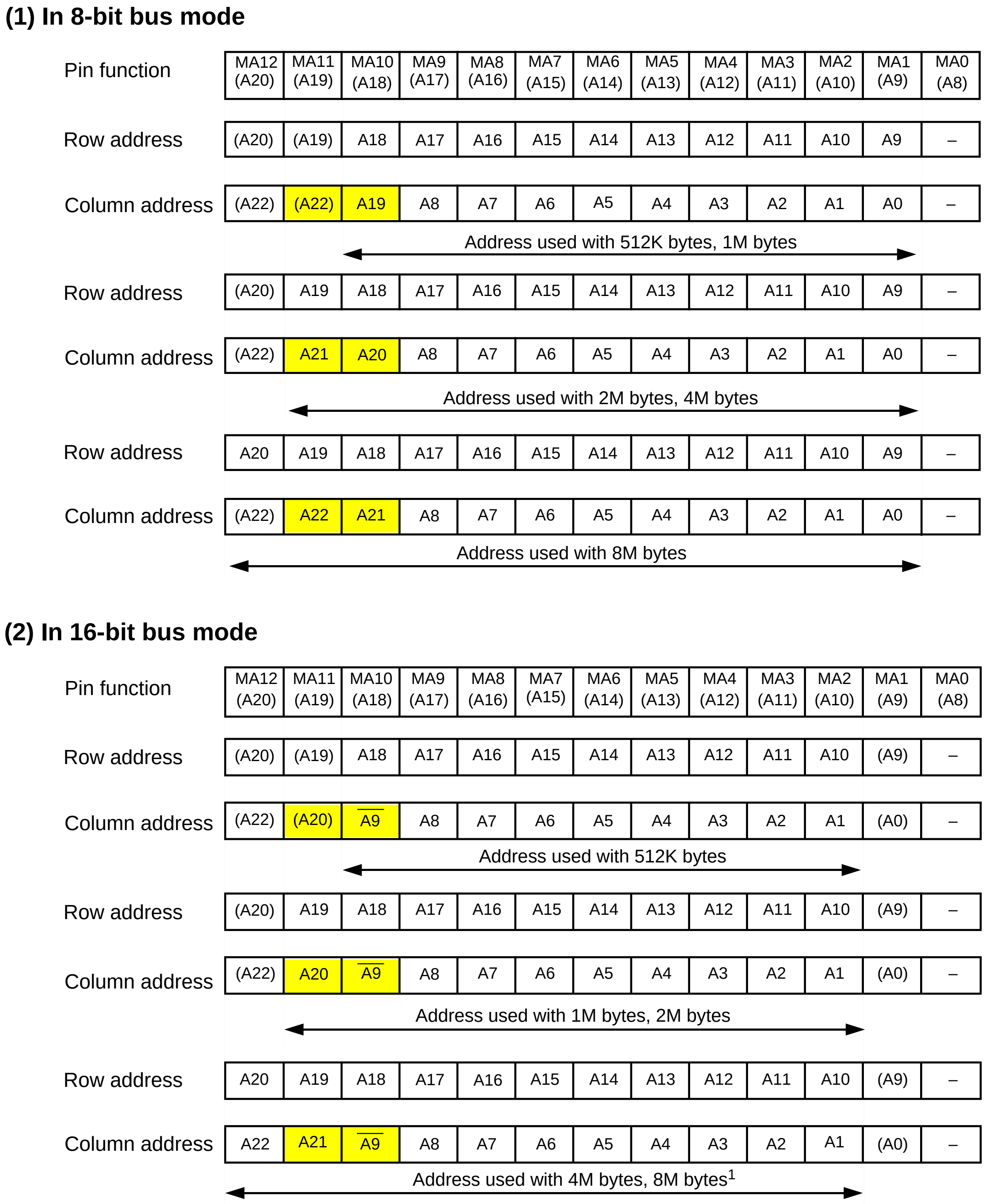
(1) setting for the self-refresh (1 wait, 4M bytes)

```
•••
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 $\overline{\text{RAS}}$ and $\overline{\text{CAS}}$ are set to "L" with the self-refresh. When other devices except the DRAM are attached and the $\overline{\text{WR}}$ signal is set to "L", any processings like setting the $\overline{\text{CS}}$ to "H" should be done.
Figures 1.27.3 to 1.27.5 show a bus timing during a DRAM access.



(1) In 8-bit bus mode

Pin function	MA12 (A20)	MA11 (A19)	MA10 (A18)	MA9 (A17)	MA8 (A16)	MA7 (A15)	MA6 (A14)	MA5 (A13)	MA4 (A12)	MA3 (A11)	MA2 (A10)	MA1 (A9)	MA0 (A8)
Row address	(A20)	(A19)	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	–
Column address	(A22)	(A22)	A19	A8	A7	A6	A5	A4	A3	A2	A1	A0	–

Address used with 512K bytes, 1M bytes

Row address	(A20)	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	–
Column address	(A22)	A21	A20	A8	A7	A6	A5	A4	A3	A2	A1	A0	–

Address used with 2M bytes, 4M bytes

Row address	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	–
Column address	(A22)	A22	A21	A8	A7	A6	A5	A4	A3	A2	A1	A0	–

Address used with 8M bytes

(2) In 16-bit bus mode

Pin function	MA12 (A20)	MA11 (A19)	MA10 (A18)	MA9 (A17)	MA8 (A16)	MA7 (A15)	MA6 (A14)	MA5 (A13)	MA4 (A12)	MA3 (A11)	MA2 (A10)	MA1 (A9)	MA0 (A8)
Row address	(A20)	(A19)	A18	A17	A16	A15	A14	A13	A12	A11	A10	(A9)	–
Column address	(A22)	(A20)	$\overline{\text{A9}}$	A8	A7	A6	A5	A4	A3	A2	A1	(A0)	–

Address used with 512K bytes

Row address	(A20)	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	(A9)	–
Column address	(A22)	A20	$\overline{\text{A9}}$	A8	A7	A6	A5	A4	A3	A2	A1	(A0)	–

Address used with 1M bytes, 2M bytes

Row address	A20	A19	A18	A17	A16	A15	A14	A13	A12	A11	A10	(A9)	–
Column address	A22	A21	$\overline{\text{A9}}$	A8	A7	A6	A5	A4	A3	A2	A1	(A0)	–

Address used with 4M bytes, 8M bytes[1]

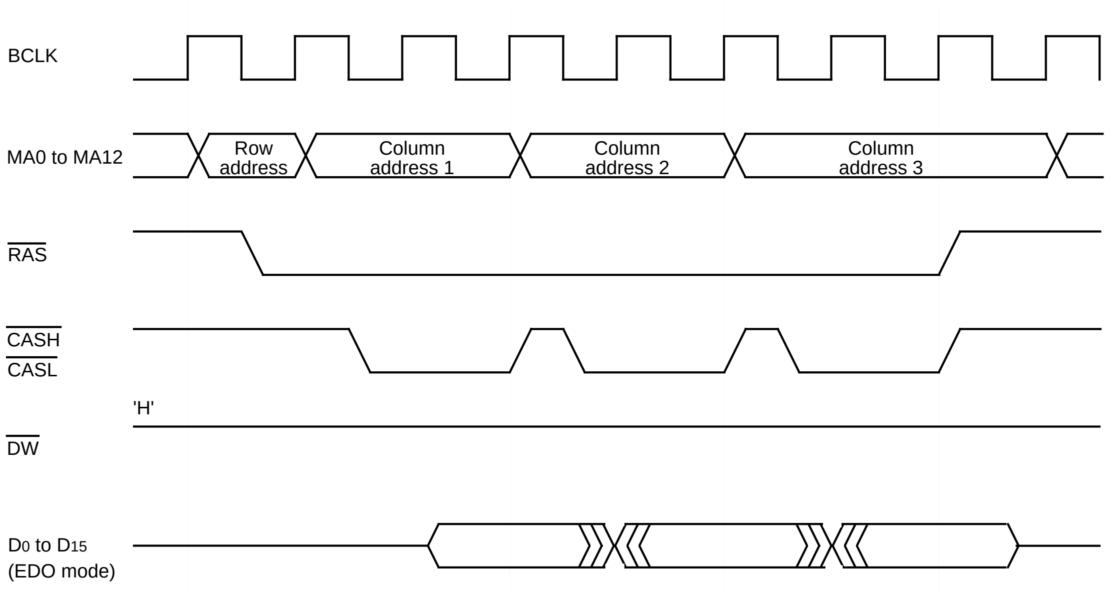
( ): bits disabled, 🟨 bits that change according to a data bus width and DRAM space
–: indeterminate

Notes:

1. The above applies to a 4Mx1 or 4Mx4 memory configuration. With a 4Mx16 configuration, use the following combinations should be used :
For row addresses, MA0 to MA12; for column addresses MA2 to MA8, MA11 and MA12.
Or for row addresses MA1 to MA12; for column addresses MA2 to MA9, MA11, MA12.

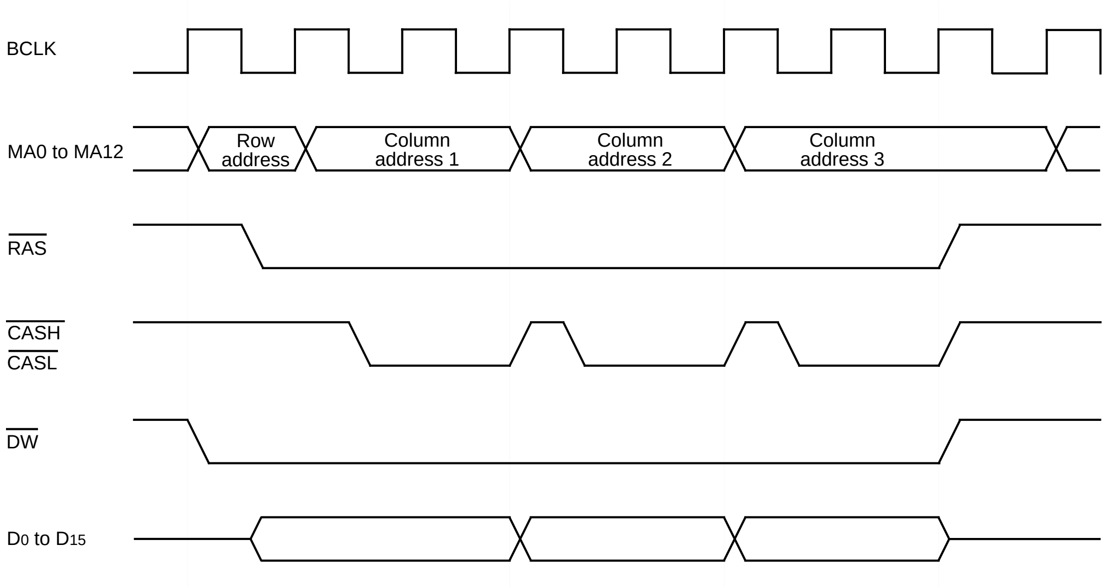
Figure 1.27.2. Address Multiplexed Output Pattern

(1) Read cycle (WT bit = 0 with 2 waits)



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.

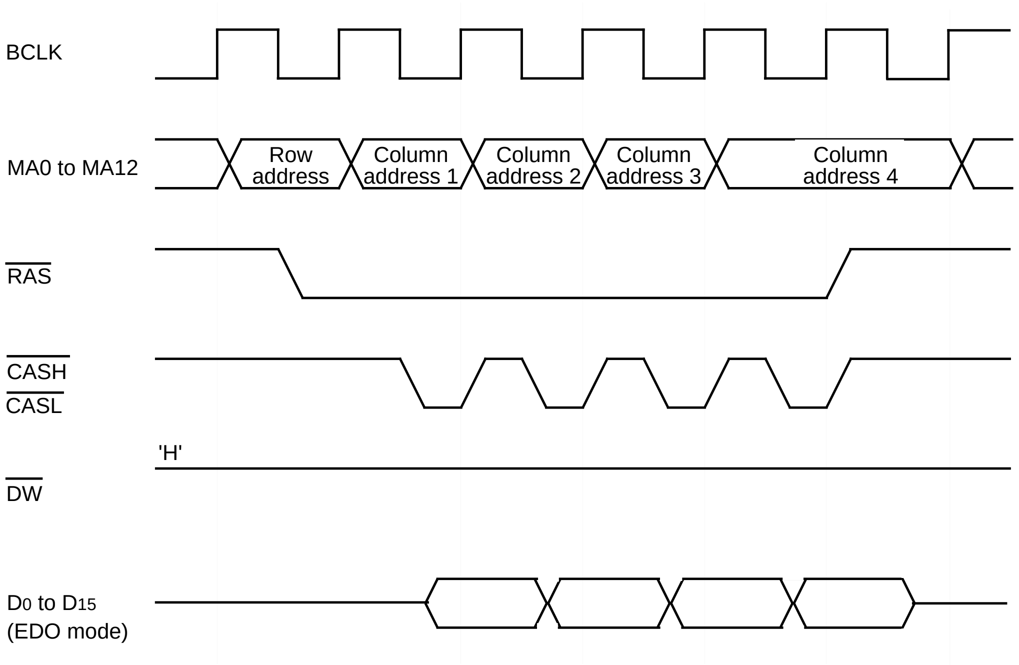
(2) Write cycle (WT bit = 0)



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.

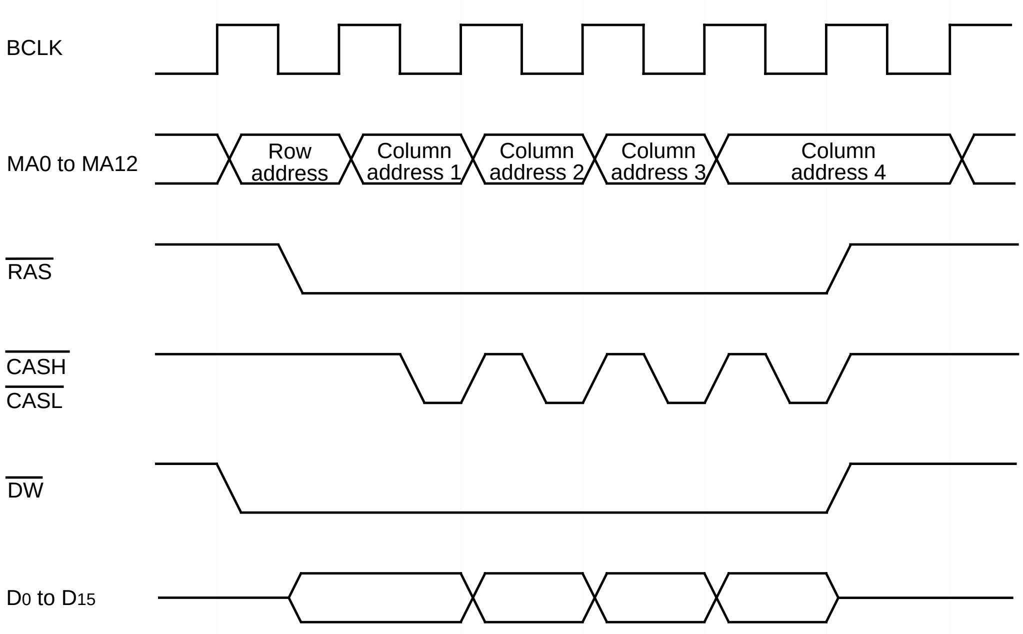
Figure 1.27.3. Bus Timing during DRAM Access (1)

(1) Read cycle (WT bit = 1 with 1 wait)



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.

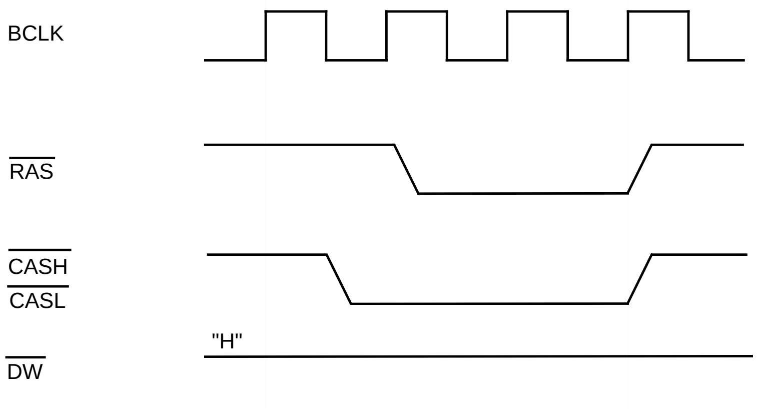
(2) Write cycle (WT bit = 1)



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 1.27.4. Bus Timing during DRAM Access (2)

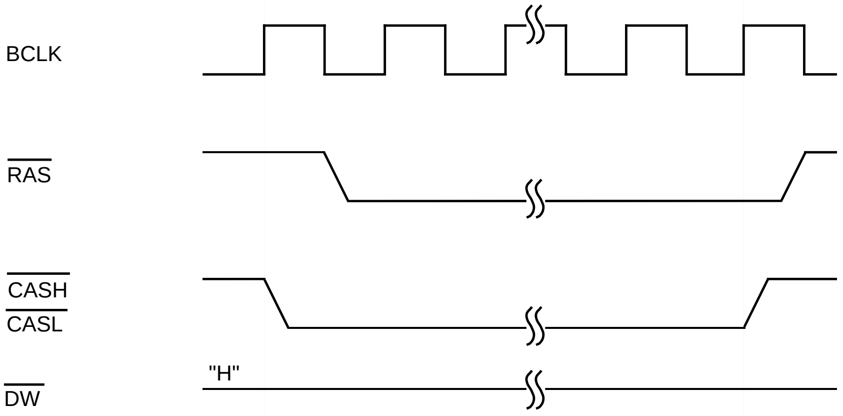
(1) $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh cycle



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 P8$_5$) are available in the 100-pin package and 123 programmable I/O ports from P0 and to P15 (excluding P8$_5$) 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. P8$_5$ 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 P8$_5$ 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 A$_{22}$, $\overline{\text{A}_{23}}$, D$_0$ to D$_{15}$, MA$_0$ to MA$_{12}$, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{WRL}}$/$\overline{\text{WR}}$/$\overline{\text{CASL}}$, $\overline{\text{WRH}}$/$\overline{\text{BHE}}$/$\overline{\text{CASH}}$, $\overline{\text{RD}}$/$\overline{\text{DW}}$, BCLK/ALE/CLK$_{\text{OUT}}$, $\overline{\text{HLDA}}$/ALE, $\overline{\text{HOLD}}$, ALE/$\overline{\text{RAS}}$, and $\overline{\text{RDY}}$) in the PDi register cannot be modified. No bit in the direction register for P8$_5$ 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 (A$_0$ to A$_{22}$, $\overline{\text{A}_{23}}$, D$_0$ to D$_{15}$, MA$_0$ to MA$_{12}$, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{WRL}}$/$\overline{\text{WR}}$/$\overline{\text{CASL}}$, $\overline{\text{WRH}}$/$\overline{\text{BHE}}$/$\overline{\text{CASH}}$, $\overline{\text{RD}}$/$\overline{\text{DW}}$, BCLK/ALE/CLK$_{\text{OUT}}$, $\overline{\text{HLDA}}$/ALE, $\overline{\text{HOLD}}$, ALE/$\overline{\text{RAS}}$, and $\overline{\text{RDY}}$) 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 PSi 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, $P4_0$ to $P4_3$ 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.

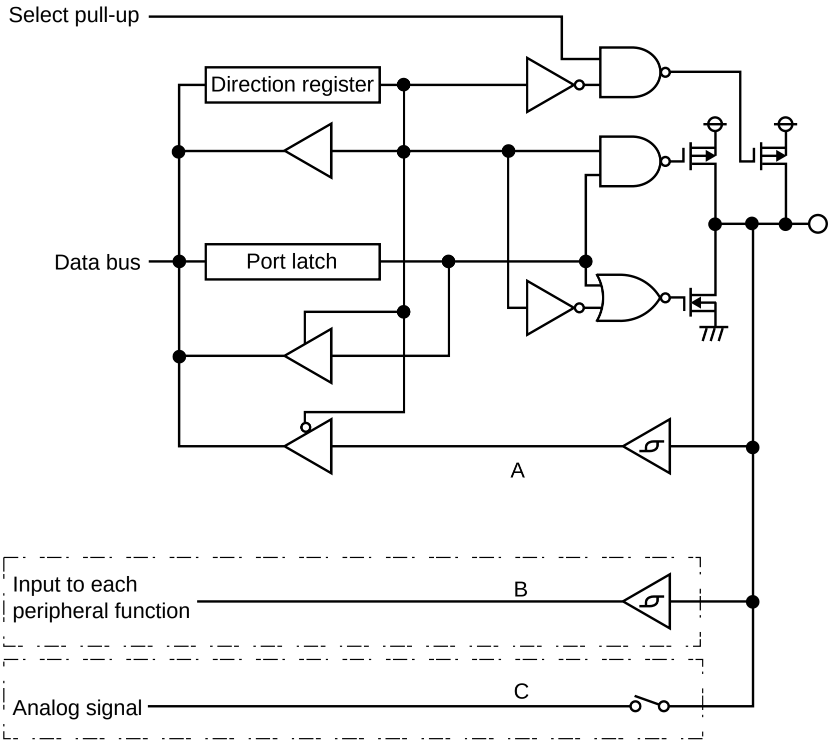
Refer to the paragraph "Analog input and other peripheral function input" about the IPS2 bit.

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, AN_4 to AN_7, $AN15_0$ to $AN15_7$), 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 $\overline{KI_0}$ to $\overline{KI_3}$ pin input levels change.

Programmable I/O ports



Option / Port	(A) Hysteresis	Circuit (B) Peripheral function input	Circuit (C) Analog I/F
$P0_0$ to $P0_7$ $P2_0$ to $P2_7$	–	–	○
$P3_0$ to $P3_7$ $P4_0$ to $P4_7$ $P5_0$ to $P5_2$ $P5_4$	–	–	–
$P5_5$	–	○	–
$P5_6$	–	–	–
$P5_7$	–	○	–
$P8_3$, $P8_4$	○	○	–
$P8_6$	–	–	–
$P8_7$	–	○	○
$P10_0$ to $P10_3$	–	–	○
$P10_4$ to $P10_7$	○	○	○
(Note 1) $P11_4$ $P14_4$ to $P14_6$	–	–	–
$P15_2$, $P15_3$ $P15_6$, $P15_7$	–	○	○

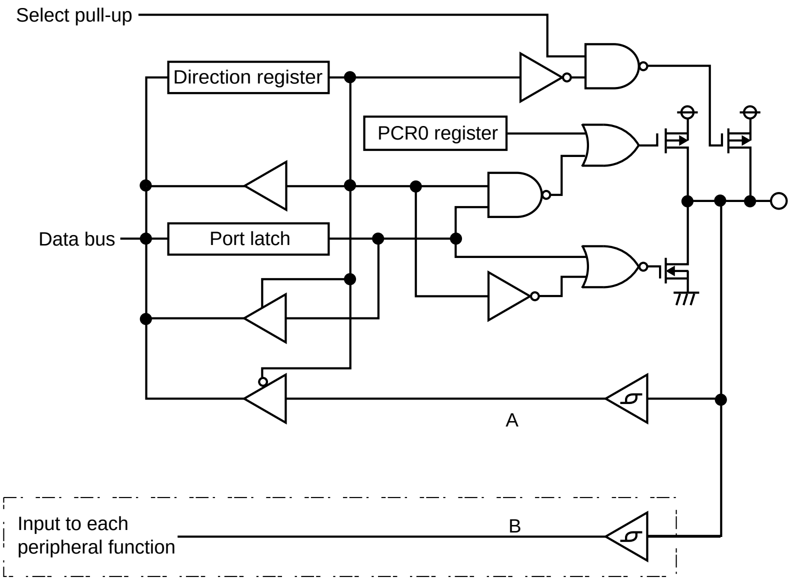
○: Available, – : Not available

Notes

1. These ports exist in the 144-pin package only.

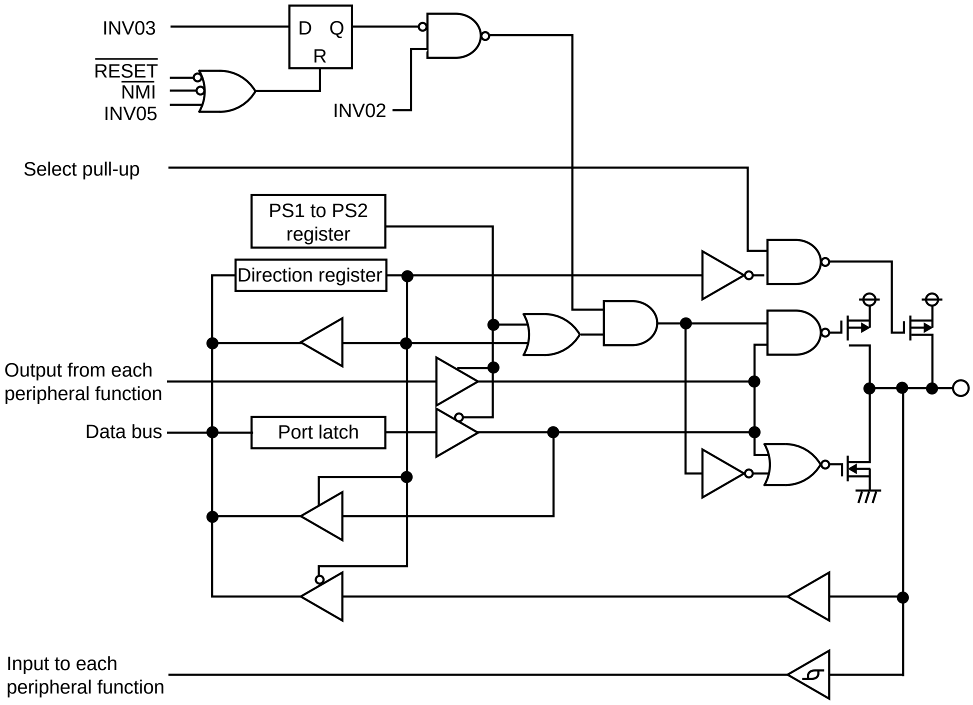
Figure 1.28.1. Programmable I/O Ports (1)

Programmable I/O ports with port control register



Option Port	(A) Hysteresis	Circuit (B) Peripheral function input
P1$_0$ to P1$_4$	–	–
P1$_5$ to P1$_7$	○	○

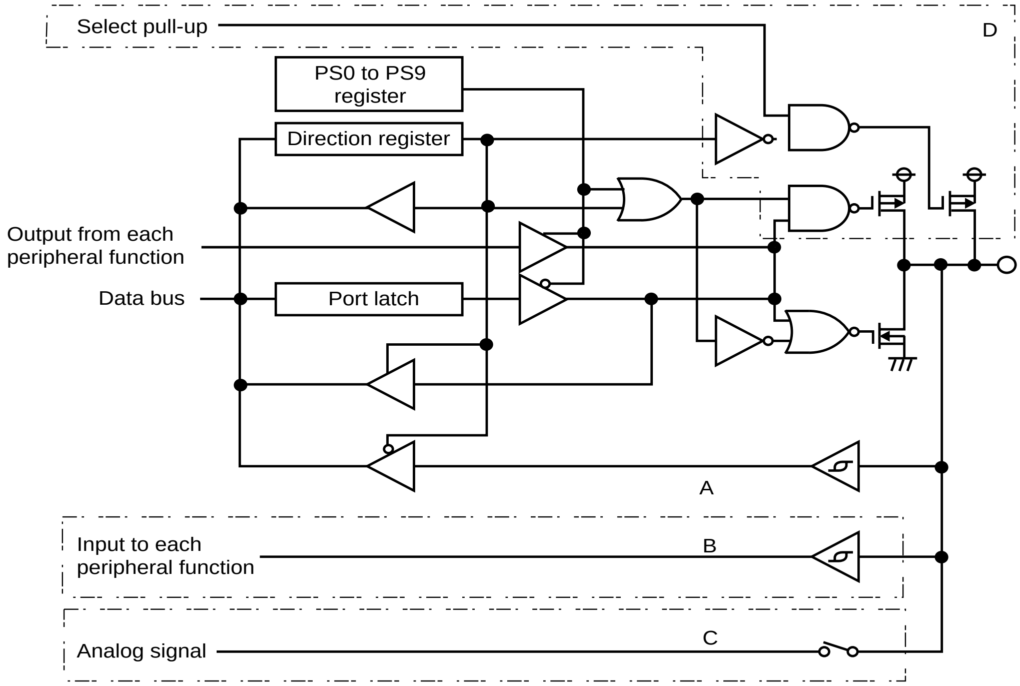
○ : Available, – : Not available



Port : P7$_2$, P7$_3$, P7$_4$, P7$_5$, P8$_0$, P8$_1$

Figure 1.28.2. Programmable I/O Ports (2)

Programmable I/O ports with function select register



	Option Port	(A) Hysteresis	Circuit (B) Peripheral function input	Circuit (C) Analog I/F	Circuit (D)
	$P5_3$	–	–	–	○
	$P6_0$ to $P6_7$	–	○	–	○
	$P7_0$, $P7_1$ (Note 1)	–	○	–	–
	$P7_6$, $P7_7$	–	○	–	○
	$P8_2$	○	○	–	○
	$P9_0$ to $P9_2$	–	○	–	○
	$P9_3$ to $P9_6$	–	○	○	○
	$P9_7$	–	○	–	○
(Note 2)	$P11_0$	–	–	–	○
	$P11_1$, $P11_2$	–	○	–	○
	$P11_3$ $P12_0$	–	–	–	○
	$P12_1$, $P12_2$	–	–	–	○
	$P12_3$ to $P12_7$ $P13_0$ to $P13_4$	–	–	–	○
	$P13_5$, $P13_6$	–	○	–	○
	$P13_7$ $P14_0$, $P14_1$	–	–	–	○
	$P14_2$, $P14_3$	–	○	–	○
	$P15_0$, $P15_1$ $P15_4$, $P15_5$	–	○	○	○

○ : Available, – : Not available

Notes:
1. $P7_0$ and $P7_1$ 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)

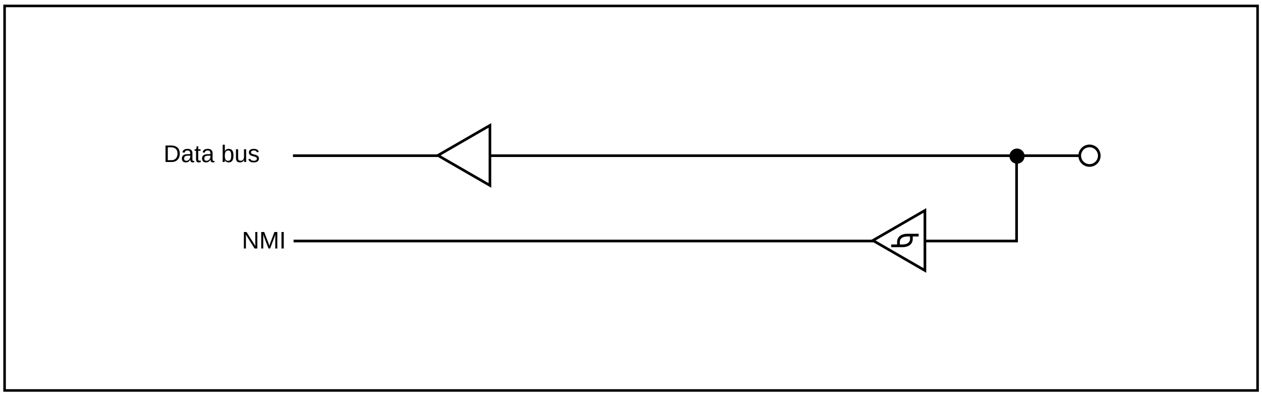
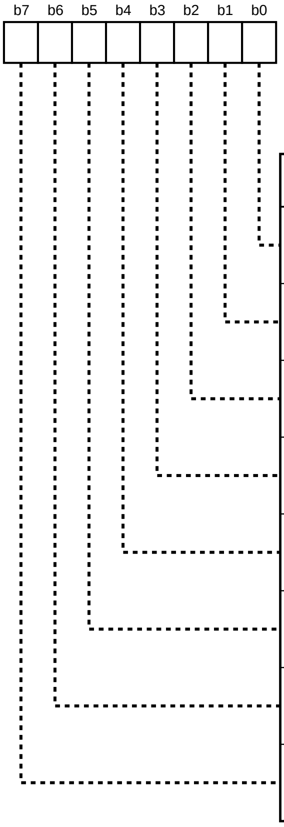


Figure 1.28.4. Programmable I/O Ports (4)

Port Pi direction register (i=0 to 15)



Symbol	Address	When reset
PD0 to PD5	03E2$_{16}$, 03E3$_{16}$, 03E6$_{16}$, 03E7$_{16}$, 03EA$_{16}$, 03EB$_{16}$[2]	00$_{16}$
PD6 to PD10	03C2$_{16}$, 03C3$_{16}$, 03C6$_{16}$[4], 03C7$_{16}$[1], 03CA$_{16}$,	00$_{16}$
PD11 to PD15	03CB$_{16}$[4], 03CE, 03CF$_{16}$, 03D2$_{16}$[4], 03D3$_{16}$[3]	00$_{16}$

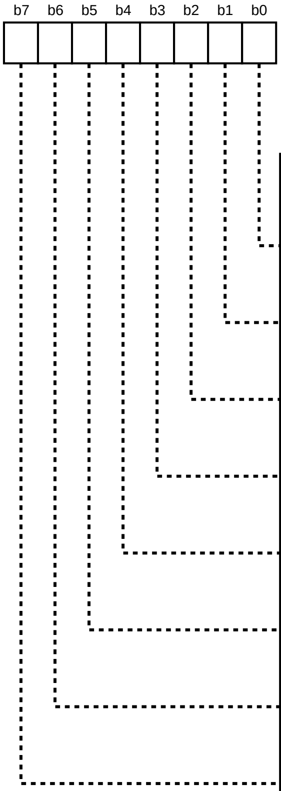
Bit Symbol	Bit name	Function	RW
PDi_0	Port Pi$_0$ direction register	0 : Input mode (Functions as input port) 1 : Output mode (Functions as output port)	RW
PDi_1	Port Pi$_1$ direction register	0 : Input mode (Functions as input port) 1 : Output mode (Functions as output port)	RW
PDi_2	Port Pi$_2$ direction register	0 : Input mode (Functions as input port) 1 : Output mode (Functions as output port)	RW
PDi_3	Port Pi$_3$ direction register	0 : Input mode (Functions as input port) 1 : Output mode (Functions as output port)	RW
PDi_4	Port Pi$_4$ direction register	0 : Input mode (Functions as input port) 1 : Output mode (Functions as output port)	RW
PDi_5	Port Pi$_5$ direction register	0 : Input mode (Functions as input port) 1 : Output mode (Functions as output port)	RW
PDi_6	Port Pi$_6$ direction register	0 : Input mode (Functions as input port) 1 : Output mode (Functions as output port)	RW
PDi_7	Port Pi$_7$ direction register	0 : Input mode (Functions as input port) 1 : Output mode (Functions as output port)	RW

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. In memory expansion and microprocessor mode, the bus control pins (A$_0$ to A$_{22}$, $\overline{\text{A}_{23}}$, D$_0$ to D$_{15}$, MA0 to MA12, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, $\overline{\text{WRL}}$/$\overline{\text{WR}}$/$\overline{\text{CASL}}$, $\overline{\text{WRH}}$/$\overline{\text{BHE}}$/$\overline{\text{CASH}}$, $\overline{\text{RD}}$/$\overline{\text{DW}}$, BCLK/ALE/CLK$_{\text{OUT}}$, $\overline{\text{HLDA}}$/ALE, $\overline{\text{HOLD}}$, ALE/$\overline{\text{RAS}}$ and $\overline{\text{RDY}}$) of the direction register cannot be changed.
3. The PD11 to PD15 registers should be set to "00$_{16}$" in the 100-pin package.
4. 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

Port Pi register (i=0 to 15)[1,2]



Symbol	Address	When reset
P0 to P5	$03E0_{16}$, $03E1_{16}$, $03E4_{16}$, $03E5_{16}$, $03E8_{16}$, $03E9_{16}$	Indeterminate
P6 to P10	$03C0_{16}$, $03C1_{16}$[3], $03C4_{16}$[4], $03C5_{16}$, $03C8_{16}$, $03C9_{16}$	Indeterminate
P11 to P15	$03C9_{16}$[5], $03CC_{16}$, $03CD_{16}$, $03D0_{16}$[5], $03D1_{16}$	Indeterminate

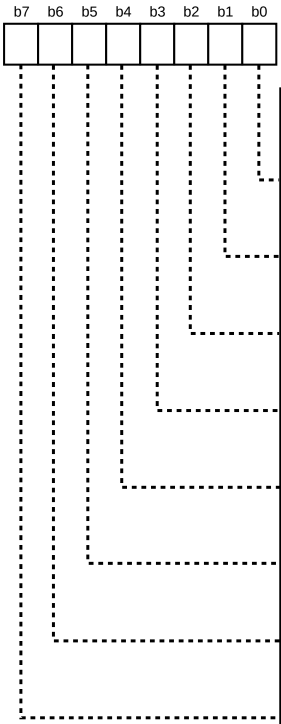
Bit symbol	Bit name	Function	RW
Pi_0	Port Pi$_0$ register	Pin levels can be read by reading bits corresponding to programmable ports set in input mode. Pin levels can be controlled by writing to bits corresponding to programmable ports set in output mode.	RW
Pi_1	Port Pi$_1$ register		RW
Pi_2	Port Pi$_2$ register		RW
Pi_3	Port Pi$_3$ register		RW
Pi_4	Port Pi$_4$ register		RW
Pi_5	Port Pi$_5$ register		RW
Pi_6	Port Pi$_6$ register		RW
Pi_7	Port Pi$_7$ register		RW

Notes:

1. In memory expansion and microprocessor mode, the bus control pins (A_0 to A_{22}, $\overline{A_{23}}$, D_0 to D_{15}, MA0 to MA12, $\overline{CS0}$ to $\overline{CS3}$, $\overline{WRL}/\overline{WR}/\overline{CASL}$, $\overline{WRH}/\overline{BHE}/\overline{CASH}$, $\overline{RD}/\overline{DW}$, BCLK/ALE/$CLK_{OUT}$, $\overline{HLDA}$/ALE, $\overline{HOLD}$, ALE/$\overline{RAS}$, and $\overline{RDY}$) of the direction register cannot be changed.
2. The P11 to P15 registers are provided in the 144-pin package only.
3. Pins go into a high-impedance state when $P7_0$ and $P7_1$ output "1" because of their N-channel open drain ports.
4. The $P8_5$ register is for read only.
5. Nothing is assigned in the P11_5 to P11_7 bits in the P11 register and the P14_7 bit in the P14 register. When write, these bits should be set to "0". When read, its content is indeterminate.

Figure 1.28.6. P0 to P15 Registers

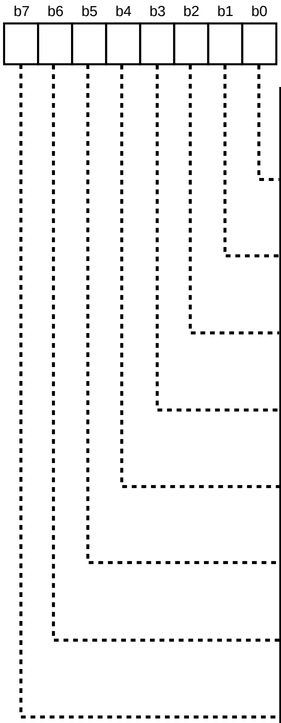
Function select register A0



Symbol	Address	When reset
PS0	$03B0_{16}$	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
PS0_0	Port $P6_0$ output function select bit	0 : I/O port 1 : $\overline{RTS0}$	RW
PS0_1	Port $P6_1$ output function select bit	0 : I/O port 1 : CLK0 output	RW
PS0_2	Port $P6_2$ output function select bit	0 : I/O port 1 : Selected in PSL0_2 bit	RW
PS0_3	Port $P6_3$ output function select bit	0 : I/O port 1 : TxD0/SDA0	RW
PS0_4	Port $P6_4$ output function select bit	0 : I/O port 1 : Selected in PSL0_4 bit	RW
PS0_5	Port $P6_5$ output function select bit	0 : I/O port 1 : CLK1 output	RW
PS0_6	Port $P6_6$ output function select bit	0 : I/O port 1 : Selected in PSL0_6 bit	RW
PS0_7	Port $P6_7$ output function select bit	0 : I/O port 1 : TxD1/SDA1	RW

Function select register A1



Symbol	Address	When reset
PS1	$03B1_{16}$	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
PS1_0	Port $P7_0$ output function select bit	0 : I/O port 1 : Selected in PSL1_0 bit	RW
PS1_1	Port $P7_1$ output function select bit	0 : I/O port 1 : Selected in PSL1_1 bit	RW
PS1_2	Port $P7_2$ output function select bit	0 : I/O port 1 : Selected in PSL1_2 bit	RW
PS1_3	Port $P7_3$ output function select bit	0 : I/O port 1 : Selected in PSL1_3 bit	RW
PS1_4	Port $P7_4$ output function select bit	0 : I/O port 1 : Selected in PSL1_4 bit	RW
PS1_5	Port $P7_5$ output function select bit	0 : I/O port 1 : Selected in PSL1_5 bit	RW
PS1_6	Port $P7_6$ output function select bit	0 : I/O port 1 : Selected in PSL1_6 bit	RW
PS1_7	Port $P7_7$ output function select bit	0 : I/O port 1 : $OUTC0_1$/ISCLK0 output	RW

Figure 1.28.7. PS0 Register and PS1 Register

Function select register A2

b7	b6	b5	b4	b3	b2	b1	b0
0	0	╳	0	0			

Symbol	Address	When reset
PS2	03B4$_{16}$	00X0 0000$_2$

Bit symbol	Bit name	Function	RW
PS2_0	Port P8_0 output function select bit	0 : I/O port 1 : Selected in PSL2_0 bit	RW
PS2_1	Port P8_1 output function select bit	0 : I/O port 1 : Selected in PSL2_1 bit	RW
PS2_2	Port P8_2 output function select bit	0 : I/O port 1 : Selected in PSL2_2 bit	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 A3[1]

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
PS3	03B5$_{16}$	0000 0000$_2$

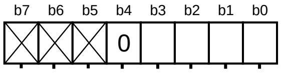
Bit symbol	Bit Name	Function	RW
PS3_0	Port P9_0 output function select bit	0 : I/O port 1 : CLK3 output	RW
PS3_1	Port P9_1 output function select bit	0 : I/O port 1 : Selected in PSL3_1 bit	RW
PS3_2	Port P9_2 output function select bit	0 : I/O port 1 : Selected in PSL3_2 bit	RW
PS3_3	Port P9_3 output function select bit	0 : I/O port 1 : $\overline{RTS3}$	RW
PS3_4	Port P9_4 output function select bit	0 : I/O port 1 : $\overline{RTS4}$	RW
PS3_5	Port P9_5 output function select bit	0 : I/O port 1 : CLK4 output	RW
PS3_6	Port P9_6 output function select bit	0 : I/O port 1 : TxD4/SDA4	RW
PS3_7	Port P9_7 output function select bit	0 : I/O port 1 : Selected in PSL3_7 bit	RW

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 1.28.8. PS2 Register and PS3 Register

Function select register A5[1]



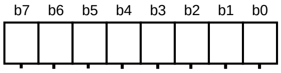
Symbol	Address	When reset
PS5	03B9$_{16}$	XXX0 0000$_2$

Bit symbol	Bit name	Function	RW
PS5_0	Port P11$_0$ output function select bit	0 : I/O port 1 : OUTC1$_0$/ IST$_X$D1/BE1$_{OUT}$	RW
PS5_1	Port P11$_1$ output function select bit	0 : I/O port 1 : OUTC1$_1$/ ISCLK1 output	RW
PS5_2	Port P11$_2$ output function select bit	0 : I/O port 1 : OUTC1$_2$	RW
PS5_3	Port P11$_3$ output function select bit	0 : I/O port 1 : OUTC1$_3$	RW
——	Reserved bit	Should set to "0"	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
——			—
——			—

Notes:

1. This register is provided in the 144-pin package only.

Function select register A6[1]



Symbol	Address	When reset
PS6	03BC$_{16}$	00$_{16}$

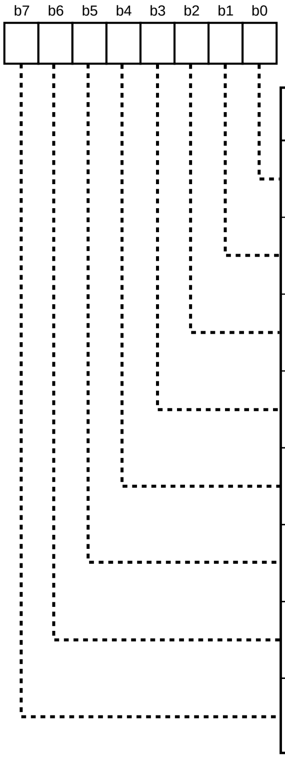
Bit symbol	Bit name	output function	RW
PS6_0	Port P12$_0$ output function select bit	0 : I/O port 1 : OUTC3$_0$/ISTxD3	RW
PS6_1	Port P12$_1$ output function select bit	0 : I/O port 1 : OUTC3$_1$/ISCLK3	RW
PS6_2	Port P12$_2$ output function select bit	0 : I/O port 1 : OUTC3$_2$/ISRxD3	RW
PS6_3	Port P12$_3$ output function select bit	0 : I/O port 1 : OUTC3$_3$	RW
PS6_4	Port P12$_4$ output function select bit	0 : I/O port 1 : OUTC3$_4$	RW
PS6_5	Port P12$_5$ output function select bit	0 : I/O port 1 : OUTC3$_5$	RW
PS6_6	Port P12$_6$ output function select bit	0 : I/O port 1 : OUTC3$_6$	RW
PS6_7	Port P12$_7$ output function select bit	0 : I/O port 1 : OUTC3$_7$	RW

Notes:

1. This register is provided in the 144-pin package only.

Figure 1.28.9. PS5 Register and PS6 Register

Function select register A7^1

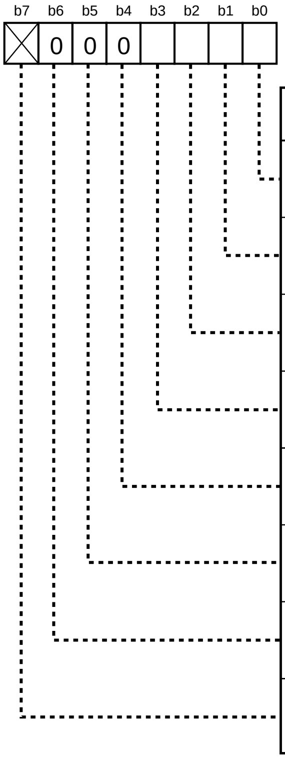


Symbol	Address	When reset
PS7	03BD$_{16}$	0000 0000$_2$

Bit symbol	Bit name	Function	RW
PS7_0	Port P13$_0$ output function select bit	0 : I/O port 1 : OUTC2$_4$	RW
PS7_1	Port P13$_1$ output function select bit	0 : I/O port 1 : OUTC2$_5$	RW
PS7_2	Port P13$_2$ output function select bit	0 : I/O port 1 : OUTC2$_6$	RW
PS7_3	Port P13$_3$ output function select bit	0 : I/O port 1 : OUTC2$_3$	RW
PS7_4	Port P13$_4$ output function select bit	0 : I/O port 1 : OUTC20/ISTXD2/IE$_{OUT}$	RW
PS7_5	Port P13$_5$ output function select bit	0 : I/O port 1 : OUTC2$_2$	RW
PS7_6	Port P13$_6$ output function select bit	0 : I/O port 1 : OUTC2$_1$/ISCLK2 output	RW
PS7_7	Port P13$_7$ output function select bit	0 : I/O port 1 : OUTC2$_7$	RW

Notes:
1. This register is provided in the 144-pin package only.

Function select register A8^1



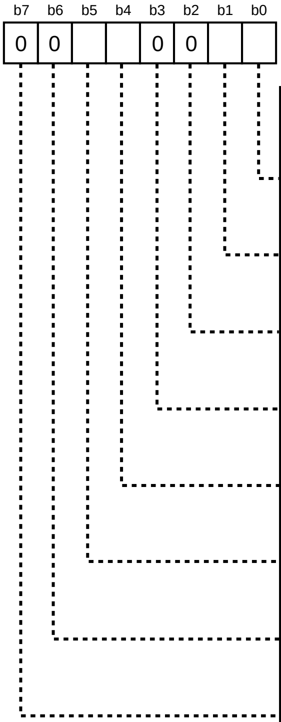
Symbol	Address	When reset
PS8	03A0$_{16}$	X000 0000$_2$

Bit symbol	Bit name	Function	RW
PS8_0	Port P14$_0$ output function select bit	0 : I/O port 1 : OUTC1$_4$	RW
PS8_1	Port P14$_1$ output function select bit	0 : I/O port 1 : OUTC1$_5$	RW
PS8_2	Port P14$_2$ output function select bit	0 : I/O port 1 : OUTC1$_6$	RW
PS8_3	Port P14$_3$ output function select bit	0 : I/O port 1 : OUTC1$_7$	RW
——	Reserved bit	Should set to "0"	RW
——			RW
——			RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—

Notes:
1: This register is provided in the 144-pin package only.

Figure 1.28.10. PS7 Register and PS8 Register

Function select register A9[1]



Symbol	Address	When reset
PS9	$03A1_{16}$	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
PS9_0	Port $P15_0$ output function select bit	0 : I/O port 1 : $OUTC0_0$/ IST_xD0/$BE0_{OUT}$	RW
PS9_1	Port $P15_1$ output function select bit	0 : I/O port 1 : $OUTC0_1$/ ISCLK0 output	RW
——	Reserved bit	Should set to "0"	RW
——			RW
PS9_4	Port $P15_4$ output function select bit	0 : I/O port 1 : $OUTC0_4$	RW
PS9_5	Port $P15_5$ output function select bit	0 : I/O port 1 : $OUTC0_5$	RW
——	Reserved bit	Should set to "0"	RW
——			RW

Notes:

1. This register is provided in the 144-pin package only.

Figure 1.28.11. PS9 Register

Function select register B0

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

Symbol	Address	When reset
PSL0	$03B2_{16}$	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
——	Reserved bit	Should set to "0"	RW
——	Reserved bit	Should set to "0"	RW
PSL0_2	Port $P6_2$ output peripheral function select bit	0 : SCL0 1 : $\mathrm{ST_XD0}$	RW
——	Reserved bit	Should set to "0"	RW
PSL0_4	Port $P6_4$ output peripheral function select bit	0 : $\overline{\mathrm{RTS1}}$ 1 : $\mathrm{OUTC2_1}$/ISCLK2 output	RW
——	Reserved bit	Should set to "0"	RW
PSL0_6	Port $P6_6$ output peripheral function select bit	0 : SCL1 output 1 : $\mathrm{ST_XD1}$	RW
——	Reserved bit	Should set to "0"	RW

Function select register B1

b7	b6	b5	b4	b3	b2	b1	b0
0							

Symbol	Address	When reset
PSL1	$03B3_{16}$	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
PSL1_0	Port $P7_0$ output peripheral function select bit	0 : Selected in PSC_0 bit 1 : $\mathrm{TA0_{OUT}}$ output	RW
PSL1_1	Port $P7_1$ output peripheral function select bit	0 : Selected in PSC_1 bit 1 : $\mathrm{ST_XD2}$	RW
PSL1_2	Port $P7_2$ output peripheral function select bit	0 : Selected in PSC_2 bit 1 : $\mathrm{TA1_{OUT}}$ output	RW
PSL1_3	Port $P7_3$ output peripheral function select bit	0 : Selected in PSC_3 bit 1 : $\overline{\mathrm{V}}$	RW
PSL1_4	Port $P7_4$ output peripheral function select bit	0 : Selected in PSC_4 bit 1 : W	RW
PSL1_5	Port $P7_5$ output peripheral function select bit	0 : $\overline{\mathrm{W}}$ 1 : $\mathrm{OUTC1_2}$	RW
PSL1_6	Port $P7_6$ output peripheral function select bit	0 : Selected in PSC_6 bit 1 : $\mathrm{TA3_{OUT}}$ output	RW
——	Reserved bit	Should set to "0"	RW

Figure 1.28.12. PSL0 Register and PSL1 Register

Function select register B2

b7	b6	b5	b4	b3	b2	b1	b0
0	0	×	0	0			

Symbol	Address	When reset
PSL2	$03B6_{16}$	$00X0\ 0000_2$

Bit symbol	Bit name	Function	RW
PSL2_0	Port P8$_0$ output peripheral function select bit	0 : TA4$_{OUT}$ output 1 : U	RW
PSL2_1	Port P8$_1$ output peripheral function select bit	0 : $\overline{U}$ 1 : OUTC3$_0$/ISTxD3	RW
PSL2_2	Port P8$_2$ output peripheral function select bit	0 : OUTC3$_2$/ISRxD3 1 : CAN$_{OUT}$	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

b7	b6	b5	b4	b3	b2	b1	b0
							0

Symbol	Address	When reset
PSL3	$03B7_{16}$	$0000\ 0000_2$

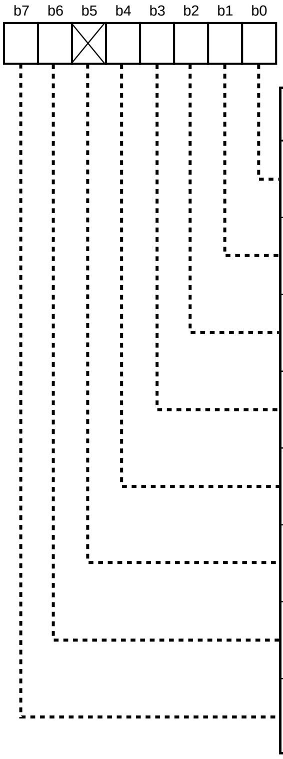
Bit symbol	Bit name	Function	RW
——	Reserved bit	Should set to "0"	RW
PSL3_1	Port P9$_1$ output peripheral function select bit	0 : SCL3 output 1 : STxD3	RW
PSL3_2	Port P9$_2$ output peripheral function select bit	0 : TxD3/SDA3 1 : OUTC2$_0$/ISTxD2/IE$_{OUT}$	RW
PSL3_3	Port P9$_3$ output peripheral function select bit	0 : Expect DA0 1 : DA0 (Note 1)	RW
PSL3_4	Port P9$_4$ output peripheral function select bit	0 : Expect DA1 1 : DA1 (Note 1)	RW
PSL3_5	Port P9$_5$ output peripheral function select bit	0 : Expect ANEX0 1 : ANEX0 (Note 1)	RW
PSL3_6	Port P9$_6$ output peripheral function select bit	0 : Expect ANEX1 1 : ANEX1 (Note 1)	RW
PSL3_7	Port P9$_7$ output peripheral function select bit	0 : SCL4 output 1 : STxD4	RW

Notes:
1. 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

Function select register C

Symbol	Address	When reset
PSC	03AF$_{16}$	00X0 0000$_2$



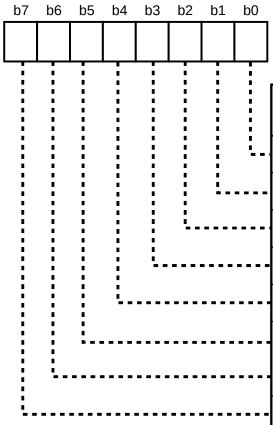
Bit symbol	Bit name	Function	RW
PSC_0	Port P7$_0$ output peripheral function select bit	0 : TxD2/SDA2 1 : OUTC2$_0$/ISTxD2/IE$_{OUT}$	RW
PSC_1	Port P7$_1$ output peripheral function select bit	0 : SCL2 output 1 : OUTC2$_2$	RW
PSC_2	Port P7$_2$ output peripheral function select bit	0 : CLK$_2$ output 1 : V	RW
PSC_3	Port P7$_3$ output peripheral function select bit	0 : $\overline{RTS2}$ 1 : OUTC1$_0$/ISTxD1/BE1$_{OUT}$	RW
PSC_4	Port P7$_4$ output peripheral function select bit	0 : TA2$_{OUT}$ output 1 : OUTC1$_1$/ISCLK1 output	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—
PSC_6	Port P7$_6$ output peripheral function select bit	0 : OUTC0$_0$/ISTxD0/BE0$_{OUT}$ 1 : CAN$_{OUT}$	RW
PSC_7	Key input interrupt disable bit	0 : P10$_4$ to P10$_7$ or $\overline{KI_0}$ to $\overline{KI_3}$ 1 : AN$_4$ to AN$_7$ (Note 1)	RW

Notes:

1: When changing the PSC_7 bit, the ILVL2 to ILVL0 bits in the KUPIC register should be set to "000$_2$" (interrupt disabled).
Although the AN$_4$ to AN$_7$ pins can be used when this bit is set to "0", the power consumption may increase.

Figure 1.28.14. PSC Register

Pull-up control register 0[1]



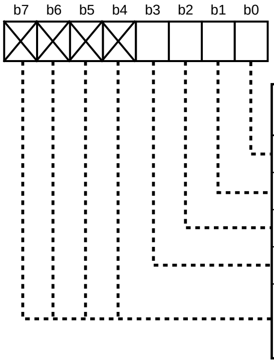
Symbol	Address	When reset
PUR0	03F0$_{16}$	0000 0000$_2$

Bit symbol	Bit name	Function	RW
PU00	$P0_0$ to $P0_3$ pull-up	Pull-up setting for corresponding port 0 : Not pulled up 1 : Pulled up	RW
PU01	$P0_4$ to $P0_7$ pull-up		RW
PU02	$P1_0$ to $P1_3$ pull-up		RW
PU03	$P1_4$ to $P1_7$ pull-up		RW
PU04	$P2_0$ to $P2_3$ pull-up		RW
PU05	$P2_4$ to $P2_7$ pull-up		RW
PU06	$P3_0$ to $P3_3$ pull-up		RW
PU07	$P3_4$ to $P3_7$ pull-up		RW

Notes:

1. Each bit in the PUR0 register should be set to "0" since ports P0 to P5 operate as address bus in the memory expansion mode and microprocessor mode. When using the ports as I/O port by the settings, either pull-up or no pull-up can be selected.

Pull-up control register 1[1]



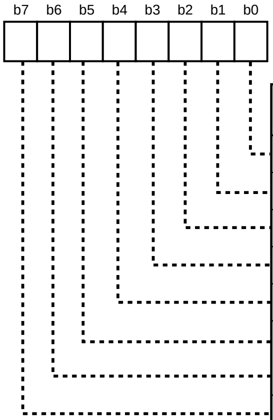
Symbol	Address	When reset
PUR1	03F1$_{16}$	XXXX 0000$_2$

Bit symbol	Bit name	Function	RW
PU10	$P4_0$ to $P4_3$ pull-up	Pull-up setting for corresponding port 0 : Not pulled up 1 : Pulled up	RW
PU11	$P4_4$ to $P4_7$ pull-up		RW
PU12	$P5_0$ to $P5_3$ pull-up		RW
PU13	$P5_4$ to $P5_7$ pull-up		RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—

Notes:

1. Each bit in the PUR0 register should be set to "0" since ports P0 to P5 operate as address bus in the memory expansion mode and microprocessor mode. When using the ports as I/O port by the settings, either pull-up or no pull-up can be selected.

Pull-up control register 2



Symbol	Address	When reset
PUR2	03DA$_{16}$	0000 0000$_2$

Bit symbol	Bit name	Function	RW
PU20	$P6_0$ to $P6_3$ pull-up	Pull-up setting for corresponding port 0 : Not pulled up 1 : Pulled up	RW
PU21	$P6_4$ to $P6_7$ pull-up		RW
PU22	$P7_2$ to $P7_3$ pull-up[1]		RW
PU23	$P7_4$ to $P7_7$ pull-up		RW
PU24	$P8_0$ to $P8_3$ pull-up		RW
PU25	$P8_4$ to $P8_7$ pull-up[2]		RW
PU26	$P9_0$ to $P9_3$ pull-up		RW
PU27	$P9_4$ to $P9_7$ pull-up		RW

Notes:

1. No pull-up in $P7_0$, $P7_1$
2. No pull-up in $P8_5$

Figure 1.28.15. PUR0 Register, PUR1 Register and PUR2 Register

Pull-up control register 3

<144-pin package>

b7	b6	b5	b4	b3	b2	b1	b0

Symbol	Address	When reset
PUR3	$03DB_{16}$	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
PU30	$P10_0$ to $P10_3$ pull-up	Pull-up setting for corresponding port 0 : Not pulled up 1 : Pulled up	RW
PU31	$P10_4$ to $P10_7$ pull-up		RW
PU32	$P11_0$ to $P11_3$ pull-up		RW
PU33	$P11_4$ pull-up		RW
PU34	$P12_0$ to $P12_3$ pull-up		RW
PU35	$P12_4$ to $P12_7$ pull-up		RW
PU36	$P13_0$ to $P13_3$ pull-up		RW
PU37	$P13_4$ to $P13_7$ pull-up		RW

Pull-up control register 3

<100-pin package>

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

Symbol	Address	When reset
PUR3	$03DB_{16}$	$0000\ 0000_2$

Bit symbol	Bit name	Function	RW
PU30	$P10_0$ to $P10_3$ pull-up	Pull-up setting for corresponding port 0 : Not pulled up 1 : Pulled up	RW
PU31	$P10_4$ to $P10_7$ pull-up		RW
——	Reserved bit	Should set to "0"	RW

Pull-up control register 4[1]

b7	b6	b5	b4	b3	b2	b1	b0
☒	☒	☒	☒				

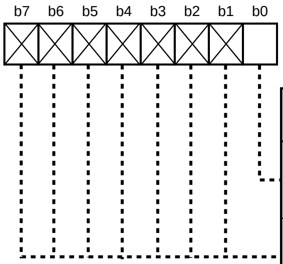
Symbol	Address	When reset
PUR4	$03DC_{16}$	$XXXX\ 0000_2$

Bit symbol	Bit name	Function	RW
PU40	$P14_0$ to $P14_3$ pull-up	Pull-up setting for corresponding port 0 : Not pulled up 1 : Pulled up	RW
PU41	$P14_4$ to $P14_7$ pull-up		RW
PU42	$P15_0$ to $P15_3$ pull-up		RW
PU43	$P15_4$ to $P15_7$ pull-up		RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		——

Note 1: This register should be set to "00_{16}" in the 100-pin package.

Figure 1.28.16. PUR3 Register and PUR4 Register

Port control register (Note 1)



Symbol PCR
Address $03FF_{16}$
When reset $XXXX\ XXX0_2$

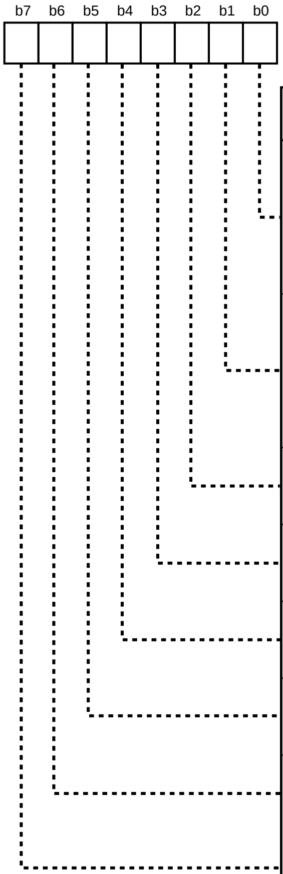
Bit symbol	Bit name	Function	RW
PCR0	Port P1 control bit	0 : CMOS port with port P1 output format 1 : N-channel open drain port output (Note 2)	RW
——	Nothing is assigned. When write, should set to "0". When read, its content is indeterminate.		—

Notes:

1. Each bit in the PUR0 register should be set to "0" since port P1 operate as address bus in memory expansion mode and microprocessor mode. When using the ports as I/O port by the settings, either pull-up or no pull-up can be selected.
2. This function is designed not to make port P1 a full open drain but to turn OFF the P channel in the CMOS port permanently.
   Absolute maximum rating of the input is from -0.3V to VCC + 3.0V.

Figure 1.28.17. PCR Register

Input function select register



Symbol IPS
Address 0178_{16}
When reset $0000\ 0000_2$

Bit symbol	Bit name	Function	RW
IPS0	Group 0 input pin select bit 0	Assigns each function of INPC0$_0$, INPC0$_1$/ISCLK0 and INPC0$_2$/ISRxD0 /BE0$_{IN}$ to the following ports. 0 : P7$_6$, P7$_7$, P8$_0$ 1 : P15$_0$, P15$_1$, P15$_2$	RW
IPS1	Group 1 input pin select bit 1	Assigns each function of INPC1$_1$ /ISCLK1 and INPC1$_2$/ISRxD1/BE1$_{IN}$ to the following ports. 0 : P7$_4$, P7$_5$ 1 : P11$_1$, P11$_2$	RW
IPS2	P15 input peripheral function select bit	0 : Except AN15 (Note 1) 1 : AN15	RW
IPS3	CAN$_{IN}$ function pin select bit	0 : P7$_7$ 1 : P8$_3$	RW
IPS4	ISRxD2/IE$_{IN}$ function pin select bit	b5 b4 0 0 : P7$_1$ 0 1 : P9$_1$ 1 0 : P13$_5$ 1 1 : Avoid this setting	RW
IPS5			RW
IPS6	ISCLK2 function pin select bit	0 : P6$_4$ 1 : P13$_6$	RW
IPS7	ISRxD3 function pin select bit	0 : P8$_1$, P8$_2$ 1 : P12$_0$, P12$_2$	RW

Notes:

1. Although the AN15$_0$ to AN15$_7$ pins can be used when this bit is set to "0", the power consumption may increase.

Figure 1.28.18. IPS Register

Table 1.28.1. Unassigned Pin Handling in Single-chip Mode

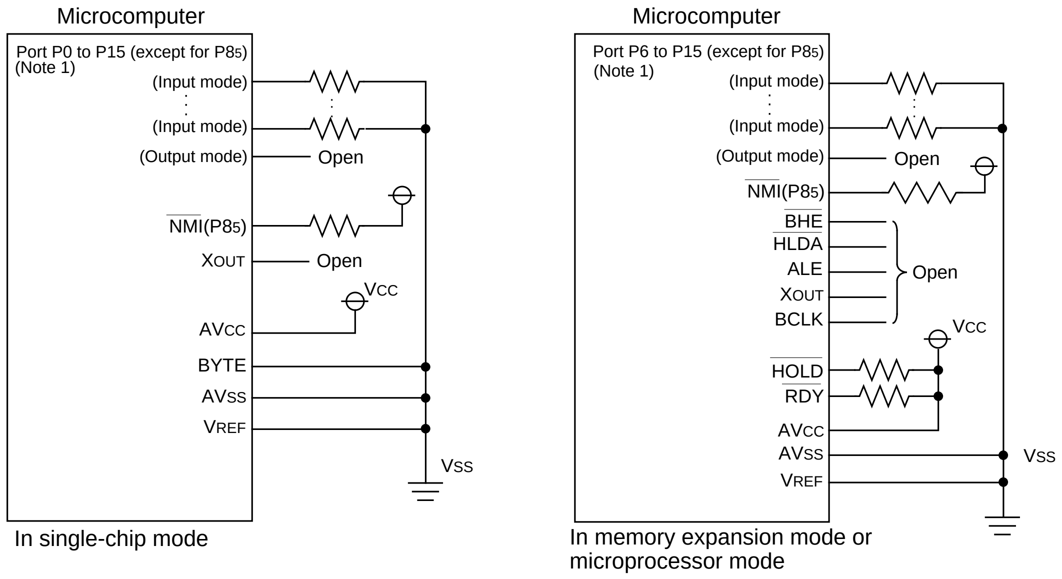
Pin name	How to handle
Ports P0 to P15 (excluding P85)[1]	After setting for input mode, connect each pin to VSS via a resistance (pull-down); or after setting for output mode, leave these pins open.
XOUT[2]	Open
$\overline{\text{NMI}}$(P85)	Connect to VCC via a resistance (pull-up)
AVCC	Connect to VCC
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)[1]	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.
$\overline{\text{BHE}}$, ALE, $\overline{\text{HLDA}}$, XOUT[2], BCLK	Open
$\overline{\text{HOLD}}$, $\overline{\text{RDY}}$, $\overline{\text{NMI}}$(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.



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

Figure 1.28.19. Unassigned Pin Handling

Table 1.28.3. Port P6 Peripheral Function Output Control

	PS0 register	PSL0 register
Bit 0	0: $P6_0/\overline{CTS0}/\overline{SS0}$ 1: $\overline{RTS0}$	Should set to "0"
Bit 1	0: $P6_1$/CLK0 input 1: CLK0 output	Should set to "0"
Bit 2	0: $P6_2$/RxD0/SCL0 input 0: Selected in PSL0 register	0: SCL0 output 1: STxD0
Bit 3	0: $P6_3$/SRxD0 1: TxD0/SDA0	Should set to "0"
Bit 4	0: $P6_4/\overline{CTS1}/\overline{SS1}$ 1: Selected in PSL0 register	0: $\overline{RTS1}$ 1: $OUTC2_1$/ISCLK2 output
Bit 5	0: $P6_5$/CLK1 input 1: CLK1 output	Should set to "0"
Bit 6	0: $P6_6$/RxD1/SCL1 input 1: Selected in PSL0 register	0: SCL1 output 1: STxD1
Bit 7	0: $P6_7$/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: $P7_0$/SRxD2/$TA0_{OUT}$ input 1: Selected in PSL1 register	0: Selected in PSC register 1: $TA0_{OUT}$ output	0: TxD2/SDA2 1: $OUTC2_0$/ISTxD2/IE_{OUT}
Bit 1	0: $P7_1$/$TB5_{IN}$/$TA0_{IN}$/RxD2/ ISRxD2/IE_{IN}/SCL2 input 1: Selected PSL1 register	0: Selected in PSC register 1: STxD2	0: SCL2 output 1: $OUTC2_2$
Bit 2	0: $P7_2$/$TA1_{OUT}$ input/ CLK2 input 1: Selected in PSL1 register	0: Selected in PSC register 1: $TA1_{OUT}$ output	0: CLK2 output 1: V
Bit 3	0: $P7_3/\overline{CTS2}/\overline{SS2}/TA1_{IN}$ 1: Selected in PSL1 register	0: Selected in PSC register 1: $\overline{V}$	0: $\overline{RTS2}$ 1: $OUTC1_0$/ISTxD1/$BE1_{OUT}$
Bit 4	0: $P7_4$/$INPC1_1$/ISCK1 input/ $TA2_{OUT}$ input 1: Selected in PSL1 register	0: Selected in PSC register 1: W	0: $TA2_{OUT}$ output 1: $OUTC1_1$/ISCLK1 output
Bit 5	0: $P7_5$/$TA2_{IN}$/$INPC1_2$/ISRxD1/ $BE1_{IN}$ 1: Selected in PSL1 register	0: $\overline{W}$ 1: $OUTC1_2$	Should set to "0"
Bit 6	0: $P7_6$/$INPC0_0$/$TA3_{OUT}$ input 1: Selected in PSL1 register	0: Selected in PSC register 1: $TA3_{OUT}$ output	0: $OUTC0_0$/ISTxD0/$BE0_{OUT}$ 1: CAN_{OUT}
Bit 7	0: $P7_7$/$TA3_{IN}$/CAN_{IN}/ISCLK0 input/$INPC0_1$ 1: $OUTC0_1$/ISCLK0	Should set to "0"	0: $P10_4$ to $P10_7$ or $\overline{KI_0}$ to $\overline{KI_3}$ 1: AN4 to AN_7 (No relation to $P7_7$)

Table 1.28.5. Port P8 Peripheral Function Output Control

	PS2 register	PSL2 register
Bit 0	0: P8$_0$/INPC0$_2$/ISRxD0/BE0$_{IN}$ /TA4$_{OUT}$ input 1: Selected in PSL2 register	0: TA4$_{OUT}$ output 1: U
Bit 1	0: P8$_1$/TA4$_{IN}$ 1: Selected in PSL2 register	0: $\overline{U}$ 1: OUTC3$_0$/ISTxD3
Bit 2	0: P8$_2$/$\overline{INT0}$ 1: Selected in PSL2 register	0: OUTC3$_2$/ISRxD3 1: CAN$_{OUT}$
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: P9$_0$/TB0$_{IN}$/CLK3 input 1: CLK3 output	Should set to "0"
Bit 1	0: P9$_1$/TB1$_{IN}$/RxD3/ISRxD2/SCL3 input/ IE$_{IN}$ 1: Selected in PSL3 register	0: SCL3 output 1: STxD3
Bit 2	0: P9$_2$/TB2$_{IN}$/SRxD3 1: Selected in PSL3 register	0: TxD3/SDA3 1: OUTC2$_0$/ISTxD2/IE$_{OUT}$
Bit 3	0: P9$_3$/TB3$_{IN}$/$\overline{CTS3}$/$\overline{SS3}$/DA0 output 1: $\overline{RTS3}$	0: Except DA0 1: DA0
Bit 4	0: P9$_4$/TB4$_{IN}$/$\overline{CTS4}$/$\overline{SS4}$/DA1 output 1: $\overline{RTS4}$	0: Except DA1 1: DA1
Bit 5	0: P9$_5$/ANEX0/CLK4 input 1: CLK4 output	0: Except ANEX0 1: ANEX0
Bit 6	0: P9$_6$/SRxD4/ANEX1 1: TxD4/SDA4	0: Except ANEX1 1: ANEX1
Bit 7	0: P9$_7$/RxD4/$\overline{AD_{TRG}}$/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: P10$_4$ to P10$_7$ or $\overline{KI_0}$ to $\overline{KI_3}$ 1: AN$_4$ to AN$_7$

Table 1.28.8. Port P11 Peripheral Function Output Control

	PS5 register
Bit 0	0: P11$_0$ 1: OUTC1$_0$/ISTxD1/BE1$_{OUT}$
Bit 1	0: P11$_1$/INPC1$_1$/ISCLK1 input 1: OUTC1$_1$/ISCLK1 output
Bit 2	0: P11$_2$/INPC1$_2$/ISRxD1/BE$_{IN}$ 1: OUTC1$_2$
Bit 3	0: P11$_3$ 1: OUTC1$_3$
Bit 4 to 7	Should set to "0"

Table 1.28.9. Port P12 Peripheral Function Output Control

	PS6 register
Bit 0	0: P12$_0$ 1: OUTC3$_0$/ISTxD3
Bit 1	0: P12$_1$/ISCLK3 input 1: OUTC3$_1$/ISCLK3 output
Bit 2	0: P12$_2$ 1: OUTC3$_2$/ISRxD3
Bit 3	0: P12$_3$ 1: OUTC3$_3$
Bit 4	0: P12$_4$ 1: OUTC3$_4$
Bit 5	0: P12$_5$ 1: OUTC3$_5$
Bit 6	0: P12$_6$ 1: OUTC3$_6$
Bit 7	0: P12$_7$ 1: OUTC3$_7$

Table 1.28.10. Port P13 Peripheral Function Output Control

	PS7 register
Bit 0	0: P13$_0$ 1: OUTC2$_4$
Bit 1	0: P13$_1$ 1: OUTC2$_5$
Bit 2	0: P13$_2$ 1: OUTC2$_6$
Bit 3	0: P13$_3$ 1: OUTC2$_3$
Bit 4	0: P13$_4$ 1: OUTC2$_0$/ISTxD2/IE$_{OUT}$
Bit 5	0: P13$_5$/ISRxD2/IE$_{IN}$ 1: OUTC2$_2$
Bit 6	0: P13$_6$/ISCLK2 input 1: OUTC2$_1$/ISCLK2 output
Bit 7	0: P13$_7$ 1: OUTC2$_7$

Table 1.28.11. Port P14 Peripheral Function Output Control

	PS8 register
Bit 0	0: P14$_0$ 1: OUTC1$_4$
Bit 1	0: P14$_1$ 1: OUTC1$_5$
Bit 2	0: P14$_2$/INPC1$_6$ 1: OUTC1$_6$
Bit 3	0: P14$_3$/INPC1$_7$ 1: OUTC1$_7$
Bit 4 to 7	Should set to "0"

Table 1.28.12. Port P15 Peripheral Function Output Control

	PS9 register
Bit 0	0: P15$_0$/INPC0$_0$/AN15$_0$ 1: OUTC0$_0$/ISTxD0/BE$_{OUT}$
Bit 1	0: P15$_1$/INPC0$_1$/AN15$_1$/ISCLK0 input 1: OUTC0$_1$/ISCLK0 output
Bit 2 to 3	Should set to "0"
Bit 4	0: P15$_4$/INPC0$_4$/AN15$_4$ 1: OUTC0$_4$
Bit 5	0: P15$_5$/INPC0$_5$/AN15$_5$ 1: OUTC0$_5$
Bit 6 to 7	Should set to "0"

Usage Precaution

Notes on the 100-pin Package

Set address space $03CB_{16}$, $03CE_{16}$, $03CF_{16}$, $03D2_{16}$ and $3D3_{16}$ to "FF_{16}" as each default value in the 100-pin package. Set address space $03DC_{16}$ to "00_{16}" in the 100-pin package.

$\overline{\text{HOLD}}$ 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 dare set to "11_2" (microprocessor mode) or "10_2" (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 "11_2" (microprocessor mode) or "10_2" (memory expansion mode), $P4_0$ to $P4_7$ (A16 to $\overline{\text{A23}}$, $\overline{\text{CS0}}$ to $\overline{\text{CS3}}$, MA8 to MA12) and $P5_0$ to $P5_2$ ($\overline{\text{RD}}$/$\overline{\text{WR}}$/$\overline{\text{BHE}}$, $\overline{\text{RD}}$/$\overline{\text{WRL}}$/$\overline{\text{WRH}}$, $\overline{\text{CASL}}$/$\overline{\text{CASH}}$/$\overline{\text{DW}}$) 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 CNV_{SS} 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_{16}" 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.

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

- The $\overline{\text{NMI}}$ interrupt cannot be obstructed. Connect (pull up) the $\overline{\text{NMI}}$ pin to V_{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 $\overline{\text{INTi}}$ 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 $\overline{\text{INTi}}$ pins. (At least 234ns when X_{IN} = 30MHz and no division.)
- When a polarity of the $\overline{\text{INTi}}$ pins is switched, the IR bit in the 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.

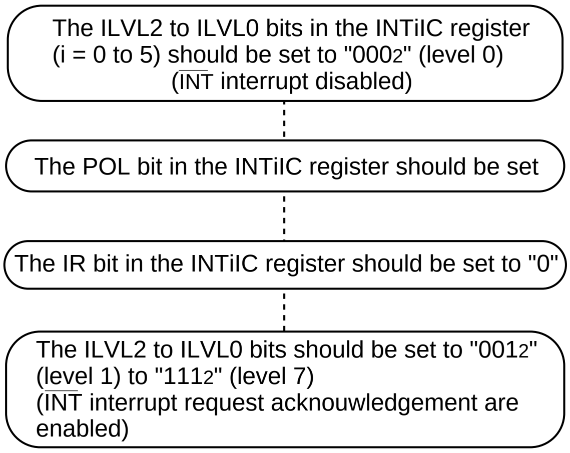

The ILVL2 to ILVL0 bits in the INTiIC register (i = 0 to 5) should be set to "000$_2$" (level 0) ($\overline{\text{INT}}$ interrupt disabled)
The POL bit in the INTiIC register should be set
The IR bit in the INTiIC register should be set to "0"
The ILVL2 to ILVL0 bits should be set to "001$_2$" (level 1) to "111$_2$" (level 7) ($\overline{\text{INT}}$ interrupt request acknouwledgement are enabled)

Figure 1.29.1. Switching Procedure for $\overline{\text{INT}}$ 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

(5) Changing IIOiIR Register (i = 0 to 11)

When bits 1 to 7 in the IIOiIR 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 "00_2" (DMA disabled). Then set the MDi1 to MDi0 bits to "01_2" (single transfer) or "11_2" (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[1], 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 "00_2" or the DCTi register is set to "0000_{16}" (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 "01_2" or "11_2". 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 $\overline{INT}$ interrupt as a DMA request), avoid setting the DCTi register to "1".

Timer A

The TAiS bit (i=1 to 4) in the TABSR register is set to "0" (stops counting) after reset. The TAiS 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 "$FFFF_{16}$" 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 "$FFFF_{16}$" in underflow and "0000_{16}" 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.

(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_{16}" to increment the counter and "$FFFF_{16}$" 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 TAi_{OUT} 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 TAi_{IN} 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 TAiS register to "0" (stops counting) while a PWM pulse is output, the followings occurs:

- The counter stops.
- If the TAi_{OUT} pin is output "H", output level is in "L" and the IR bit is set to "1".
- If the TAi_{OUT} 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.

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 TBiIC 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 se 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 BTSR 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 in1 µs 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 ADiIC register indicates whether the A-D conversion is completed. Verify the IR bit in the ADiIC 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(XIN) is faster than 10 MHz, keep the Ø AD frequency 10 MHz or less by dividing f(XIN).

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 $\overline{\text{NMI}}$ 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 $\overline{\text{NMI}}$ pin input level and output pin states.

Table 1.29.1. INV00 Register and $\overline{\text{NMI}}$ Pin

Setting Value of INV00 register		input level to $\overline{\text{NMI}}$ pin	Sates of P7$_2$ to P7$_5$, P8$_0$, and P8$_1$ 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)^1	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 0.1µF) 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 V$_{REF}$ connection). When using the A-D convertor, wait at least 1 µs to start the A-D convertor after setting the VCUT bit to "1" (V$_{REF}$ connection).

(2) To use AN$_4$ (P10$_4$) to AN$_7$ (P10$_7$), 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 P10$_4$ to P10$_7$ 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 P10$_4$ to P10$_7$ 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$_{16}$".

(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 P10$_4$ to P10$_7$ 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.

Table 1.29.2. Register Having Indeterminate Bits when Read

Register	Address	Register	Address
WDTS register	$000E_{16}$	U2BRG register	0339_{16}
G0RI register	$00EC_{16}$	U2TB register	$033B_{16}$, $033A_{16}$
G1RI register	$012C_{16}$	UDF register	0344_{16}
G2TB register	$016D_{16}$, $016C_{16}$	TA0 register^1	0347_{16}, 0346_{16}
U4BRG register	$02F9_{16}$	TA1 register^1	0349_{16}, 0348_{16}
U4TB register	$02FB_{16}$, $02FA_{16}$	TA2 register^1	$034B_{16}$, $034A_{16}$
TA11 register	0303_{16}, 0302_{16}	TA3 register^1	$034D_{16}$, $034C_{16}$
TA21 register	0305_{16}, 0304_{16}	TA4 register^1	$034F_{16}$, $034E_{16}$
TA41 register	0307_{16}, 0306_{16}	U0BRG register	0369_{16}
DTT register	$030C_{16}$	U0TB register	$036B_{16}$, $036A_{16}$
ICTB2 register	$030D_{16}$	U1BRG register	$02E9_{16}$
U3BRG register	0329_{16}	U1TB register	$02EB_{16}$, $02EA_{16}$
U3TB register	$032B_{16}$, $032A_{16}$	AD0CON2 register	0394_{16}

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 ($0001B_{16}$) to "$0F_{16}$". With the power supply above 3.3V, avoid setting the VDC0 register.

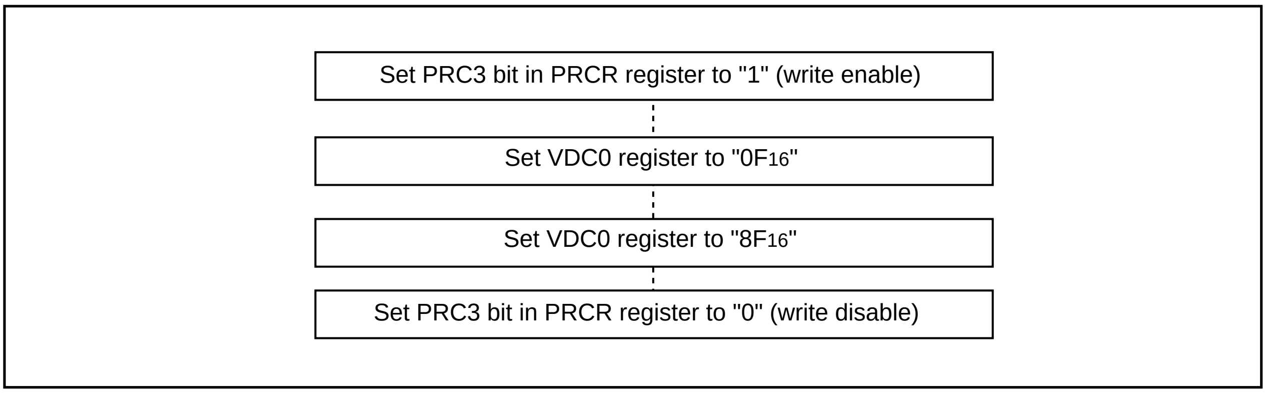

Set PRC3 bit in PRCR register to "1" (write enable)
Set VDC0 register to "0F16"
Set VDC0 register to "8F16"
Set PRC3 bit in PRCR register to "0" (write disable)

Figure 1.29.2. VDC Separating Procedure

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(X_{IN})=30MHz, with no wait) 3.0V to 5.5V (f(X_{IN})=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)[1]
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 excuting 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

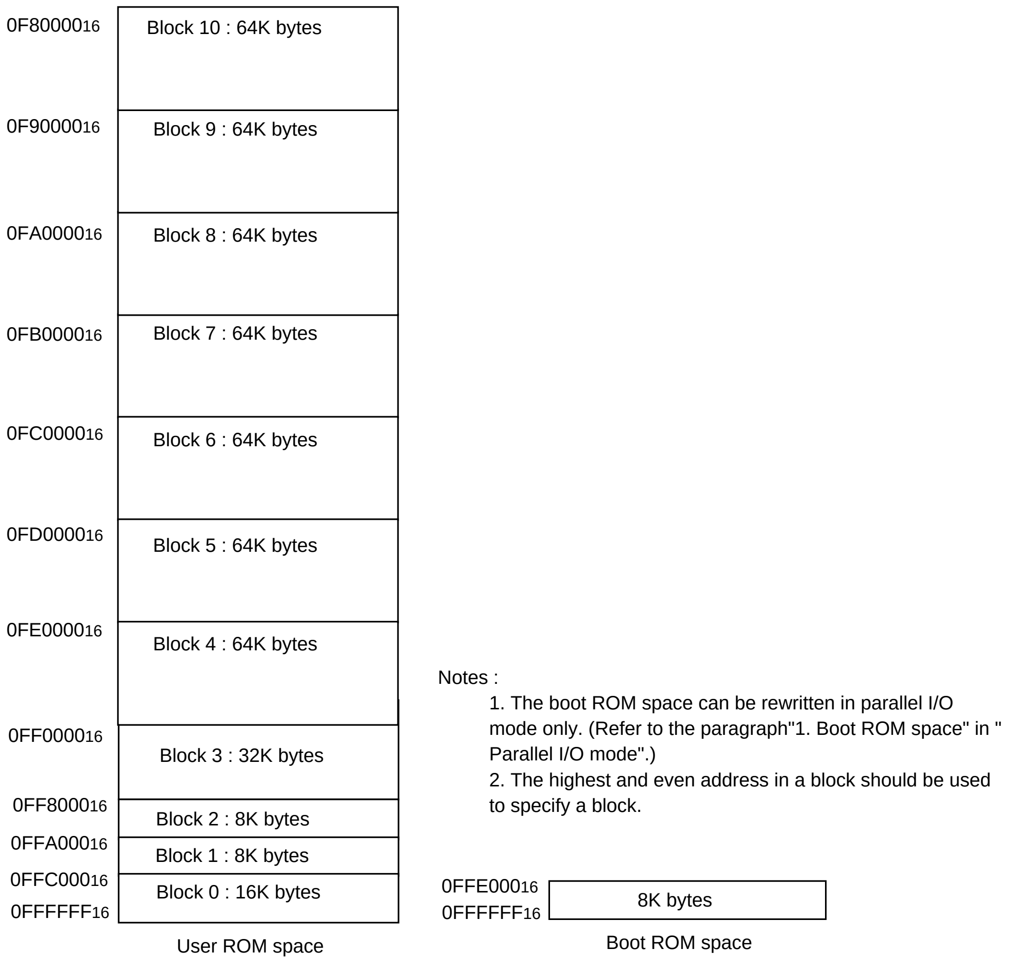
1. Memory Map

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 CNV_{SS} pin and the $P5_0$ pin and "L" to the $P5_5$ pin, a program in the boot ROM space is executed (Refer to the paragraph "Boot mode"). When resetting with "L" input to the CNV_{SS} pin, a program in the user ROM space is executed and the boot ROM space cannot be read.



Notes :

1. The boot ROM space can be rewritten in parallel I/O mode only. (Refer to the paragraph"1. Boot ROM space" in " Parallel I/O mode".)
2. The highest and even address in a block should be used to specify a block.

Figure 1.30.1. Flash Memory Block Diagram

2. Boot Mode

The microcomputer enters boot mode when resetting with "H" input to the CNV$_{SS}$ and P5$_0$ pins and "L" to the P5$_5$ 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.

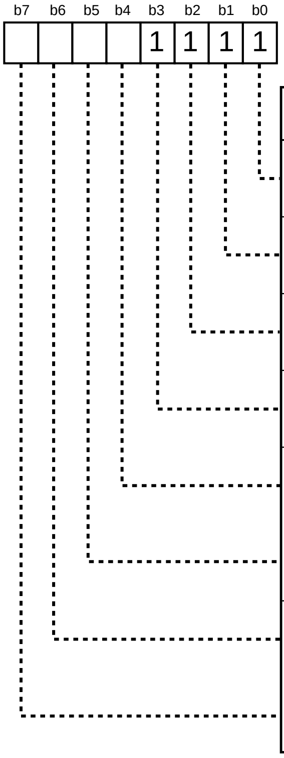
- **ROM Code Protect Function**

  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 "00$_2$" (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$_{16}$, 0FFFFE3$_{16}$, 0FFFFEB$_{16}$, 0FFFFEF$_{16}$, 0FFFFF3$_{16}$, 0FFFFF7$_{16}$ and 0FFFFFB$_{16}$. A program with ID code set in these addresses should be written to the flash memory.

ROM code protect control address[3]



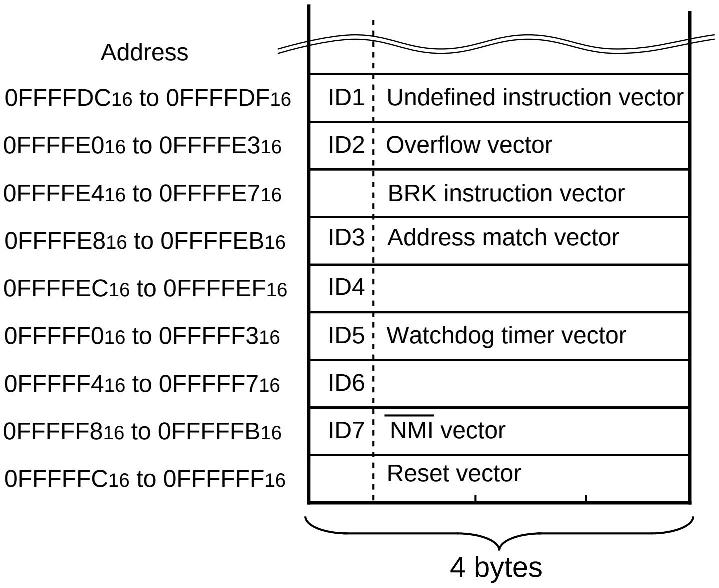
Symbol ROMCP
Address $0FFFFFF_{16}$
When reset FF_{16}

Bit symbol	Bit name	Function	RW
——	Reserved bit	Should set to "1"	RW
——			RW
——			RW
——			RW
ROMCR	ROM code protect reset bit[1]	b5 b4 0 0 : Protect removed 0 1 : ROMCP bit enabled 1 0 : ROMCP bit enabled 1 1 : ROMCP bit enabled	RW
			RW
ROMCP1	ROM code protect level 1 set bit[2]	b7 b6 0 0 : Protect enable 0 1 : Protect enable 1 0 : Protect enable 1 1 : Protect disable	RW
			RW

Notes :

1. When setting the ROMCR bit to "00_2", the ROM code protect level 1 is reset. When the ROM code protection is enabled, the ROMCR bit cannot be changed in parallel I/O mode. The ROMCR bit should be changed in standard serial I/O mode etc.
2. When the ROM code protection is enabled, the flash memory is protected against reading or rewriting in parallel I/O mode.

Figure 1.30.2. ROMCP Register



Address		
$0FFFFDC_{16}$ to $0FFFFDF_{16}$	ID1	Undefined instruction vector
$0FFFFE0_{16}$ to $0FFFFE3_{16}$	ID2	Overflow vector
$0FFFFE4_{16}$ to $0FFFFE7_{16}$		BRK instruction vector
$0FFFFE8_{16}$ to $0FFFFEB_{16}$	ID3	Address match vector
$0FFFFEC_{16}$ to $0FFFFEF_{16}$	ID4	
$0FFFFF0_{16}$ to $0FFFFF3_{16}$	ID5	Watchdog timer vector
$0FFFFF4_{16}$ to $0FFFFF7_{16}$	ID6	
$0FFFFF8_{16}$ to $0FFFFFB_{16}$	ID7	$\overline{\text{NMI}}$ vector
$0FFFFFC_{16}$ to $0FFFFFF_{16}$		Reset vector

Figure 1.30.3. Address for ID code stored

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 (D_{15} to D_8) are ignored.

Table 1.30.3. Software commands

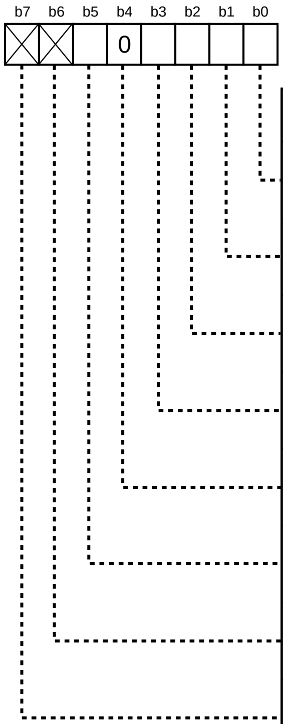
Software command	First bus cycle			Second bus cycle			Third bus cycle		
	Mode	Address	Data (D_{15} to D_0)	Mode	Address	Data (D_{15} to D_0)	Mode	Address	Data (D_{15} to D_0)
Read array	Write	X	$xxFF_{16}$						
Read status register	Write	X	$xx70_{16}$	Read	X	SRD			
Clear status register	Write	X	$xx50_{16}$						
Page program	Write	X	$xx41_{16}$	Write	WA	WD	Write	WA+2	WD
Block erase	Write	X	$xx20_{16}$	Write	BA	$xxD0_{16}$			
All unlock block erase	Write	X	$xxA7_{16}$	Write	X	$xxD0_{16}$			
Lock bit program	Write	X	$xx77_{16}$	Write	BA	$xxD0_{16}$			
Read lock bit status	Write	X	$xx71_{16}$	Read	BA	D_6			

SRD : Data of the status register (D_7 to D_0)
WA : Address to be written (increment A_7 to A_0 by 2 from "00_{16}" to "FE_{16}")
WD : Data to write (16 bits)
BA : Highest-order address of a block ($A_0 = 0$)
D_6 : Lock bit ($D_6 = 1$: unlock, $D_6 = 0$: locked)
X : Any even address in user ROM space
xx : 8 high-order bits of command code (ignored)

1. Flash Memory Control Register 0 (FMR0 Register)

Figure 1.30.4 shows the FMR0 register.

Flash memory control register 0



Symbol	Address	When reset
FMR0	0057_{16}	$XX00\ 0001_2$

Bit symbol	Bit name	Function	RW
FMR00	RY/$\overline{\text{BY}}$ signal status bit	0 : BUSY(auto-writing, auto-erasing) 1 : READY(auto-write completed, auto-erasure completed)	RO
FMR01	CPU rewrite mode select bit	0 : CPU rewrite mode disabled[1,6] 1 : CPU rewrite mode enabled[1,6]	RW
FMR02	Lock bit disable bit	0 : lock bit enabled[1,3,4] 1 : lock bit disabled[1,3,4]	RW
FMR03	Flash memory reset bit[5]	When setting the bit to "0" following "1", flash memory access is interrupted to reset the flash memory control circuit	RW
——	Reserved bit	Should set to"0"	RW
FMR05	User ROM space select bit (Available in boot mode only)[2]	0: Boot ROM space access[6] 1: User ROM space access[6]	RW
——	Noting is assigned. When write, should set to "0". When read, its content is indeterminate.		—
——			—

Notes :

1. This bit should be set to "1" following "0". Avoid generating an interrupt and operating a DMA transfer until setting the bit to "1", after setting the bit to "0". "0" should be set to set to "0".
2. Set this bit by program in other space than the flash memory.
3. When the FMR01 bit is set to "0", this bit is also set to "0" simultaneously.
4. The bit can be written only when the FMR01 bit is set to "1".
5. When the FMR01 bit is set "0", the FMR03 bit should be set to "0".
   The FMR03 bit should be accessed when setting the FMR01 bit to "1".
6. The bit should be set while setting the $\overline{\text{NMI}}$ pin to "H".

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 "80_{16}". (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 programing 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)

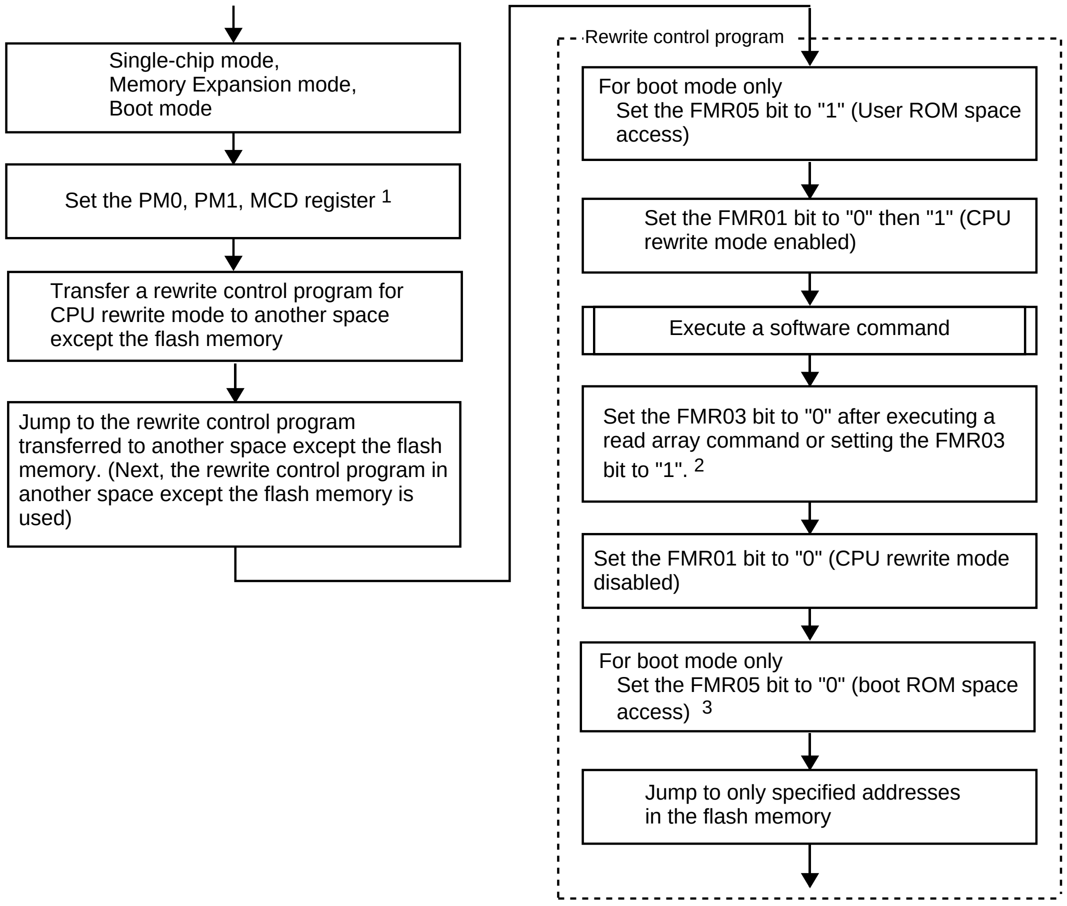
The lock bit is set to "0" (locked) when executing the lock bit program command. The lock bit is set to "1" (unlocked) when erasing a block. The lock bit cannot be set to "1" by command.
Lock bit status can be read by the read lock bit status command.

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.

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.


Single-chip mode,
Memory Expansion mode,
Boot mode
Set the PM0, PM1, MCD register 1
Transfer a rewrite control program for CPU rewrite mode to another space except the flash memory
Jump to the rewrite control program transferred to another space except the flash memory. (Next, the rewrite control program in another space except the flash memory is used)
Rewrite control program
For boot mode only
Set the FMR05 bit to "1" (User ROM space access)
Set the FMR01 bit to "0" then "1" (CPU rewrite mode enabled)
Execute a software command
Set the FMR03 bit to "0" after executing a read array command or setting the FMR03 bit to "1". 2
Set the FMR01 bit to "0" (CPU rewrite mode disabled)
For boot mode only
Set the FMR05 bit to "0" (boot ROM space access) 3
Jump to only specified addresses in the flash memory

Notes :

1. A main clock frequency should be set in the MCED register as follows:
   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)
2. CPU rewrite mode should be exited after excuting a read array command or reseting the flash memory.
3. When CPU rewrite mode is exited with setting the FMR05 bit is "1", the user ROM space can be accessed.

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 (D_{15} to D_8) are disabled.

- **Read Array**

  The read array command reads the flash memory.
  Command code "xxFF$_{16}$" 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 "xx70$_{16}$" 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 "xx50$_{16}$" 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 "xx41$_{16}$" 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 (A_7 to A_0) of an address to be written should be incremented by 2 from "00$_{16}$" to "FF$_{16}$".
  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 programing. 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".)

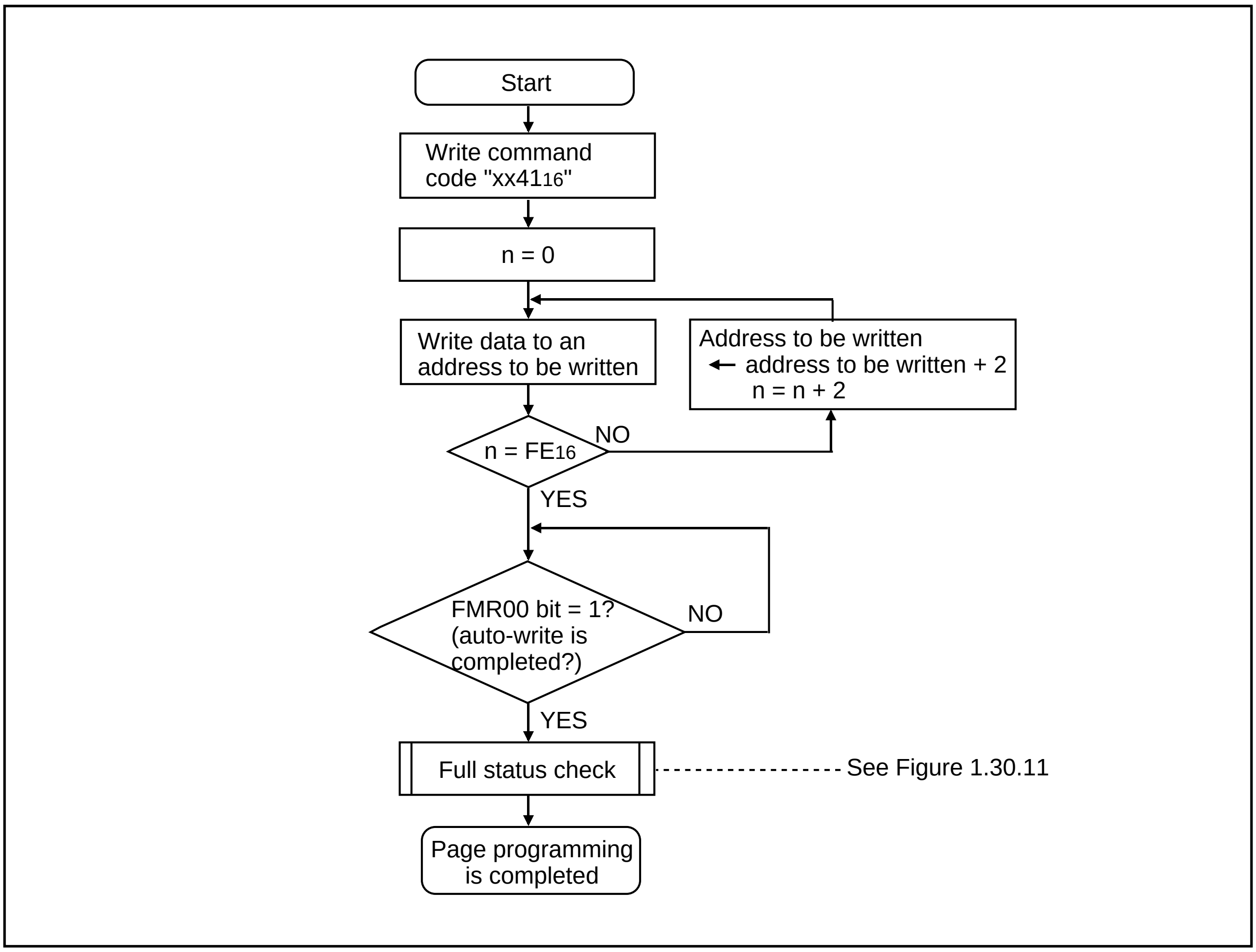

Start
Write command code "xx41₁₆"
n = 0
Write data to an address to be written
Address to be written ← address to be written + 2
n = n + 2
n = FE₁₆
NO
YES
FMR00 bit = 1? (auto-write is completed?)
NO
YES
Full status check
See Figure 1.30.11
Page programming is completed

Figure 1.30.6. Page Program Command

- **Block Erase**

The block erase command erases blocks by block.
Command code "xx20_{16}" should be written in the first bus cycle and "xx$D0_{16}$" 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".)

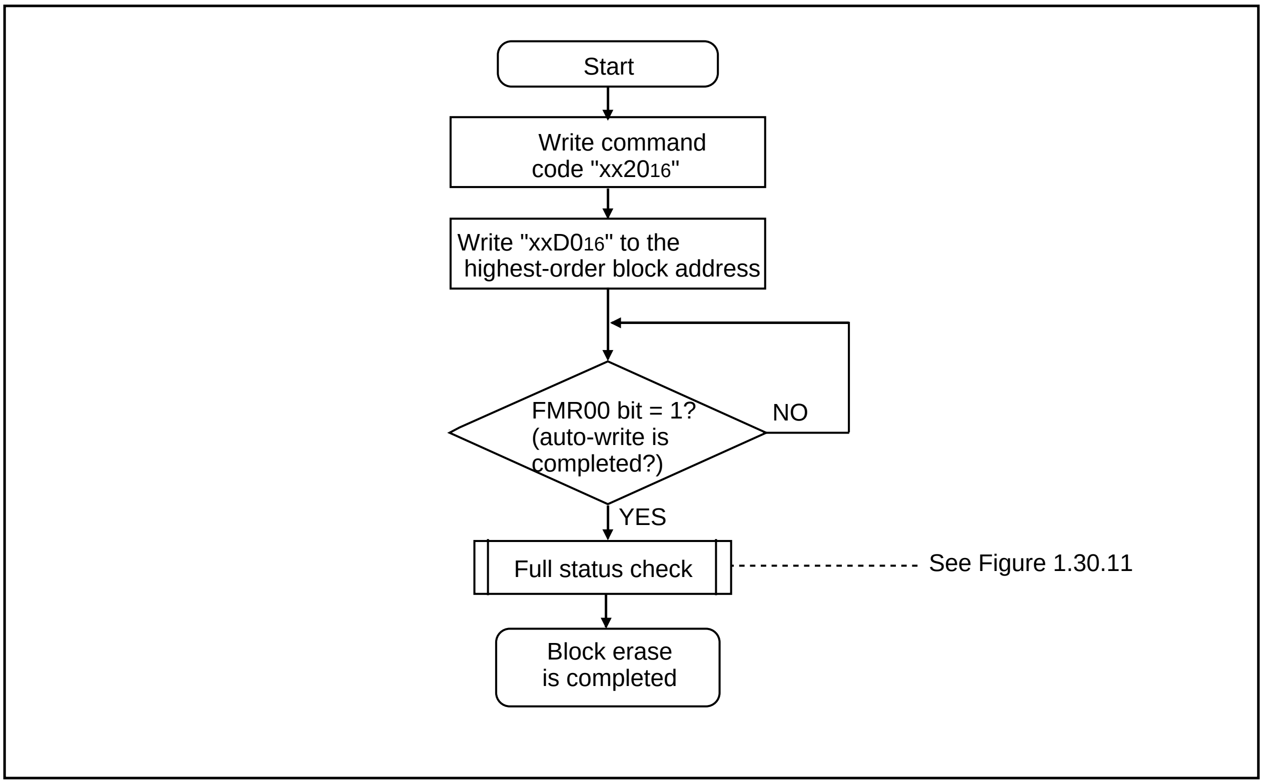

Start
Write command code "xx2016"
Write "xxD016" to the highest-order block address
FMR00 bit = 1? (auto-write is completed?)
NO
YES
Full status check
See Figure 1.30.11
Block erase is completed

Figure 1.30.7. Block Erase command

- **All Unlock Block Erase**

The all unlock block erase command erases all blocks.

Command code "xxA7$_{16}$" should be written in the first bus cycle and "xxD0$_{16}$" 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".)

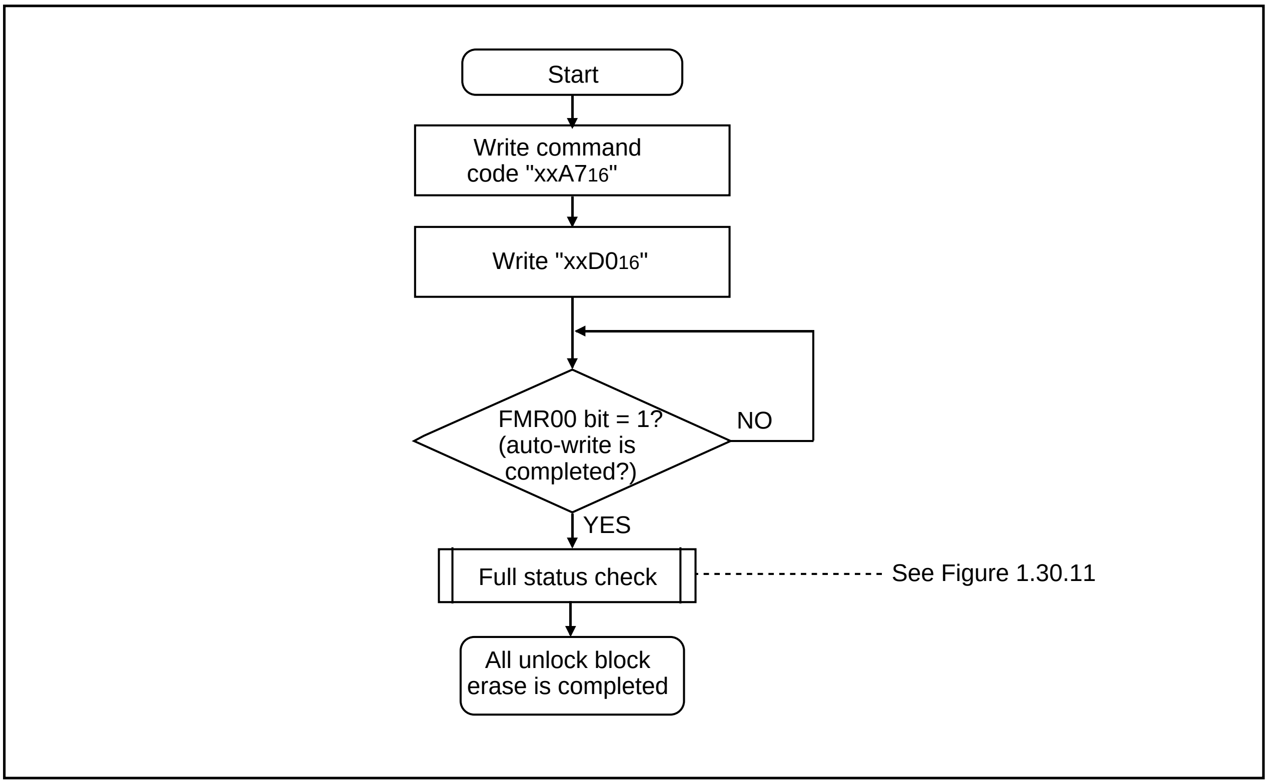

Start
Write command code "xxA7₁₆"
Write "xxD0₁₆"
FMR00 bit = 1? (auto-write is completed?)
NO
YES
Full status check
See Figure 1.30.11
All unlock block erase is completed

Figure 1.30.8. All Unlock Block Erase Command

- **Lock Bit Program**

The look bit program command sets the lock bit in a specified block to "0" (locked).
Command code "xx77_{16}" should be written in the first bus cycle and "xx$D0_{16}$" 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-erasure is in progress.
The FMR00 bit in the FMR0 register determines whether auto-erasure is completed.
The SDR register indicates auto-erasure log after auto-erasure 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.

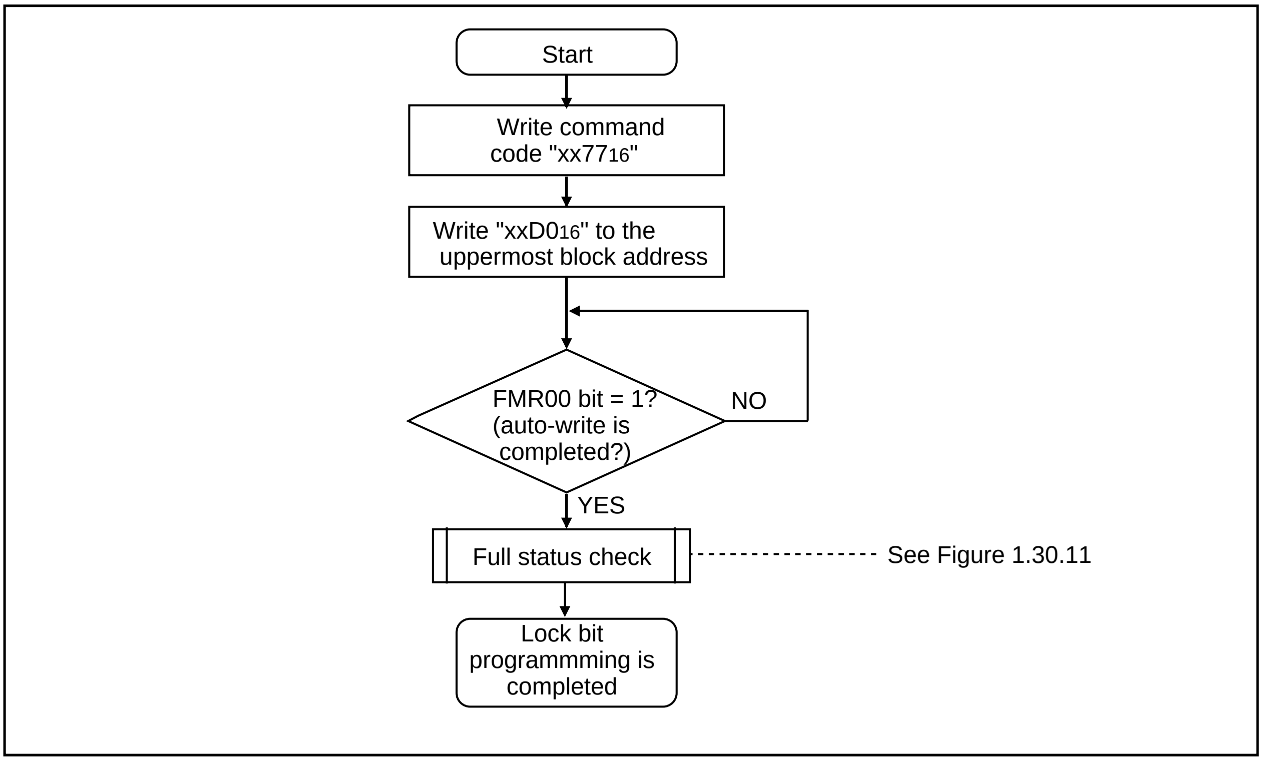

Start
Write command code "xx7716"
Write "xxD016" to the uppermost block address
FMR00 bit = 1? (auto-write is completed?)
NO
YES
Full status check
See Figure 1.30.11
Lock bit programmming is completed

Figure 1.30.9. Lock Bit Program Command

- **Lock Bit Read Status**

The lock bit read status command reads a lock bit status in a specified block.

Command code "xx71_{16}" should be written in the first bus cycle and "xx$D0_{16}$" 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 (D_6).

Figure 1.30.10 shows a flow chart of the lock bit read program.

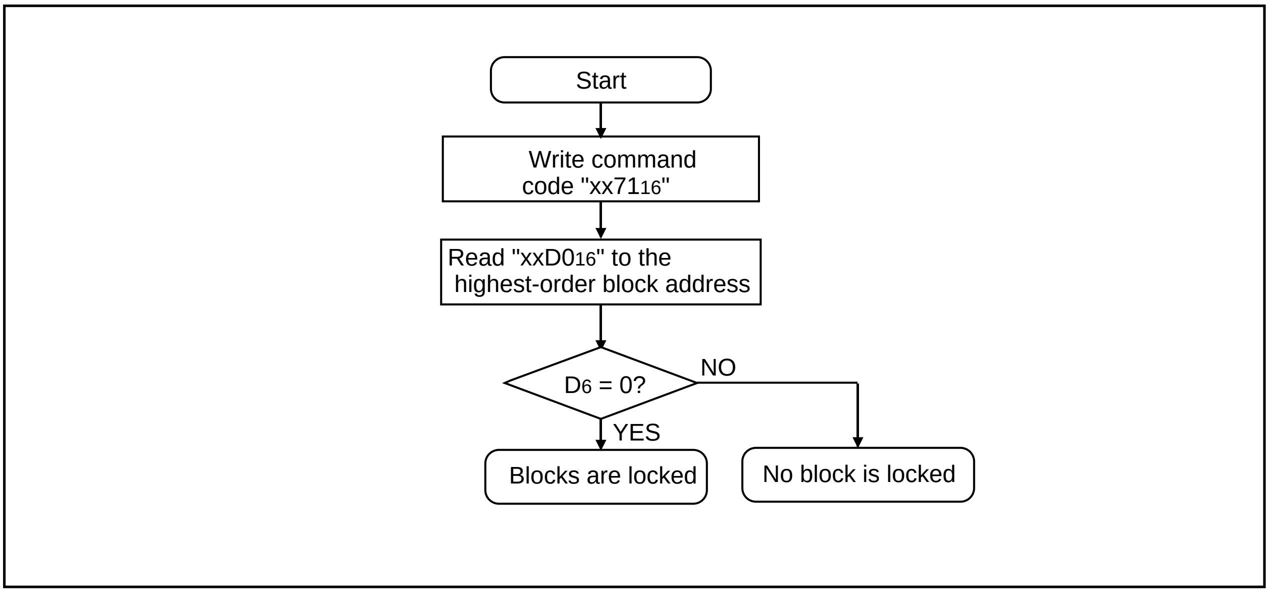

Start
Write command code "xx7116"
Read "xxD016" to the highest-order block address
D6 = 0?
NO
YES
Blocks are locked
No block is locked

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).

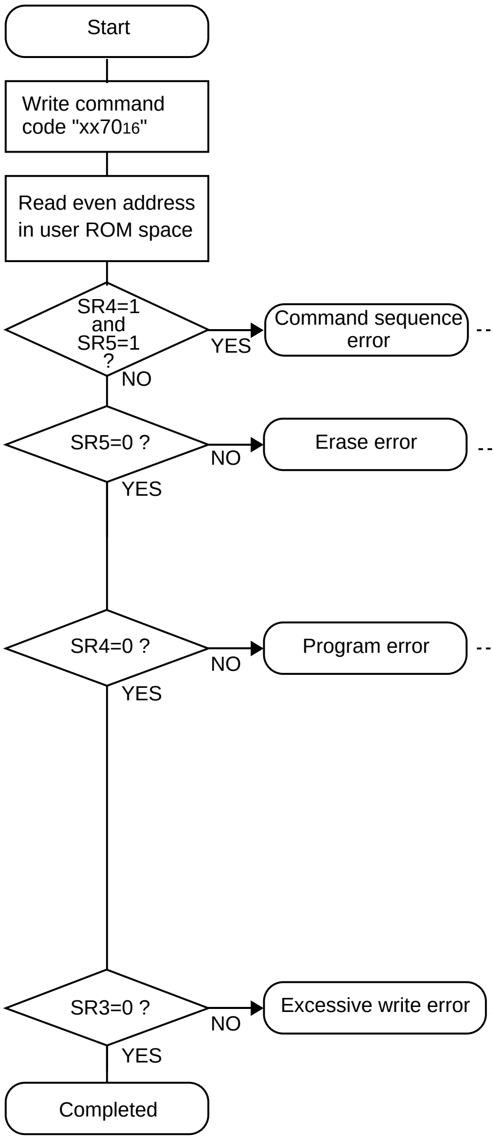
Table 1.30.5 lists errors and the SRD register (SR3 to SR5). Figure 1.30.11 shows a flow chart of the full status check and how to handle each error.

Table 1.30.5. Error and SRD Register

SRD register			Error	Error occurs
SR5	SR4	SR3		
1	1	0	Command sequence error	• When writing an incorrect command • When writing data without valid values ("xxD0$_{16}$" or "xxFF$_{16}$") in the second bus cycle of the lock bit program, block erase or all unlock block erase command[1]
1	0	0	Erase error	• When executing the block erase command on a locked block[2] • When executing the block erase or the all unlock block erase commands on an unlock block and erasing a block incorrectly
0	1	0	Program error	• When executing the page program command on a page in a locked block[2] • When executing the page program command on a page in an unlocked block and programming incorrectly • When exectuing the lock bit program command and programming incorrectly
0	0	1	Excessive write error	When writing excessively after the page program command is executed

Notes :

1. When writing command code "xxFF$_{16}$" 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.



Command sequence error:

(1) Execute the clear status register command to set the SR4 and SR5 bits to "0".
(2) Execute a correct command again.
Note : When a similar error occurs, that block cannot use.

Erase error:

(1) Execute the clear status register command to set the SR5 bit to "0".
(2) Execute the lock bit read status command to set the FMR02 bit to "1" if a lock bit in a block that an error occurs is set to "0".
(3) Execute the block erase or all unlocked block erase command again.
Note : When a similar error occurs, that block cannot be used.
If the lock bit is set to "1" in (2) above, that block cannot be used.

Program error:

[When the page program is executed]
(1) Execute the clear status register command to set the SR4 bit to "0".
(2) Execute the read lock bit status command to set the FMR02 bit to "1" if a lock bit in a block that an error occurs is set to "0".
(3) Execute the page program command again.
Note : When a similar error occurs, that block cannot be used.
If the lock bit is set to "1" in (2) above, that block cannot be used.
[When the lock bit program is executed]
(1) Execute the clear status register command to set the SR4 bit to "0".
(2) Set the FMR02 bit to "1".
(3) Execute the block erase command to erase a block that an error occurs.
(4) Execute the lock bit program command again.
Note : When a similar error occurs, that block cannot be used.

Excessive write error:

(1) Execute the clear status register command to set the SR3 bit to "0".
(2) Execute the block erase command to erase a block that an error occurs.
(3) Execute the page program command again.
Note : When a similar error occurs, that block cannot be used.

Note : When either of the SR5 to SR3 bits is set to "1", the page program, block erase, all unlocked block erase and lock bit program commands cannot be accepted.
The clear status register command should be executed before executing each command.

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 relocatabale 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 P8$_5$/$\overline{\text{NMI}}$ 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.

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 $0FFFFDF_{16}$, $0FFFFE3_{16}$, $0FFFFEB_{16}$, $0FFFFEF_{16}$, $0FFFFF3_{16}$, $0FFFFF7_{16}$ and $0FFFFFB_{16}$. A program set with ID code in these addresses should be written to the flash memory.

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
$\overline{\text{RESET}}$	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
$P0_0$ to $P0_7$	Input port P0	I	Input "H" or "L" or open
$P1_0$ to $P1_7$	Input port P1	I	Input "H" or "L" or open
$P2_0$ to $P2_7$	Input port P2	I	Input "H" or "L" or open
$P3_0$ to $P3_7$	Input port P3	I	Input "H" or "L" or open
$P4_0$ to $P4_7$	Input port P4	I	Input "H" or "L" or open
$P5_0$	$\overline{\text{CE}}$ input	I	Input "H"
$P5_5$	$\overline{\text{EPM}}$ input	I	Input "L"
$P5_1$ to $P5_4$, $P5_6$, $P5_7$	Input port P5	I	Input "H" or "L" or open
$P6_0$ to $P6_3$	Input port P6	I	Input "H" or "L" or open
$P6_4$	BUSY output	O	Standard serial mode 1: BUSY signal output pin Standard serial mode 2: Monitors to check a program operation
$P6_5$	SCLK input	I	Standard serial mode 1: Serial clock input pin Standard serial mode 2: Input "L"
$P6_6$	Data input RxD	I	Serial data input pin
$P6_7$	Data ouput TxD	O	Serial data output pin[1]
$P7_0$ to $P7_7$	Input port P7	I	Input "H" or "L" or open
$P8_0$ to $P8_4$, $P8_6$, $P8_7$	Input port P8	I	Input "H" or "L" or open
$P8_5$	$\overline{\text{NMI}}$ input	I	Connect $P8_5$ to Vcc
$P9_0$ to $P9_7$	Input port P9	I	Input "H" or "L" or open
$P10_0$ to $P10_7$	Input port P10	I	Input "H" or "L" or open
$P11_0$ to $P11_4$	Input port P11	I	Input "H" or "L" or open[2]
$P12_0$ to $P12_7$	Input port P12	I	Input "H" or "L" or open[2]
$P13_0$ to $P13_7$	Input port P13	I	Input "H" or "L" or open[2]
$P14_0$ to $P14_6$	Input port P14	I	Input "H" or "L" or open[2]
$P15_0$ to $P15_7$	Input port P15	I	Input "H" or "L" or open[2]

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.

Mode settings

Signal	Value
CNVss	Vcc
$\overline{\text{EPM}}$	Vss
$\overline{\text{RESET}}$	Vss >> Vcc
$\overline{\text{CE}}$	Vcc

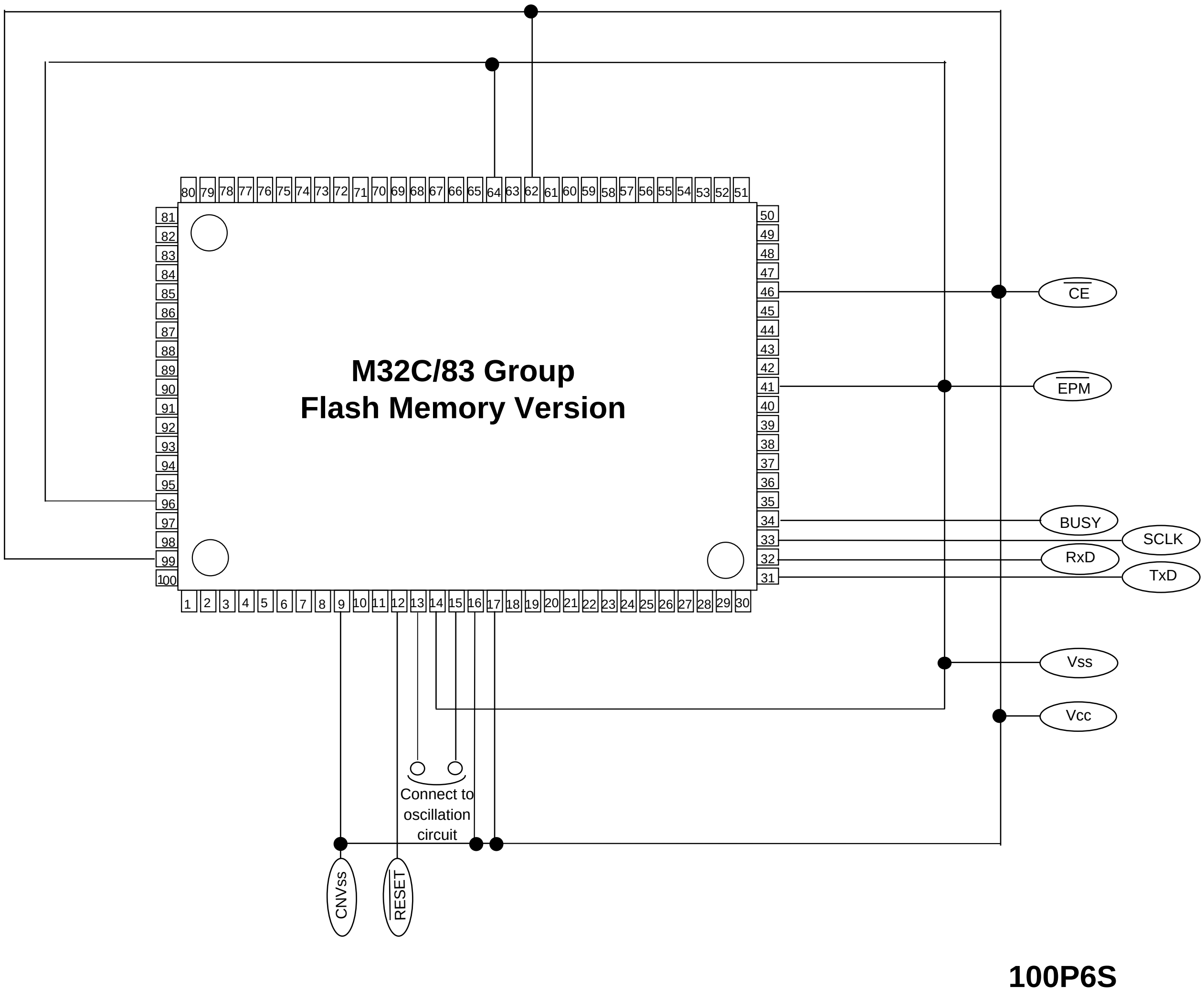
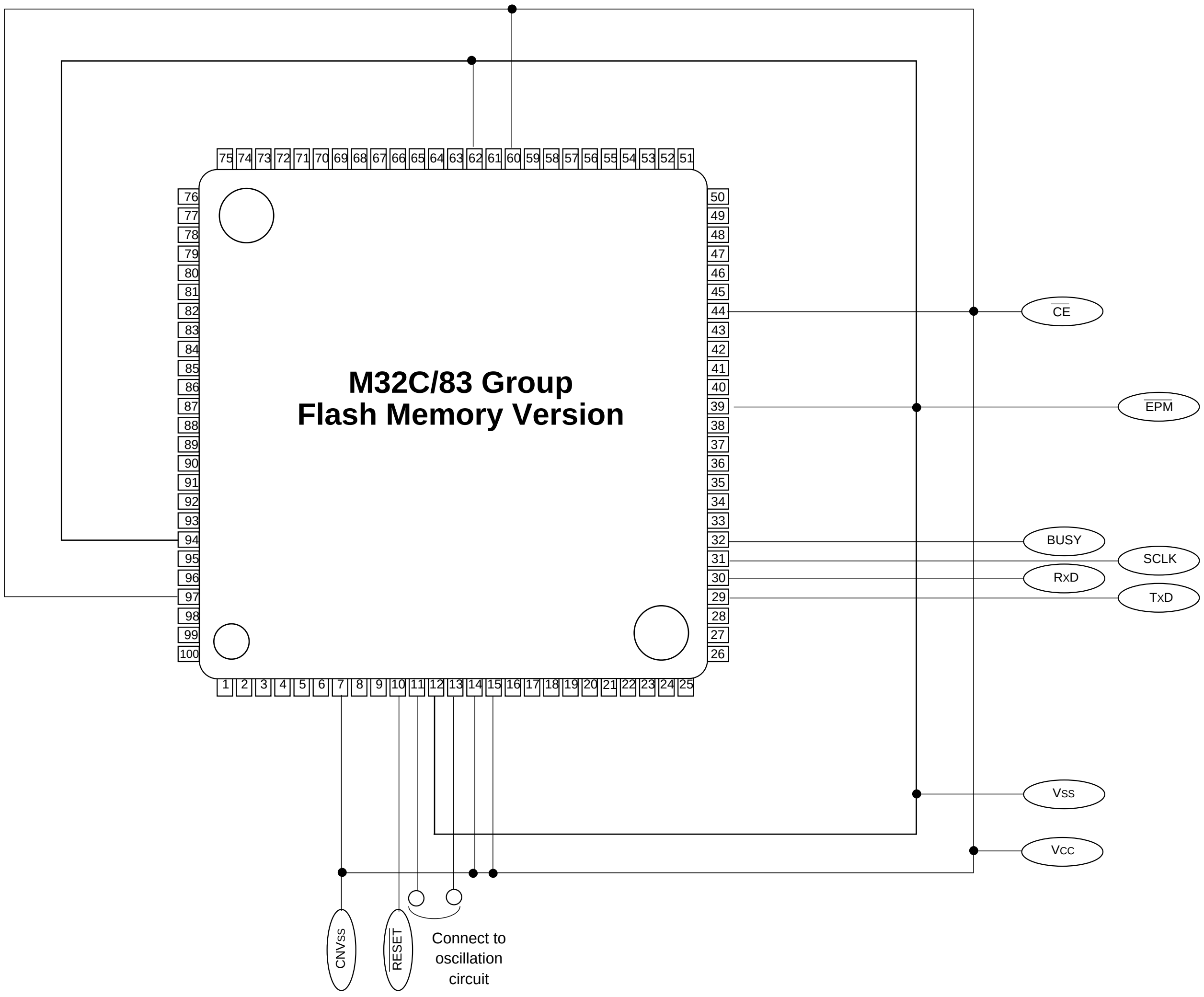


Figure 1.30.12. Pin Connections in Standard Serial I/O Mode (1)

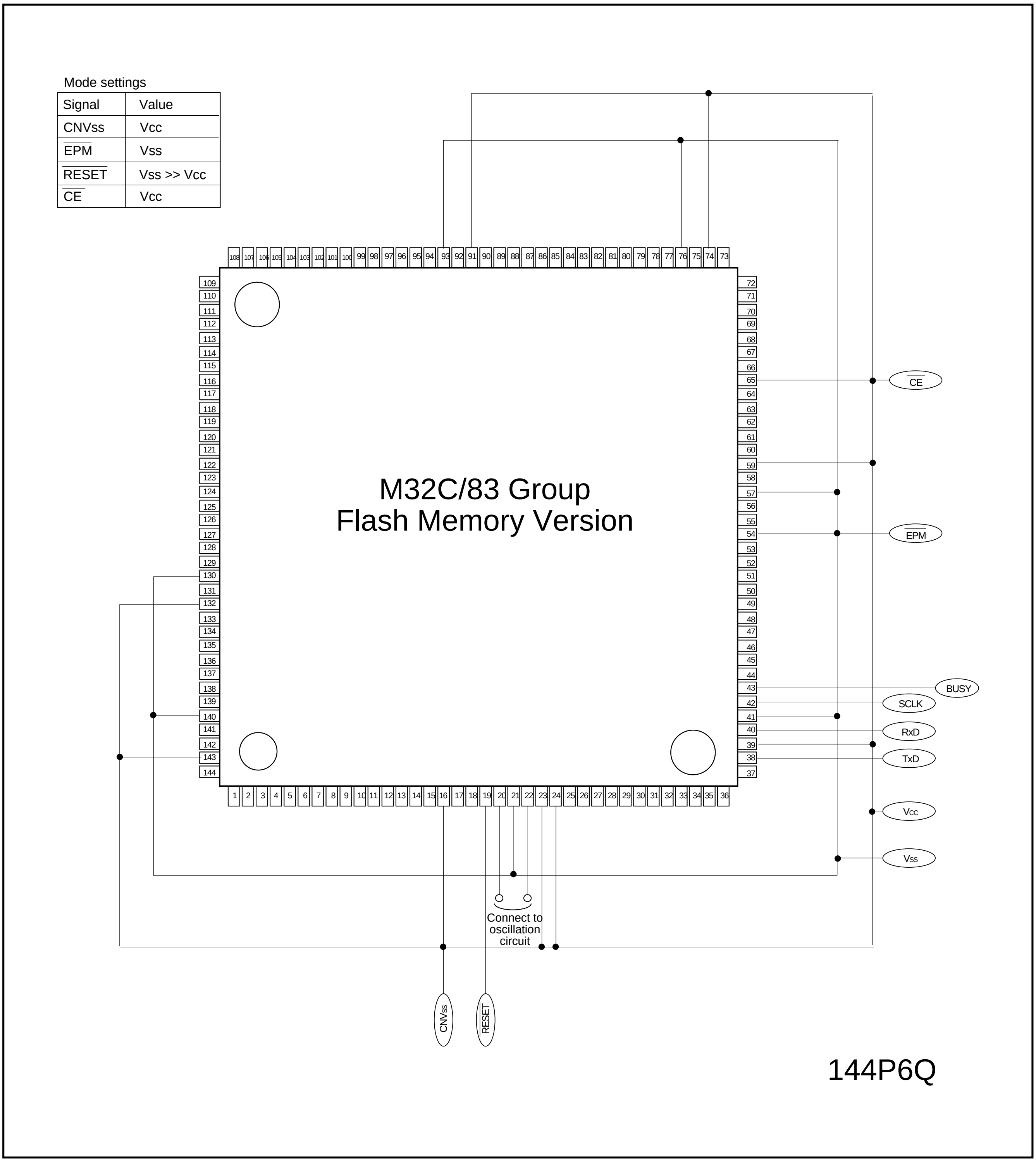
Mode settings

Signal	Value
CNVss	Vcc
$\overline{\text{EPM}}$	Vss
$\overline{\text{RESET}}$	Vss >> Vcc
$\overline{\text{CE}}$	Vcc



100P6Q

Figure 1.30.13. Pin Connections in Standard Serial I/O Mode (2)



Mode settings

Signal	Value
CNVss	Vcc
$\overline{\text{EPM}}$	Vss
$\overline{\text{RESET}}$	Vss >> Vcc
$\overline{\text{CE}}$	Vcc

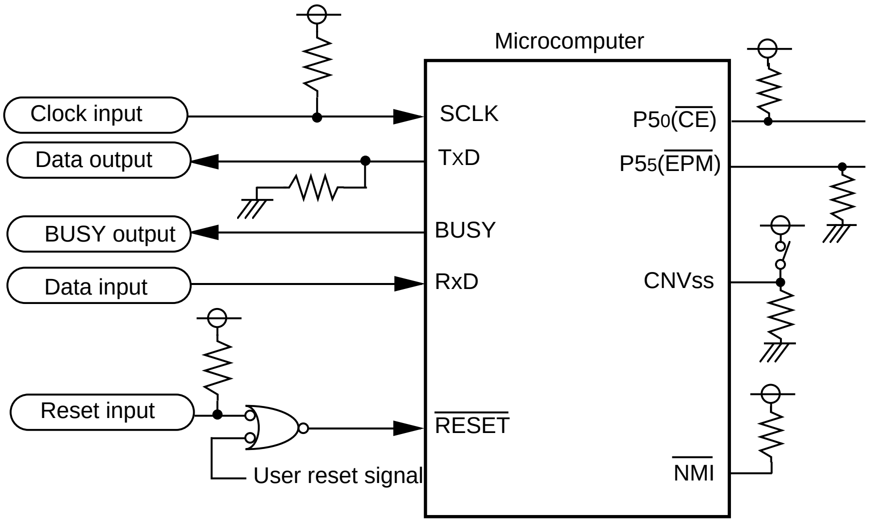
Figure 1.30.14. Pin Connections in Standard Serial I/O Mode (3)

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.

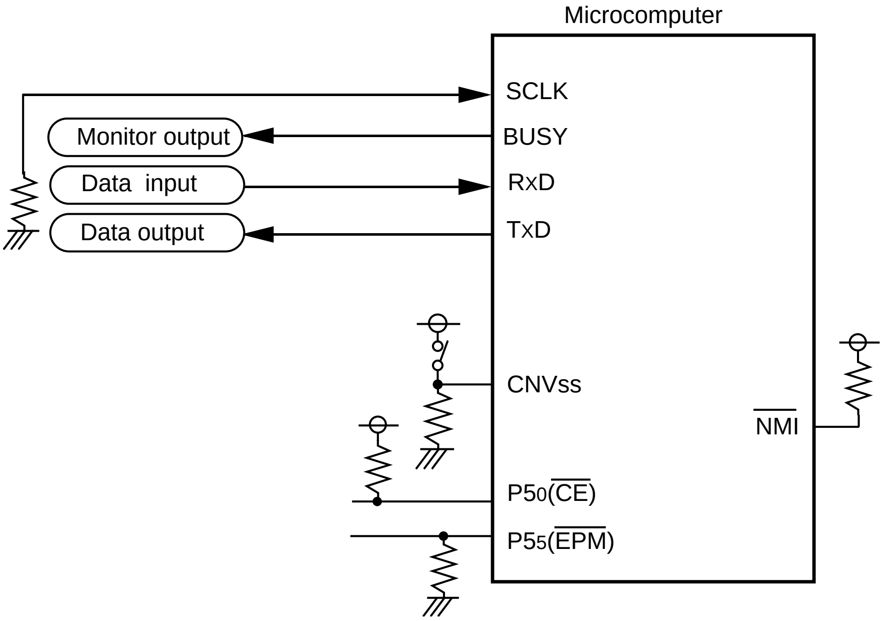
Example of Circuit Application in the Standard Serial I/O Mode

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.



(1) Control pins and external circuitry varies depending on a serial writer. Refer to the user's manual for the serial writer.
(2) In this example, a selector changes over between single-chip mode and standard serial I/O mode.
(3) When an user reset signal may be set to "L" in standard serial I/O mode, a connection between an user reset signal and the $\overline{\text{RESET}}$ pin should be broken by, for example, a jumper switch.

Figure 1.30.15. Circuit Application in Standard Serial I/O Mode 1



In this example, a selector changes over between single-chip mode and standard serial I/O mode.

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 $0FFE000_{16}$ to $0FFFFFF_{16}$. When rewriting the boot ROM space, rewrite this address range only. (Avoid accessing other than addresses $0FFE000_{16}$ to $0FFFFFF_{16}$.)

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 "00_2" (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

Table 1.31.1. Absolute Maximum Ratings

Symbol	Parameter		Condition	Value	Unit
V_{CC}	Supply voltage		V_{CC}=AV_{CC}	-0.3 to 6.0	V
AV_{CC}	Analog supply voltage		V_{CC}=AV_{CC}	-0.3 to 6.0	V
V_I	Input voltage	$\overline{RESET}$, CNVss, BYTE, $P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_2$-$P7_7$, $P8_0$-$P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$, $P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$[1], V_{REF}, X_{IN}		-0.3 to Vcc+0.3	V
		$P7_0$, $P7_1$		-0.3 to 6.0	V
V_O	Output voltage	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_2$-$P7_7$, $P8_0$-$P8_4$, $P8_6$, $P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$, $P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$[1], V_{REF}, X_{IN}		-0.3 to Vcc+0.3	V
		$P7_0$, $P7_1$		-0.3 to 6.0	V
Pd	Power consumption		Topr=25°C	500	mW
Topr	Operating ambient temperature			-20 to 85	°C
Tstg	Storage temperature			-65 to 150	°C

Notes :

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

Table 1.31.2. Recommended Operating Conditions (VCC = 3.0V to 5.5V at Topr = – 20 to 85°C)

Symbol	Parameter			Standard			Unit
				Min	Typ	Max	
VCC	Supply voltage(When VDC-ON)			3.0	5.0	5.5	V
	Supply voltage(When VDC-pass through)			3.0	3.3	3.6	V
AVCC	Analog supply voltage				VCC		V
VSS	Supply voltage				0		V
AVSS	Analog supply voltage				0		V
VIH	Input "H" voltage	P2$_0$-P2$_7$, P3$_0$-P3$_7$, P4$_0$-P4$_7$, P5$_0$-P5$_7$, P6$_0$-P6$_7$, P7$_2$-P7$_7$, P8$_0$-P8$_7$[3], P9$_0$-P9$_7$, P10$_0$-P10$_7$, P11$_0$-P11$_4$, P12$_0$-P12$_7$, P13$_0$-P13$_7$, P14$_0$-P14$_6$, P15$_0$-P15$_7$[4], XIN, $\overline{\text{RESET}}$, CNVss, BYTE		0.8Vcc		Vcc	V
		P7$_0$, P7$_1$		0.8Vcc		6.0	V
		P0$_0$-P0$_7$, P1$_0$-P1$_7$ (In single-chip mode)		0.8Vcc		Vcc	V
		P0$_0$-P0$_7$, P1$_0$-P1$_7$ (In memory expansion and microprocessor modes)		0.5Vcc		Vcc	V
VIL	Input "L" voltage	P2$_0$-P2$_7$, P3$_0$-P3$_7$, P4$_0$-P4$_7$, P5$_0$-P5$_7$, P6$_0$-P6$_7$, P7$_0$-P7$_7$, P8$_0$-P8$_7$[3], P9$_0$-P9$_7$, P10$_0$-P10$_7$, P11$_0$-P11$_4$, P12$_0$-P12$_7$, P13$_0$-P13$_7$, P14$_0$-P14$_6$, P15$_0$-P15$_7$[4], XIN, $\overline{\text{RESET}}$, CNVss, BYTE		0		0.2Vcc	V
		P0$_0$-P0$_7$, P1$_0$-P1$_7$ (In single-chip mode)		0		0.2Vcc	V
		P0$_0$-P0$_7$, P1$_0$-P1$_7$ (In memory expansion and microprocessor modes)		0		0.16Vcc	V
IOH(peak)	Peak output "H" current[2]	P0$_0$-P0$_7$, P1$_0$-P1$_7$, P2$_0$-P2$_7$, P3$_0$-P3$_7$, P4$_0$-P4$_7$, P5$_0$-P5$_7$, P6$_0$-P6$_7$, P7$_2$-P7$_7$, P8$_0$-P8$_4$, P8$_6$, P8$_7$, P9$_0$-P9$_7$, P10$_0$-P10$_7$, P11$_0$-P11$_4$, P12$_0$-P12$_7$, P13$_0$-P13$_7$, P14$_0$-P14$_6$, P15$_0$-P15$_7$[4]				-10.0	mA
IOH(avg)	Average output "H" current[1]	P0$_0$-P0$_7$, P1$_0$-P1$_7$, P2$_0$-P2$_7$, P3$_0$-P3$_7$, P4$_0$-P4$_7$, P5$_0$-P5$_7$, P6$_0$-P6$_7$, P7$_2$-P7$_7$, P8$_0$-P8$_4$, P8$_6$, P8$_7$, P9$_0$-P9$_7$, P10$_0$-P10$_7$, P11$_0$-P11$_4$, P12$_0$-P12$_7$, P13$_0$-P13$_7$, P14$_0$-P14$_6$, P15$_0$-P15$_7$[4]				-5.0	mA
IOL(peak)	Peak output "L" current[2]	P0$_0$-P0$_7$, P1$_0$-P1$_7$, P2$_0$-P2$_7$, P3$_0$-P3$_7$, P4$_0$-P4$_7$, P5$_0$-P5$_7$, P6$_0$-P6$_7$, P7$_0$-P7$_7$, P8$_0$-P8$_4$, P8$_6$, P8$_7$, P9$_0$-P9$_7$, P10$_0$-P10$_7$, P11$_0$-P11$_4$, P12$_0$-P12$_7$, P13$_0$-P13$_7$, P14$_0$-P14$_6$, P15$_0$-P15$_7$[4]				10.0	mA
IOL(avg)	Average output "L" current[1]	P0$_0$-P0$_7$, P1$_0$-P1$_7$, P2$_0$-P2$_7$, P3$_0$-P3$_7$, P4$_0$-P4$_7$, P5$_0$-P5$_7$, P6$_0$-P6$_7$, P7$_0$-P7$_7$, P8$_0$-P8$_4$, P8$_6$, P8$_7$, P9$_0$-P9$_7$, P10$_0$-P10$_7$, P11$_0$-P11$_4$, P12$_0$-P12$_7$, P13$_0$-P13$_7$, P14$_0$-P14$_6$, P15$_0$-P15$_7$[4]				5.0	mA
f(XIN)	Main clock input frequency	VDC-ON	Vcc=4.2 to 5.5V	0		30	MHz
			Vcc=3.0 to 4.2V	0		20	MHz
		VDC-pass through	Vcc=3.0 to 3.6V	0		20	MHz
f(XCIN)	Sub clock oscillation frequency				32.768		kHz

Notes :

1. Output current is averaged within 100ms.
2. Total IOL (peak) for ports P0, P1, P2, P8$_6$, P8$_7$, P9, P10, P11, P14 and P15 must be 80mA and below. Total IOH (peak) for ports P0, P1, P2, P8$_6$, P8$_7$, P9, P10, P11, P14 and P15 must be -80mA and below. Total IOL (peak) for ports P3, P4, P5, P6, P7,P8$_0$ to P8$_4$, P12 and P13 must be 80mA and below. Total IOH (peak) for ports P3, P4, P5, P6, P7$_2$ to P7$_7$, P8$_0$ to P8$_4$, P12 and P13 must be -80mA and below.
3. VIH and VIL reference for P8$_7$ applies to use P8$_7$ as a programmable input ports. It do not apply to use P8$_7$ as XCIN.
4. Ports P11 to P15 are provided in the 144-pin package only.

Table 1.31.3. Electrical Characteristics (V_{CC}=4.2 to 5.5V, V_{SS}=0V at Topr= –20 to 85°C, f(X_{IN})=30MHz unless otherwise specified)

V_{CC} = 5V

Symbol	Parameter		Condition	Standard			Unit
				Min	Typ	Max	
V_{OH}	Output "H" voltage	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_0$-$P7_7$, $P8_0$-$P8_4$, $P8_6$, $P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$, $P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$[1]	I_{OH}=-5mA	3.0			V
		$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_0$-$P7_7$, $P8_0$-$P8_4$, $P8_6$, $P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$, $P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$[1]	I_{OH}=-200µA	4.7			V
		X_{OUT}	I_{OH}=-1mA	3.0			V
		X_{COUT}	No load applied		3.3		V
V_{OL}	Output "L" voltage	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_0$-$P7_7$, $P8_0$-$P8_4$, $P8_6$, $P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$, $P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$[1]	I_{OL}=5mA			2.0	V
		$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_0$-$P7_7$, $P8_0$-$P8_4$, $P8_6$, $P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$, $P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$[1]	I_{OL}=200µA			0.45	V
		X_{OUT}	I_{OL}=1mA			2.0	V
		X_{COUT}	No load applied		0		V
V_{T+}-V_{T-}	Hysteresis	$\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$-$TA4_{IN}$, $TB0_{IN}$-$TB5_{IN}$, $\overline{INT0}$-$\overline{INT5}$, $\overline{AD_{TRG}}$, $\overline{CTS0}$-$\overline{CTS4}$, CLK0-CLK4, $TA0_{OUT}$-$TA4_{OUT}$, $\overline{NMI}$, $\overline{KI0}$-$\overline{KI3}$, RxD0-RxD4, SCL0-SCL4, SDA0-SDA4		0.2		1.0	V
		$\overline{RESET}$		0.2		1.8	V
I_{IH}	Input "H" current	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_2$-$P7_7$, $P8_0$-$P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$, $P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$[1], X_{IN}, $\overline{RESET}$, CNVss, BYTE	V_I=5V			5.0	µA
I_{IL}	Input "L" current	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_2$-$P7_7$, $P8_0$-$P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$, $P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$[1], X_{IN}, $\overline{RESET}$, CNVss, BYTE	V_I=0V			-5.0	µA
R_{PULLUP}	Pull-up resistance	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_2$-$P7_7$, $P8_0$-$P8_4$, $P8_6$, $P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$, $P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$[1]	V_I=0V	30	50	167	kΩ
R_{fXIN}	Feedback resistance	X_{IN}			1.5		MΩ
R_{fXCIN}	Feedback resistance	X_{CIN}			10		MΩ
V_{RAM}	RAM retention voltage	VDC-ON		2.5			V
I_{CC}	Supply current	Measurment condition: In sigle-chip mode, output pins are open and other pins are connected to Vss.	f(X_{IN})=30MHz, square wave, no division		38	54	mA
			f(X_{CIN})=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.

$V_{CC} = 5V$

Table 1.31.4. A-D Conversion Characteristics ($V_{CC} = AV_{CC} = V_{REF}$ = 4.2 to 5.5V, Vss = AV_{SS} = 0V at Topr = –20 to 85°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	AN_0 to AN_7 AN_{EX0}, AN_{EX1}			±3	LSB
			External op-amp connection mode			±7	LSB
DNL	Differential nonlinearity error					±1	LSB
-	Offset error					±3	LSB
-	Gain error					±3	LSB
R_{LADDER}	Ladder resistance	$V_{REF} = V_{CC}$		8		40	kΩ
t_{CONV}	Conversion time(10bit)			3.3			µs
t_{CONV}	Conversion time(8bit)			2.8			µs
t_{SAMP}	Sampling time			0.3			µs
V_{REF}	Reference voltage			2		V_{CC}	V
V_{IA}	Analog input voltage			0		V_{REF}	V

Notes :

1. Divide f(X_{IN}), if exceeding 10 MHz, to keep ØAD 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 Topr = –20 to 85°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	µs
R_O	Output resistance		4	10	20	kΩ
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 "00_{16}".
   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).

VCC = 5V

Timing Requirements (VCC = 4.2 to 5.5V, VSS = 0V at Topr = –20 to 85°C unless otherwise specified)

Table 1.31.6. External Clock Input

Symbol	Parameter	Standard		Unit
		Min	Max	
t_c	External clock input cycle time	33		ns
$t_{w(H)}$	External clock input "H" pulse width	13		ns
$t_{w(L)}$	External clock input "L" pulse width	13		ns
t_r	External clock rising-edge time		5	ns
t_f	External clock falling-edge time		5	ns

Table 1.31.7. Memory Expansion and Microprocessor Modes

Symbol	Parameter	Standard		Unit
		Min	Max	
$t_{ac1(RD-DB)}$	Data input access time (RD standard, no wait)		(Note 1)	ns
$t_{ac1(AD-DB)}$	Data input access time (AD standard, CS standard, no wait)		(Note 1)	ns
$t_{ac2(RD-DB)}$	Data input access time (RD standard, with wait)		(Note 1)	ns
$t_{ac2(AD-DB)}$	Data input access time (AD standard, CS standard, with wait)		(Note 1)	ns
$t_{ac3(RD-DB)}$	Data input access time (RD standard, when accessing multiplex bus space)		(Note 1)	ns
$t_{ac3(AD-DB)}$	Data input access time (AD standard, CS standard, when accessing multiplex bus space)		(Note 1)	ns
$t_{ac4(RAS-DB)}$	Data input access time (RAS standard, DRAM access)		(Note 1)	ns
$t_{ac4(CAS-DB)}$	Data input access time (CAS standard, DRAM access)		(Note 1)	ns
$t_{ac4(CAD-DB)}$	Data input access time (CAD standard, DRAM access)		(Note 1)	ns
$t_{su(DB-BCLK)}$	Data input setup time	26		ns
$t_{su(RDY-BCLK)}$	$\overline{\text{RDY}}$ input setup time	26		ns
$t_{su(HOLD-BCLK)}$	$\overline{\text{HOLD}}$ input setup time	30		ns
$t_{h(RD-DB)}$	Data input hold time	0		ns
$t_{h(CAS-DB)}$	Data input hold time	0		ns
$t_{h(BCLK-RDY)}$	$\overline{\text{RDY}}$ input hold time	0		ns
$t_{h(BCLK-HOLD)}$	$\overline{\text{HOLD}}$ input hold time	0		ns
$t_{d(BCLK-HLDA)}$	$\overline{\text{HLDA}}$ output delay time		25	ns

Notes :

1. A vlaue 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.

$$t_{ac1(RD-DB)} = \frac{10^9}{f_{(BCLK)} \times 2} - 35 \quad \text{[ns]}$$

$$t_{ac1(AD-DB)} = \frac{10^9}{f_{(BCLK)}} - 35 \quad \text{[ns]}$$

$$t_{ac2(RD-DB)} = \frac{10^9 \times m}{f_{(BCLK)} \times 2} - 35 \quad \text{[ns] (m=3 when 1 wait, m=5 and when 2 waits and m=7 when 3 waits)}$$

$$t_{ac2(AD-DB)} = \frac{10^9 \times n}{f_{(BCLK)}} - 35 \quad \text{[ns] (n=2 when 1 wait, n=3 when 2 waits and n=4 when 3 waits)}$$

$$t_{ac3(RD-DB)} = \frac{10^9 \times m}{f_{(BCLK)} \times 2} - 35 \quad \text{[ns] (m=3 when 2 waits and m=5 when 3 waits)}$$

$$t_{ac3(AD-DB)} = \frac{10^9 \times n}{f_{(BCLK)} \times 2} - 35 \quad \text{[ns] (n=5 when 2 waits and n=7 when 3 waits)}$$

$$t_{ac4(RAS-DB)} = \frac{10^9 \times m}{f_{(BCLK)} \times 2} - 35 \quad \text{[ns] (m=3 when 1 wait and m=5 when 2 waits)}$$

$$t_{ac4(CAS-DB)} = \frac{10^9 \times n}{f_{(BCLK)} \times 2} - 35 \quad \text{[ns] (n=1 when 1 wait and n=3 when 2 waits)}$$

$$t_{ac4(CAD-DB)} = \frac{10^9 \times l}{f_{(BCLK)}} - 35 \quad \text{[ns] (l=1 when 1 wait and l=2 when 2 waits)}$$

VCC = 5V

Timing Requirements
(VCC = 4.2 to 5.5V, VSS = 0V at Topr = –20 to 85°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

VCC = 5V

Timing Requirements

(VCC = 4.2 to 5.5V, VSS = 0V at Topr = –20 to 85°C unless otherwise specified)

Table 1.31.13. Timer B Input (Count Source Input in eEvent Counter Mode)

Symbol	Parameter	Standard		Unit
		Min	Max	
$t_{c(TB)}$	TBi$_{IN}$ input cycle time (counted on one edge)	100		ns
$t_{w(TBH)}$	TBi$_{IN}$ input "H" pulse width (counted on one edge)	40		ns
$t_{w(TBL)}$	TBi$_{IN}$ input "L" pulse width (counted on one edge)	40		ns
$t_{c(TB)}$	TBi$_{IN}$ input cycle time (counted on both edges)	200		ns
$t_{w(TBH)}$	TBi$_{IN}$ input "H" pulse width (counted on both edges)	80		ns
$t_{w(TBL)}$	TBi$_{IN}$ 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	
$t_{c(TB)}$	TBi$_{IN}$ input cycle time	400		ns
$t_{w(TBH)}$	TBi$_{IN}$ input "H" pulse width	200		ns
$t_{w(TBL)}$	TBi$_{IN}$ input "L" pulse width	200		ns

Table 1.31.15. Timer B Input (Pulse Width Measurement Mode)

Symbol	Parameter	Standard		Unit
		Min	Max	
$t_{c(TB)}$	TBi$_{IN}$ input cycle time	400		ns
$t_{w(TBH)}$	TBi$_{IN}$ input "H" pulse width	200		ns
$t_{w(TBL)}$	TBi$_{IN}$ input "L" pulse width	200		ns

Table 1.31.16. A-D trigger Input

Symbol	Parameter	Standard		Unit
		Min	Max	
$t_{c(AD)}$	$\overline{\text{AD}_{\text{TRG}}}$ input cycle time (trigger available at minimum and above)	1000		ns
$t_{w(ADL)}$	$\overline{\text{AD}_{\text{TRG}}}$ 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 $\overline{\text{INTi}}$ Input

Symbol	Parameter	Standard		Unit
		Min	Max	
$t_{w(INH)}$	$\overline{\text{INTi}}$ input "H" pulse width	250		ns
$t_{w(INL)}$	$\overline{\text{INTi}}$ input "L" pulse width	250		ns

VCC = 5V

Switching Characteristics

(VCC = 4.2 to 5.5V, VSS = 0V at Topr = –20 to 85°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.

$$t_{d(DB-WR)} = \frac{10^9}{f_{(BCLK)}} - 20 \quad [ns]$$

$$t_{h(WR-DB)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [ns]$$

$$t_{h(WR-AD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [ns]$$

$$t_{h(WR-CS)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [ns]$$

$$t_{w(WR)} = \frac{10^9}{f_{(BCLK)} \times 2} - 15 \quad [ns]$$

Switching Characteristics
(V_{CC} = 4.2 to 5.5V, V_{SS} = 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	Measurment 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.

$$t_{d(DB-WR)} = \frac{10^9 \times n}{f_{(BCLK)}} - 20 \quad [ns]$$ (n=1 when 1 wait n=2 when 2 waits and n=3 when 3 waits)

$$t_{h(WR-DB)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [ns]$$

$$t_{h(WR-AD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [ns]$$

$$t_{h(WR-CS)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [ns]$$

$$t_{w(WR)} = \frac{10^9 \times n}{f_{(BCLK)} \times 2} - 15 \quad [ns]$$ (n=1 when 1 wait n=3 when 2 waits and n=5 when 3 waits)

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	
$t_{d(BCLK-AD)}$	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)		(Note 1)		ns
$t_{h(WR-AD)}$	Address output hold time (WR standard)		(Note 1)		ns
$t_{d(BCLK-CS)}$	Chip-select output delay time			18	ns
$t_{h(BCLK-CS)}$	Chip-select output hold time (BCLK standard)		-3		ns
$t_{h(RD-CS)}$	Chip-select output hold time (RD standard)		(Note 1)		ns
$t_{h(WR-CS)}$	Chip-select output hold time (WR standard)		(Note 1)		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			18	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		-5		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			18	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		-3		ns
$t_{d(DB-WR)}$	Data output delay time (WR standard)		(Note 1)		ns
$t_{h(WR-DB)}$	Data output hold time (WR standard)		(Note 1)		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time (BCLK standard)			18	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time (BCLK standard)		– 2		ns
$t_{d(AD-ALE)}$	ALE signal output delay time (address standard)		(Note 1)		ns
$t_{h(ALE-AD)}$	ALE signal output hold time (address standard)		(Note 1)		ns
$t_{dz(RD-AD)}$	Address output high-impedance start time			8	ns

Notes :

1. A value can be obtained from the following expressions, according to BCLK frequency.

$$t_{h(RD-AD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [ns]$$

$$t_{h(WR-AD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [ns]$$

$$t_{h(RD-CS)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [ns]$$

$$t_{h(WR-CS)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [ns]$$

$$t_{d(DB-WR)} = \frac{10^9 \times m}{f_{(BCLK)} \times 2} - 25 \quad [ns] \quad \text{(m=3 when 2 waits and m=5 when 3 waits)}$$

$$t_{h(WR-DB)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [ns]$$

$$t_{d(AD-ALE)} = \frac{10^9}{f_{(BCLK)} \times 2} - 20 \quad [ns]$$

$$t_{h(ALE-AD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [ns]$$

VCC = 5V

Switching Characteristics
(V_{CC} = 4.2 to 5.5V, V_{SS} = 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	
$t_{d(BCLK-RAD)}$	Row address output delay time	Figure 1.31.1		18	ns
$t_{h(BCLK-RAD)}$	Row address output hold time (BCLK standard)		-3		ns
$t_{d(BCLK-CAD)}$	Column address output delay time			18	ns
$t_{h(BCLK-CAD)}$	Column address output hold time (BCLK standard)		-3		ns
$t_{h(RAS-RAD)}$	Row address output hold time after RAS output		(Note 1)		ns
$t_{d(BCLK-RAS)}$	RAS output delay time (BCLK standard)			18	ns
$t_{h(BCLK-RAS)}$	RAS output hold time (BCLK standard)		-3		ns
t_{RP}	RAS "H" hold time		(Note 1)		ns
$t_{d(BCLK-CAS)}$	CAS output delay time (BCLK standard)			18	ns
$t_{h(BCLK-CAS)}$	CAS output hold time (BCLK standard)		-3		ns
$t_{d(BCLK-DW)}$	DW output delay time (BCLK standard)			18	ns
$t_{h(BCLK-DW)}$	DW output hold time (BCLK standard)		-5		ns
$t_{su(DB-CAS)}$	CAS output setup time after DB output		(Note 1)		ns
$t_{h(BCLK-DB)}$	DB signal output hold time (BCLK standard)		-7		ns
$t_{su(CAS-RAS)}$	CAS output setup time before RAS output (refresh)		(Note 1)		ns

Notes :

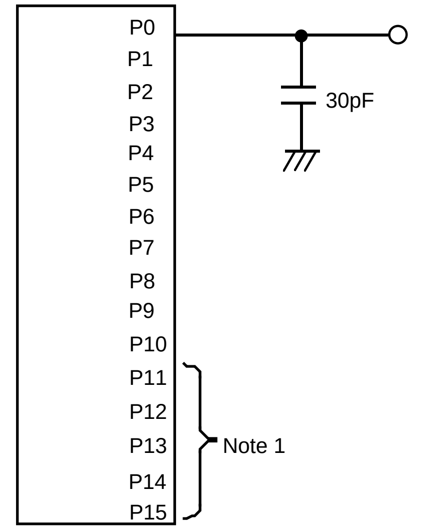
1. A value can be obtained from the following expressions, according to BCLK frequency.

$$t_{h(RAS-RAD)} = \frac{10^9}{f(BCLK) \times 2} - 13 \quad [ns]$$

$$t_{RP} = \frac{10^9}{f(BCLK) \times 2} \times 3 - 20 \quad [ns]$$

$$t_{su(DB-CAS)} = \frac{10^9}{f(BCLK)} - 20 \quad [ns]$$

$$t_{su(CAS-RAS)} = \frac{10^9}{f(BCLK) \times 2} - 13 \quad [ns]$$



Notes :

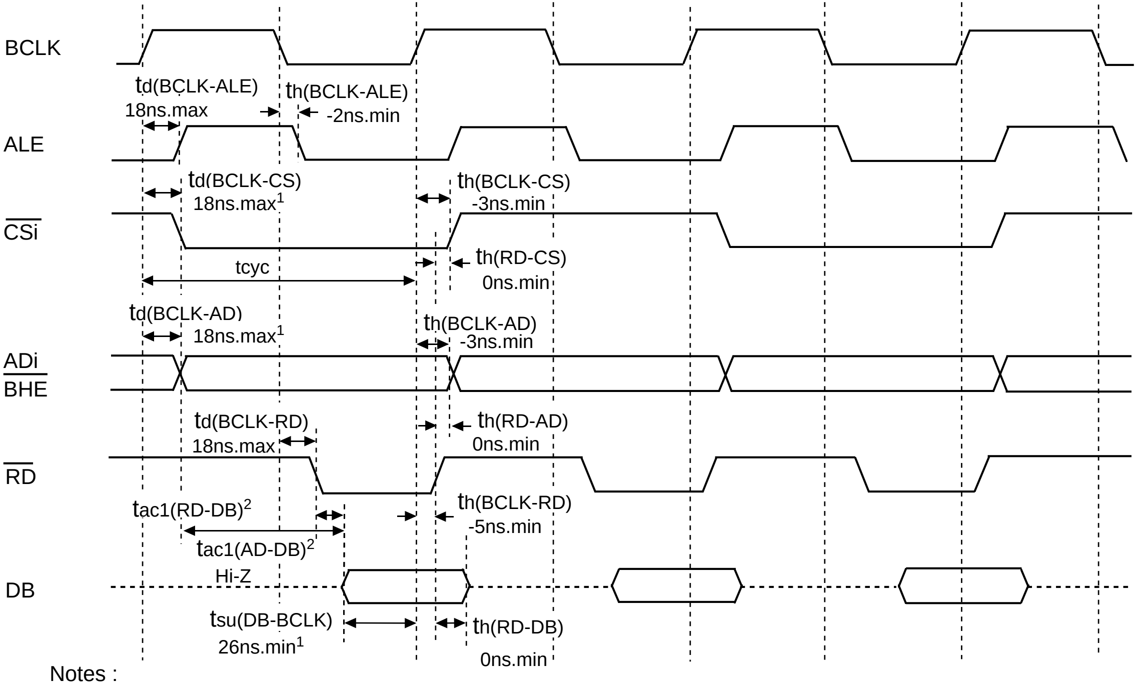
1. P11 to P15 are provided in the 144-pin package only.

Figure 1.31.1. Ports P0 to P15 Measurement Circuit

Vcc=5V

Memory expansion mode and microprocessor mode (without wait)

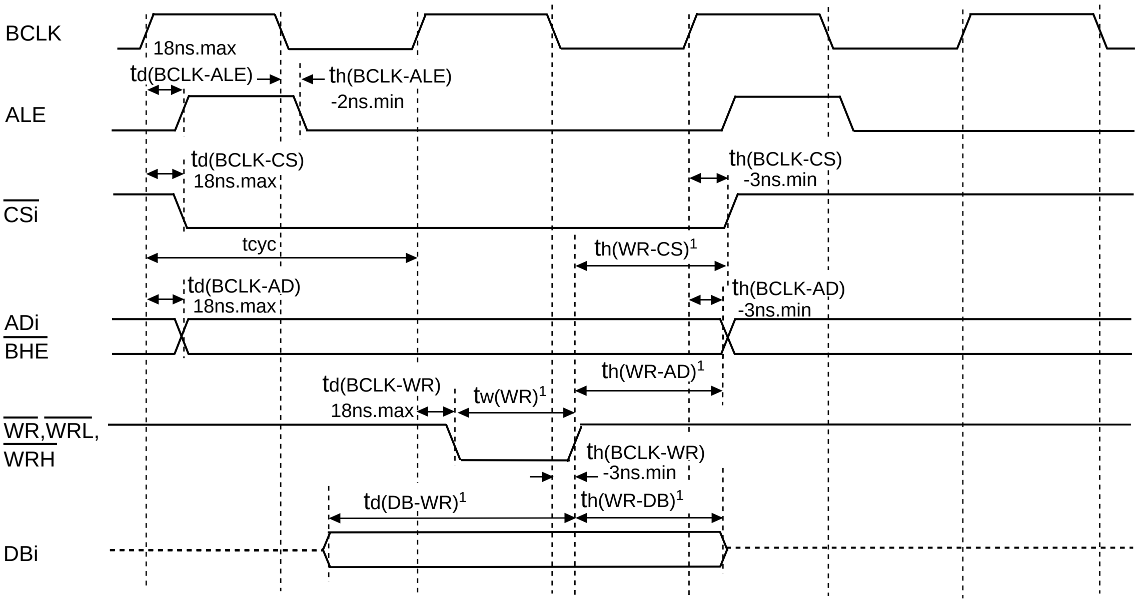
Read timing



Notes :

1. A value is guaranteed with no external factor. Maximum 35ns is guaranteed for $t_{d(BCLK-AD)}$+$t_{su(DB-BCLK)}$.
2. It depends on operation frequency.
   $t_{ac1(RD-DB)}$=(tcyc/2-35)ns.max
   $t_{ac1(AD-DB)}$=(tcyc-35)ns.max

Write timing (written in 2 cycles without wait)



Notes :

1. It depends on operation frequency.
   $t_{d(DB-WR)}$=(tcyc-20)ns.min
   $t_{h(WR-DB)}$=(tcyc/2-10)ns.min
   $t_{h(WR-AD)}$=(tcyc/2-10)ns.min
   $t_{h(WR-CS)}$=(tcyc/2-10)ns.min
   $t_{w(WR)}$=(tcyc/2-15)ns.min

Measurement conditions

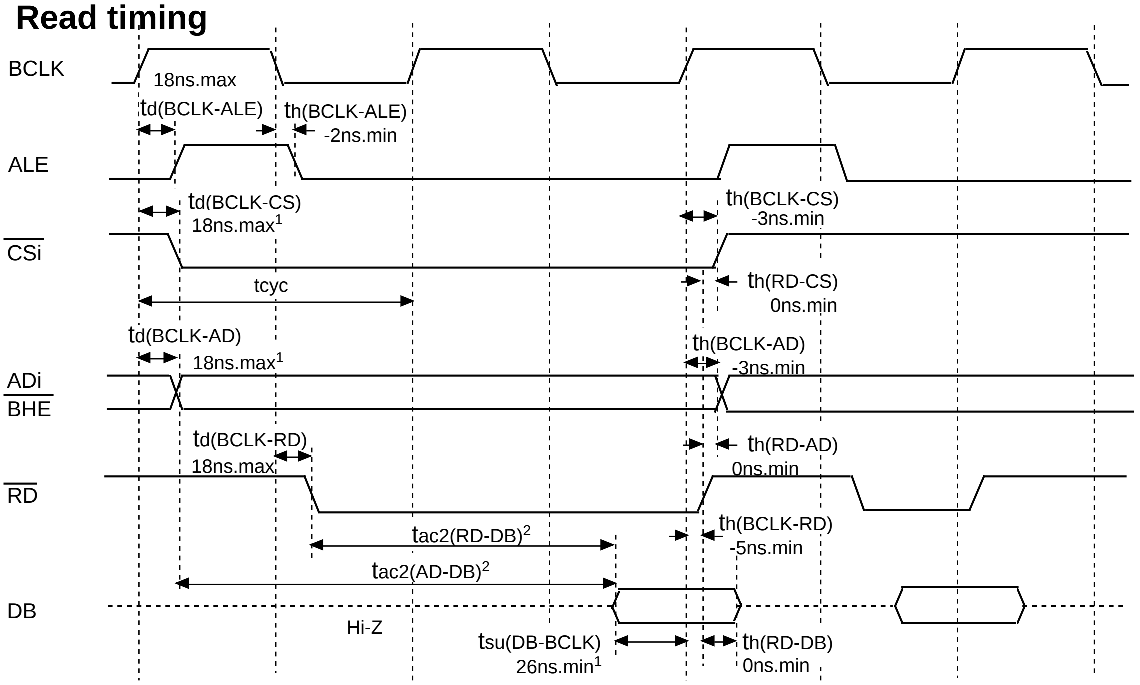
- V_{CC}=4.2 to 5.5V
- Input high or low voltage : V_{IH}=2.5V, V_{IL}=0.8V
- Output high or low voltage : V_{OH}=2.0V, V_{OL}=0.8V

Figure 1.31.2. V_{CC}=5V Timing Diagram (1)

Vcc=5V

Memory expansion mode and microprocessor mode (with a wait)

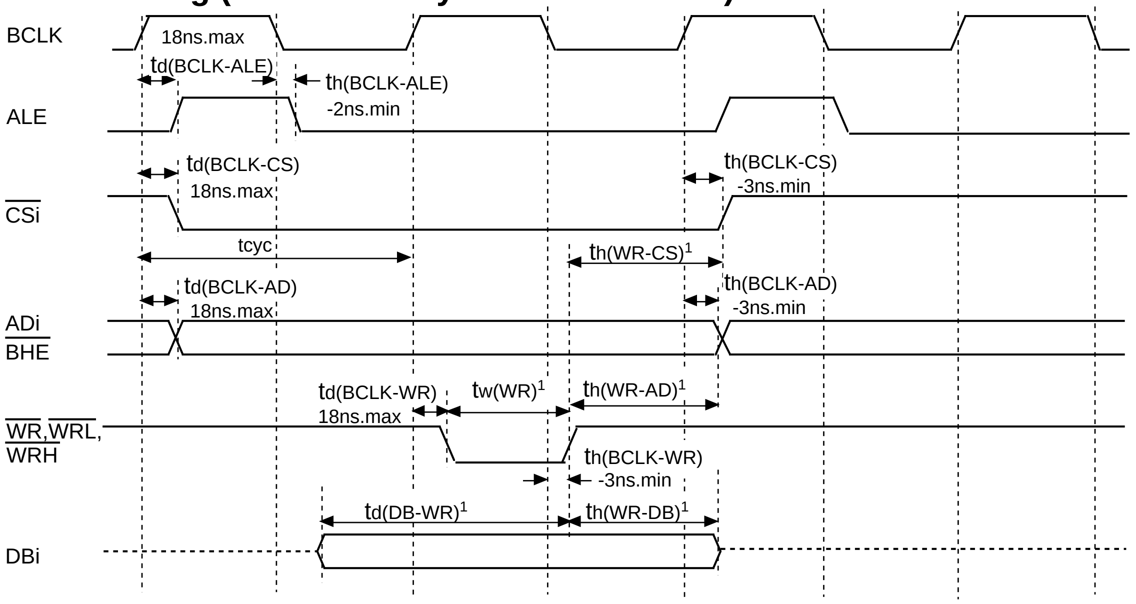
Read timing



Notes :

1. A value is guaranteed with no external factor. Maximum 35ns is guaranteed for $t_{d(BCLK-AD)}+t_{su(DB-BCLK)}$.
2. It depends on operation frequency.
   $t_{ac2(RD-DB)}$=(tcyc/2 x m-35)ns.max (m=3 when 1 wait, m=5 when 2 waits and m=7 when 3 waits.)
   $t_{ac2(AD-DB)}$=(tcyc x n-35)ns.max (n=2 when 1 wait, n=3 when 2 waits and n=4 when 3 waits.)

Write timing (written in 2 cycles without wait)



Notes :

1. It depends on operation frequency.
   $t_{d(DB-WR)}$=(tcyc x n-20)ns.min
   (n=1 when 1 wait, n=2 when 2 waits and n=3 when 3 waits)
   $t_{h(WR-DB)}$=(tcyc/2-10)ns.min
   $t_{h(WR-AD)}$=(tcyc/2-10)ns.min
   $t_{h(WR-CS)}$=(tcyc/2-10)ns.min
   $t_{w(WR)}$=(tcyc/2 x n-15)ns.min
   (n=1 when 1 wait, n=3 when 2 waits and n=5 when 3 waits)

Measurement conditions
- V_{CC}=4.2 to 5.5V
- Input high or low voltage
  : V_{IH}=2.5V, V_{IL}=0.8
- Output high or low voltage
  : V_{OH}=2.0V, V_{OL}=0.8V

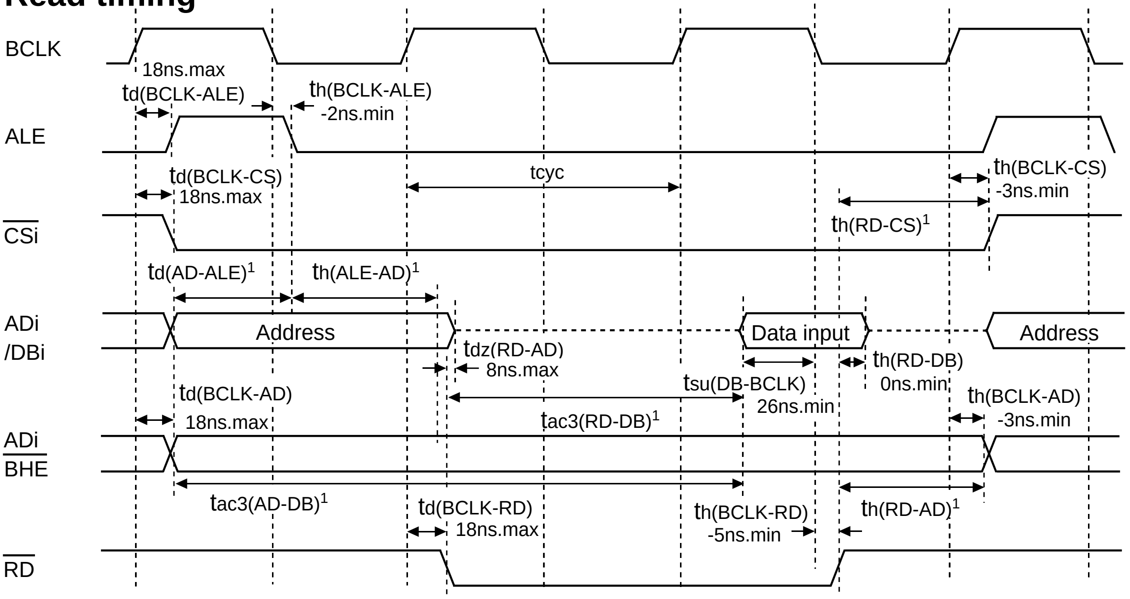
Figure 1.31.3. Vcc=5V Timing Diagram (2)

Memory expansion mode and microprocessor mode

Vcc=5V

(When accessing external memory space with a wait and multiplex bus)

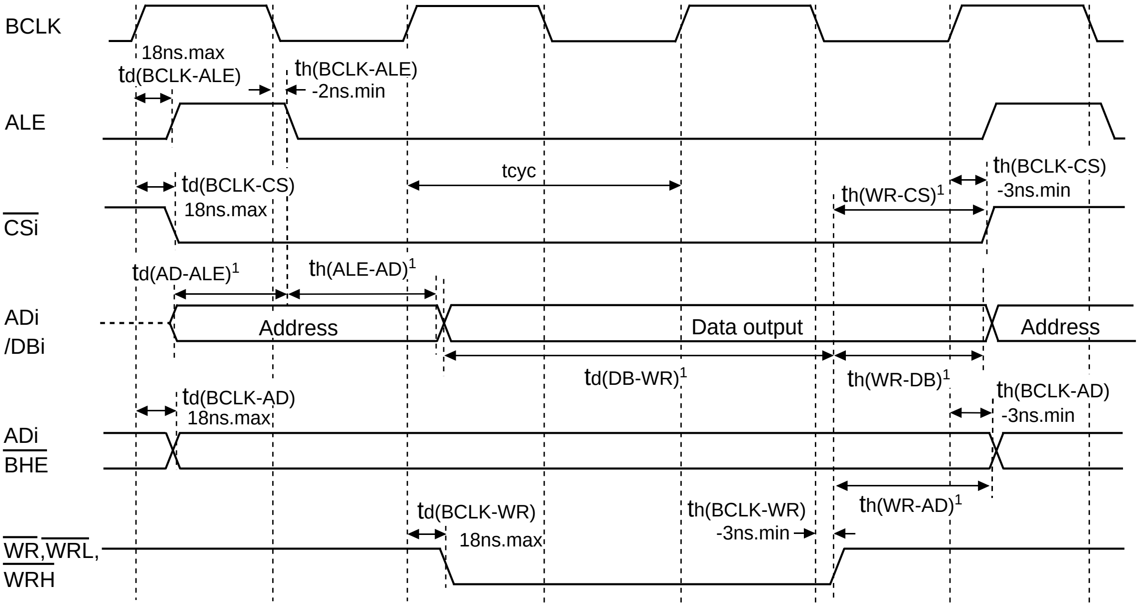
Read timing



Notes :

1. It depends on operation frequency.
   $t_{d(AD-ALE)}$=(tcyc/2-20)ns.min
   $t_{h(ALE-AD)}$=(tcyc/2-10)ns.min, $t_{h(RD-AD)}$=(tcyc/2-10)ns.min, $t_{h(RD-CS)}$=(tcyc/2-10)ns.min
   $t_{ac3(RD-DB)}$=(tcyc/2 x m-35)ns.max (m=3 when 2waits and m=5 when 3 waits.)
   $t_{ac3(AD-DB)}$=(tcyc/2 x n-35)ns.max (n=5 when 2 waits and n=7 when 3 waits.)

Write timing (written in 2 cycles without wait)



Notes :

1. It depends on operation frequency.
   $t_{d(AD-ALE)}$=(tcyc/2-20)ns.min
   $t_{h(ALE-AD)}$=(tcyc/2-10)ns.min, $t_{h(WR-AD)}$=(tcyc/2-10)ns.min
   $t_{h(WR-CS)}$=(tcyc/2-10)ns.min, $t_{h(WR-DB)}$=(tcyc/2-10)ns.min
   $t_{d(DB-WR)}$=(tcyc/2 x m-25)ns.min
   (m=3 when 2 waits and m=5 when 3 waits.)

Measurement conditions

- V_{CC}=4.2 to 5.5V
- Input high or low voltage
  : V_{IH}=2.5V, V_{IL}=0.8V
- Output high or low voltage
  : V_{OH}=2.0V, V_{OL}=0.8V

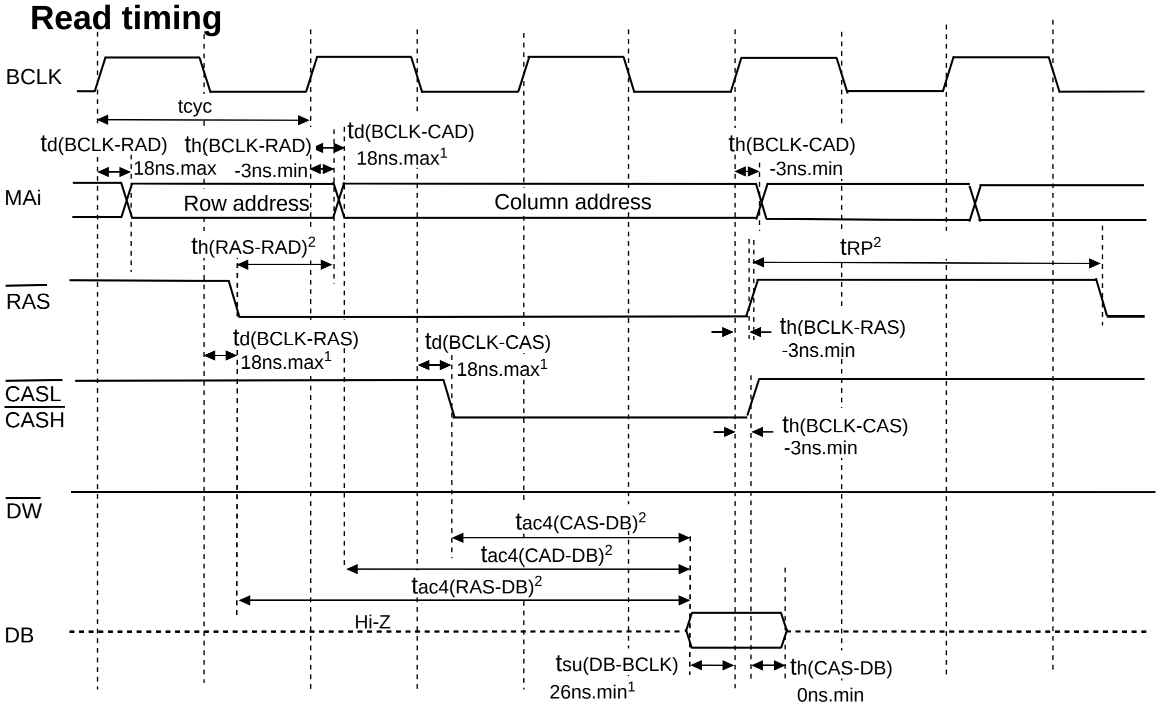
Figure 1.31.4. VCC=5V Timing Diagram (3)

Memory expansion mode and microprocessor mode

(When accessing DRAM space)

Vcc=5V

Read timing



Notes:

1. A value is guaranteed with no external factor. Maximum 35ns is guaranteed for $t_{d(BCLK-AD)}+t_{su(DB-BCLK)}$.
   $t_{d(BCLK-RAS)} + t_{su(DB-BCLK)}$
   $t_{d(BCLK-CAS)} + t_{su(DB-BCLK)}$
   $t_{d(BCLK-CAD)} + t_{su(DB-BCLK)}$
2. Depends on operation frequency
   $t_{ac4(RAS-DB)}$=(tcyc/2 x m-35)ns.max (m=3 when 1 wait and m=5 when 2 waits.)
   $t_{ac4(CAS-DB)}$=(tcyc/2 x n-35)ns.max (n=1 when 1 wait and n=3 when 2 waits.)
   $t_{ac4(CAD-DB)}$=(tcyc x l-35)ns.max (l=1 when 1 wait and l=2 when 2 waits.)
   $t_{h(RAS-RAD)}$=(tcyc/2-13)ns.min
   t_{RP}=(tcyc/2 x 3-20)ns.min

Measurement conditions

- V_{CC}=4.2 to 5.5V
- Input high or low voltage
  : V_{IH}=2.5V, V_{IL}=0.8V
- Output high or low voltage
  : V_{OH}=2.0V, V_{OL}=0.8V

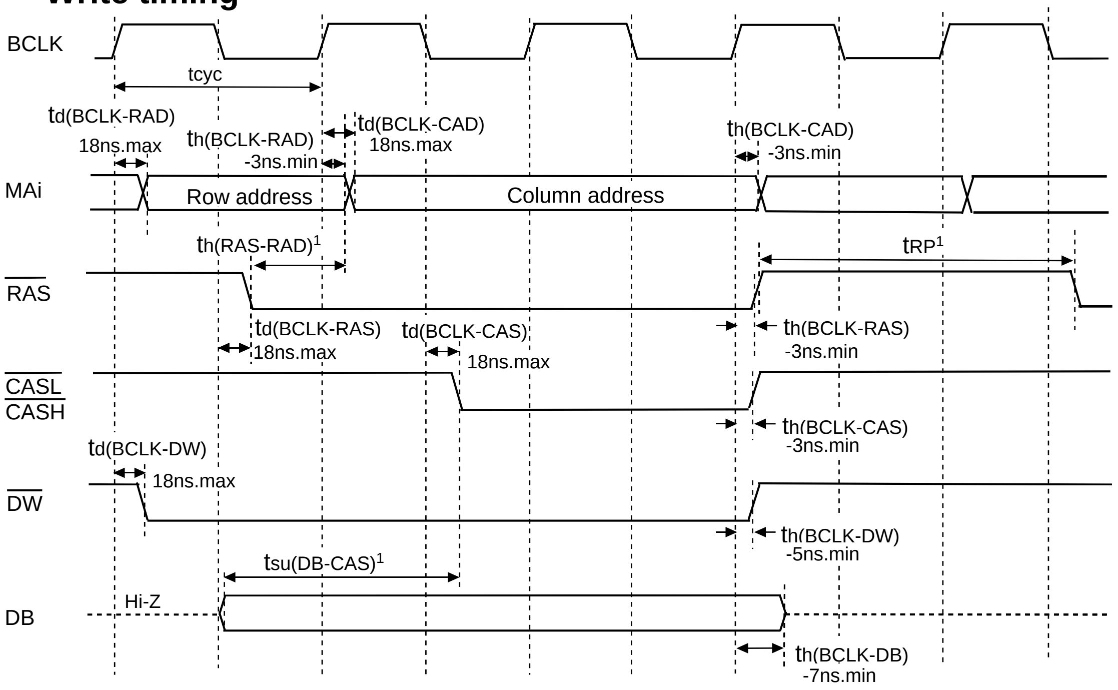
Figure 131.5. VCC=5V Timing Diagram (4)

Memory expansion mode and microprocessor mode

Vcc=5V

(When accessing DRAM space)

Write timing



Notes :

1. It depends on operation frequency.
   $t_{h(RAS-RAD)}$=(tcyc/2-13)ns.min
   t_{RP}=(tcyc/2 x 3-20)ns.min
   $t_{su(DB-CAS)}$=(tcyc-20)ns.min

Measurement conditions

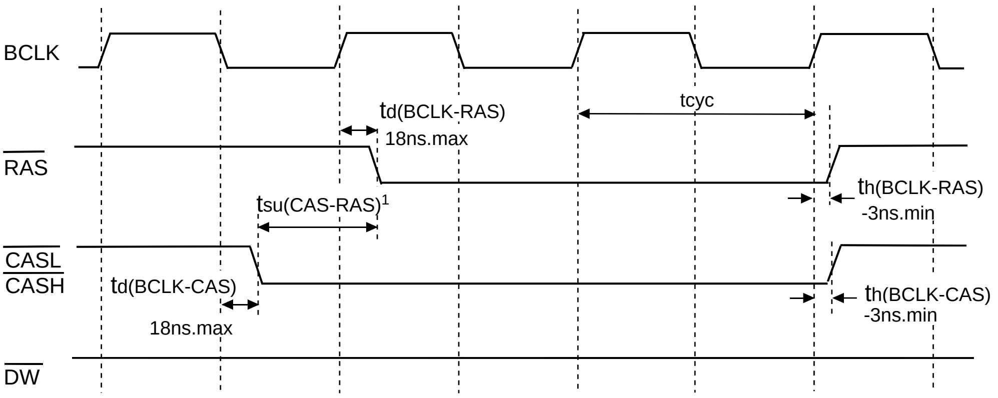
- V_{CC}=4.2 to 5.5V
- Input high or low voltage
  : V_{IH}=2.5V, V_{IL}=0.8V
- Output high or low voltage
  : V_{OH}=2.0V, V_{OL}=0.8V

Figure 1.31.6. VCC=5V Timing Diagram (5)

Memory expansion mode and microprocessor mode

Vcc=5V

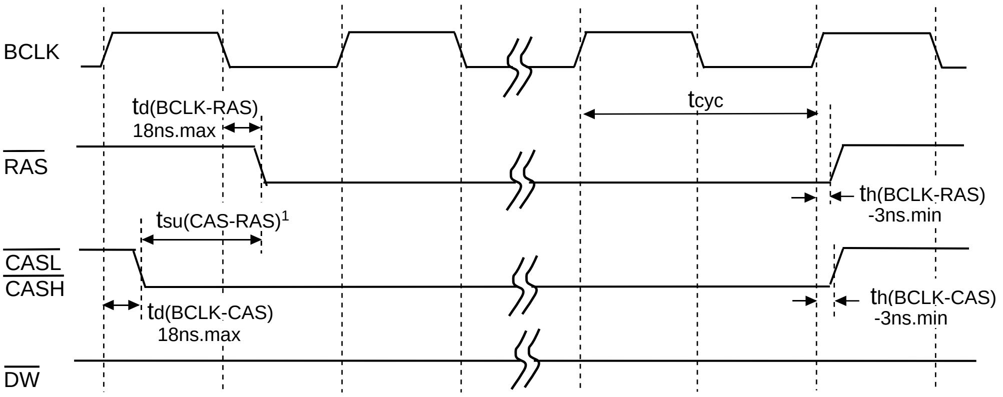
Refresh timing ($\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh)



Notes :

1. It depends on operation frequency.
$t_{su(CAS-RAS)}$=(tcyc/2-13)ns.min

Refresh timing (Self-refresh)



Notes :

1. It depends on operation frequency.
$t_{su(CAS-RAS)}$=(tcyc/2-13)ns.min

Measurement conditions
- V_{CC}=4.2 to 5.5V
- Input high or low voltage
  : V_{IH}=2.5V, V_{IL}=0.8V
- Output high or low voltage
  : V_{OH}=2.0V, V_{OL}=0.8V

Figure 1.31.7. Vcc=5V Timing Diagram (6)

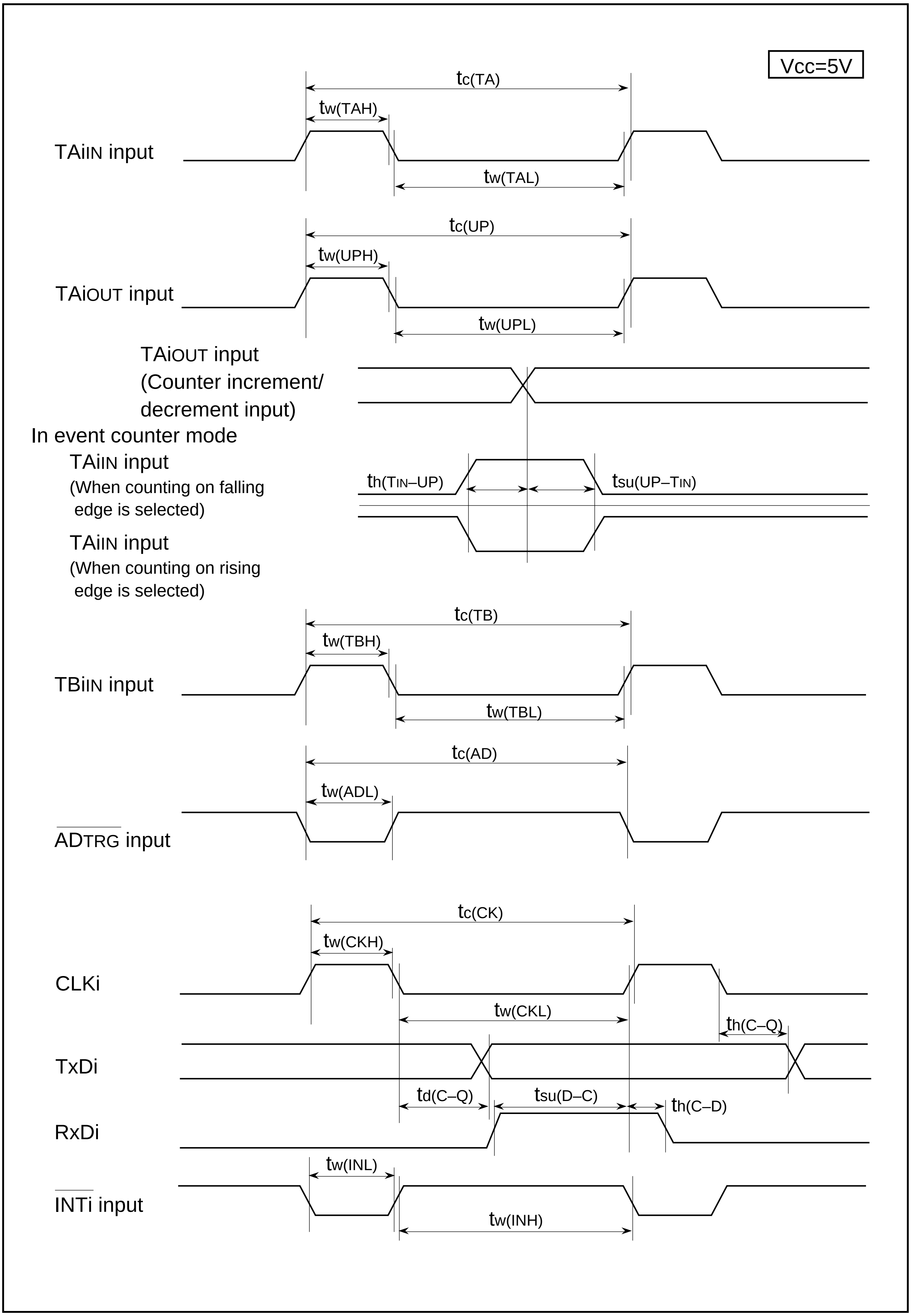
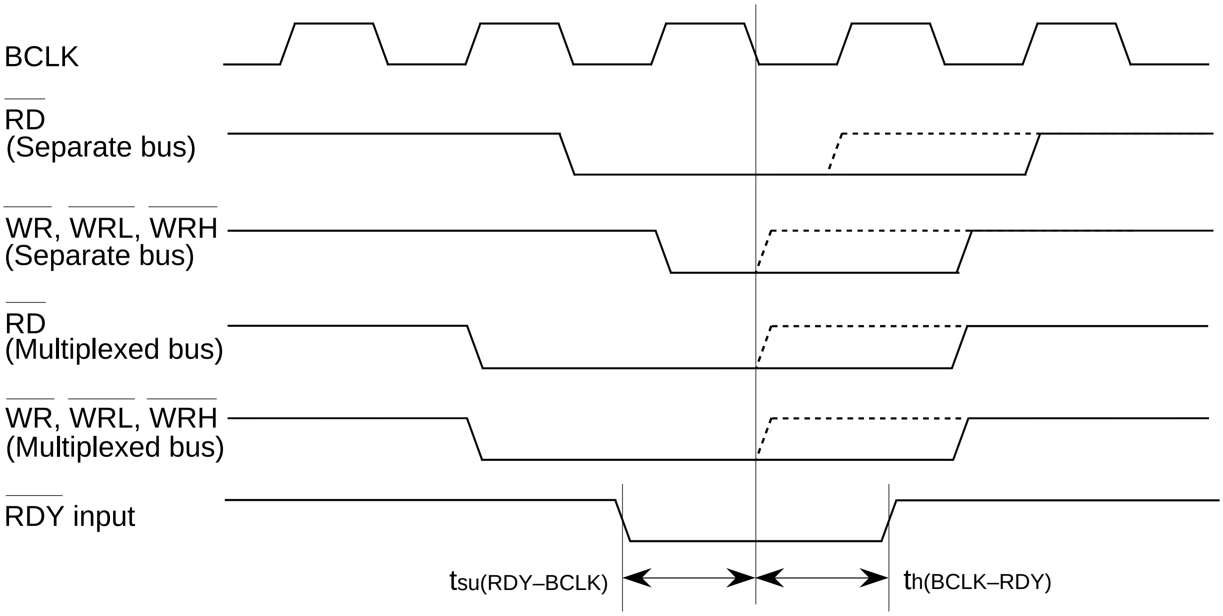


Figure 1.31.8. Vcc=5V Timing Diagram (7)

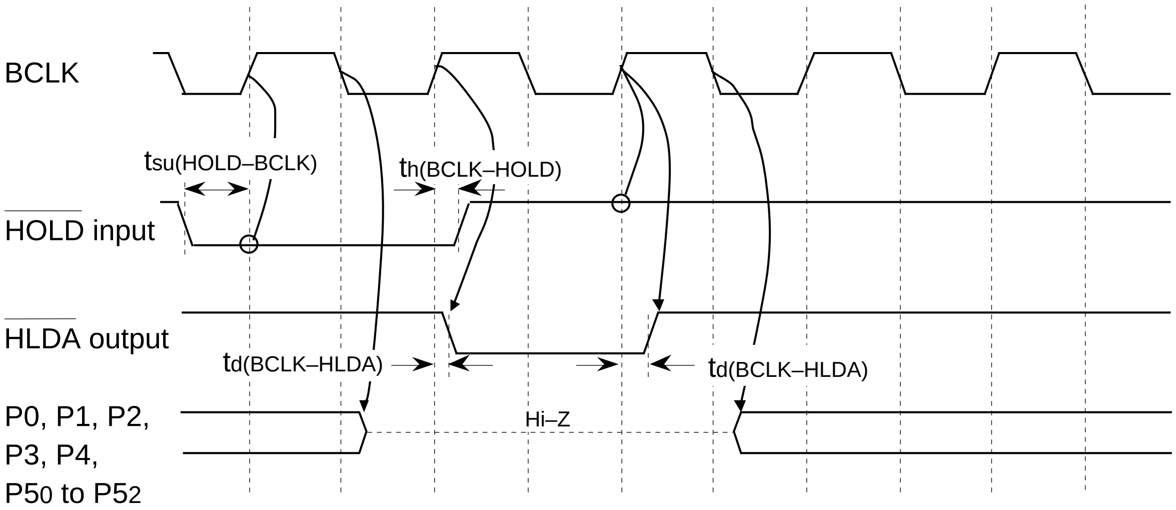
Vcc=5V

Memory expansion mode and microprocessor mode

(Valid only with a wait)



(Valid with or without wait)



Measurement conditions :
- V_{CC}=4.2 to 5.5V
- Input high or low voltage : V_{IH}=4.0V, V_{IL}=1.0V
- Output high or low voltage : V_{OH}=2.5V, V_{OL}=2.5V

Figure 1.31.9. Vcc=5V Timing Diagram (8)

V_{CC} = 3.3V

Table 1.31.23. Electrical Characteristics (V_{CC}=3.0 to 3.6V, V_{SS}=0V at Topr = –20 to 85°C, f(X_{IN})=20MHz unless otherwise specified)

Symbol	Parameter		Condition	Standard			Unit
				Min	Typ	Max	
V_{OH}	Output "H" voltage	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_0$-$P7_7$, $P8_0$-$P8_4$, $P8_6$, $P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$, $P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$[1]	I_{OH}=-1mA	2.7			V
		X_{OUT}	I_{OH}=-0.1mA	2.7			V
		X_{COUT}	No load applied		3.3		V
V_{OL}	Output "L" voltage	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_0$-$P7_7$, $P8_0$-$P8_4$, $P8_6$, $P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$, $P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$[1]	I_{OL}=1mA			0.5	
		X_{OUT}	I_{OL}=0.1mA			0.5	V
		X_{COUT}	No load applied		0		V
V_{T+}-V_{T-}	Hysteresis	$\overline{HOLD}$, $\overline{RDY}$, $TA0_{IN}$-$TA4_{IN}$, $TB0_{IN}$-$TB5_{IN}$, $\overline{INT0}$-$\overline{INT5}$, $\overline{AD}_{TRG}$, $\overline{CTS0}$-$\overline{CTS4}$, CLK0-CLK4, $TA0_{OUT}$-$TA4_{OUT}$, $\overline{NMI}$, $\overline{KI0}$-$\overline{KI3}$, RxD0-RxD4, SCL0-SCL4, SDA0-SDA4		0.2		1.0	V
		$\overline{RESET}$	0.2		1.8	V	
I_{IH}	Input "H" current	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_2$-$P7_7$, $P8_0$-$P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$, $P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$[1], X_{IN}, $\overline{RESET}$, CNVss, BYTE	V_I=3V			4.0	µA
I_{IL}	Input "L" current	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_2$-$P7_7$, $P8_0$-$P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$, $P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$[1], X_{IN}, $\overline{RESET}$, CNVss, BYTE	V_I=0V			-4.0	µA
R_{PULLUP}	Pull-up resistance	$P0_0$-$P0_7$, $P1_0$-$P1_7$, $P2_0$-$P2_7$, $P3_0$-$P3_7$, $P4_0$-$P4_7$, $P5_0$-$P5_7$, $P6_0$-$P6_7$, $P7_2$-$P7_7$, $P8_0$-$P8_4$, $P8_6$, $P8_7$, $P9_0$-$P9_7$, $P10_0$-$P10_7$, $P11_0$-$P11_4$, $P12_0$-$P12_7$, $P13_0$-$P13_7$, $P14_0$-$P14_6$, $P15_0$-$P15_7$[1]	V_I=0V	66	120	500	kΩ
R_{fXIN}	Feedback resistance	X_{IN}			3.0		MΩ
R_{fXCIN}	Feedback resistance	X_{CIN}			20.0		MΩ
V_{RAM}	RAM retention voltage	VDC-ON		2.5			V
		VDC-pass through		2.0			V
I_{CC}	Power supply current	Measuring condition: In sigle-chip mode, the output pins are open and other pins are connected to Vss.	f(X_{IN})=20MHz, square wave, no division		26	38	mA
			f(X_{CIN})=32kHz, with a wait, VDC-pass through, Topr=25°C		5.0		µA
			f(X_{CIN})=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.

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 Topr = –20 to 85°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			±2	LSB
DSL	Differential nonlinearity error	No S&H function(8-bit)				±1	LSB
-	Offset error	No S&H function(8-bit)				±2	LSB
-	Gain error	No S&H function(8-bit)				±2	LSB
R_{LADDER}	Ladder resistance		V_{REF} = V_{CC}	8		40	kΩ
t_{CONV}	Conversion time(8bit)			9.8			µs
V_{REF}	Reference voltage			3.0		V_{CC}	V
V_{IA}	Analog input voltage			0		V_{REF}	V

S&H: Sample and hold

Notes :

1. Divide f(X_{IN}), if exceeding 10MHz, to keep ØAD 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 Topr = –20 to 85°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	µs
R_O	Output resistance		4	10	20	kΩ
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 "00_{16}". 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).

VCC = 3.3V

Timing Requirements (VCC = 3.0 to 3.6V, VSS = 0V at Topr = –20 to 85°C unless otherwise specified)

Table 1.31.26. External Clock Input

Symbol	Parameter	Standard		Unit
		Min	Max	
t_c	External clock input cycle time	50		ns
$t_{w(H)}$	External clock input "H" pulse width	22		ns
$t_{w(L)}$	External clock input "L" pulse width	22		ns
t_r	External clock rising-edge time		5	ns
t_f	External clock falling-edge time		5	ns

Table 1.31.27. Memory Expansion and Microprocessor Modes

Symbol	Parameter	Standard		Unit
		Min	Max	
$t_{ac1(RD-DB)}$	Data input access time (RD standard, without wait)		(Note 1)	ns
$t_{ac1(AD-DB)}$	Data input access time (AD standard, CS standard, without wait)		(Note 1)	ns
$t_{ac2(RD-DB)}$	Data input access time (RD standard, with a wait)		(Note 1)	ns
$t_{ac2(AD-DB)}$	Data input access time (AD standard, CS standard, with a wait)		(Note 1)	ns
$t_{ac3(RD-DB)}$	Data input access time (RD standard, when accessing multiplex bus space)		(Note 1)	ns
$t_{ac3(AD-DB)}$	Data input access time (AD standard, CS standard, when accessing multiplex bus space)		(Note 1)	ns
$t_{ac4(RAS-DB)}$	Data input access time (RAS standard, DRAM access)		(Note 1)	ns
$t_{ac4(CAS-DB)}$	Data input access time (CAS standard, DRAM access)		(Note 1)	ns
$t_{ac4(CAD-DB)}$	Data input access time (CAD standard, DRAM access)		(Note 1)	ns
$t_{su(DB-BCLK)}$	Data input setup time	30		ns
$t_{su(RDY-BCLK)}$	$\overline{\mathrm{RDY}}$ input setup time	40		ns
$t_{su(HOLD-BCLK)}$	$\overline{\mathrm{HOLD}}$ input setup time	60		ns
$t_{h(RD-DB)}$	Data input hold time	0		ns
$t_{h(CAS-DB)}$	Data input hold time	0		ns
$t_{h(BCLK-RDY)}$	$\overline{\mathrm{RDY}}$ input hold time	0		ns
$t_{h(BCLK-HOLD)}$	$\overline{\mathrm{HOLD}}$ input hold time	0		ns
$t_{d(BCLK-HLDA)}$	$\overline{\mathrm{HLDA}}$ 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.

$$t_{ac1(RD-DB)} = \frac{10^9}{f_{(BCLK)} \text{ X } 2} - 35 \quad \text{[ns]}$$

$$t_{ac1(AD-DB)} = \frac{10^9}{f_{(BCLK)}} - 35 \quad \text{[ns]}$$

$$t_{ac2(RD-DB)} = \frac{10^9 \text{ X } m}{f_{(BCLK)} \text{ X } 2} - 35 \quad \text{[ns] (m=3 when 1 wait, m=5 when 2 waits and m=7 when 3waits.)}$$

$$t_{ac2(AD-DB)} = \frac{10^9 \text{ X } n}{f_{(BCLK)}} - 35 \quad \text{[ns] (n=2 when 1 wait, n=3 when 2 waits and n=4 when 3 waits.)}$$

$$t_{ac3(RD-DB)} = \frac{10^9 \text{ X } m}{f_{(BCLK)} \text{ X } 2} - 35 \quad \text{[ns] (m=3 when 2 waits and m=5 when 3 waits.)}$$

$$t_{ac3(AD-DB)} = \frac{10^9 \text{ X } n}{f_{(BCLK)} \text{ X } 2} - 35 \quad \text{[ns] (n=5 when 2 waits and n=7 when 3 waits.)}$$

$$t_{ac4(RAS-DB)} = \frac{10^9 \text{ X } m}{f_{(BCLK)} \text{ X } 2} - 35 \quad \text{[ns] (m=3 when 1 wait and m=5 when 2 waits.)}$$

$$t_{ac4(CAS-DB)} = \frac{10^9 \text{ X } n}{f_{(BCLK)} \text{ X } 2} - 35 \quad \text{[ns] (n=1 when 1 wait and n=3 when 2 waits.)}$$

$$t_{ac4(CAD-DB)} = \frac{10^9 \text{ X } l}{f_{(BCLK)}} - 35 \quad \text{[ns] (l=1 when 1 wait and l=2 when 2 waits.)}$$

V_{CC} = 3.3V

Timing Requirements

(V_{CC} = 3.0 to 3.6V, V_{SS} = 0V at Topr = –20 to 85°C unless otherwise specified)

Table 1.31.28. Timer A Input (Count Source Input in Event Counter Mode)

Symbol	Parameter	Standard		Unit
		Min	Max	
$t_{c(TA)}$	TAi_{IN} input cycle time	100		ns
$t_{w(TAH)}$	TAi_{IN} input "H" pulse width	40		ns
$t_{w(TAL)}$	TAi_{IN} input "L" pulse width	40		ns

Table 1.31.29. Timer A Input (Gate Input in Timer Mode)

Symbol	Parameter	Standard		Unit
		Min	Max	
$t_{c(TA)}$	TAi_{IN} input cycle time	400		ns
$t_{w(TAH)}$	TAi_{IN} input "H" pulse width	200		ns
$t_{w(TAL)}$	TAi_{IN} 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	
$t_{c(TA)}$	TAi_{IN} input cycle time	200		ns
$t_{w(TAH)}$	TAi_{IN} input "H" pulse width	100		ns
$t_{w(TAL)}$	TAi_{IN} 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	
$t_{w(TAH)}$	TAi_{IN} input "H" pulse width	100		ns
$t_{w(TAL)}$	TAi_{IN} 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	
$t_{c(UP)}$	TAi_{OUT} input cycle time	2000		ns
$t_{w(UPH)}$	TAi_{OUT} input "H" pulse width	1000		ns
$t_{w(UPL)}$	TAi_{OUT} input "L" pulse width	1000		ns
$t_{su(UP-TIN)}$	TAi_{OUT} input setup time	400		ns
$t_{h(TIN-UP)}$	TAi_{OUT} input hold time	400		ns

VCC = 3.3V

Timing Requirements

(VCC = 3.0 to 3.6V, VSS = 0V at Topr = –20 to 85°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	
$t_{c(AD)}$	$\overline{\text{ADTRG}}$ input cycle time (trigger available at minimum and above)	1000		ns
$t_{w(ADL)}$	$\overline{\text{ADTRG}}$ 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 $\overline{\text{INTi}}$ input

Symbol	Parameter	Standard		Unit
		Min	Max	
$t_{w(INH)}$	$\overline{\text{INTi}}$ input "H" pulse width	250		ns
$t_{w(INL)}$	$\overline{\text{INTi}}$ input "L" pulse width	250		ns

VCC = 3.3V

Switching Characteristics

(VCC = 3.0 to 3.6V, VSS = 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	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.31.1		18	ns
$t_{h(BCLK-AD)}$	Address output hold time (BCLK standard)		0		ns
$t_{h(RD-AD)}$	Address output hold time (RD standard)		0		ns
$t_{h(WR-AD)}$	Address output hold time (WR standard)		(Note 1)		ns
$t_{d(BCLK-CS)}$	Chip-select output delay time			18	ns
$t_{h(BCLK-CS)}$	Chip-select output hold time (BCLK standard)		0		ns
$t_{h(RD-CS)}$	Chip-select output hold time (RD standard)		0		ns
$t_{h(WR-CS)}$	Chip-select output hold time (WR standard)		(Note 1)		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		– 3		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			18	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(DB-WR)}$	Data output delay time (WR standard)		(Note 1)		ns
$t_{h(WR-DB)}$	Data output hold time (WR standard)		(Note 1)		ns
$t_{w(WR)}$	Write pulse width		(Note 1)		ns

Notes :

1. A value can be obtained from the following expressions, according to BCLK frequency.

$$t_{d(DB-WR)} = \frac{10^9}{f_{(BCLK)}} - 20 \quad [\text{ns}]$$

$$t_{h(WR-DB)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$t_{h(WR-AD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$t_{h(WR-CS)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$t_{w(WR)} = \frac{10^9}{f_{(BCLK)} \times 2} - 15 \quad [\text{ns}]$$

VCC = 3.3V

Switching Characteristics

(VCC = 3.0 to 3.6V, VSS = 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	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.31.1		18	ns
$t_{h(BCLK-AD)}$	Address output hold time (BCLK standard)		0		ns
$t_{h(RD-AD)}$	Address output hold time (RD standard)		0		ns
$t_{h(WR-AD)}$	Address output hold time (WR standard)		(Note 1)		ns
$t_{d(BCLK-CS)}$	Chip-select output delay time			18	ns
$t_{h(BCLK-CS)}$	Chip-select output hold time (BCLK standard)		0		ns
$t_{h(RD-CS)}$	Chip-select output hold time (RD standard)		0		ns
$t_{h(WR-CS)}$	Chip-select output hold time (WR standard)		(Note 1)		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		– 3		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			18	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(DB-WR)}$	Data output delay time (WR standard)		(Note 1)		ns
$t_{h(WR-DB)}$	Data output hold time (WR standard)		(Note 1)		ns
$t_{w(WR)}$	Write pulse width		(Note 1)		ns

Notes :

1. A value can be obtained from the following expressions, according to BCLK frequency.

$$t_{d(DB-WR)} = \frac{10^{9} \times n}{f_{(BCLK)}} - 20 \quad \text{[ns]} \quad \text{(n=1 when 1 wait, n=2 when 2 waits and n=3 when 3 waits.)}$$

$$t_{h(WR-DB)} = \frac{10^{9}}{f_{(BCLK)} \times 2} - 10 \quad \text{[ns]}$$

$$t_{h(WR-AD)} = \frac{10^{9}}{f_{(BCLK)} \times 2} - 10 \quad \text{[ns]}$$

$$t_{h(WR-CS)} = \frac{10^{9}}{f_{(BCLK)} \times 2} - 10 \quad \text{[ns]}$$

$$t_{w(WR)} = \frac{10^{9} \times n}{f_{(BCLK)} \times 2} - 15 \quad \text{[ns]} \quad \text{(n=1 when 1 wait, n=3 when 2 waits and n=5 when 3 waits.)}$$

VCC = 3.3V

Switching Characteristics

(VCC = 3.0 to 3.6V, VSS = 0V at Topr = –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	
$t_{d(BCLK-AD)}$	Address output delay time	Figure 1.31.1		18	ns
$t_{h(BCLK-AD)}$	Address output hold time (BCLK standard)		0		ns
$t_{h(RD-AD)}$	Address output hold time (RD standard)		(Note 1)		ns
$t_{h(WR-AD)}$	Address output hold time (WR standard)		(Note 1)		ns
$t_{d(BCLK-CS)}$	Chip-select output delay time			18	ns
$t_{h(BCLK-CS)}$	Chip-select output hold time (BCLK standard)		0		ns
$t_{h(RD-CS)}$	Chip-select output hold time (RD standard)		(Note 1)		ns
$t_{h(WR-CS)}$	Chip-select output hold time (WR standard)		(Note 1)		ns
$t_{d(BCLK-RD)}$	RD signal output delay time			18	ns
$t_{h(BCLK-RD)}$	RD signal output hold time		– 3		ns
$t_{d(BCLK-WR)}$	WR signal output delay time			18	ns
$t_{h(BCLK-WR)}$	WR signal output hold time		0		ns
$t_{d(DB-WR)}$	Data output delay time (WR standard)		(Note 1)		ns
$t_{h(WR-DB)}$	Data output hold time (WR standard)		(Note 1)		ns
$t_{d(BCLK-ALE)}$	ALE signal output delay time (BCLK standard)			18	ns
$t_{h(BCLK-ALE)}$	ALE signal output hold time (BCLK standard)		– 2		ns
$t_{d(AD-ALE)}$	ALE signal output delay time (address standard)		(Note 1)		ns
$t_{h(ALE-AD)}$	ALE signal output hold time (address standard)		(Note 1)		ns
$t_{dz(RD-AD)}$	Address output high-impedance start time			8	ns

Notes :

1. A value can be obtained from the following formula, according to BCLK frequency.

$$t_{h(RD-AD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$t_{h(WR-AD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$t_{h(RD-CS)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$t_{h(WR-CS)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$t_{d(DB-WR)} = \frac{10^9 \times m}{f_{(BCLK)} \times 2} - 25 \quad [\text{ns}] \quad \text{(m=3 when 2 waits and m=5 when 3 waits.)}$$

$$t_{h(WR-DB)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

$$t_{d(AD-ALE)} = \frac{10^9}{f_{(BCLK)} \times 2} - 20 \quad [\text{ns}]$$

$$t_{h(ALE-AD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 10 \quad [\text{ns}]$$

VCC = 3.3V

Switching Characteristics

(VCC = 3.0 to 3.6V, VSS = 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	
$t_{d(BCLK-RAD)}$	Row address output delay time	Figure 1.31.1		18	ns
$t_{h(BCLK-RAD)}$	Row address output hold time (BCLK standard)		0		ns
$t_{d(BCLK-CAD)}$	Column address output delay time			18	ns
$t_{h(BCLK-CAD)}$	Column address output hold time (BCLK standard)		0		ns
$t_{h(RAS-RAD)}$	Row address output hold time after RAS output		(Note 1)		ns
$t_{d(BCLK-RAS)}$	RAS output delay time (BCLK standard)			18	ns
$t_{h(BCLK-RAS)}$	RAS output hold time (BCLK standard)		0		ns
t_{RP}	RAS "H" hold time		(Note 1)		ns
$t_{d(BCLK-CAS)}$	CAS output delay time (BCLK standard)			18	ns
$t_{h(BCLK-CAS)}$	CAS output hold time (BCLK standard)		0		ns
$t_{d(BCLK-DW)}$	Data output delay time (BCLK standard)			18	ns
$t_{h(BCLK-DW)}$	Data output hold time (BCLK standard)		– 3		ns
$t_{su(DB-CAS)}$	CAS after DB output setup time		(Note 1)		ns
$t_{h(BCLK-DB)}$	DB signal output hold time (BCLK standard)		– 7		ns
$t_{su(CAS-RAS)}$	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.

$$t_{h(RAS-RAD)} = \frac{10^9}{f_{(BCLK)} \times 2} - 13 \quad [ns]$$

$$t_{RP} = \frac{10^9 \times 3}{f_{(BCLK)} \times 2} - 20 \quad [ns]$$

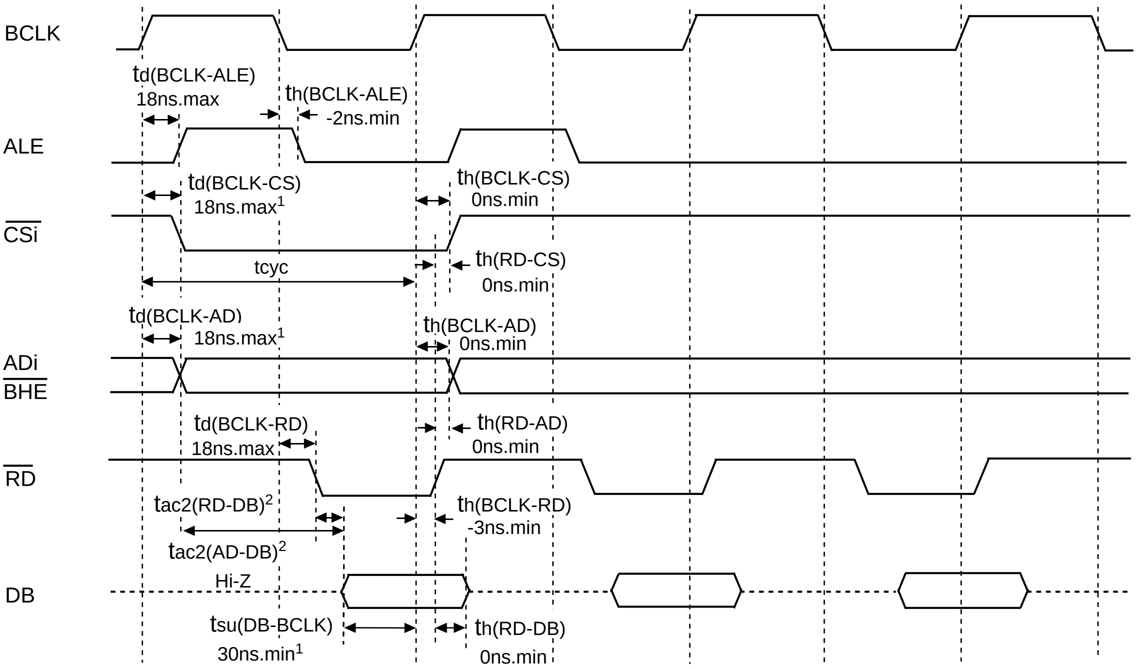
$$t_{su(DB-CAS)} = \frac{10^9}{f_{(BCLK)}} - 20 \quad [ns]$$

$$t_{su(CAS-RAS)} = \frac{10^9}{f_{(BCLK)} \times 2} - 13 \quad [ns]$$

Vcc=3.3V

Memory expansion mode and microprocessor mode (without wait)

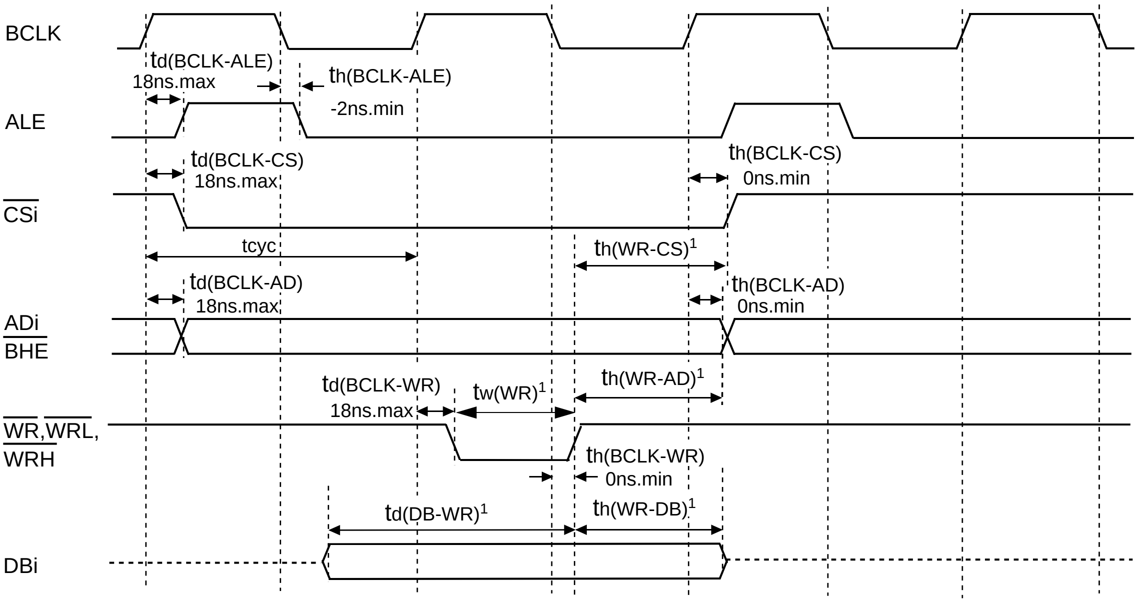
Read timing



Notes :

1. A value is guarantee with no external factor. Maximum 35ns is garanteed for $t_{d(BCLK-AD)}+t_{su(DB-BCLK)}$.
2. It depends on operation frequency.
   $t_{ac2(RD-DB)}$=(tcyc/2-35)ns.max
   $t_{ac2(AD-DB)}$=(tcyc-35)ns.max

Write timing



Notes :

1. It depends on operation frequency.
   $t_{d(DB-WR)}$=(tcyc-20)ns.min
   $t_{h(WR-DB)}$=(tcyc/2-10)ns.min
   $t_{h(WR-AD)}$=(tcyc/2-10)ns.min
   $t_{h(WR-CS)}$=(tcyc/2-10)ns.min
   $t_{w(WR)}$=(tcyc/2-15)ns.min

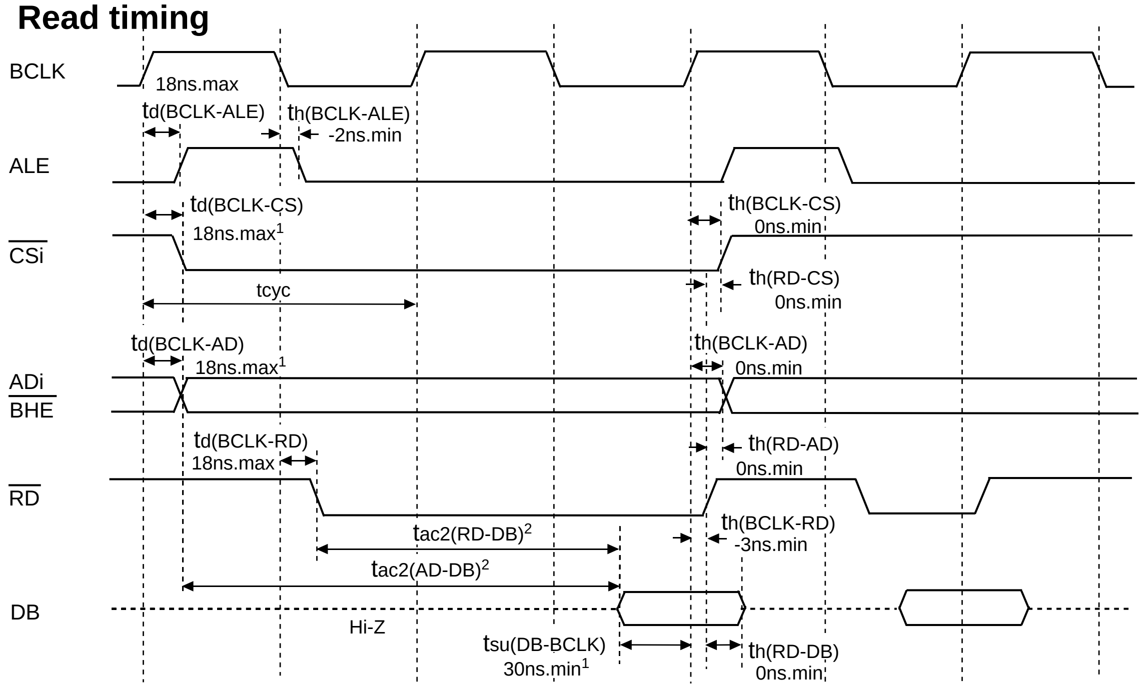
Measurement conditions

- V_{CC}=3.0 to 3.6V
- Input high or low voltage : V_{IH}=1.5V, V_{IL}=0.5V
- Output high or low voltage : V_{OH}=1.5V, V_{OL}=1.5V

Figure 1.31.10. V_{CC}=3.3V Timing Diagram (1)

Memory expansion mode and microprocessor mode (with a wait) Vcc=3.3V

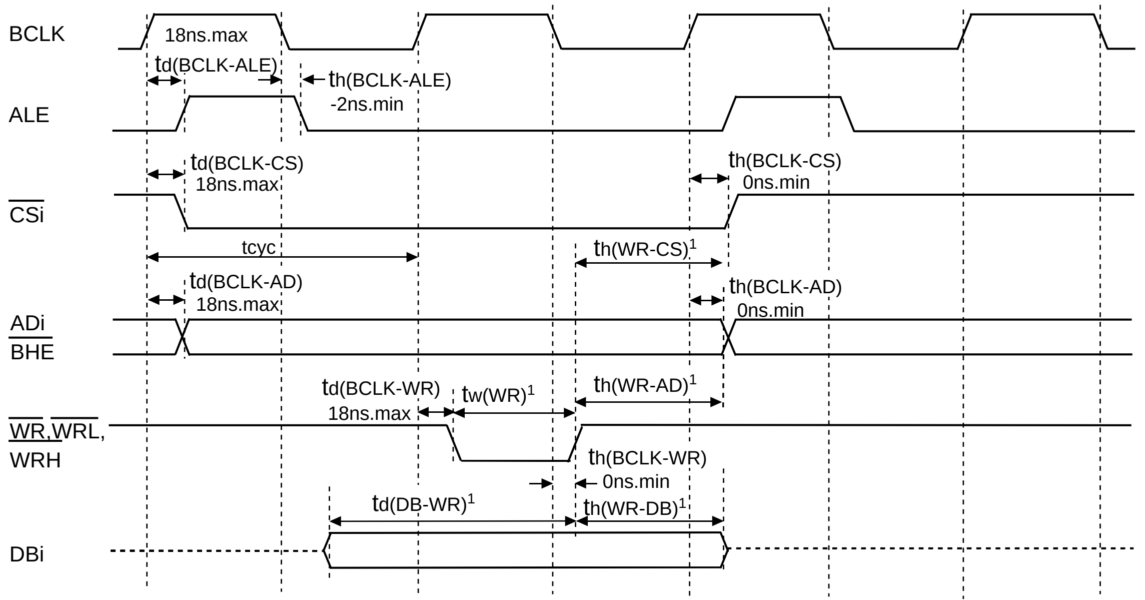
Read timing



Notes :

1. A value is guarantee with no external factor. Maximum 35ns is garanteed for $t_{d(BCLK-AD)}+t_{su(DB-BCLK)}$.
2. It depends on operation frequency.
   $t_{ac2(RD-DB)}$=(tcyc/2 x m-35)ns.max (m=3 when 1 wait, m=5 when 2 waits and m=7 when 3 waits.)
   $t_{ac2(AD-DB)}$=(tcyc x n-35)ns.max (n=2 when 1 wait, n=3 when 2 waits and n=4 when 3 waits.)

Write timing



Notes :

1. It depends on operation frequency.
   $t_{d(DB-WR)}$=(tcyc x n-20)ns.min
   (n=1 when 1 wait, n=2 when 2 waits and n=3 when 3 waits.)
   $t_{h(WR-DB)}$=(tcyc/2-10)ns.min
   $t_{h(WR-AD)}$=(tcyc/2-10)ns.min
   $t_{h(WR-CS)}$=(tcyc/2-10)ns.min
   $t_{w(WR)}$=(tcyc/2 x n-15)ns.min
   (n=1 when 1 wait, n=3 when 2 waits and n=5 when 3 waits.)

Measurement conditions

- V_{CC}=3.0 to 3.6V
- Input high or low voltage
  : V_{IH}=1.5V, V_{IL}=0.5V
- Output high or low voltage
  : V_{OH}=1.5V, V_{OL}=1.5V

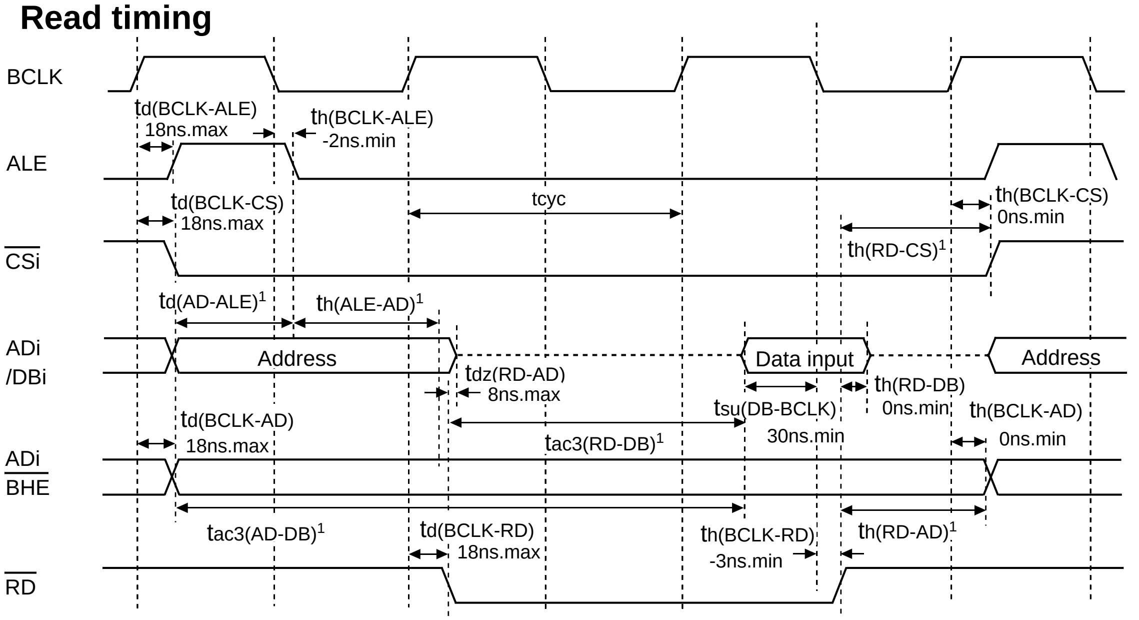
Figure 1.31.11. V_{CC}=3.3V Timing Diagram (2)

Memory expansion mode and microprocessor mode

Vcc=3.3V

(When accessing external memory space with a wait, and multiplexed bus)

Read timing



Notes :

1. It depends on operation frequency.

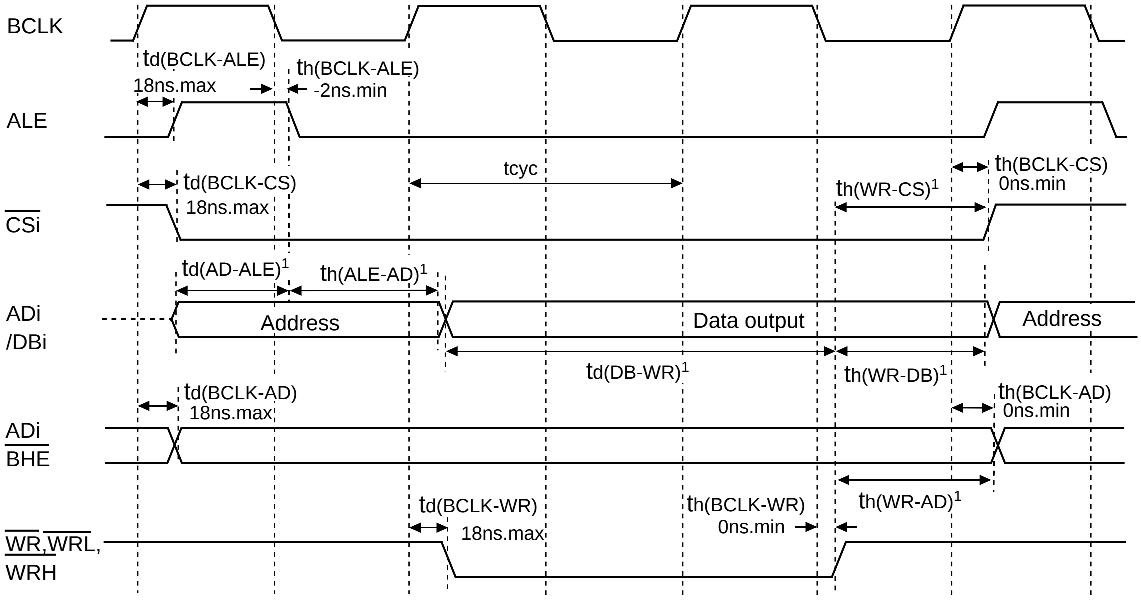
$t_{d(AD-ALE)}$=(tcyc/2-20)ns.min

$t_{h(ALE-AD)}$=(tcyc/2-10)ns.min, $t_{h(RD-AD)}$=(tcyc/2-10)ns.min, $t_{h(RD-CS)}$=(tcyc/2-10)ns.min

$t_{ac3(RD-DB)}$=(tcyc/2 x m-35)ns.max (m=3 when 2 waits and m=5 when 3 waits.)

$t_{ac3(AD-DB)}$=(tcyc/2 x n-35)ns.max (n=5 when 2 waits and n=7 when 3 waits.)

Write Timing



Notes :

1. It depends on operation frequency.

$t_{d(AD-ALE)}$=(tcyc/2-20)ns.min
$t_{h(ALE-AD)}$=(tcyc/2-10)ns.min, $t_{h(WR-AD)}$=(tcyc/2-10)ns.min
$t_{h(WR-CS)}$=(tcyc/2-10)ns.min, $t_{h(WR-DB)}$=(tcyc/2-10)ns.min
$t_{d(DB-WR)}$=(tcyc/2 x m-25)ns.min
(m=3 when 2 waits amd ,m=5 when 3 waits.)

Measurement conditions
- V_{CC}=3.0 to 3.6V
- Input high or low voltage
  : V_{IH}=1.5V, V_{IL}=0.5V
- Output high or low voltage
  : V_{OH}=1.5V, V_{OL}=1.5V

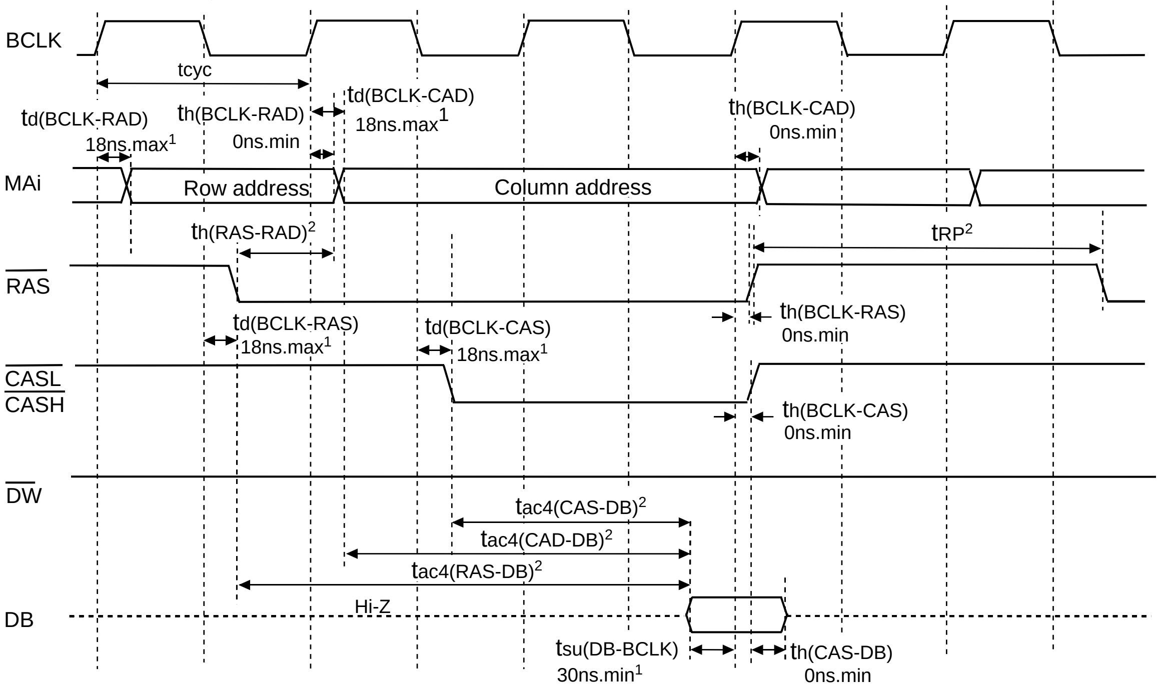
Figure 1.31.12. V_{CC}=3.3V Timing Diagram (3)

Memory expansion mode and microprocessor mode

(When accessing DRAM space with 2 waits)

Vcc=3.3V

Read timing



Notes :

1. A value is guarantee with no external factor. Maximum 35ns is garanteed for follows:
   - $t_{d(BCLK-RAS)} + t_{su(DB-BCLK)}$
   - $t_{d(BCLK-CAS)} + t_{su(DB-BCLK)}$
   - $t_{d(BCLK-CAD)} + t_{su(DB-BCLK)}$
2. It depends on operation frequency.
   - $t_{ac4(RAS-DB)}$=(tcyc/2 x m-35)ns.max (m=3 when 1 wait and m=5 when 2 waits.)
   - $t_{ac4(CAS-DB)}$=(tcyc/2 x n-35)ns.max (n=1 when 1 wait and n=3 when 2 waits.)
   - $t_{ac4(CAD-DB)}$=(tcyc x l-35)ns.max (l=1 when 1 wait and l=2 when 2 waits.)
   - $t_{h(RAS-RAD)}$=(tcyc/2-13)ns.min
   - t_{RP}=(tcyc/2 x 3-20)ns.min

Measurement conditions

- V_{CC}=3.0 to 3.6V
- Input high or low voltage
  : V_{IH}=1.5V, V_{IL}=0.5V
- Output high or low voltage
  : V_{OH}=1.5V, V_{OL}=1.5V

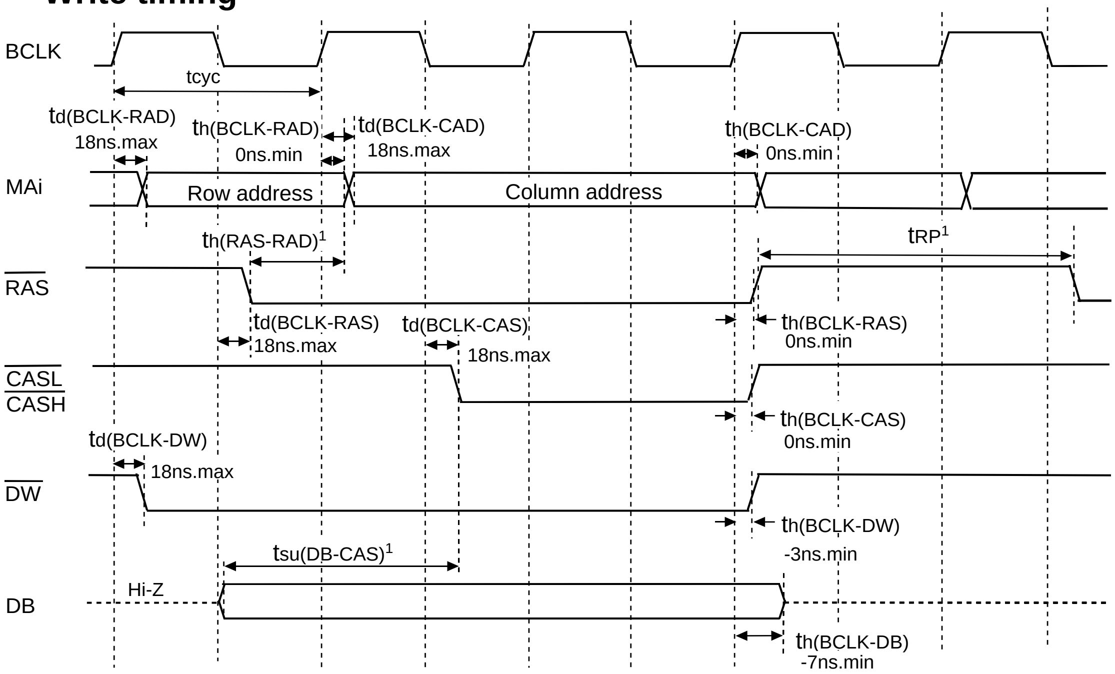
Figure 1.31.13. V_{CC}=3.3V Timing Diagram (4)

Memory expansion mode and microprocessor mode

Vcc=3.3V

(When accessing DRAM space with 2 waits)

Write timing



Notes :

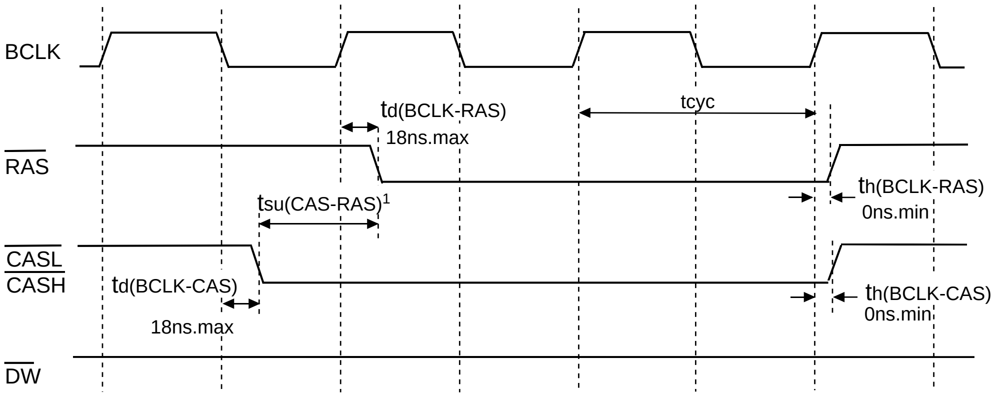
1. It depends on operation frequency.
   $t_{h(RAS-RAD)}$=(tcyc/2-13)ns.min
   t_{RP}=(tcyc/2 x 3-20)ns.min
   $t_{su(DB-CAS)}$=(tcyc-20)ns.min

Measurement conditions
- V_{CC}=3.0 to 3.6V
- Input high or low voltage
  : V_{IH}=1.5V, V_{IL}=0.5V
- Output high or low voltage
  : V_{OH}=1.5V, V_{OL}=1.5V

Figure 1.31.14. V_{CC}=3.3V Timing Diagram (5)

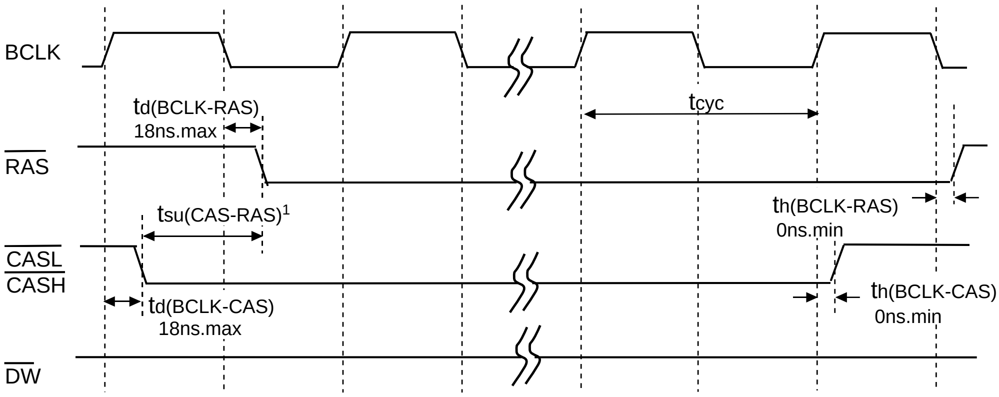
Memory expansion mode and microprocessor mode Vcc=3.3V

Refresh timing ($\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh)



Notes :
1. It depends on operation frequency.
$t_{su(CAS-RAS)}$=(tcyc/2-13)ns.min

Refresh timing (Self-refresh)



Notes :
1. It depends on operation frequency.
$t_{su(CAS-RAS)}$=(tcyc/2-13)ns.min

Measurement conditions
- V_{CC}=3.0 to 3.6V
- Input high or low voltage
  : V_{IH}=1.5V, V_{IL}=0.5V
- Output high or low voltage
  : V_{OH}=1.5V, V_{OL}=1.5V

Figure 1.31.15. V_{CC}=3.3V Timing Diagram (6)

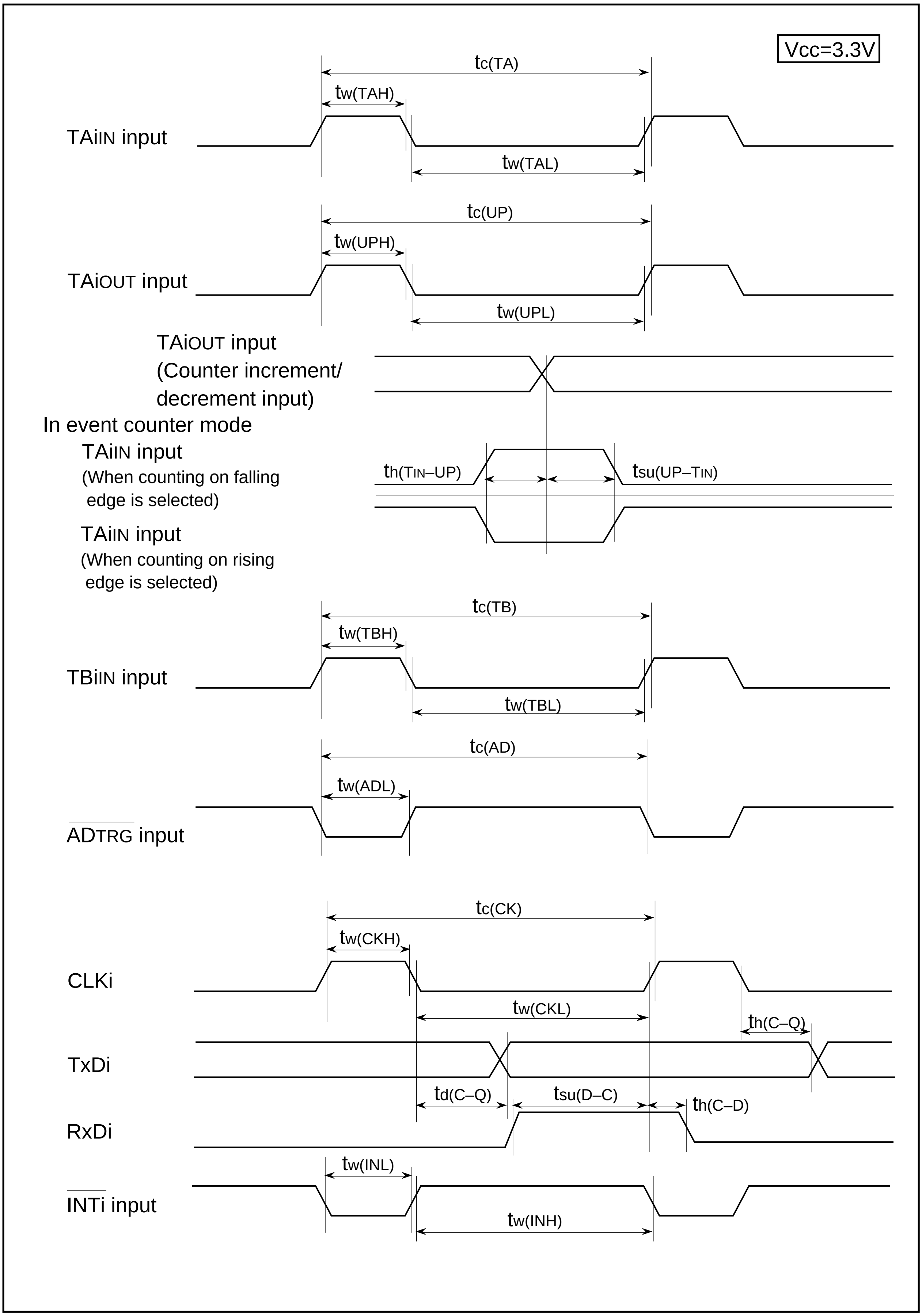
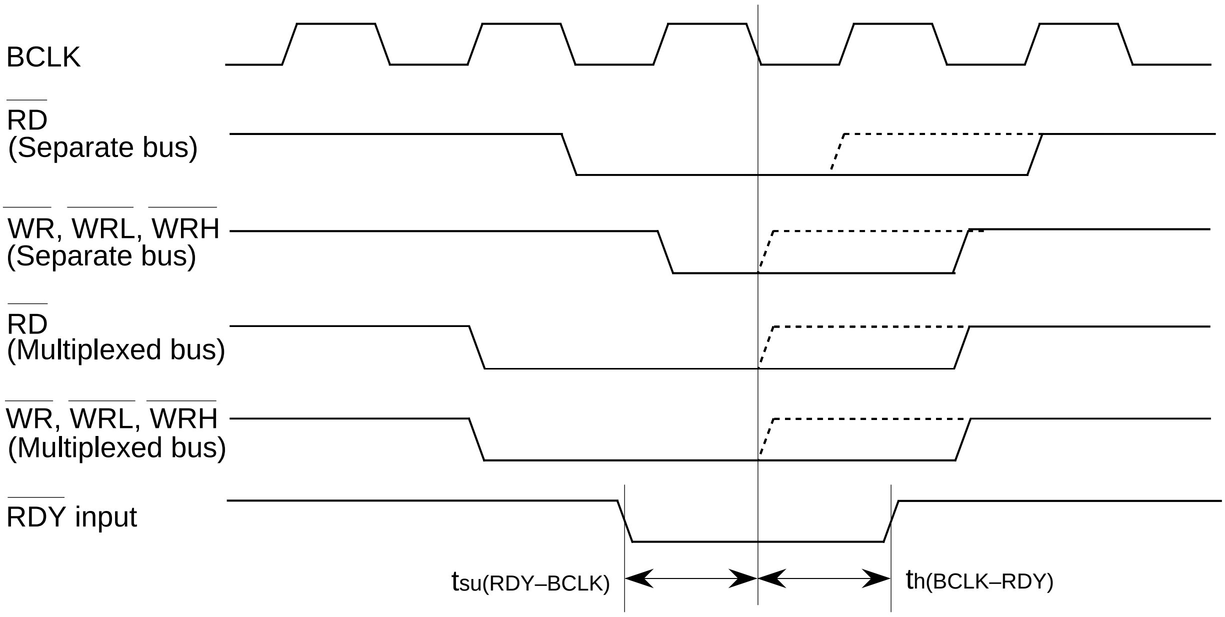


Figure 1.31.16. V_{CC}=3.3V Timing Diagram (7)

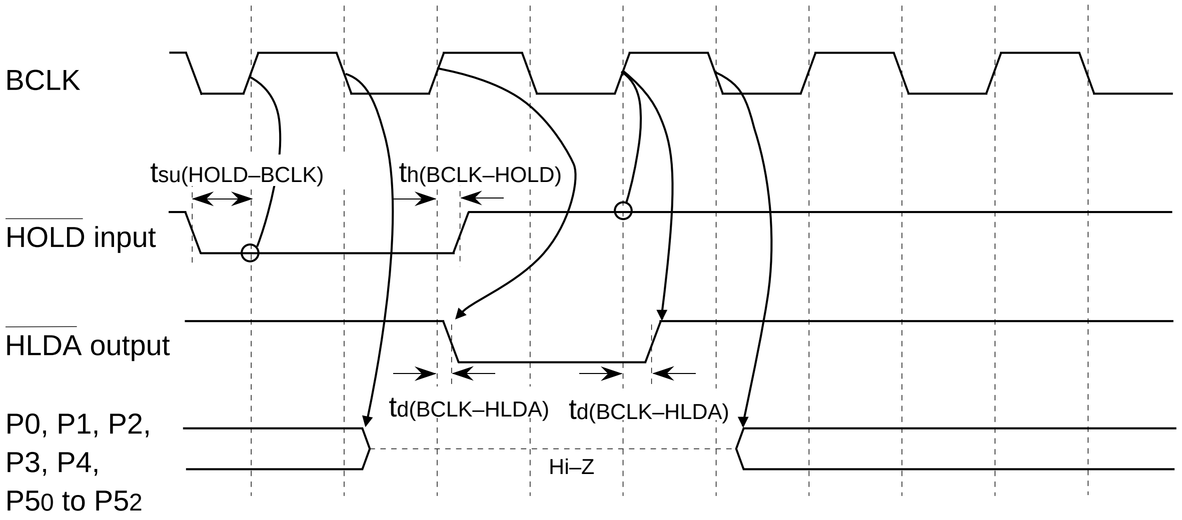
Vcc=3.3V

Memory Expansion Mode and Microprocessor Mode

(Valid only with a wait)



(Valid with or without wait)



Measurement conditions :
- VCC=3.0 to 3.6V
- Input high or low voltage : VIH=2.4V, VIL=0.6V
- Output high or low voltage : VOH=1.5V, VOL=1.5V

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**

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**

PLC1 **73**
PLV **72**
PM0 **49**
PM1 **50**
PRCR **87**
PS0 **349**
PS1 **349**
PS2 **350**
PS3 **350**
PS5 **351**
PS6 **351**
PS7 **352**
PS8 **352**
PS9 **353**
PSC **356**
PSL0 **354**
PSL1 **354**
PSL2 **355**
PSL3 **355**
PUR0 **357**
PUR1 **357**
PUR2 **357**
PUR3 **358**
PUR4 **358**

R

R0 **21**
R1 **21**
R2 **21**
R3 **21**
REFCNT **335**
RLVL **97**, **127**
RMAD0 to RMAD3 **106**
ROMCP **375**

S

SB **21**
SVF **21**
SVP **21**

T

TA0 to TA4 **138**
TA0MR to TA4MR **139**, **144**, **147**, **150**, **152**
TA1, TA2, TA4, TA11, TA21, TA41 **169**
TA1MR, TA2MR, TA4MR **171**
TABSR **139**, **155**, **170**
TB0 to TB5 **154**
TB0MR to TB5MR **155**, **157**, **159**, **161**
TB2 **170**
TB2MR **171**
TB2SC **169**
TBSR **156**
TCSPR **71**, **141**
TRGSR **141**, **170**

U

U0BRG to U4BRG **177**
U0C0 to U4C0 **178**
U0C1 to U4C1 **179**
U0MR to U4MR **177**
U0RB to U4RB **176**
U0SMR to U4SMR **179**
U0SMR2 to U4SMR2 **180**
U0SMR3 to U4SMR3 **181**
U0SMR4 to U4SMR4 **182**
U0TB to U4TB **176**
UDF **140**
USP **21**

V

VCT **21**
VDC0 **371**

W

WCR **58**
WDC **113**
WDTS **113**

X

X0R to X15R **331**
XYC **331**

Y

Y0R to Y15R **331**

Rev.	Date	Description	
		Page	Summary
1.01	12/2002	All	**Full-fledged revision** • Modify the notation system of registers and bits
		23	**Reset** • Delete the figure “Device’s internal status after a reset is cleared”.
		65	**System Clock** • Modify the figure “Clock Generation Circuit”. • Add descriptions about the ‘PLL clock’. • Modify the figure “Status Transition”.
		88	**Interrupt** • Modify the figure “Intelligent I/O Interrupt and CAN Interrupt”. • Add tables ‘registers to be used and settings’. • Change symbols of the bits in the interrupt request register. • Change symbols of the bits in the interrupt enable register.
		137	**Timer A** • Modify the figure “Timer A Configuration”. • Add tables ‘registers to be used and settings’.
		154	**Timer B** • Modify the figure “Timer B Configuration”. • Add tables ‘registers to be used and settings’.
		163	**Three-Phase Control Timer Function** • Change the bit name, the ‘INV17bit’ in the INVC1 register to reserved bit.
		174	**Serial I/O** • Modify the figure “UARTi Block Diagram”. • Add the table ‘registers to be used and settings’ in each mode. • Add distributions about the ‘clock-divided synchronous function (GCI mode)’. • Add descriptions about the ‘bus conflict detect function (IE mode)’.
		264	**Intelligent I/O** • Modify the figure “Intelligent I/O Group 0 Block Diagram”. • Modify the figure “Intelligent I/O Group 1 Block Diagram”. • Modify the figure “Intelligent I/O Group 2 Block Diagram”. • Modify the figure “Intelligent I/O Group 3 Block Diagram”. • Add the table ‘registers and settings’ associated with each function and mode. • Add a bit function of ‘the BCK0 to BCK1 bit in the G0BCR0 to G3BCR0 register’. -Group 0 and 1 • Add descriptions about the ‘HDLC data processing mode’. -Group 0 and 1 • Add distributions about the ‘IEBus mode’. -Group2 • Add descriptions about the ‘8-bit and 16-bit clock synchronous serial I/O function’ -Group3
		338	**A-D Convertor** • Modify the figure “A-D Convertor Block Diagram”. • Add the table ‘pin settings’.
		355	**D-A Convertor** • Add the table ‘pin settings’.
		394	**Usage Precaution** • Add descriptions about the ‘PLL synthesizer’. • Add descriptions about the ‘Timer A’ and ‘Timer B’. • Add descriptions about the ‘Low-Voltage Operation’.

Rev.	Date	Description	
		Page	Summary
1.02	1/2003		**Overview**
		2-3	• Add -40 to 85°C to ‘Operating ambient temperature’ row in Table 1.1.1 and 1.1.2.
		3	• Delete 8-bit or 16-bit clock synchronous serial I/O:1 channel (group3) on ‘Peripheral function’ row in Table 1.1.2.
			SFR
		33	• Modify 00?0 X000$_2$ to 0000 X000$_2$ on ‘value after RESET’ column on ‘017B$_{16}$’ row.
			System Clock
		78	• Modify 0 to 1 on ‘PLC00’ column and ‘10MHz’ row in Table 1.8.2.
		78	• 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.
		80	• Modify 1 to 0 on ‘CM00’ column and ‘BCLK output’ row in Table 1.8.5.
			DMAC
		117	• Add the note 3 in Figure 1.11.2.
			Timer
		141	• Modify TA4 and TA1 to TA0 and TA2 on the TA1TGL and TA1TGH in the top figure of Table 1.14.5.
			• Modify TA4 and TA1 to TA1 and TA3 on the TA2TGL and TA2TGH in the top figure of Table 1.14.5.
			• Modify TA4 and TA1 to TA2 and TA4 on the TA3TGL and TA3TGH in the top figure of Table 1.14.5.
			• Modify TA4 and TA1 to TA3 and TA0 on the TA4TGL and TA4TGH in the top figure of Table 1.14.5.
			Serial I/O
		186	• Modify PD7_0=0 to PD7_2=0 on ‘PD7 register’ column and ‘CLK2 input’ row in Table 1.18.4.
		192	• Modify PD7_0=0 to PD7_2=0 on ‘PD7 register’ column and ‘CLK2 input’ row in Table 1.19.4.
		206	• Modify a function discription on ‘UiRRM’ row in Table 1.20.9.
		207	• Modify PD7_2=0 to PD7_0=0 on ‘PD7 register’ column and ‘SRxD2 input’ row in Table 1.20.11.
			• Modify PD7_0=0 to PD7_2=0 on ‘PD7 register’ column and ‘CLK2 input’ row in Table 1.20.11.
		216	• Modify PS3_4=0 to PS3_5=0 on ‘PS3 register’ column and ‘CLK4 input’ row in Table 1.20.23.
			CAN Module
		226	• Modify PSL2_2=0 to PSL2_1=0 on ‘PSL1 and PSL2 registers’ column and ‘P8$_2$’ row in Table 1.21.2.
			Intelligent I/O
		296	• 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.
		304	• Modify RxD to ISRxD on ‘IPOL’ row and TxD to ISTxD on ‘OPOL’ row in Figure 1.22.33.
		315	• Modify IPS=1 to IPS1=1 on IPS registers column and ‘P11$_2$’ row in Table 1.22.26.
		317	• Modify TCRCRC to TCRCE on ‘CRC’ row in Table 1.22.28.
			• Delete SIOiTR and SIOiRR and add SRTiR in note 3 in Table 1.22.28.
		320	• Modify IER to OER in note 1 in the second figure of Figure 1.22.42.

Rev.	Date	Description	
		Page	Summary
1.02	1/2003	324	• Modify SIOiTR to SIO2TR and SIO5RR to SIO2RR in Table 1.22.30 and 1.22.36.
		334	• Modify GiCR to G3CR in Table 1.22.41.
			DRAMC
		364	• Modify SRDF to SREF in note 3 in Figure 1.27.1.
		385	• Modify IOUTC1$_0$ to OUTC1$_0$ on ‘PSC_3’ row in Figure 1.28.14.
		388	• Modify P0 to P5 to P1 in note 1 in Table 1.28.17.
			Programable I/O Port
		390	• Modify INPC$_1$ to INPC1$_1$ on ‘PS1 register’ column and ‘Bit 4’ row in Table 1.28.4.
		391	• Modify INPC$_0$ to INPC0$_2$ on ‘PS2 register’ column and ‘Bit 0’ row in table 1.28.5.
		393	• Modify ISCLK input to ISCLK0 input on ‘Bit 1’ row in table 1.28.12.
			Usage Precaution
		394	• Modify PM0 to PM00 in “$\overline{\text{HOLD}}$ Signal” • Modify all SP to ISP in (1) SP Setting of “Interrupts”.
		398	• Modify all TAi to TBi in 1. Timer Mode and Event Counter Mode of “Timer B”.
		400	• Modify the CAN module to the microcomputer in “Resetting CNVSS Pin with H”. • Delete a discription of ‘Difference between Flash Memory version and Masked ROM’
			Electric Caractistics
		429	• Modify IOH=5mA to IOL=5mA on ‘VOL’ row and ‘Mesurement Condition’ column in Table 1.31.3.

MITSUBISHI SEMICONDUCTORS
M32C/83 Group DATA SHEET REV. 1.02

December First Edition 2002

Editioned by
Committee of editing of Mitsubishi Semiconductor DATA SHEET

Published by
Mitsubishi Electric Corp., Kitaitami Works